ISSN 2395-8618

Dictionary and Corpus-Based Study of Lexical Functions in Spanish

Olga Kolesnikova and Alexander Gelbukh

***Abstract*—In this work we study semantic and contextual characteristics of four types of verb-noun collocations in Spanish. Each type corresponds to a different lexical function defined in the works of Žolkovskij and Mel'čuk [1, 2, 3] and further elaborated by Apresjan [4, 5]. First, we explain how the typology of lexical functions can be viewed as a consistent way to classify collocations according to their semantic and syntactic patterns. Then, using four lexical functions as well as free word combinations as classes of verb-noun pairs, we examine how they can be identified automatically by supervised learning methods. To build a semantic representation of verb-noun pairs, we used WordNet hypernyms of the verb and the noun. To study contextual properties of the classes, we experimented on a corpus of news. The highest F1-score achieved in the experiments was 0.81 for CausFunc$_1$ using hypernyms. We found that contextual characteristics were not powerful enough to discriminate among subtle semantic differences of lexical functions: the best F1-score of 0.62 for Real$_1$ was achieved by GaussianProcessClassifier using raw frequency of context words after removing stopwords from the corpus. Discussing our results, we looked for features which could account for higher or lower results.**

***Index Terms*—Verb-noun collocations, lexical functions, hypernyms, context, supervised learning.**

I. Introduction

EXTRACTING meaning and relations between words has been central to research in computational linguistics and its more technical counterpart, natural language processing (NLP). The majority of techniques are based on statistics obtained from a corpus: words are represented as vectors in a vector space with frequency of context words as vector features. To mine word associations, the distance between vectors is computed: the less the distance, the stronger the relation between the respective words.

Traditionally, word associations are discovered at the paradigmatic and syntagmatic levels. At the paradigmatic level, such associations or lexical relations as synonymy, antonymy, hyponymy/hypernymy, and meronymy/holonymy are defined. It can be noted that their definitions are semantic-driven, i.e., it is possible to make a meaningful abstraction of associations belonging to the same type and express it in simple terms and patterns, e.g., as in WordNet Reference Manual[1] [6]:

X is a hyponym of Y if X is a (kind of) Y
Y is a hypernym of X if X is a (kind of) Y
X is a meronym of Y if X is a part of Y
Y is a holonym of X if X is a part of Y

Likewise, synonymy and antonymy can be defined[2]:

X is a synonym of Y if X is the same as Y
X is an antonym of Y if X is the opposite of Y

Concerning syntagmatic word associations, they are more numerous and diverse: the central notion here is syntactic and semantic combinability or compatibility; a word can be characterized or "portrayed" by other words it typically collocates with. Excellent examples of such combinatorial "portraits" are word sketches generated by Sketch Engine, an online corpus-based language processing and lexicographic tool[3] [7]. A sketch includes a set of wordlists, where each list is comprised of words with a certain grammatical relation to the query word. For example, if the query word is a noun, then its sketch displays the relations object_of, subject_of, modifier, modifiers, and/or, etc. Figure 1 is a partial representation of the sketch for the noun *control* generated on the British Academic Written English Corpus (BAWE)[4].

Now we will take a closer look at the column of the relation object_of containing verbs used with *control* as the direct object forming verbal phrases: *exercise control*, *regain control*, *maintain control*, etc. Reviewing the verbs, it can be noted that, on the one hand, they have different meaning, but on the other hand, they can be grouped in sets according to similar semantics:

{*achieve*, *gain*, *regain*},
{*exercise*, *exert*},
{*maintain*, *retain*},

Manuscript received on June 12, 2019, accepted for publication on September 20, 2019, published on December 30, 2019.

Olga Kolesnikova is with the Escuela Superior de Cómputo, Instituto Politécnico Nacional, Mexico (e-mail: kolesolga@gmail.com).

Alexander Gelbukh is with the Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico (web: www.gelbukh.com).

[1] http://ccl.pku.edu.cn/doubtfire/Semantics/WordNet/Manual.html

[2] https://en.wiktionary.org/wiki/Wiktionary:Semantic_relations

[3] https://www.sketchengine.eu

[4] BAWE was developed at the Universities of Warwick, Reading and Oxford Brookes under the directorship of Hilary Nesi and Sheena Gardner (formerly of the Centre for Applied Linguistics, Warwick), Paul Thompson (formerly of the Department of Applied Linguistics, Reading) and Paul Wickens (School of Education, Oxford Brookes), with funding from the ESRC (RES-000-23-0800). The corpus includes 2,761 academic works with about 7 million words written at the universities in the UK.

ISSN 2395-8618

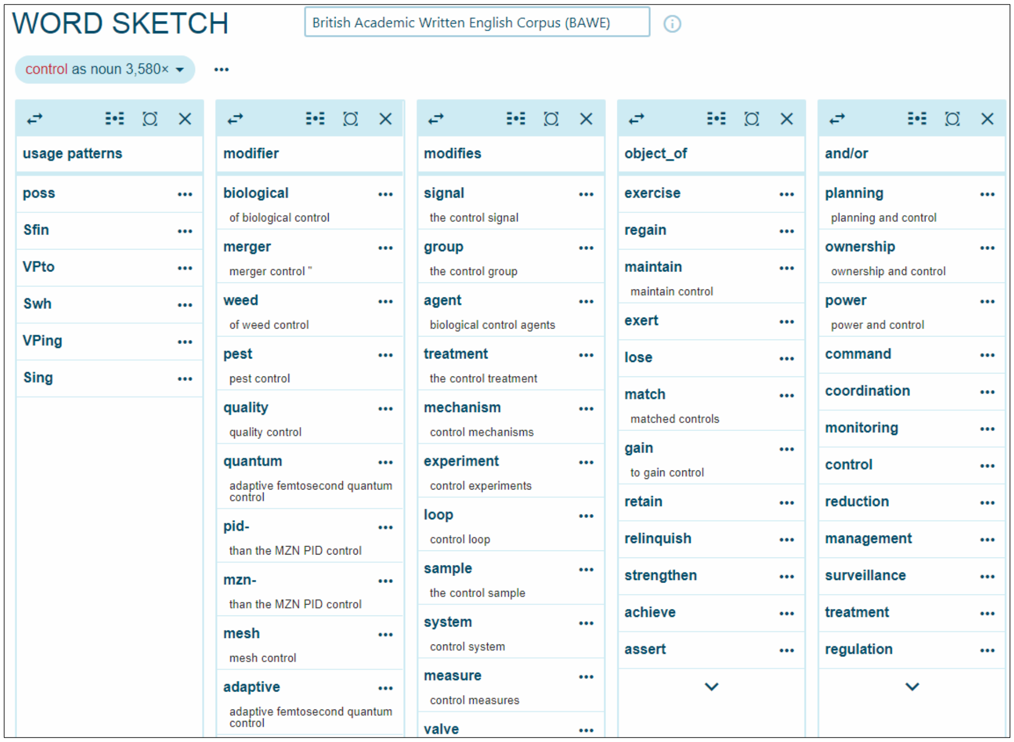

Fig. 1. Word sketch of the noun *control*.

{*strengthen*, *assert*},
{*relinquish*, *lose*}.

The verbs in each set convey some unique concept which can be possibly formalized as follows:

{*achieve*, *gain*, *regain*}:	begin_to_carry_out (*control*)
{exercise, *exert*}:	carry_out (*control*)
{maintain, *retain*}:	continue_to_carry_out (*control*)
{strengthen, *assert*}:	carry_out_to_a_greater_extent (*control*)
{*relinquish*, *lose*}:	terminate_to_carry_out (*control*)

In fact, the same semantic concepts can be found in verb-noun relations across different nouns. In Table 1 we present a number of verbal concepts exemplified with the verbs we looked up in the object_of column in sketches for the five nouns: *control*, *support*, *obstacle*, *favour*, and *attention*. The sketches were generated by Sketch Engine on the aforementioned BAWE corpus and the ukWac corpus[5] (Ferraresi, Zanchetta, Baroni, & Bernardini, 2008). At the beginning of each row, a formalization of the respective semantics is proposed.

