Supervised Learning Algorithms Evaluation on Recognizing Semantic Types of Spanish Verb-Noun Collocations

Alexander Gelbukh and Olga Kolesnikova

Centro de Investigación en Computación, Instituto Politécnico Nacional, México DF, Mexico

gelbukh@gelbukh.com, kolesolga@gmail.com

Abstract. The meaning of such verb-noun collocations as *the wind blows*, *time flies, the day passes by* can be generalized as 'what is designated by the noun exists'. Likewise, the meaning of *make a decision*, *provide support, write a letter* can be generalized as 'make what is designated by the noun'. These generalizations represent the meaning of certain groups of collocations and may be used as semantic annotation. Our objective is to evaluate the performance of some existing supervised machine learning methods on the task of annotating Spanish collocations with generalized meanings, some of which are exemplified above. The experimental results have demonstrated that supervised learning methods achieve significant accuracy allowing them to be used in high quality semantic annotation.

Keywords. Collocations, semantic annotation, supervised machine learning.

Evaluación de algoritmos de aprendizaje supervisado para reconocimiento de las clases semánticas de colocaciones verbo-sustantivo en español

Resumen. El significado de colocaciones de tipo verbo-sustantivo tales como *the wind blows*, el viento sopla, *time flies*, el tiempo vuela, *the day passes*, el día pasa, se puede generalizar y presentar con el patrón 'existe lo que indica el sustantivo'. Análogamente, el significado de *make a decisión*, tomar la decisión, *provide support*, proporcionar apoyo, *write a letter*, escribir una carta, se puede generalizar como 'hacer lo que señala el sustantivo'. Estas generalizaciones representan el significado de ciertos grupos de colocaciones y se pueden utilizar como anotación semántica. Nuestro objetivo es evaluar los algoritmos de aprendizaje de máquina supervisado para etiquetar colocaciones de tipo verbo-sustantivo en español con la propuesta anotación semántica. Los resultados obtenidos muestran que los métodos utilizados logran una precisión alta y se pueden usar para etiquetar colocaciones con la información semántica representada por el significado generalizado.

Palabras clave. Colocaciones, anotación semántica, aprendizaje de máquina supervisado.

1 Introduction

Collocation is such a word combination in which one word, called **the base**, is used in its typical sense and the other word, called **the collocate**, is not used in its typical and well-predicted sense but acquires another meaning depending on the base. For example, in the collocation to take a look, to take does not mean 'to get into one's hands or into one's possession, power, or control' but 'to undertake and make, do, or perform'. On the contrary, in free word combinations, words are used in their typical senses. Examples of free word combinations with to take are to take a book, to take a cup, to take a flower, etc. Collocations present a challenge for natural language processing because the choice of collocates is not motivated semantically but depends on lexical preferences of their respective bases. One and the same meaning can be expressed by different words depending on the base. By saying to take a look, we mean 'to look, or to "perform" a look', but if we want to convey the semantics of "to perform" as applied to the noun laugh, we will say to give a laugh, but not *to take a laugh.

One way to deal with collocations is to expand existing dictionaries by including such meanings of words that are acquired when these words function as collocates. However, it requires

a lot of manual work, is time and money-consuming.

Another way of treating collocations is to store them in special dictionaries of collocations and use such dictionaries alongside with word dictionaries in language applications. The challenge of this approach is to create collocational dictionaries automatically, and it involves automatic extraction of collocations and their semantic annotation. In Section 2 of this paper we will give a brief discussion of the types of semantic annotation, and in Section 3 propose a new type of semantic annotation which we call the generalized meaning. The latter is related to the phenomenon termed lexical functions; a brief explanation of this topic will be presented as state-of-the-art in Section 4. Then in Sections 5 and 6 we study the performance of some algorithms representing four basic types of machine learning techniques, namely, Bayesian classification, rule induction, decision tree construction and the nearest neighbor method, on the task of annotating Spanish verb-noun collocations with seven generalized meanings. Conclusions and future work will be outlined in Section 7.

Some parts of this work were presented at the Mexican International Congress of Artificial Intelligence MICAI-2010, and published in its minutes.

2 Semantic Annotation

Semantic annotation of words in a corpus or in a wordlist is tagging words with names, attributes, comments, descriptions, or other labels which give additional information about the words. Semantic annotation resolves ambiguity of natural language by representing certain concepts in a formal language. The following kinds of semantic tags are commonly used in natural language applications:

1. **Semantic or thematic roles**. These tags generalize the semantics of verbal arguments with the purpose to map them to the syntactic frames. Semantic roles are used in such projects as FrameNet, PropBank, UNL, SIL, EAGLES, among others.

2. **Levin verb classes**. Verb classification by Levin (1993) is based on the ability or inability of a verb to occur in pairs of meaning preserving syntactic frames (diathesis alternations) and on similar meanings. This classification is built on the assumption that syntactic frames reflect the underlying semantics. Verb classes of Levin are implemented as a means of data organization in VerbNet, the largest on-line verb lexicon currently available for English.

