Master thesis on Interactive Intelligent Systems
Universitat Pompeu Fabra

Automatic Identification and Classification of Collocations with Word Embeddings

Jessica Perez Guijarro

Supervisor: Leo Wanner

Co-Supervisor: Luis Espinosa

July 2019

upf. Universitat
Pompeu Fabra
Barcelona

Master thesis on Interactive Intelligent Systems
Universitat Pompeu Fabra

Automatic Identification and Classification of Collocations with Word Embeddings

Jessica Perez Guijarro

Supervisor: Leo Wanner

Co-Supervisor: Luis Espinosa

July 2019

Contents

List of Figures **1**

List of Tables **2**

1 Introduction **3**

1.1 Objectives 4

1.2 Structure of the Thesis 5

2 Fundamentals **6**

2.1 Collocations 6

2.2 Theoretical perspective of collocations 6

2.2.1 The Firthian tradition 7

2.2.2 The Lexicological perspective 7

2.3 Collocations classification 8

2.4 Word Vectorisation Techniques 9

2.4.1 Word Embeddings 9

2.4.2 Relation Vectors 10

3 State of the Art **12**

4 Methods **14**

4.1 Overview of the data process 14

4.2 Preprocessing 14

4.3 Classification model 16

4.4 Evaluation metrics 17

5 Results 19

5.1 Data 19

5.2 Word vector approach 21

5.2.1 Experimental setup 21

5.2.2 Results 22

5.3 Relation vector approach 23

5.3.1 Experimental setup 23

5.3.2 Results 24

5.4 Results discussion 24

6 Conclusions and future work 27

6.1 Conclusions 27

6.2 Future work 28

Bibliography 29

A Word2Vec Approach: Extended Lexical Function Dataset Results 32

B Word2Vec Approach: Aggregated Lexical Function Dataset Results 35

C Relational Vector Approach: Extended Lexical Function Dataset Results 37

D Relational Vector Approach: Aggregated Lexical Function Dataset Results 40

I am never stagnant; I rise from my worst disasters, I turn, I change.

Virginia Woolf, The Waves

Acknowledgement

I would like to express my sincere gratitude to:

- My supervisor: Leo Wanner
- My co-supervisor: Luis Espinosa
- My Soulmate: You know, I know, Who else needs to know?

Abstract

Collocations are very important to the field of Natural Language Processing because being able to modeling, processing and extract them properly can be advantageous for improving the quality of multiple NLP applications. Hence, the main objective of this thesis is to improve the current baseline results for collocation identification and lexical function classification with the supervised technique Support Vector Machine (SVM). In order to do that, different experiments are conducted where several SVM multiclass classifiers are trained based on three main factors. The first one is the chosen dataset to perform the experiment. The second one is the vector operation applied to the base and collocate vectors conforming each one of the training samples. These operations are: multiplication, addition, subtraction, and concatenation. Finally, the type of feature used to train the classifiers being these: word embeddings (word2vec) or word embeddings plus relation vectors.

Keywords: Collocation; Lexical Function; Classification; Word Embeddings

List of Figures

1 Results of classification [1] 4

2 Diagram illustrating the CBOW model, inspired by Mikolov [2] . . . 10
3 Diagram illustrating the Skip-Gram model, inspired by Mikolov [2] . . 11

4 Data Workflow diagram. 15
5 Spacy Parser inputs and outputs. 16

List of Tables

1 Lexical Functions Examples 8

2 Extended Lexical Functions dataset. 20
3 Proposed Generic LF Aggregation. 21
4 Aggregated Lexical Functions dataset. 21
5 Extended Lexical Functions Dataset Results for English. 22
6 Extended Lexical Functions Dataset Results for Spanish. 23
7 Aggregated Lexical Functions Dataset Results for English. 23
8 Aggregated Lexical Functions Dataset Results for Spanish. 23
9 Extended Lexical Functions Dataset Results for English. 24
10 Extended Lexical Functions Dataset Results for Spanish. 24
11 Aggregated Lexical Functions Dataset Results for English. 25
12 Aggregated Lexical Functions Dataset Results for Spanish. 25

Chapter 1

Introduction

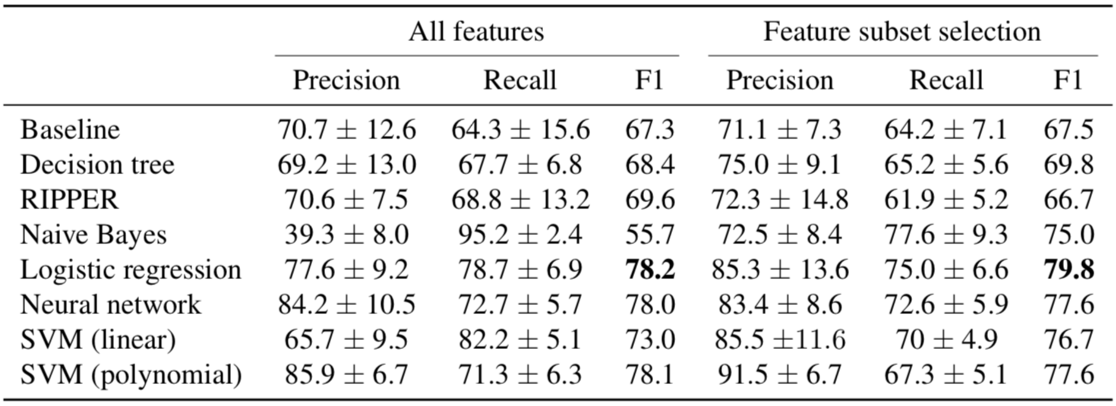
Give a walk is a literal translation of *dar un paseo* in Spanish, but despite being identical grammatical structures that apparently have the same lexical meaning the truth is that the English sentence is not correct. The reason why this phenomenon is because we are dealing with collocations.