The concepts specified in Table 1 and many others alike can be used to characterize and classify the various syntagmatic relations between words thus enabling meaningful generalizations of diverse phrase types. A powerful abstraction of these semantic concepts is lexical function, a formalism proposed and developed in the works of Žolkovskij and Mel'čuk [1, 2, 3] and further elaborated by Apresjan [4, 5] to represent numerous lexical semantic relations between words in a unified and consistent way.

Lexical function (LF) is defined similarly to a function in mathematics: it is an abstraction of the dependency relation between a word w of a vocabulary V and a set W of words $\{w_1', \dots w_n'\}$, $W \subseteq V$. The word w is the argument to the lexical function, and the set W is its value: LF(w) = W. Each LF represents a specific lexical semantic relation between the LF argument and each word in the LF value set. About 60 lexical functions have been defined on both the paradigmatic and syntagmatic levels, their detailed descriptions can be found in [9]. Table 2 shows some examples borrowed from [9-13].

This work is a study of four syntagmatic lexical functions most frequently observed in verb-noun collocations. These functions are as follows:

1. Oper_1, from Latin *operari* 'do, carry out', means 'to perform the action given by the noun', e.g. *make a decision*, *make a step*, *take a shower*, *take a walk*, *commit suicide*, *do an exercise*, *give a talk*, *give a smile*, *have breakfast*, *pay a visit*, *lend support*. The number in the subscript means that the action is realized by the agent, the first argument of the verb

[5] The ukWac contains texts retrieved by crawling the .uk domain and includes more than 2 billion words.

ISSN 2395-8618

TABLE I
SEMANTIC CLASSES OF VERBS THAT COLLOCATE WITH NOUNS *CONTROL*, *SUPPORT*, *OBSTACLE*, *FAVOUR*, AND *ATTENTION*.

Noun	***control***	***support***	***obstacle***	***favour***	***attention***
Sense[6]	an act or instance of controlling; power or authority to guide or manage	the act or process of supporting, the condition of being supported	something that impedes progress or achievement	friendly regard shown toward another especially by a superior; approving consideration or attention	the act or state of applying the mind to something
cause	*enforce, enhance, ensure, establish, impose, offer, provide, set*	*encourage, ensure, generate, give, lend, offer, provide*	*cause, create, establish, place, pose, present, provide, raise*	*bestow, confer, do, give, grant, offer, pay*	*attract, awaken, bring, captivate, capture, catch, direct, draw, grab, grip, point, pull, trigger*
begin_ to_carry_out	*achieve, acquire, gain, get, obtain, regain, resume, take*	*attract, find, gain, get, obtain, receive, win*	*address, confront, encounter, face, meet*	*accept, earn, find, gain, get, obtain, receive, win*	*arrest, center, concentrate, fix, gain, garner, get, give, place, put, turn*
carry_out	*exercise, exert, have, hold*	*enjoy, use*	*deal with, experience, handle, tackle*	*have, experience, enjoy, use*	*dedicate, devote, enjoy, exercise, focus, give, occupy, pay, relish*
continue_ to_carry_out	*continue, develop, ensure, keep, maintain, preserve, retain*	*continue, maintain*	*remain*	*keep*	*continue, develop, hold, maintain, sustain*
make_visible	*demonstrate, exhibit*	*demonstrate, express, reveal, show*	*identify, show*	*exhibit, show*	*display, reflect, show*
carry_out_ to_a_greater_ extent	*assert, extend, increase, strengthen*	*extend*	*increase*	*shower, spread*	*broaden, extend, force, grow, heighten, increase, widen*
carry_out_ to_a_lesser_ extent	*decrease, ease, limit, loose, reduce, relax*	*limit, reduce, remove*	*minimise, reduce*	*reduce*	*avert, confine, decrease, diminish, discourage, divide, limit, minimize, reduce, restrict, shift, split*
terminate_ to_carry_out	*abolish, lose, relinquish, remove, surrender*	*end, lose, refuse, withdraw, withhold*	*avoid, eliminate, fix, ignore, overcome, remove, resolve*	*lose, withdraw, withhold*	*deflect, detract, distract, divert, escape, lose, remove, seize, withdraw*

2. Real_1, from Latin *realis* ‘real’, means ‘to fulfill the requirement imposed by the noun or performing an action typical for the noun’, the action is also carried out by the agent, e.g. *drive a bus*, *follow advice*, *spread a sail*, *prove an accusation*, *succumb to illness*, *turn back an obstacle*.
3. CausFunc_0 is a complex LF comprised of two semantic units: Caus, from Latin *causare* ‘cause’ and Func_0 from Latin *functionare* ‘function’; CausFunc_0 means ‘to cause the action/event denoted by the noun to happen, occur’, zero in the subscript means that the action is viewed as happening without respect to its agent or that there is no agent, e.g. *bring about the crisis*, *create/present a difficulty*, *call elections*, *establish a system*, *produce an effect*.
4. CausFunc_1 is another complex LF meaning ‘to cause the event of someone performing the action denoted by the noun’, e.g. *open a perspective*, *raise hope*, *open a way*, *cause damage*, *instill a habit* (*into someone*).

II. MATERIALS AND METHODOLOGY

A. Dataset of Lexical Functions

The objective of this work is to study semantic and contextual properties of the four syntagmatic lexical functions described in the previous section. We intend to examine how these properties would allow for detecting LFs automatically with supervised learning methods. The study was performed on Spanish verb-noun combinations annotated with lexical functions. Table 3 presents a few instances of our dataset. For each LF we borrowed 60 samples from the list of Spanish verb-noun collocations annotated with LFs [14] in order to make our dataset balanced.

[6] Definitions of senses are borrowed from https://www.merriam-webster.com/

ISSN 2395-8618

TABLE II
EXAMPLES OF LEXICAL FUNCTIONS, ARG IS THE LF ARGUMENT, VALUE IS THE LF VALUE.

Paradigmatic		**Syntagmatic**	
LF	**Definition**	**LF**	**Definition**
Syn(*car*) = *vehicle* Syn(*modify*) = *change*	Synonym	Bon(*lecture*) = *informative* Bon(*meal*) = *exquisite*	From Lat. *bonus*, good; positive property of Arg
Anti(*open*) = *close* Anti(*high*) = *low*	Antonym	Degrad(*milk*) = *sour* Degrad(*tooth*) = *decay*	Degrade, become permanently worse or bad
Conv$_{21}$(*give*) = *receive* Conv$_{21}$(*include*) = *belong*	From Lat. *conversivum*, conversive; the same action viewed as performed by the agent (Arg) and as performed by the recipient (Value)	Liqu(*file*) = *delete* Liqu(*law*) = *annul*	Liquidate Arg, cause Arg not to be
Gener(*table*) = *furniture* Gener(*rose*) = *flower*	Generic concept of Arg	Magn(*love*) = *deep* Magn(*patience*) = *infinite*	From Lat. *magnus*. great; intensification of Arg: very, to a high degree, intense, intensely
Sing(*fleet*) = *ship* Sing(*sand*) = *grain*	A singular instance, unit of Arg	Son(*ass*) = *bray* Son(*bell*) = *chime*	From Lat. *sonare*. to sound; typical sound or noise of Arg
Mult(*cattle*) = *herd* Mult(*bee*) = *swarm*	Multitude of Arg	Manif(*amazement*) = *lurk* Manif(*joy*) = *explode*	From Lat. *manifestare*, to manifest; Arg manifests itself (in something)

We achieve our objective by determining the extent to which lexical functions can be automatically identified by supervised learning methods, first, using semantic information obtained from WordNet [15] and, second, using contextual data retrieved from a corpus of 1,131 issues of Excélsior newspaper within the period from April 1996 to June 1999. We explain both methods in two subsections which follow.

B. Semantic Approach

To take advantage of the semantic information provided by WordNet, we extracted all hypernyms of the verb and noun for each verb-noun collocation. As an example consider the collocation *tomar una decisión* (make a decision) annotated with Oper$_1$.

Hypernyms of *tomar* and *decisión* can be viewed in Table 4 together with sense glosses, below each Spanish synset its corresponding synset in the English WordNet is given, numbers following the word and underscore are sense numbers. The synsets were retrieved from the Multilingual Central Repository version 3.0 [16]. Synsets containing *tomar* and *decisión* were also included as the zero-level hypernyms.