3 Proposed Semantic Annotation: Generalized Meaning

The meaning of individual words can be described by definitions in conventional dictionaries for human usage like *Longman Dictionary of Contemporary English* [14] or the *Merriam-Webster English Dictionary* [18] (available online, see References). Often, most frequent words have many senses. For example, *Longman Dictionary of Contemporary English* [14] gives 47 senses for the verb *to take*, 44 for *to make*, the number of senses for *to have* reaches 49, but *to play* looks very poor with only 10 senses!

Having taken a careful look at definitions of the previously given verbs, one will notice that these verbs have a particular meaning aspect in common and we are going to present this fact below. Note, that in this section we use word definitions from the *Longman Dictionary of Contemporary English* [14] mentioned above. Therefore, when referring to the dictionary we mean the *Longman Dictionary of Contemporary English* [14].

Now we will show how some facets of meaning are repeated in verb definitions. We will do this by considering a few examples of polysemic verbs that have the meaning `do sth` (sth = something) included in the lists of their senses.

First, let us consider the verb *to take*. The dictionary gives the following definition of *to take* in the sense `do sth`: ‘a word meaning to do something used with many different nouns to form a phrase that means: ‘do the actions connected with the nouns’: *take a walk* / *take a bath* / *take a breath* / *take a vacation*.’

Another example is the verb *to make*. In the dictionary, it also has the sense `do sth` followed by the comment: 'used with some nouns to mean that someone performs the action of the noun: *make a decision* / *mistake*.'

Now let us consider the verb *to play*. One of its meanings given in the dictionary is 'to take part in a game or sport' like golf, chess, etc. Though the exact phrase *do sth* or the exact word *do* is not encountered in the definition of *to play*, we look for the definition of *to take part* in the dictionary and find the following: *to take part* is 'to do an activity, sport etc. with other people'. Therefore, it can be affirmed that *to play* also has `do` as one of its senses, because in the definition of *to play*, we can substitute *to take part* by 'to do an activity, sport etc. with other people'.

We will call the meaning `do sth`, or just `do`, the generalized meaning of the verbs *to take*, *to make*, *to have*, and *to play*, since *do* is used in the first, more general, part of verb definitions. Likewise, the generalized meanings `make`, `begin`, `continue`, `exist`, `act accordingly`, `undergo`, `cause` can be determined.

Here we present examples of the generalized meanings mentioned in the previous paragraph:

- `make`: *to create*: Her behavior was creating a lot of problems, *to build*: Are they going to build on this land?, *to produce*: Gas can be produced from coal, *to write* (a book, poem);
- `begin`: *to start*: start learning German, *to enter*: Andrea is studying law as a preparation for entering politics, *to introduce*: The death of Pericles in 429 BC introduced a darker period in Athenian history, *to launch*: launch a campaign / appeal / inquiry, *to become*: He became King at the age of 17;
- `continue`: *to keep*: No, we're going to keep the house in Vermont and rent it out, *to maintain*: Britain wants to maintain its position as a world power, *to pursue*: Kristin pursued her acting career with great determination, *to sustain*: The teacher tried hard to sustain the children's interest, *to run*: The contract runs for a year;
- `exist`: *the possibility exists*, *time flies*, *the day passes by*, *a doubt arises*, *joy fills* (*somebody*), *the wind blows*, *an accident happens*, *the rain falls*;
- `act accordingly`: *to use a tool*, *to correct an error*, *to reach a level*, *to fulfill the obligation*;
- `undergo`: *to get a benefit*, *to have an attack* (of a disease), *to receive treatment*, *to gain attention*.

It is a generally accepted fact that the meaning of an individual word depends on its context, i.e. the surrounding words in corpora. This fact is also true in the case of generalized meanings that we have determined. Verbs acquire these meanings when collocate with nouns belonging to a particular semantic group, for example, the group denoting actions. If verb-noun combinations are annotated with generalized meanings `do`, `make`, `begin`, `continue`, `exist`, `act accordingly`, `undergo`, etc., such annotation disambiguates both the verb and the noun. Word sense disambiguation is one of the most important and difficult tasks of natural language processing; therefore, semantic annotation of verb-noun combinations is a task of significant relevance. Collocations tagged with semantic information (and the generalized meaning certainly is semantic information) may be a valuable lexical resource for natural language applications including text analysis, text generation, machine translation, computer assisted language learning, etc.

4 Related Work

4.1 Lexical Functions

It should be noted here that the proposed concept of the generalized meaning is close to the notion of lexical functions developed by the Meaning-Text Theory [15].

Lexical function is a mapping from one word (called **the keyword**, for example, *decision*) to another it collocates with in corpora (called **the lexical function value,** for the word *decision*, the value of one of the lexical functions is the verb *to make*). This mapping is further characterized by the meaning of semantically homogeneous groups of values and by typical syntactic patterns in which lexical function values are used with their respective keywords in texts. For the keyword

decision, the lexical function $Oper_1$, meaning 'do, perform, carry out', gives the value *to make*. That is, to express the meaning 'do, or perform, a decision', one says in English *to make a decision*.