A collocation is defined as a pair of words that co-occur frequently and where one of the words depends on the other to express its lexical meaning. Collocations are unpredictable lexical items present in a language that are hard to guess for individuals that are non-native speakers of that language. Then, it is not surprising that collocations are one of the many challenges that face the field of Natural Language Processing.

Modeling collocations properly can be useful for improving the quality of multiple NLP applications. Specifically, be able to process and extract collocations from corpora could be very useful for Natural Language Processing tasks as for example text summarization [3], text generation [3] and text understanding [3]. From a pedagogic point of view, knowing the inner mechanics of how collocations work could be useful for students learning a foreign language in order to speak and write the studied language in a more natural way. On the other hand, teachers can also benefit from this collocation knowledge because they could design more effective language teaching materials and methods for their students [3][4].

1.1 Objectives

The classification task of collocations with Machine Learning techniques has been addressed by different approaches. For instance, [5] carry out different experiments to evaluate the performance of three Machine Learning Techniques being these: Neural Networks, Naïve Bayes and Decision Trees, over the task of Lexical Functions[1] classification. Another study evaluates different classification algorithms for collocations extraction in Croatian [1]. In Figure 1, the obtained results for each algorithm are shown.



	All features			Feature subset selection		
	Precision	Recall	F1	Precision	Recall	F1
Baseline	70.7 ± 12.6	64.3 ± 15.6	67.3	71.1 ± 7.3	64.2 ± 7.1	67.5
Decision tree	69.2 ± 13.0	67.7 ± 6.8	68.4	75.0 ± 9.1	65.2 ± 5.6	69.8
RIPPER	70.6 ± 7.5	68.8 ± 13.2	69.6	72.3 ± 14.8	61.9 ± 5.2	66.7
Naive Bayes	39.3 ± 8.0	95.2 ± 2.4	55.7	72.5 ± 8.4	77.6 ± 9.3	75.0
Logistic regression	77.6 ± 9.2	78.7 ± 6.9	**78.2**	85.3 ± 13.6	75.0 ± 6.6	**79.8**
Neural network	84.2 ± 10.5	72.7 ± 5.7	78.0	83.4 ± 8.6	72.6 ± 5.9	77.6
SVM (linear)	65.7 ± 9.5	82.2 ± 5.1	73.0	85.5 ±11.6	70 ± 4.9	76.7
SVM (polynomial)	85.9 ± 6.7	71.3 ± 6.3	78.1	91.5 ± 6.7	67.3 ± 5.1	77.6

Figure 1: Results of classification [1]

However, there is not a system capable of automatically discriminate collocations of a corpus.

Therefore, the main goal of this thesis is to improve the current baseline results for collocation identification and lexical function classification with supervised machine Learning methods. Specifically, we train several Support Vector Machine (SVM) multiclass classifiers with two different types of features, word embeddings and relation vectors, and compare the performance results of each one of them. By using relation vector, we go one step further considering as a combined unit the elements of a collocation. Hypothetically, we expect that those classifiers trained with relation vectors have better performance than those trained with word embeddings

[1] As explain in section 2.3, a lexical function is an abstract semantic relation between the elements that form a collocation

thus, in both cases is expected that lexical functions like *Antibon* or *Antimagn* are more difficult to predict than others like *Bon* or *Magn*. Moreover, we do not expect any significant difference between the results of the classifiers of different datasets or languages[2].

1.2 Structure of the Thesis

The present section is an introduction to the thesis and the importance of investigate the field of collocations. The rest of this thesis is structured as follows: in Section 2, the theoretical background related to collocations is presented. In Section 3, are presented the different methods that are used nowadays to abroad approach the problem. In Section 4, are described the preprocessing steps performed over the data and the architecture of the classification model. In Section 5, the different experiments carried out as well as the discussion of the obtained results are presented. Finally, the conclusions and future improvements related to the field are presented in Section 6.

Furthermore, an Appendix has been added with all the detailed results obtained for each one of the performed experiments.

[2] The languages considered for this thesis are English and Spanish

Chapter 2

Fundamentals

2.1 Collocations

Collocations are binary restricted idiosyncratic co-occurrences of lexical items (Hausmann 1984; Cowie 1994; Mel'čuk 1996) where the item known as base is freely chosen by the speaker and the other item known as collocate depends on the base and on the concrete meaning that the speaker wants to express with the combination of both elements (Garcia et al. 2017) [6]. For example, the collocation *strong tea* holds an intensity semantic relation between *strong* (the collocate) and *tea*, the base.