C. Contextual Approach

Contextual data was obtained taking four words to the left of the verb and four words to the right of the noun, words between the verb and the noun were not taken into account in this work. We also studied the impact of stopwords by keeping or removing them from the corpus thus obtaining two context representations: with and without stopwords. The bag of words model was applied in our experiments, i.e., the word order was disregarded, only word frequencies were considered.

As an example, let us take the same collocation *tomar decisión* considered in section II.B to see its context in the following segment:

> Ahora *le corresponde el turno a la microeconomía: su gobierno debe* ***tomar la*** decisión *sin vacilar en ningún momento de ser factor de unidad por sus acciones determinantes en beneficio de la micro y pequeña empresa* (borrowed from the article *Propuesta a Zedillo Sobre la Cartera Vencida* (A proposal to Zedillo about overdue loans), Excélsior, April 6, 1996; literal word-for-word translation: Now to it corresponds the turn to the microeconomics: its government must make the decision without hesitate in no moment to be factor of unity for its actions decisive for benefit of the micro and small business).

The context of *tomar decisión* using the option of keeping stopwords is the set {*microeconomía*, *su*, *gobierno*, *deber*, *sin*, *vacilar*, *en*, *ningún*}, and the context of this collocation after stopwords elimination becomes {*turno*, *microeconomía*, *gobierno*, *deber*, *vacilar*, *momento*, *factor*, *unidad*}. Such two types of sets were generated for all occurrences of *tomar decision*, the sets of each type were united to represent the context of this collocation.

ISSN 2395-8618

TABLE III
EXAMPLES OF SPANISH LEXICAL FUNCTIONS, EACH SPANISH COLLOCATION IS FOLLOWED BY ITS ENGLISH TRANSLATION.

$Oper_1$	**$Real_1$**	**$CausFunc_0$**	**$CausFunc_1$**
dar un beso give a kiss	*alcanzar una meta* reach a goal	*convocar un concurso* call for a contest	*dar sentido* give sense
ejercer una función exercise a function	*aprovechar la oportunidad* use the opportunity	*crear un sistema* create a system	*abrir un espacio* open a space
hacer cálculo do calculation	*contestar la pregunta* answer the question	*declarar guerra* declare war	*ofrecer la oportunidad* offer the opportunity
jugar un papel play a role	*cumplir el requisito* fulfill the requirement	*encontrar el camino* *find the way*	*prestar ayuda* give help
presentar una dificultad present a difficulty	*lograr el objetivo* achieve the objective	*establecer un criterio* establish a criterion	*reservar el derecho* reserve the right
realizar una tarea do a task	*recorrer un camino* walk along a road	*producir un aumento* produce an increase	*poner un límite* put a limit
tener sabor have taste (about food)	*seguir la instrucción* follow the instruction	*provocar un cambio* cause a change	*hacer realidad* make (sth) a reality

TABLE IV
HYPERNYMS OF *TOMAR* AND *DECISIÓN* USED AS FEATURES TO REPRESENT THE MEANING OF THE COLLOCATION *TOMAR UNA DECISIÓN* (MAKE A DECISION). BELOW EACH SPANISH SYNSET, THE CORRESPONDING ENGLISH SYNSET IS GIVEN.

Word	**Synset**	**Synset gloss**
tomar (lit. take)	{coger_1 escoger_1 seleccionar_1 elegir_1 triar_1 decantar_1 optar_1 tomar_2} {choose_1 take_10 select_1 pick_out_1}	pick out, select, or choose from a number of alternatives
	{decidir_2 determinar_1 resolver_3 decidirse_1 concluir_4} {decide_1 make_up_one's_mind_1 determine_5}	reach, make, or come to a decision about something
decisión (decision)	{decisión_2 determinación_3 resolución_3} {decision_1 determination_5 conclusion_9}	the act of making up your mind about something
	{ elección_2 selección_1} {choice_2 selection_1 option_3 pick_9}	the act of choosing or selecting
	{acción_1 acto_1 hecho_1} {action_1}	something done (usually as opposed to something said)
	{acción_5 acto_2 actividad_humana_1 acción_humana_1} {act_2 deed_2 human_action_1 human_activity_1}	something that people do or cause to happen
	{evento_1 suceso_1} {event_1}	something that happens at a given place and time
	{rasgo_psicológico_1} {psychological_feature_1}	a feature of the mental life of a living organism
	{abstracción_2} {abstraction_6 abstract_entity_1}	a general concept formed by extracting common features from specific examples
	{entidad_1 ente_1} {entity_1}	that which is perceived or known or inferred to have its own distinct existence (living or nonliving)

D. Supervised Learning

To apply the supervised learning methods chosen for our experiments, we represent lists of hypernyms (semantic approach) and context words (contextual approach) of verb-noun pairs as vectors of features in a vector space model. In the semantic approach, binary feature representation was used: 1 signifies that a given hypernym is present among hypernyms of a verb-noun pair, and 0 signifies that it is absent. Within the contextual approach, word counts (raw frequencies) were used as vector features.

We also experimented with another vector representation using tf-idf values for context words as vector features. Binary features in hypernym vectors can also be viewed as numbers, so we computed tf-idf values for these features. Tf-idf is a widely used function to assign weights to words such that the importance of rare words for meaning discrimination is increased, while the influence of very frequent or common words is decreased. Frequent and common words can be found in the context of words with very different semantics; thus, they do not help in distinguishing among different senses, and moreover, they introduce noise into the dataset.

ISSN 2395-8618

In total, we built six vector representations:

1. Vectors of binary features for hypernyms,
2. Vectors of tf-idf values for hypernyms,
3. Vectors of context word counts on the original corpus,
4. Vectors of context words counts on the corpus after stopwords removal,
5. Vectors of tf-idf values for context words on the original corpus,
6. Vectors of tf-idf values for context words on the corpus after stopwords removal.

We defined the task of automatic identification of lexical functions in verb-noun collocations as a classification task, in which collocations are to be classified into four classes corresponding to the four chosen syntagmatic LFs: Oper_1, Real_2, $\mathrm{CausFunc}_0$, and $\mathrm{CausFunc}_1$.

To the four classes mentioned above, we added free verb-noun combinations as another class to see how they can be detected in contrast to lexical functions. Free word combinations are phrases whose meaning can be derived as a combination of individual word meanings, e.g., *cook a meal*, *give a pen*, *take a box*. On the other hand, the meaning of restricted word combinations or collocations cannot be interpreted using the same compositional approach, e.g., *cook the books*, *give a smile*, *take a bite*.

Concerning supervised learning methods, we selected techniques commonly used in NLP tasks and compatible with our vector representations; we applied them as implemented in the Scikit-learn package for Python with default parameters [17]. In what follows we list the chosen methods, for each method its name in the Scikit-learn implementation is given in parenthesis:

Multinomial Naïve Bayes (MultinomialNB),
Gaussian Naïve Bayes (GaussianNB)
Gaussian processes for probabilistic classification (GaussianProcessClassifier),
K-nearest neighbors vote (KNeighborsClassifier),
Support vector machine (LinearSVC),
Decision tree multi-class classification (DecisionTreeClassifier),
Random forest algorithm (RandomForestClassifier),
Multi-layer perceptron (MLPClassifier),

In the experiments, 50% of the dataset was used for training, and the other 50% was used for validation. In the next section we present the results of our experiments.

III. Results

In this section, we give the results of classifying verb-noun collocations according to the four syntagmatic lexical functions explained in the introduction and exemplified in section II.A. Free verb-noun combinations were also included as a class. For classification, we used supervised learning methods selected in section II.D. The results are presented in terms of precision, recall, and F1-score. For classification purposes, precision (P) is defined as the number of true positives (Tp) divided by the sum of the number of true positives and the number of false positives (Fp); recall (R) is the number of true positives divided by the sum of the number of true positives and the number of false negatives (Fn), F1-score (F1) is the harmonic mean of precision and recall:

$$P = \frac{Tp}{Tp+Fp}\ ,\ R = \frac{Tp}{Tp+Fn}\ ,\ F1 = \frac{2*P*R}{P+R}\ .$$

A. Experiments with Semantic Representation

As it was explained in section II.A, hypernyms of the verb and the noun in a verb-noun pair were used as binary (Boolean) features in vectors. A feature in a vector had a value of either 1 (if a hypernym is present among the hypernyms of a verb-noun pair) or 0 (otherwise). In fact, 1 and 0 can be interpreted numerically as counts of the number of times a hypernym occurs in the set of all hypernyms of a verb-noun pair, thus, for each count, tf-idf measure can be computed.