The formalism of lexical functions is intended to represent fixed word combinations, or collocations like *to make a decision*, *to give a lecture*, *to lend support*, etc. For more information on lexical functions, consult [17].

We do not apply the formalism of lexical functions as it is. Our purpose is to annotate verb-noun collocations with generalized meanings, and the meanings we have chosen are not exactly the meanings of lexical functions though have some resemblance to them. Another difference is that lexical functions describe collocations, but generalized meanings are present in collocations as well as in free word combinations. However, the research is made for collocations, not for free word combinations, and this focus is motivated by the importance of collocations in natural language processing as explained in the previous sections.

4.2 Automatic Tagging of Collocations with Lexical Functions

A few attempts have been made to annotate collocations with lexical functions automatically. One of the attempts is reported in [26, 27] where semantic annotation of Spanish verb-noun collocations was viewed as a classification task. Classes were represented by nine lexical functions chosen for experimentation. These lexical functions had the meaning 'perform, experience, carry out something', 'cause the existence of something', 'begin to perform something', 'continue to perform something', etc.

Concerning linguistic data, the authors of [26, 27] used two groups of Spanish verb-noun collocations. In the first group, the nouns belonged to the semantic field of emotions; in the second group, the nouns were field-independent.

For classifying collocations according to lexical functions, the following supervised learning algorithms were applied: Nearest Neighbor technique, Naïve Bayesian network, Tree-Augmented Network Classification technique and a decision tree classification technique based on the ID3-algorithm.

As a source of information for building the training and test sets, the hyperonym hierarchy of the Spanish part of EuroWordNet [25] was used.

A hyperonym of a word A is a word B such that B is a kind of A. For example, flower is a generic concept for rose, daisy, tulip, orchid, so flower is a hyperonym to each of those words. In its turn, a hyperonym of flower is plant, and a hyperonym of plant is living thing, and a hyperonym of living thing is entity. Thus, hyperonyms of a single word form a chain (rose → flower → plant → living thing → entity), and all words connected by the relation kind-of, or hyperonymy, form a tree.

Beside hyperonyms and synonyms, the hyperonym hierarchy in EuroWordNet also includes Base Concepts and Top Concepts. Base Concepts are labels of semantic fields like 'feeling', 'motion', 'possession'. Top Concepts, for example, 'Dynamic', 'Mental', 'Social', are words selected to further characterize the Base Concepts.

Hyperonyms, Base Concepts and Top Concepts were used as features to represent the meaning of verb-noun collocations in [26, 27] and the classification procedure was as follows.

Each lexical function selected for the experiments had its own list of instances, on the basis of which the prototypical instance was calculated. A candidate instance was assigned the lexical function whose prototype value was the most similar to the instance. Similarity was measured using path length in the hyperonym hierarchy.

The average F-measure of about 0.700 was achieved in these experiments. A more detailed analysis of the results obtained in [26, 27] is given in Section 6, where state-of-the-art results are discussed together with the results of our experiments.

5 Experimental Procedure

The objective of our work is to study performance of supervised machine learning methods on the task of annotating Spanish verb-noun collocations with generalized meanings. We have chosen methods which are characteristic of various commonly used approaches in machine learning: Bayesian classification, trees, rules, nearest

neighbor technique, and kernel methods. We train the selected classifiers on a manually compiled corpus of verb-noun collocations tagged with the generalized meanings and the Spanish WordNet senses. After classification models have been built on the training data, the models are tested on annotating unseen data with the meanings. The tests are performed on the training set using 10-fold cross-validation technique.

5.1 Data

Verb-noun collocations for the training sets were extracted automatically from *the Spanish Web Corpus* (available online, see References) by the Sketch Engine [11]. Other tools can also be used for extracting verb-noun pairs, e.g. [2, 22].

From the list of the collocations extracted, we selected those collocations that had the meanings `do`, `make`, `begin`, `continue`, `exist`, `act accordingly`, and `undergo`. Then we annotated such collocations with word senses of the Spanish WordNet [25]. The collected data included 266 collocations with the meaning do, the meaning make was represented by 109 collocations, 24 for begin, 16 for continue, also 16 collocations with the meaning exist, 60 collocations with the meaning act accordingly, the meaning undergo was encountered in 28 collocations. Thus, the total number of verb-noun collocations annotated with seven meanings was 519.

All 519 collocations were included in the training set. Table 1 demonstrates examples of the data. The examples are given as they are encountered in the list compiled automatically, so the nouns are used without articles or quantifiers.

For machine learning methods to be applied, each data instance should be represented by a set of features characteristic for this instance. Hyperonyms were chosen as data features in our experiments. Therefore, the meaning of each noun and each verb was represented as a set of all hyperonyms of this noun or verb. Hyperonyms were extracted from the Spanish WordNet [25]. The meaning of a verb-noun collocation was thus represented as the union of the set of all hyperonyms of the verb and the set of all hyperonyms of the noun. Sets of hyperonyms also included both constituents of verb-noun collocations, i.e., collocational constituents were considered as zero-level hyperonyms.