Table 5 displays the results of classifying verb-noun pairs into five classes: four lexical functions (Oper_1, Real_1, $\mathrm{CausFunc}_0$, $\mathrm{CausFunc}_1$) and free verb-noun combinations (FC) with supervised learning methods selected in section II.C. We decided to incorporate verb-noun pairs which are not collocations but free word combinations in order to see how they can be distinguished as opposed to lexical functions.

Table 5 is divided vertically into two sections: the left section entitled as counts gives results for the case where vectors include binary-valued features; the right section of the table entitled as tf-idf contains results for the case where vectors include tf-idf measure calculated for each binary value interpreted numerically. For each classifier and for each class, values of precision (P), recall (R), and F1-score (F1) are given. For convenience, F1-scores are in bold. The best F1-score for each class and for each feature representation is underlined (in other words, it is the best F1-score in each column). The lowest part of the table contains average values of precision, recall, and F1-score for each class. In this row, the best F1-score among all classes and both feature representations is underlined, i.e., it is the highest value among all F1-score values in this row.

Interestingly enough, the best classifiers in this experiment were support vector machine (LinearSVC) and DecisionTreeClassifier. LinearSVC distinguished successfully among lexical functions and free word combinations on binary-valued features and DecisionTreeClassifier showed more efficiency on tf-idf values. Comparing all best F1-score values for classes, it can be noted that the highest value of 0.81 was achieved by LinearSVC on $\mathrm{CausFunc}_1$ using counts as vector features. The best average F1-score was also shown for $\mathrm{CausFunc}_1$ on counts.

ISSN 2395-8618

TABLE V
RESULTS OF CLASSIFICATION USING HYPERNYMS AS FEATURES.

Classifier	Metrics	Oper$_1$	Real$_1$	Caus Func$_0$	Caus Func$_1$	FC	Oper$_1$	Real$_1$	Caus Func$_0$	Caus Func$_1$	FC
		counts					tf-idf				
Multinomial NB	P	0.54	0.80	0.42	0.71	0.45	0.45	1.00	0.42	0.90	0.38
	R	0.70	0.33	0.61	0.52	0.61	0.67	0.14	0.55	0.55	0.74
	F1	**0.61**	**0.47**	**0.50**	**0.60**	**0.52**	**0.54**	**0.24**	**0.48**	**0.68**	**0.50**
Gaussian NB	P	0.57	0.70	0.50	0.48	0.47	0.52	0.61	0.47	0.47	0.40
	R	0.48	0.53	0.45	0.61	0.61	0.48	0.47	0.45	0.45	0.61
	F1	**0.52**	**0.60**	**0.47**	**0.53**	**0.53**	**0.50**	**0.53**	**0.46**	**0.46**	**0.48**
Gaussian Process Classifier	P	0.44	0.74	0.42	0.70	0.47	0.50	1.00	0.50	0.90	0.31
	R	0.78	0.39	0.35	0.58	0.61	0.70	0.17	0.45	0.55	0.78
	F1	**0.56**	**0.51**	**0.39**	**0.63**	**0.53**	**0.58**	**0.29**	**0.47**	**0.68**	**0.44**
KNeighbors Classifier	P	0.36	0.60	0.33	0.68	0.44	0.50	0.93	0.50	0.68	0.41
	R	0.67	0.25	0.35	0.52	0.52	0.59	0.39	0.65	0.64	0.57
	F1	**0.47**	**0.35**	**0.34**	**0.59**	**0.48**	**0.54**	**0.55**	**0.56**	**0.66**	**0.47**
Linear SVC	P	0.59	1.00	0.62	0.96	0.48	0.57	0.84	0.53	0.88	0.43
	R	0.85	0.61	0.68	0.70	0.65	0.74	0.44	0.58	0.67	0.70
	F1	**0.70**	**0.76**	**0.65**	**0.81**	**0.56**	**0.65**	**0.58**	**0.55**	**0.76**	**0.53**
Decision Tree Classifier	P	0.69	0.75	0.51	0.85	0.52	0.68	1.00	0.56	0.92	0.35
	R	0.74	0.58	0.68	0.70	0.57	0.63	0.58	0.58	0.67	0.74
	F1	**0.71**	**0.66**	**0.58**	**0.77**	**0.54**	**0.65**	**0.74**	**0.57**	**0.77**	**0.48**
Random Forest Classifier	P	0.59	0.76	0.58	0.84	0.58	0.67	0.83	0.47	0.86	0.39
	R	0.89	0.61	0.68	0.64	0.48	0.59	0.53	0.65	0.58	0.65
	F1	**0.71**	**0.68**	**0.63**	**0.72**	**0.52**	**0.63**	**0.64**	**0.54**	**0.69**	**0.49**
MLP Classifier	P	0.51	0.79	0.45	0.87	0.40	0.49	0.77	0.44	0.77	0.38
	R	0.67	0.53	0.48	0.61	0.61	0.67	0.47	0.45	0.52	0.61
	F1	**0.58**	**0.63**	**0.47**	**0.71**	**0.48**	**0.56**	**0.59**	**0.44**	**0.62**	**0.47**
Average	P	0.54	0.76	0.48	0.76	0.48	0.55	0.87	0.49	0.80	0.38
	R	0.72	0.48	0.53	0.61	0.58	0.63	0.40	0.55	0.58	0.68
	F1	**0.61**	**0.58**	**0.50**	**0.67**	**0.52**	**0.58**	**0.52**	**0.51**	**0.66**	**0.48**

TABLE VI
RESULTS OF CLASSIFICATION USING THE ORIGINAL CONTEXT OF VERB-NOUN PAIRS.

Classifier	Metrics	Oper$_1$	Real$_1$	Caus Func$_0$	Caus Func$_1$	FC	Oper$_1$	Real$_1$	Caus Func$_0$	Caus Func$_1$	FC
		counts					tf-idf				
Multinomial NB	P	0.18	0.39	0.00	0.75	0.00	0.15	0.00	0.00	0.00	0.00
	R	0.82	0.63	0.00	0.09	0.00	1.00	0.00	0.00	0.00	0.00
	F1	**0.30**	**0.48**	**0.00**	**0.16**	**0.00**	**0.26**	**0.00**	**0.00**	**0.00**	**0.00**
Gaussian NB	P	0.15	0.38	1.00	0.14	0.00	0.15	0.36	1.00	0.13	0.00
	R	0.73	0.37	0.03	0.06	0.00	0.68	0.37	0.03	0.06	0.00
	F1	**0.25**	**0.38**	**0.06**	**0.08**	**0.00**	**0.24**	**0.36**	**0.06**	**0.08**	**0.00**
Gaussian Process Classifier	P	0.00	0.18	0.00	0.00	0.00	0.20	0.42	0.00	1.00	0.00
	R	0.00	1.00	0.00	0.00	0.00	0.91	0.70	0.00	0.06	0.00
	F1	**0.00**	**0.31**	**0.00**	**0.00**	**0.00**	**0.33**	**0.53**	**0.00**	**0.11**	**0.00**
KNeighbors Classifier	P	0.23	0.26	0.27	0.50	0.30	0.23	0.42	0.67	0.36	0.40
	R	0.32	0.33	0.34	0.15	0.27	0.73	0.59	0.06	0.24	0.20
	F1	**0.26**	**0.29**	**0.30**	**0.23**	**0.28**	**0.35**	**0.49**	**0.11**	**0.29**	**0.27**
Linear SVC	P	0.34	0.37	0.44	0.39	0.28	0.29	0.48	0.53	0.53	0.25
	R	0.50	0.37	0.43	0.26	0.30	0.59	0.81	0.26	0.50	0.07
	F1	**0.41**	**0.37**	**0.43**	**0.32**	**0.29**	**0.39**	**0.60**	**0.35**	**0.52**	**0.11**