It should be noted here, that the Spanish WordNet is structured the same way as the Princeton WordNet [6]. In the latter, nouns, verbs, adjectives, and adverbs are organized into synonym sets, or synsets, each representing one underlying lexical concept. When hyperonyms are extracted from the Spanish WordNet. what we actually obtain are synsets of hyperonyms. Each synset has its identification number, and every word in a synset is tagged with a sense number.

Let us consider an example. Suppose we want to build a meaning representation for the collocation *recibir ayuda*, to receive help. Since all collocations in the training set are labeled with the senses of their components, hyperonyms of *recibir*_1 and *ayuda*_1 must be looked for.

First, the synset containing *ayuda*_1 is retrieved: 00782440n *asistencia*_1 *ayuda*_1 (assistance, help) where 00782440n is the synset's identification number. There are two hyperonym synsets for the synset with *ayuda*_1: 00261466n *actividad*_1 (activity) and 0017487n *acto*_2 *acción*_6 (act, action). Therefore, the meaning representation of *ayuda*_1 is the set {*asistencia*_1 *ayuda*_1; *actividad*_1; *acto*_2 *acción*_6}.

Likewise the verb's meaning representation is constructed which is the set {*recibir*_1; *conseguir*_1 *tomar*_1 *sacar*_1 *obtener*_1}.

Lastly, the meaning representation of *recibir*_1 *ayuda*_1 is build and we get the set {*asistencia*_1 *ayuda*_1; *actividad*_1; *acto*_2 *acción*_6; *recibir*_1; *conseguir*_1 *tomar*_1 *sacar*_1 *obtener*_1}. In this set, each hyperonym synset is considered a feature.

5.2 Methodology

Two types of experiments were carried out. First, we studied the performance of diverse machine learning algorithms on the task of annotating Spanish verb-noun collocations with generalized meanings. The purpose was to identify algorithms that operated best. The task was viewed as a binary classification problem; i.e., predicting if a particular collocation belongs to a given class or not. Each of the seven generalized meaning was represented as a class variable with two possible

Table 1. Examples of verb-noun collocations

Generalized meaning	Collocations	
	Spanish	English lit. translation
do	*hacer justicia* *dar beso*	*do justice* *give kiss*
make	*hacer ruido* *establecer criterio*	*make noise* *establish criterion*
begin	*iniciar proceso* *tomar iniciativa*	*initialize process* *take initiative*
continue	*mantener control* *llevar vida*	*maintain control* *lead life*
exist	*relación existe* *año pasa*	*relation exists* *year passes*
act accordingly	*alcanzar meta* *cumplir requisito*	*reach aim* *fulfill requirement*
undergo	*recibir ayuda* *sufrir daño*	*receive help* *suffer damage*

values: 1 if a given collocation is of that class and 0 if it is not. Experiments were fulfilled on 42 supervised machine learning algorithms using WEKA version 3-6-2 software [9, 24, 28];

Class bayes: AODE, AODEsr, BayesianLogisticRegression (BLR), BayesNet, HNB, NaiveBayes, NaiveBayesSimple, NaiveBayesUpdateable, WAODE.

Class functions: LibSVM, Logistic, RBFNetwork, SimpleLogistic, SMO, VotedPerceptron, Winnow.

Class lazy: IB1, IBk, KStar, LWL.

Class rules: ConjunctiveRule, DecisionTable, JRip, NNge, OneR, PART, Prism, Ridor, ZeroR.

Class trees: ADTree, BFTree, DecisionStump, FT, Id3, J48, J48graft, LADTree, RandomForest, RandomTree, REPTree, SimpleCart.

Secondly, the task of annotating Spanish verb-noun collocations with generalized meanings was viewed as a *k*-class classification problem. Each meaning was seen a category, thus we had 7-class classification. To perform such classification, we chose a number of methods that may be called characteristic of various commonly used approaches in machine learning: Bayesian classification (NaiveBayes), rule induction (PART, JRip, Prism, Ridor), decision tree construction (BFTree, SimpleCart, FT, REPTree), the nearest technique (IB1), and kernel methods (SMO).

In both types of experiments, the performance of algorithms was evaluated on the training set using 10-fold cross-validation.

6 Results and Discussion

Tables 2 and 3 present the results of the performance of algorithms chosen for the first type of experiments as explained in Section 5.2. Five best algorithms were identified for each of the seven generalized meanings; in Tables 2, 3 they are listed ranked by the values of F-measure. For each generalized meaning, the average F-measure is given as well.

It is seen from Tables 2 and 3, six of the seven best algorithms tested on the task of annotating Spanish verb-noun combinations with generalized meanings `do`, `make`, `begin`, `continue`, `exist`, `act accordingly`, and `undergo` belong to the category of rule-based techniques. The purpose of algorithms based on rules is to examine data and construct rules which are first-order conditional statements (if…then…).

Rule-based methods acquire and use conceptual knowledge which is human-readable and easy to understand Witten and Frank, [28]. Indeed, a concept consists of a number of features that are necessary and sufficient for description of an abstract idea. It appears that verb-noun collocations as specific linguistic data can be well distinguished by rules that involve hyperonym information. Since hyperonyms are more generic words than a given word, they are able to represent the generalized meaning. It should be remarked here that though hyperonyms and generalized meanings both depict the abstract meaning typical for a significant number of words and in that sense both possess a "generalized" nature, they are different in some important respects. However, this issue has more to do with pure linguistics than with natural language engineering, so we will not consider it here.

The best result is shown by the rule-based method PART with the F-measure value of 0.877 for the meaning `do`. The second best result is achieved by Ridor (F-measure of 0.813 for the

meaning `continue`). The third and the fourth places are held by Prism (0.781 for the meaning `act accordingly` and 0.757 for the meaning `begin`). The resting best algorithms are JRip reaching the value of F-measure of 0.716 for `make`, PART (0.706 for `undergo`), and the last best result of 0.696 is shown by BFTree for the meaning `exist`. Note, that BFTree is the only decision tree learning algorithm mentioned in this paragraph, the other six algorithms which were found to be best on annotating collocations with the generalized meanings are based on rules as it was said earlier in this section. Concerning decision tree algorithms, we will comment on them in the paragraph that follows.

It is also seen from Tables 2 and 3 that the second best type of machine learning algorithms on the task on annotating verb-noun collocations with the generalized meanings is trees. Decision tree learning is a technique whose purpose is to identify a discrete-valued target function as precisely as possible, and the function is represented by a decision tree T.M. Mitchell, [19]. Trees can also be put in the form of rules to make them more human readable.

Tables 4 and 5 present the results of the second type of experiments as explained in Section 5.2., i.e., the results of the performance of algorithms which showed to be best in the first type of experiments: PART, JRip, Prism, and Ridor representing rule induction algorithms, and BFTree. To obtain more evidence of operation of decision tree methods, we also experimented with another two tree-constructing algorithms: SimpleCart, FT, and REPTree.

We also studied the performance of NaiveBayes (NB in Tables 4 and 5), a classical probabilistic Bayesian classifier, as well as the performance of IB1, a basic nearest neighbor instance based learner using one nearest neighbor for classification, and SMO (Sequential Minimal Optimization), an implementation of support vector machine. The highest F-measure for each generalized meaning is in bold. For each classifier, Table 5 also gives the weighted average of F-measure over seven generalized meanings, and the best weighted average is in bold type.

The best result in Tables 4 and 5 is shown by SMO. This algorithm was able to reach the F-measure of 0.916 for predicting the meaning do. It is also the best among methods indicated in

Table 2. Best-performing learning algorithms for the meanings `do, make, begin, continue`

DO		**MAKE**	
rules.PART	**0.877**	rules.JRip	**0.716**
trees.SimpleCart	0.876	trees.SimpleCart	0.708
bayes.BLR	0.874	trees.LADTree	0.706
trees.BFTree	0.869	tress.REPTree	0.704
functions.SMO	0.864	trees.BFTree	0.699
Average	0.872	Average	0.707
BEGIN		**CONTINUE**	
rules.Prism	**0.757**	rules.Ridor	**0.813**
trees.FT	0.711	trees.REPTree	0.800
functions.SMO	0.683	lazy.LWL	0.800
rules.NNge	0.682	functionsLogistic	0.786
trees.Id3	0.667	rules.Prism	0.783
Average	0.700	Average	0.796

Table 3. Best-performing learning algorithms for the meanings `exist`, `act accordingly`, `undergo`

EXIST		**ACT ACCORDINGLY**	
trees.BFTree	**0.696**	rules.Prism	**0.781**
trees.Id3	0.640	bayes.BLR	0.650
trees.J48	0.636	functions.SMO	0.627
lazy.LWL	0.632	trees.FT	0.598
bayes.BLR	0.600	rules.NNge	0.593
Average	0.641	Average	0.650
UNDERGO			
rules.PART	**0.706**		
trees.J48	0.706		
trees.LADTree	0.667		
rules.JRip	0.629		
trees.SimpleCart	0.625		
Average	0.667		

Table 4. Performance of some algorithms using 7-class approach

Algorithm	DO	MAKE	BEGIN	CONTINUE
rules.PART: 21 rules	0.894	0.783	0.524	0.774
rules.JRip: 26 rules	0.878	0.800	0.634	0.800
rules.Prism: 222 rules	0.896	0.840	0.647	0.720
rules.Ridor: 31 rules	0.888	0.709	0.667	0.774
trees.BFTree	0.908	0.814	0.605	0.800
trees.SimpleCart	0.915	0.798	0.605	0.800
trees.FT	0.915	**0.863**	0.714	**0.875**
trees.REPTree	0.893	0.746	0.632	0.759
bayes.NB	0.759	0.698	0.000	0.000
lazy.IB1	0.783	0.620	0.378	0.519
functions.SMO	**0.916**	0.843	**0.773**	0.839

Table 5. Performance of some algorithms using 7-class approach, UND. stands for `undergo`, w.a. stands for weighted average