ISSN 2395-8618

Decision Tree Classifier	P	0.20	0.28	0.24	0.33	0.05	0.26	0.22	0.28	0.33	0.15
	R	0.36	0.30	0.23	0.26	0.03	0.41	0.26	0.29	0.18	0.13
	F1	**0.26**	**0.29**	**0.24**	**0.30**	**0.04**	**0.32**	**0.24**	**0.28**	**0.23**	**0.14**
Random Forest Classifier	P	0.25	0.23	0.42	0.48	0.30	0.14	0.23	0.32	0.32	0.18
	R	0.36	0.22	0.46	0.35	0.27	0.27	0.22	0.29	0.24	0.13
	F1	**0.30**	**0.23**	**0.44**	**0.41**	**0.28**	**0.18**	**0.23**	**0.30**	**0.27**	**0.15**
MLP Classifier	P	0.22	0.25	0.39	0.62	0.32	0.17	0.50	0.44	0.55	0.25
	R	0.09	0.67	0.37	0.15	0.27	0.64	0.59	0.11	0.35	0.03
	F1	**0.13**	**0.36**	**0.38**	**0.24**	**0.29**	**0.27**	**0.54**	**0.18**	**0.43**	**0.06**
Average	**P**	0.20	0.29	0.34	0.40	0.16	0.20	0.33	0.40	0.40	0.15
	R	0.40	0.49	0.23	0.16	0.14	0.65	0.44	0.13	0.20	0.05
	F1	**0.24**	**0.34**	**0.23**	**0.22**	**0.15**	**0.29**	**0.37**	**0.16**	**0.24**	**0.09**

TABLE VII

RESULTS OF CLASSIFICATION USING THE CONTEXT OF VERB-NOUN PAIRS AFTER STOPWORDS ELIMINATION

Classifier	**Metrics**	**$Oper_1$**	**$Real_1$**	**Caus $Func_0$**	**Caus $Func_1$**	**FC**	**$Oper_1$**	**$Real_1$**	**Caus $Func_0$**	**Caus $Func_1$**	**FC**
		counts					**tf-idf**				
Multinomial NB	P	0.00	0.46	0.12	0.00	0.27	0.00	0.00	0.18	0.00	0.00
	R	0.00	0.51	0.29	0.00	0.18	0.00	0.00	1.00	0.00	0.00
	F1	**0.00**	**0.49**	**0.17**	**0.00**	**0.21**	**0.00**	**0.00**	**0.30**	**0.00**	**0.00**
Gaussian NB	P	0.08	0.38	0.24	0.00	0.00	0.08	0.40	0.21	0.00	0.00
	R	0.25	0.43	0.47	0.00	0.00	0.25	0.51	0.35	0.00	0.00
	F1	**0.12**	**0.41**	**0.31**	**0.00**	**0.00**	**0.12**	**0.45**	**0.26**	**0.00**	**0.00**
Gaussian Process Classifier	P	0.00	0.45	0.00	0.60	0.00	0.00	0.00	0.14	0.24	0.00
	R	0.00	0.97	0.00	0.45	0.00	0.00	0.00	0.29	0.70	0.00
	F1	**0.00**	**0.62**	**0.00**	**0.51**	**0.00**	**0.00**	**0.00**	**0.19**	**0.35**	**0.00**
KNeighbors Classifier	P	0.00	0.83	0.12	0.25	0.33	0.00	1.00	0.00	0.20	0.43
	R	0.00	0.14	0.18	0.70	0.12	0.00	0.05	0.00	0.85	0.18
	F1	**0.00**	**0.23**	**0.14**	**0.36**	**0.17**	**0.00**	**0.10**	**0.00**	**0.32**	**0.25**
Linear SVC	P	0.00	0.57	0.21	0.29	0.50	0.00	0.46	0.20	0.26	0.50
	R	0.00	0.22	0.24	0.60	0.59	0.00	0.16	0.29	0.60	0.29
	F1	**0.00**	**0.31**	**0.22**	**0.39**	**0.54**	**0.00**	**0.24**	**0.24**	**0.36**	**0.37**
Decision Tree Classifier	P	0.00	0.36	0.21	0.36	0.22	0.00	0.33	0.19	0.55	0.19
	R	0.00	0.11	0.29	0.50	0.35	0.00	0.22	0.24	0.60	0.29
	F1	**0.00**	**0.17**	**0.24**	**0.42**	**0.27**	**0.00**	**0.26**	**0.21**	**0.57**	**0.23**
Random Forest Classifier	P	0.00	0.56	0.16	0.40	0.11	0.00	0.57	0.23	0.30	0.27
	R	0.00	0.24	0.29	0.60	0.12	0.00	0.22	0.35	0.50	0.35
	F1	**0.00**	**0.34**	**0.21**	**0.48**	**0.11**	**0.00**	**0.31**	**0.28**	**0.38**	**0.31**
MLP Classifier	P	0.00	0.00	0.23	0.36	0.28	0.00	0.37	0.18	0.27	0.70
	R	0.00	0.00	0.18	0.70	0.71	0.00	0.19	0.18	0.65	0.41
	F1	**0.00**	**0.00**	**0.20**	**0.47**	**0.40**	**0.00**	**0.25**	**0.18**	**0.38**	**0.52**
Average	**P**	0.01	0.45	0.16	0.28	0.21	0.01	0.39	0.17	0.23	0.26
	R	0.03	0.33	0.24	0.44	0.26	0.03	0.17	0.34	0.49	0.19
	F1	**0.02**	**0.32**	**0.19**	**0.33**	**0.21**	**0.02**	**0.20**	**0.21**	**0.30**	**0.21**

B. Experiments with Contextual Representation

In this section, we see how lexical functions and free word combinations can be distinguished by their context. We can also observe the importance of stopwords for distinguishing among classes. Table 6 presents the results of our experiments on the original context of verb-noun pairs, i.e., with stopwords preserved. Table 7 gives the classification results using context after stopwords elimination. The structure and notation of Tables 6 and 7 are the same as those of Table 5, described in section III.A.

First, let us observe the classification results using the original context of verb-noun pairs, i.e., extracting it from the original corpus, without previous stopwords elimination. Concerning the numbers in general, it stands out that they are much lower than those in Table 6, where we used hypernyms as vector features in the experiments. The highest average F1-score in Table 6 is 0.37 for $Real_1$ using tf-idf for context words, while the highest average F1-score in Table 5 is 0.67 for $CausFunc_1$ using binary valued features for hypernyms, almost two times bigger. Also, comparing best F1-score values in the

ISSN 2395-8618

columns, we see that in Table 6 it is 0.60 for $Real_1$ using tf-idf for context words (the same class and configuration with the best average F1-score), and in Table 5 it is 0.81 for $CausFunc_1$ on binary valued features for hypernyms. However, it is the same classifier which gave the largest number of best results—LinearSVC—and in both experiments this largest number is the same.

Beside LinearSVC, there were other methods with high results for some classes and vector feature configurations: RandomForestClassifier was efficient on distinguishing $CausFunc_0$ and $CausFunc_1$ using counts (0.44 and 0.41, respectively), Multinomial NB showed good results on $Real_1$ using counts (0.48).

Table 7 presents the results of classification using context words of verb-noun pairs on the corpus after stopwords elimination, so beside our study of the effect of numerical feature representation (counts and tf-idf) on the efficiency of classification, we can observe the importance of stopwords for this task.

On the whole, the numbers in Table 7 are lower than in Table 6, so in comparison with Table 5 where we used hypernyms as vector features, they are very low. Also, there are many zeros in Table 7: some classifiers could not distinguish some classes at all. $Oper_1$ was hardly distinguished by GaussianNB (F1-score value as low as 0.12), and this value was the same on counts and tf-idf. MLPClassifier could not distinguish $Oper_1$ using counts and tf-idf as well as $Real_1$ using counts. However, this classifier achieved 0.25 for $Real_1$ with tf-idf.