Algorithm	EXIST	ACT ACC.	UND.	w.a.
rules.PART: 21 rules	0.903	0.685	0.643	0.812
rules.JRip: 26 rules	0.903	0.686	0.667	0.815
rules.Prism: 222 rules	0.889	0.744	0.682	0.841
rules.Ridor: 31 rules	0.710	0.576	0.618	0.780
trees.BFTree	0.875	0.672	0.667	0.830
trees.SimpleCart	0.875	0.672	0.656	0.829
trees.FT	0.903	**0.757**	**0.733**	**0.865**
trees.REPTree	0.759	0.529	0.677	0.788
bayes.NB	0.000	0.119	0.000	0.549
lazy.IB1	0.688	0.462	0.444	0.664
functions.SMO	**0.933**	0.739	0.714	0.861

Tables 4 and 5 for the meanings begin and exist. SMO achieved the second best weighted average for all seven generalized meanings. As it was mentioned above, SMO is an implementation of support vector machine [3], a non-probabilistic binary linear classifier. For a given instance of training data, it predicts which of the two possible classes the instance belongs to. A support vector machine model is a representation of the examples as points in space, mapped so that the examples of the separate classes are divided by a clear gap. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. SVM have been used successfully for many NLP tasks, for example, word sense disambiguation, part-of-speech tagging, language identification of names, text categorization, and others. As our results demonstrate, it is also effective for annotating collocations with the generalized meanings.

However, FT (Functional Tree), a generalization of multivariate trees able to explore multiple representation languages by using decision tests based on a combination of attributes [7] was more successful than SMO for predicting the meanings `make`, `continue`, `act accordingly`, and `undergo` for which FT acquired the highest value of F-measure.

For the rule induction algorithms, Tables 4 and 5 include the number of rules generated by each technique. It appears that PART and JRip are more effective since they show high values of F-measure, 0.812 and 0.815, respectively, and generate quite a modest number of rules (21 and 26) compared with Prism which in spite of a higher F-measure of 0.841 generates as much as 222 rules.

NaiveBayes [10] and IB1 [1] showed a well-noted tendency to have low F-measure for all meanings in the experiments. The failure of NaiveBayes may be explained by the fact that statistical methods are limited by the assumption that all features in data are equally important in contributing to the decision of assigning a particular class to an example and that all features are independent of one another. However, it is a rather simplistic view of data, because in many cases data features are not equally important or independent. The latter is certainly true for linguistic data, especially for such a language phenomenon as hyperonyms (remember, the meaning of collocations is represented by hyperonyms in our training sets). Hyperonyms in the Spanish WordNet form a hierarchic structure where every hyperonym has its ancestor, except for the most general hyperonyms at the top of the hierarchy, and daughter(s), except for most specific hyperonyms at the end of the hierarchy.

However, Naive Bayes is one of the most common algorithms used in natural language processing; it is effective in text classification [5], word sense disambiguation [20], and information retrieval [13]. In spite of that, it could hardly distinguish the generalized meanings of collocations in our experiments. In the previous paragraph some reasons for this failure are suggested.

Low results of IB1 demonstrates that the normalized Euclidean distance used in this technique to find the training instance closest to the given test instance does not approximate well the target classification function. Another reason of low performance can be the fact that if more than one instances have the same smallest distance to the test instance under examination, the first one found is used, which can be erroneous.

Since all best classifiers for predicting the generalized meaning are rule-based, we can suppose that semantics of collocations is better distinguished by rules than on the basis of probabilistic knowledge learned from the training data.

A better performance of rule-based methods on predicting collocational semantics in the form of the generalized meaning proposed in our work leads to an important issue which has a big impact on how natural language applications can be developed. Certainly, computer applications make use of linguistic knowledge, but this knowledge should be carefully selected and proved to be necessary and sufficient for the computer to effectively analyze and generate texts in natural language.

At present, there are two types of knowledge taken into account when designing language applications, namely, the statistical knowledge and the symbolic one. According to these two types of information, two approaches to natural language processing have emerged. The first approach aims at building systems applying linguistic rules, which can be rather numerous and sophisticated. The second approach takes advantage of statistic information like word frequencies and distributions.

As it was observed above, rule-based methods outperformed statistical methods in distinguishing among the meanings of collocations in our experiments. It can be concluded that collocations are analyzed better by rules than by frequency counts; in other words, rules tell us more of what collocations are than frequency counts do and that knowledge in terms of rules is more informative than knowledge in terms of numbers.

Another conclusion that can be drawn from a better performance of rule-based methods concerns theoretical aspects of linguistics, in particular, the definition of collocation. However, this article is oriented towards the computational side of computational linguistics, so we leave this issue without further discussion. A more linguistically-oriented reader may consult the paper by L. Wanner [26] for an overview of the dispute between two “camps”: the adherents of the statistical approach to the definition of collocations beginning with M.A.K. Halliday, see his definition of collocation in [8], and those who

claim that the semantic criterion is crucial for distinguishing collocations from all other word combinations as in I. A. Mel'čuk [16].

The representation of collocational semantic content in the form of generalized meaning as explained in Section 3 is a new way to view lexical and semantic information that can be disclosed by analyzing word co-occurrences. Our attempt to annotate collocations with generalized meanings has been quite satisfactory.