FC were not identified at all by GaussianNB and Gaussian-ProcessClassifier using counts and tf-idf, MultinomialNB distinguished FC with an F1-score of 0.21 on counts, but it was completely unable to distinguish this class with tf-idf. KNeighborsClassifier did not detect $CausFunc_0$ using tf-idf. MultinomialNB turned out to be most inefficient among all classifiers: it could not detect $Oper_1$ and $CausFunc_1$ using counts or tf-idf, also failed to identify $Real_1$ and FC using tf-idf.

Although in many cases the classes were not detected adequately, the best F1-score among all classes and both feature representations in Table 7 is almost the same as the best F1-score in Table 6: a value of 0.62 was reached by GaussianProcessClassifier on $Real_1$ using counts. Remember, the best value in Table 6 was 0.60 showed by LinearSVC for $Real_1$ (the same lexical function!) using tf-idf. Concerning the best average F1-score values, they are also quite close: 0.33 in Table 7 for $CausFunc_1$ using counts and 0.37 for $Real_1$ in Table 6 using tf-idf.

IV. RESULTS

In this section we expose some insights we could get analyzing the results given in section III. On the one hand, we focused on semantic and syntactic characteristics of the five classes used in our experiments: four verb-noun lexical functions ($Oper_1$, $Real_1$, $CausFunc_0$, and $CausFunc_1$) and the class which includes free verb-noun combinations (FC). Our intention was to find out how such characteristics allow for better or, perhaps, problematic automatic discrimination among the classes by supervised machine learning methods. On the other hand, on the basis of our results, one could also observe how classifiers differed in their performance with respect to the classification task.

While discussing the results, we present them in a concise and graphical form for a more convenient observation. Tables and diagrams in this section will help the reader to take notice of correlations between various properties of lexical functions and features as well as methods found to be most effective for classification. In the first subsection we discuss classifiers' performance with respect to the classes and their feature representation, and in the next subsection we consider the classification results.

A. Classifiers and Feature Representation

This subsection presents a summary of classifiers' efficiency on our classification task. To make further discussion on lexical functions more detailed, we decided to first analyze the performance of each classifier tested experimentally.

Table 8 gives the values of precision and recall averaged over all classes for each feature representation; to compute these values, the numbers in Tables 5-7 we used. However, the F1-score values in Table 8 were not computed as averaged F1-score values borrowed from Tables 5-7, as in such case the mean F1-score value would not represent the relation between the mean values of precision and recall fairly. To give fair F1-score values, we computed them from the precision and recall values in Table 8. The highest F1-score for each classifier is underlined.

Among all classifiers, LinearSVC (support vector machine implemented in the Scikit-learn package, Pedregosa et al., 2011) stands out as it achieved the best F1-score of 0.75 on hypernym counts. The second-best method is RandomForestClassifier with an F1-score of 0.70 also shown on hypernym counts. The lowest F1-score of 0.07 was showed by MultinomialNB on tf-idf values computed for words in the original context of verb-noun pairs. The other technique with the same lowest F1-score was GaussianProcessClassifier tested on the original context word counts.

B. Classification

Figures 2-6 present precision, recall, and F1-score averaged over all classifiers for each class: $Oper_1$, $Real_1$, $CausFunc_0$, $CausFunc_1$, and FC. The bar diagrams in the figures show how classification results depend on the feature representation types given by the following numbers: 1 stands for hypernym counts (binary features), 2 stands for hypernym tf-idf values, 3 represents word counts in the original context of verb-noun pairs, 4 denotes the representation comprised of tf-idf values computed for words in the original context of verb-noun pairs, 5 stands for word counts in the context after stopwords elimination, and 6 represents tf-idf for words in the context after stopwords elimination.

ISSN 2395-8618

TABLE VII
CLASSIFIERS' PERFORMANCE EVALUATION.

Classifier	Metrics	Features					
		Hypernyms		Original context		Context without stopwords	
		counts	tf-idf	counts	tf-idf	counts	tf-idf
Multinomial NB	P	0.62	0.69	0.33	0.04	0.15	0.05
	R	0.54	0.48	0.38	0.25	0.20	0.25
	F1	**<u>0.58</u>**	**0.57**	**0.35**	**0.07**	**0.17**	**0.33**
Gaussian NB	P	0.56	0.52	0.42	0.41	0.18	0.17
	R	0.52	0.46	0.30	0.28	0.29	0.28
	F1	**<u>0.54</u>**	**0.49**	**0.35**	**0.33**	**0.22**	**0.21**
Gaussian Process Classifier	P	0.58	0.72	0.04	0.40	0.26	0.10
	R	0.52	0.47	0.25	0.42	0.36	0.25
	F1	**0.55**	**<u>0.57</u>**	**0.07**	**0.41**	**0.30**	**0.14**
KNeighbors Classifier	P	0.49	0.65	0.32	0.42	0.30	0.30
	R	0.45	0.57	0.28	0.40	0.26	0.22
	F1	**0.47**	**<u>0.61</u>**	**0.30**	**0.41**	**0.28**	**0.25**
Linear SVC	P	0.79	0.70	0.38	0.46	0.27	0.23
	R	0.71	0.61	0.39	0.54	0.27	0.26
	F1	**<u>0.75</u>**	**0.65**	**0.38**	**0.50**	**0.27**	**0.24**
Decision Tree Classifier	P	0.70	0.80	0.26	0.27	0.23	0.27
	R	0.68	0.62	0.29	0.28	0.22	0.27
	F1	**0.69**	**<u>0.70</u>**	**0.27**	**0.27**	**0.22**	**0.27**
Random Forest Classifier	P	0.69	0.71	0.34	0.25	0.28	0.28
	R	0.71	0.59	0.35	0.26	0.28	0.27
	F1	**<u>0.70</u>**	**0.64**	**0.34**	**0.25**	**0.28**	**0.27**
MLP Classifier	P	0.66	0.62	0.37	0.42	0.15	0.21
	R	0.57	0.53	0.32	0.42	0.22	0.26
	F1	**<u>0.61</u>**	**0.57**	**0.34**	**0.42**	**0.18**	**0.23**

Table 9 presents the best values of precision, recall and F1-score (not average values as in Figures 2-6) for each lexical functions and free verb-noun combinations in order to see with what method and feature representation each class was identified best. F1-score values are in bold for convenience. The data in Table 9 can also have a practical application: if a high precision or a high recall is required for a natural language system or tool to function in a more robust manner, the numbers in this table can help a language engineer to choose an adequate method and feature representation.

Among all classes, the best F1-score value in Table 9 is 0.81 (underlined). It was achieved by LinearSVC for $\mathrm{CausFunc}_1$ on hypernym counts as vector features. Actually, it is clear from Table 9 that all best F1-scores were always obtained based on hypernym counts, i.e., binary features, though applying different classifiers. Another interesting observation is that for all lexical functions, context works better in terms of recall with the only exception of free verb-noun combinations for whose detection hypernym information is needed. Concerning precision, higher values were attained by taking advantage of hypernym relations as carriers of semantic information in the case of Oper_1 and Real_1, for the other classes—$\mathrm{CausFunc}_0$, $\mathrm{CausFunc}_1$, and FC—context worked really well.

According to Table 9, the best classifier in our experiments was LinearSVC (support vector machine); its best F1-score result of 0.81 was demonstrated on $\mathrm{CausFunc}_1$ using hypernym counts. Now let us compare it with the results for the other lexical functions and free verb-noun combinations. The goal is to get some insights into properties of lexical functions which influence the degree of success in their automatic identification.

Table 10 gives the confusion matrix for classification with LinearSVC using hypernym counts. It can be seen there that $\mathrm{CausFunc}_1$ is mostly confused with $\mathrm{CausFunc}_0$, which is in fact not surprising because both of them include a causative semantic element and, consequently, share hypernyms. As an example, consider two $\mathrm{CausFunc}_1$ collocations: *proporcionar un servicio*, *crear un sistema*, and two $\mathrm{CausFunc}_0$ collocations: *ofrecer una posibilidad*, *abrir un espacio*. Their hypernyms are presented in Table 11, where the synsets of the words in these collocations are considered as zero-level hypernyms, English translation is given for each word, common hypernyms are underlined.