The only research we can refer to when considering our results is Wanner *et al*. [27] because the concept of lexical function is similar to the generalized meaning proposed in this work. It was mentioned in Section 4.2 that while discussing our results, a more detailed presentation and analysis of state-of-the-art results [26] and [27] would be given. Three points should be mentioned here. Firstly, we will compare state-of-the-art results for those lexical functions (explained in Section 4.1) whose semantics is closest to the generalized meanings used in our experiments. Secondly, in [26] and [27], the experiments were carried out on two sets of verb-noun collocations, as it was explained in Section 4.2. The first set included verbal collocations with emotion nouns, in the second set, the nouns were field-independent. Collocations in our corpus are field-independent, so we will compare only the results for field-independent collocations from [26, 27] with our experimental results. Thirdly, we run the experiments on data other than in [26, 27] and moreover, our data is annotated with the generalized meanings as described in Section 5.1 but not with lexical functions. Due to inequality of data sets, it is not fair to compare the results. However, we take the liberty to make such a comparison as a way of presenting the results achieved in the area of automatic semantic annotation.

Therefore, Tables 6 and 7 present some best results reported in [26, 27] together with our results obtained in the experiments of the first type. In these tables, LF stands for lexical function, GM stands for the generalized meaning, W04 and W06 for [26] and [27] respectively; F stands for F-measure, M stands for method, # signifies the number of instances for a given LF, NB is Naive Bayes, NN is the nearest neighbor algorithm. Results in W04 are demonstrated by the nearest neighbor method. The results in W06 are given in terms of precision and recall but here we present them as values of F-measure which is the harmonic mean of precision and recall. F-measure was computed by us to make the comparison with W04 and our results easier.

Another remark is important here. Data representation in our work is different than of [26, 27]. Section 4.2 explained that in order to make the meaning of collocations accessible to supervised classifiers, the collocations were represented in [26] and [27] as sets of hyperonyms, Base Concepts and Top Concepts. In our research, only hyperonyms were included in the data sets. However, the results of our experiments are better although such features as Base Concepts and Top Concepts were absent in our data representation. It seems that these features do not assist in distinguishing among generalized meanings. Nevertheless, for the meanings undergo and make state-of-the-art results are higher. Therefore, further research is necessary to determine the importance of Base Concepts and Top Concepts as features in distinguishing among generalized meanings.

7 Conclusions and Future Work

It has been demonstrated that it is feasible to apply machine learning methods for predicting the semantics of Spanish verb-noun collocations in the form of the generalized meaning proposed in this work. In particular, we studied the performance of learning algorithms on the task of assigning the generalized meanings `do`, `make`, `begin`, `continue`, `exist`, `act accordingly`, and `undergo` to a previously unseen verb-noun pair.

It has also been demonstrated that hyperonym information is sufficient for distinguishing among the generalized meanings. The best F-measure achieved in our experiments is 0.877 using the training set and 10-fold cross-validation technique. This result can be compared with the results on the task of classification of collocations according to lexical functions since the concept of lexical function is similar to the concept of generalized meaning. The highest F-

Table 6. State-of-the-art and our results

LF / GM	# in W04	Result in W04, F	# in W06
$Oper_1$ / do	50	0.609	87
$Oper_2$ / undergo	48	0.759	48
$CausFunc_0$ / make	53	0.766	53
$Real_1$ / act accord.	52	0.741	52
Average		0.719	

Table 7. State-of-the-art and our results (contd.)

LF / GM	Result in W06		# in our work	Our result	
	F	M		F	M
$Oper_1$ / do	0.737	NB	266	0.877	rules. PART
$Oper_2$ / undergo	0.662	NN	28	0.706	rules. PART
$CausFunc_0$ / make	0.676	NN	109	0.716	rules. JRip
$Real_1$ / act accord.	0.500	NN	60	0.781	rules. Prism
Average	0.644			0.770	

measure achieved on classifying collocations using the taxonomy of lexical functions was 0.760. However, such a comparison is not fair due to differences in theoretical grounds and data.

In the future, we plan to test other semantic representations, for example, word space models, and explore the effect of other data features, such as WordNet glosses. We also plan to examine how the techniques of automatic selection of the best classification in Pranckeviciene *et al*. [21] and Escalante *et al*. [4] can be applied to the task of semantic annotation of collocations with generalized meanings. Another intention is to verify classification models on a test set and experiment with different ratios between the training set and the test set.

Acknowledgements

We are grateful to Adam Kilgarriff and Vojtěch Kovář for providing us a list of 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.

A shorter version of the paper has already appeared in MICAI-2010.

References

1. **Aha, D.W., Kibler, D., & Albert, M.C.** (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37–66.
2. **Castro-Sánchez, N.A. & Sidorov, G. (2010).** Analysis of Definitions of Verbs in an Explanatory Dictionary for Automatic Extraction of Actants Based on Detection of Patterns. *Natural Language Processing and Information Systems. Lecture Notes in Computer Science,* 6177, 233–239.
3. **Cortes, C. & Vapnic, V. (1995).** Support-Vector Networks. *Machine Learning*, 20(3), 273–297.
4. **Escalante, H.J., Montes, M., & Sucar, L.E. (2009).** Particle swarm model selection. *Journal of Machine Learning Research*, 10, 405–440.
5. **Eyheramendy, S., Lewis, D., & Madigan, D. (2003).** On the Naive Bayes Model for Text Categorization. *Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, Florida, USA, 332–339.
6. **Fellbaum, C. (1998).** *WordNet: An Electronic Lexical Database*. Cambridge, Mass.: MIT Press.
7. **Gama, J. (2004).** Functional Trees. *Machine Learning*, 55(3), 219–250.
8. **Halliday, M. A. K. (1961).** Categories of the Theory of Grammar. *Word*, 17(3), 241–292.
9. **Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten I. H. (2009).** The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1), 10–18.
10. **Jiang, L., Cai, Z., & Wang, D. (2010).** Improving naive Bayes for classification. *International Journal of Computers and Applications*, 32(3), 328–332.
11. **Kilgarriff, A., Rychly, P., Smrz, P. & Tugwell, D. (2004).** The Sketch Engine. *11th EURALEX International Congress*, 105–116.

12. **Levin, B. (1993).** *English Verb Classes and Alternation: A Preliminary Investigation*. Chicago: University of Chicago Press.
13. **Lewis, D.D. (1998).** Naive (Bayes) at forty: The independence assumption in information retrieval. In Nédellec, C. & Rouveirol, C. (Eds.), *10TH European Conference on Machine Learning*, 1398, 4–15.
14. ***Longman Corpus Network* (1995).** *Longman Dictionary of Contemporary English (*3rd Edition). Harlow, Essex, England: Longman Group Ltd.
15. **Mel'čuk, I. A. (1974).** Opyt teorii lingvističeskix modelej "Smysl ↔ Tekst" [Towards a Theory of Meaning–Text Linguistic Models, in Russian|. Moscow: Nauka.
16. **Mel'čuk, I.A. (1995).** 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*, 167–232. Hillsdale, NJ: Lea Lawrence Erlbaum.
17. **Mel'čuk, I.A. (1996).** 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*, 37–102. Amsterdam, Philadelphia: John Benjamins Academic Publishing.
18. **Merriam-Webster Open Dictionary**. http://www3.merriam-webster.com/opendictionary/
19. **Mitchell, T.M. (1997).** *Machine Learning*. New York: McGraw Hill.
20. **Pedersen, T. (2000).** A simple approach to building ensembles of naive bayesian classifiers for word sense disambiguation *1st North American Chapter of the Association for Computational Linguistics Conference (NAACL 2000)*, Seattle, WA, USA, 63–69.
21. **Pranckeviciene, E., Somorjai, R. & Tran, M.N. (2007).** Feature/model selection by the linear programming combined with state-of-art classifiers: What can we learn about the data. *International Joint Conference on Neural Networks (IJCNN 2007)*, Orlando, Florida, USA, 1627–1632.
22. **Sidorov, G. (1996).** Lemmatization in automatized system for compilation of personal style dictionaries of literature writers. *Word of Dostoyevsky* (266–300). Moscow, Russia: Russian Academy of Sciences.
23. **Spanish Web Corpus.** http://trac.sketchengine.co.uk/wiki/Corpora/SpanishWebCorpus/
24. **The University of Waikato Computer Science Department Machine Learning Group. WEKA download** at http://www.cs.waikato.ac.nz/ml/weka/index_downloading.html/
25. **Vossen, P. (1998).** *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Dordrecht: Kluwer Academic Publishers.
26. **Wanner, L. (2004).** Towards automatic fine-grained classification of verb-noun collocations. *Natural Language Engineering*, 10(2), 95–143.
27. **Wanner, L., Bohnet, B. & Giereth, M. (2006).** What is beyond Collocations? Insights from Machine Learning Experiments. *12th EURALEX International Congress, Turin, Italy*, 1071–1084.
28. **Witten, I. H. & Frank, E. (2005).** *Data Mining: Practical machine learning tools and techniques*. Amsterdam, Boston, MA : Morgan Kaufmann

Alexander Gelbukh holds M.Sc. degree in mathematics y Ph.D. degree in computer science. Since 1997 he leads the Natural Language Processing of the Computing Research Center of the National Polytechnic Institute (CIC-IPN), Mexico. He is academician of the Mexican Academy of Sciences, National Researcher of Mexico of excellence level 2, and the Secretary of the Mexican Society for Artificial Intelligence (SMIA). He is author or editor of more than 440 publications and co-author of three books in the areas of natural language processing and artificial intelligence. More information about him can be found on his personal page www.Gelbukh.com.

Olga Kolesnikova obtained her M.Sc. degree in Linguistics from Novosibirsk State Pedagogical Institute, Russia, and her Ph.D. in computer science from the Computing Research Center of the National Polytechnic Institute (CIC-IPN), Mexico. Her research is in the area of computational linguistics; in particular, she is interested in text semantic analysis.

Article received on 07/02/2011; accepted on 27/09/2011.