ISSN 2395-8618

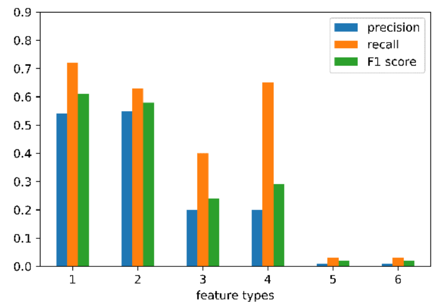

Fig. 2. $Oper_1$.

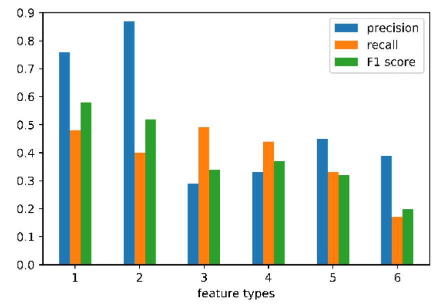

Fig. 3. $Real_1$.

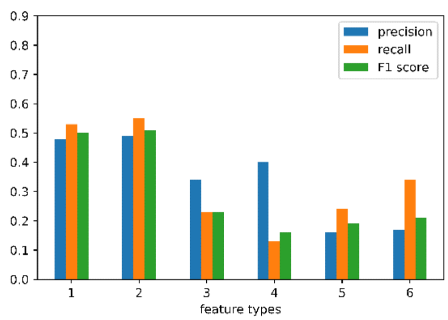

Fig. 4. $CausFunc_0$.

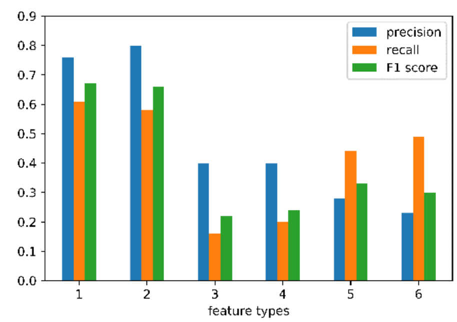

Fig. 5. $CausFunc_1$.

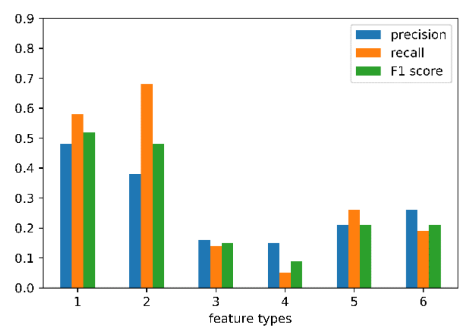

Fig. 6. Free verb-noun combinations (FC).

Now let us compare the confusion matrix resulting from LinearSVC with another confusion matrix for the same classifier but on another feature representation: tf-idf for words in the original context, i.e., without stopwords elimination. Let us remark at this point that tf-idf in many cases works better than counts (raw frequency) due to non-uniform frequency of the classes in the corpus, see Table 12. The confusion matrix for LinearSVC referred to previously in this paragraph is displayed in Table 12. For $CausFunc_1$, this classifier showed the second best F1-score of 0.52, the first best F1-score was 0.60 for $Real_1$.

It is seen in Table 13 that half of $CausFunc_1$ samples (17 of 34) were classified as $Real_1$. In the hypernym representation, no $CausFunc_1$ pair was classified as $Real_1$, it was confused not with $Real_1$ but with $CausFunc_0$. $CausFunc_1$ and $Real_1$ are different in meaning, however, due to this confusion we can suppose that their contexts are similar. Indeed, in the corpus we used in the experiments, the eight-word window context of 60 $CausFunc_1$ samples contained 10,449 unique words (we do not consider word frequencies here), the context of $Real_1$ included 9,705 unique words, and it turned out that both contexts had 5,133 unique words in common.

This high similarity of contexts for two different lexical functions is an interesting detail which does not agree with the distributional hypothesis of word meaning proposed by Harris [18] who assumed that differences in context signal differences in meaning. In fact, this assumption has been widely recognized and applied showing good results in many natural language processing tasks such as topic mining [19], text classification [20], word sense disambiguation [21], sentiment detection [22], authorship attribution [23], among others. However, in our experiments, more subtle semantic differences among lexical functions were not reflected well enough in the context for the classifiers to identify them.

ISSN 2395-8618

TABLE IX
BEST RESULTS FOR EACH LEXICAL FUNCTION AND FREE VERB-NOUN COMBINATIONS (FC).

LF	**Metrics**	**Value**	**Feature**	**Classifier**
$Oper_1$	P	0.69	hypernyms, counts	DecisionTreeClassifier
	R	1.00	original context, tf-idf	MultinomialNB
	F1	**0.71**	hypernyms, counts	DecisionTreeClassifier
				RandomForestClassifier
$Real_1$	P	1.00	hypernyms, tf-idf	DecisionTreeClassifier
				MultinomialNB
				GaussianProcessClassifier
			hypernyms, counts	LinearSVC
			context after stopwords elimination, tf-idf	KNeighborsClassifier
	R	1.00	original context, counts	GaussianProcessClassifier
	F1	**0.76**	hypernyms, counts	LinearSVC
$CausFunc_0$	P	1.00	original context, counts	GaussianNB
			original context, tf-idf	
	R	1.00	context after stopwords elimination, tf-idf	MultinomialNB
	F1	**0.65**	hypernyms, counts	LinearSVC
$CausFunc_1$	P	1.00	original context, tf-idf	GaussianProcessClassifier
	R	0.85	context after stopwords elimination, tf-idf	KNeighborsClassifier
	F1	**0.81**	hypernyms, counts	LinearSVC
FC	P	0.70	context after stopwords elimination, tf-idf	MLPClassifier
	R	0.78	hypernyms, tf-idf	GaussianProcessClassifier
	F1	**0.56**	hypernyms, counts	LinearSVC

TABLE X
CONFUSION MATRIX FOR LINEARSVC ON HYPERNYM COUNTS.

		Predicted class				
		$Oper_1$	$Real_1$	Caus $Func_0$	Caus $Func_1$	FC
Known class	$Oper_1$	23	0	2	0	2
	$Real_1$	7	22	2	0	5
	$CausFunc_0$	1	0	21	1	8
	$CausFunc_1$	3	0	6	23	1
	FC	5	0	3	0	15

TABLE XI
HYPERNYMS OF VERBS AND NOUNS IN COLLOCATIONS *PROPORCIONAR SERVICIO*, *CREAR SISTEMA*, *OFRECER POSIBILIDAD*, *ABRIR ESPACIO*.

proporcionar	provide	*proporcionar, facilitar, surtir, suministrar, dar, transferir*	provide, facilitate, supply, deliver, give, transfer
servicio	service	*servicio, prestación, trabajo, actividad, acto, acción*	service, benefit, work, activity, act, action
crear	create	*crear, realizar, causar*	create, realize, cause
sistema	system	*sistema, método, habilidad, pericia, capacidad, ingenio, poder, cognición, saber, conocimiento*	system, method, ability, skill, capacity, ingenuity, power, cognition, knowledge, wisdom
ofrecer	offer	*ofrecer, proporcionar, facilitar, surtir, suministrar, dar, transferir*	offer, provide, facilitate, supply, deliver, give, transfer
posibilidad	possibility	*posibilidad, expectativa, convicción, creencia, contenido mental, cognición, saber, conocimiento*	possibility, expectation, conviction, belief, mental content, cognition, knowledge, wisdom
abrir	open	*abrir, iniciar, desarrollar, ceder, proporcionar, facilitar, surtir, suministrar, dar, transferir*	open, initiate, develop, yield, provide, facilitate, supply, deliver, give, transfer
espacio	space	*espacio, área, región, lugar, cosa, objeto inanimado, objeto físico, objeto, entidad*	space, area, region, place, thing, inanimate object, physical object, object, entity

ISSN 2395-8618

TABLE XII
FREQUENCY OF CLASSES IN CORPUS.

Class	Frequency (number of occurrences in corpus)
$Oper_1$	63,642
$Real_1$	34,250
$CausFunc_0$	33,830
$CausFunc_1$	46,465
FC	708,159

TABLE XIII
CONFUSION MATRIX FOR LINEARSVC ON TF-IDF OF WORDS IN THE ORIGINAL CONTEXT

		Predicted class				
		$Oper_1$	$Real_1$	Caus $Func_0$	Caus $Func_1$	FC
Known class	$Oper_1$	1	2	1	13	5
	$Real_1$	1	0	0	4	22
	$CausFunc_0$	9	5	4	8	9
	$CausFunc_1$	4	17	1	6	6
	FC	2	8	2	14	4

On the other hand, $Real_1$ was not detected at all by LinearSVC: 22 of 27 $Real_1$ verb-noun pairs were attributed to free verb-noun combinations. As to their contexts, $Real_1$ context included 9,705 unique words, FC context included 9,212 unique words, and 4,632 unique words were shared by both contexts. Therefore, due to such contextual lexical similarity of lexical functions and free verb-noun combinations, other feature representations and computational methods are to be looked for in future.

V. CONCLUSION

In this work we studied semantic and contextual characteristics of four syntagmatic lexical functions in Spanish. Our objective was to determine their potential to allow for automatic detection of lexical functions by supervised learning methods. We defined the latter as a classification task.

For experiments, we chose verb-noun collocations of $Oper_1$, $Real_1$, $CuasFunc_0$, $CausFunc_1$, as well as free verb-noun combinations, having a total of five classes. WordNet Hypernyms and context words in a corpus of Spanish news were used as features in a vector space model. The features were represented as their raw frequency and tf-idf values. Also, we studied the impact of stopwords on lexical function detection, so we experimented with the original corpus and the same corpus after stopwords removal.

Concerning supervised learning methods, we chose eight techniques as implemented in the Scikit-learn package for Python. We reported the classification results in terms of precision, recall, and F1-score. The highest F1-score achieved in the experiments was 0.81 for $CausFunc_1$ using hypernyms. We found that contextual characteristics were not powerful enough to discriminate among subtle semantic differences of lexical functions: the best F1-score of 0.62 for $Real_1$ was achieved by GaussianProcessClassifier using context word counts after stopwords removal.

In future, other representations and methods are to be designed in order to attain higher results on the task of lexical function detection.

ACKNOWLEDGEMENT

The work was done under partial support of Mexican Government: SNI, COFAA-IPN, BEIFIIPN, CONACYT grant A1-S-47854, and SIP-IPN grants 20196021, 20196437.

REFERENCES

[1] A. Zolkovskij and I. Mel'čuk, "On a possible method and tools for semantic synthesis" (in Russian), *Naučno-Texničeskaja Informacija,* vol. 5, pp. 23–28, 1965.

[2] A. Zolkovskij and I. Mel'čuk, "On a system for semantic synthesis. I. Structure of the dictionary" (in Russian), *Naučno-Texničeskaja Informacija,* vol. 11, pp. 48–55, 1966.

[3] A. Zolkovskij and I. Mel'čuk, "On semantic synthesis" (in Russian), *Problemy kibernetiki*, vol. 19, pp. 177–238, 1967. Translated into French as: A. Zholkovskij and I. Mel'chuk, "Sur la synthèse sémantique", *T.A.Informations*, vol. 2, pp. 1–85, 1970.

[4] J. D. Apresjan, "On semantic motivation of lexical functions in collocations" (in Russian), *Voprosy Jazykoznanija*, vol. 5, pp. 3–33, 2008.

[5] J. D. Apresjan, "The theory of lexical functions: An update", in *Proc. Fourth Int. Conf. on Meaning-Text Theory.* Montréal: OLST, 2009, pp. 1–14.

[6] C. Fellbaum, Ed., "WordNet: An electronic lexical database". Cambridge, MA: MIT Press, 1998.

[7] A. Kilgarriff and D. Tugwell, "Sketching words", in *Lexicography and Natural Language Processing: A Festschrift in Honour of B. T. S. Atkins*, M.-H. Corréard, Ed. Euralex, 2002, pp. 125–137.

[8] A. Ferraresi, E. Zanchetta, M. Baroni and S. Bernardini, "Introducing and evaluating ukWaC, a very large web-derived corpus of English", in *Proc. 4th Web as Corpus Workshop (WAC-4) Can We Beat Google*, 2008, pp. 47–54.

[9] I. A. Mel'čuk, "Lexical functions: A tool for the description of lexical relations in a lexicon", in *Lexical Functions in Lexicography and Natural Language Processing*, L. Wanner, Ed. Amsterdam, Philadelphia, PA: Benjamins Academic Publishers, 1996, pp. 37–102.

[10] I. A. Mel'čuk, "Collocations and lexical functions", in *Phraseology. Theory, Analysis, and Applications*, A. P. Cowie, Ed. Oxford: Clarendon Press, 1998, pp. 25–53.

[11] D. Heylen, K. G. Maxwell and M. Verhagen, "Lexical functions and machine translation", in *Proc. 15th Conf. on Computational Linguistics*, Kyoto, 1994, pp. 1240–1244.

[12] T. Fontenelle, "Ergativity, collocations and lexical functions", in *Proc. EUROLEX*, M. Gellerstam et al., Eds, 1996, pp. 209–222.

[13] S. H. Song, "Zur Korrespondenz der NV-Kollokationen im Deutschen und Koreanischen", 언어학, vol. 44, pp. 37–57, 2006.

[14] A. Gelbukh and O. Kolesnikova, "Supervised learning for semantic classification of Spanish collocations", in: *Advances in Pattern Recognition*, Springer Berlin Heidelberg, 2010, pp. 362–371. The lexical resource (list of Spanish lexical functions) can be downloaded at http://www.gelbukh.com/ lexical-functions/

[15] G. A. Miller, C. Leacock, R. Tengi and R. T. Bunker, "A semantic concordance", in *Proc. Workshop on Human Language Technology*, Association for Computational Linguistics, 1993, pp. 303–308.

[16] A. Gonzalez-Agirre, E. Laparra, G. Rigau and B. C. Donostia, "Multilingual central repository version 3.0: Upgrading a very large

ISSN 2395-8618

lexical knowledge base", in *Proc. GWC 2012 6th International Global Wordnet Conference*, 2012, pp. 118–125.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg and J. Vanderplas, "Scikit-learn: Machine learning in Python", *J. of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] Z. S. Harris, "Distributional structure", *Word*, vol. 10, no. 23, pp. 146–162, 1954.

[19] C. Huang, Q. Wang, D. Yang and F. Xu, "Topic mining of tourist attractions based on a seasonal context aware LDA model", *Intelligent Data Analysis*, vol. 22, no. 2, pp. 383–405, 2018.

[20] G. Feng, S. Li, T. Sun and B. Zhang, "A probabilistic model derived term weighting scheme for text classification", *Pattern Recognition Letters*, vol. 110, pp. 23–29, 2018.

[21] D. Ustalov, D. Teslenko, A. Panchenko, M. Chernoskutov, C. Biemann and S. P. Ponzetto, "An unsupervised word sense disambiguation system for under-resourced languages", *arXiv preprint arXiv:1804.10686*, 2018.

[22] H. Saif, Y. He, M. Fernandez and H. Alani, "Contextual semantics for sentiment analysis of Twitter", *Information Processing & Management*, vol. 52, no. 1, pp. 5–19, 2016.

[23] B, Alhijawi, S. Hriez and A. Awajan, "Text-based Authorship Identification-A survey", in *Proc. Fifth Int. Symposium on Innovation in Information and Communication Technology*, IEEE, 2018, pp. 1–7.

[24] K. A. Overmann, and L. Malafouris, "Situated cognition," in *The International Encyclopedia of Anthropology*, 2018, pp. 1–8.

[25] F. Engelmann, S. Granlund, J. Kolak, M. Szreder, B. Ambridge, J. Pine, A. Theakston, and E. Lieven, "How the input shapes the acquisition of verb morphology: Elicited production and computational modelling in two highly inflected languages," *Cognitive Psychology*, vol. 110, pp. 30–69, 2019.

[26] R. H. Baayen, Y. Y. Chuang, E. Shafaei-Bajestan, and J. P. Blevins, "The discriminative lexicon: A unified computational model for the lexicon and lexical processing in comprehension and production grounded not in (de) composition but in linear discriminative learning," *Complexity*, 2019.