Collocations can be considered the nucleus of a language's lexicon and can be found transversally in most languages Mel'čuk (1998) [7] [8]. Therefore they are a key aspect in multiple NLP applications such as Machine Translation[3], Natural Language Generation[3], Word Sense Disambiguation[3] or Second Language Learning [3][9]. However, collocations pose many challenges due to the fact that it is very difficult to predict the combination of collocate - base for expressing a certain meaning.

2.2 Theoretical perspective of collocations

In the next sections, the two main perspectives related to the definition of collocations as well as the most important authors of each tradition are explained.

2.2.1 The Firthian tradition

The Firthian tradition is known for being the most prominent statistical tradition, which defines collocations as language patterns that appear on texts, understanding language pattern as the relation that is formed between those lexical items that have the highest probability to be associated. The reference authors of this tradition are Firth, Halliday, and Sinclair [10][11] [12] [3].

Firth's first reference to the importance of collocations can be found in the paper 'Modes of meaning' [10]. According to Firth, the relation between a word and its context is deterministic for the meaning of such a word. Specifically, he defines collocations as words that co-occur frequently in a context.

Following the same line of thought, Halliday describes, in his first paper related to this topic, collocations in terms of frequencies and co-occurrences of words:

> *[..] highly abstract relation of structure, in which the value of an element depends on complex factors in no sense reducible to simple sequence, lexis seems to require the recognition merely of linear co-occurrence together with some measure of significant proximity, either a scale or at least a cut-off point. It is this syntagmatic relation which is referred to as 'collocation'.* [11]

Sinclair defines in his paper 'Beginning the study of lexis' [12] a collocation in terms of '*tendencies of items to collocate with each other*'. Moreover, he developed several methods with the aim to study the nature of collocations. Specifically, in his work *The OSTI Report* he suggests a method that evaluates if a combination of a pair of words forms a collocation or not based on the type of pattern showing in the words that co-occur, the frequency of the words appearing together, the frequency of appearing by itself and finally, if there is a relation between the two items.

2.2.2 The Lexicological perspective

Unlike the statistical approach where collocations are defined as words with high frequency of co-occurrence, the lexicological tradition, whose reference authors are

Hausmann and Mel'čuk, defines a collocation as a binary restricted idiosyncratic co-occurrence of lexical items where one of the items, called base, is freely chosen by the speaker while the other one, called collocate, depends completely on the base to be able to express a certain meaning (Hausmann 1985) [13]. Specifically, Mel'čuk (Mel'čuk 1998) [7] (Mel'čuk 1996) [14] proposes in his work 'Meaning-Text Theory' (MTT) that collocations are binary combinations of words on which one is dependent on the other.

2.3 Collocations classification

The Meaning Text Theory [14] is the framework that introduced the concept of Lexical Function (LF), an abstract semantic relation between the elements that form a collocation.

Lexical Functions can be classified into Simple Lexical Functions and Complex Lexical Functions. The first one corresponds to those LFs that express one meaning as for example, '*heavy rain*' **Magn** express intensity. On the other hand, Complex LF are those formed by the combination of simple LF as for example, '*scattered applause*' **AntiMagn** express the negation of intense [not $(\sim Anti) intensity (\sim Magn)$].

In the following Table, more examples of Lexical Functions are shown. Note that all of them follow the same notation, the LF's name, a Latin abbreviation referred to the meaning of the collocation, followed by a number, referred to the *actant*, *a complement of the word, describing a necessary participants of the situation referred to by the word* [15].

Table 1: Lexical Functions Examples

Name	Meaning	Base	Collocate
Magn	Lat. magnus, 'big, great', intensity	rain	heavy
Bon	Lat. bonus 'good'	bread	fresh
Oper1	Lat. operari '[to] do, carry out'	resistance	meet
Real1	Lat. realis 'real'	hint	take

2.4 Word Vectorisation Techniques

Words can be represented as vectors in a multidimensional space where each word could have a different dimension. In order to transform a word into a vector is necessary to define a size window of words around the word to be transformed.

In the following, we show the two main techniques related to word vectors used for carried out the classification experiments.

2.4.1 Word Embeddings

Word Embeddings is a feature learning technique that maps words to vectors. Concretely, this method assumes that words are distributed in the corpora in such a way that words with similar meaning will share similar contexts [16]. There are different models to construct those word vectors; we have chosen the word2vec model of Mikolov [17] [2] [18] to the scope of this thesis.

There are two different architectures for building the word2vec model: the Continuous Bag of Words or CBOW and the Skip - gram, both based on Neural Networks techniques.

The CBOW model takes as an input the context of a target word and then average the vectors of each context word to form the Hidden Layer of the Neural Network with the objective of predicting the target word related to the input context (Figure 2).

On the other hand, the Skip gram model tries to achieve the contrary of CBOW model, meaning that by given as an input a target word like Fox, the model tries to predict its surrounding context words in a text (Figure 3).

In this case, we chose the Skip-gram model architecture to train the Word Embeddings model used in our experiments.

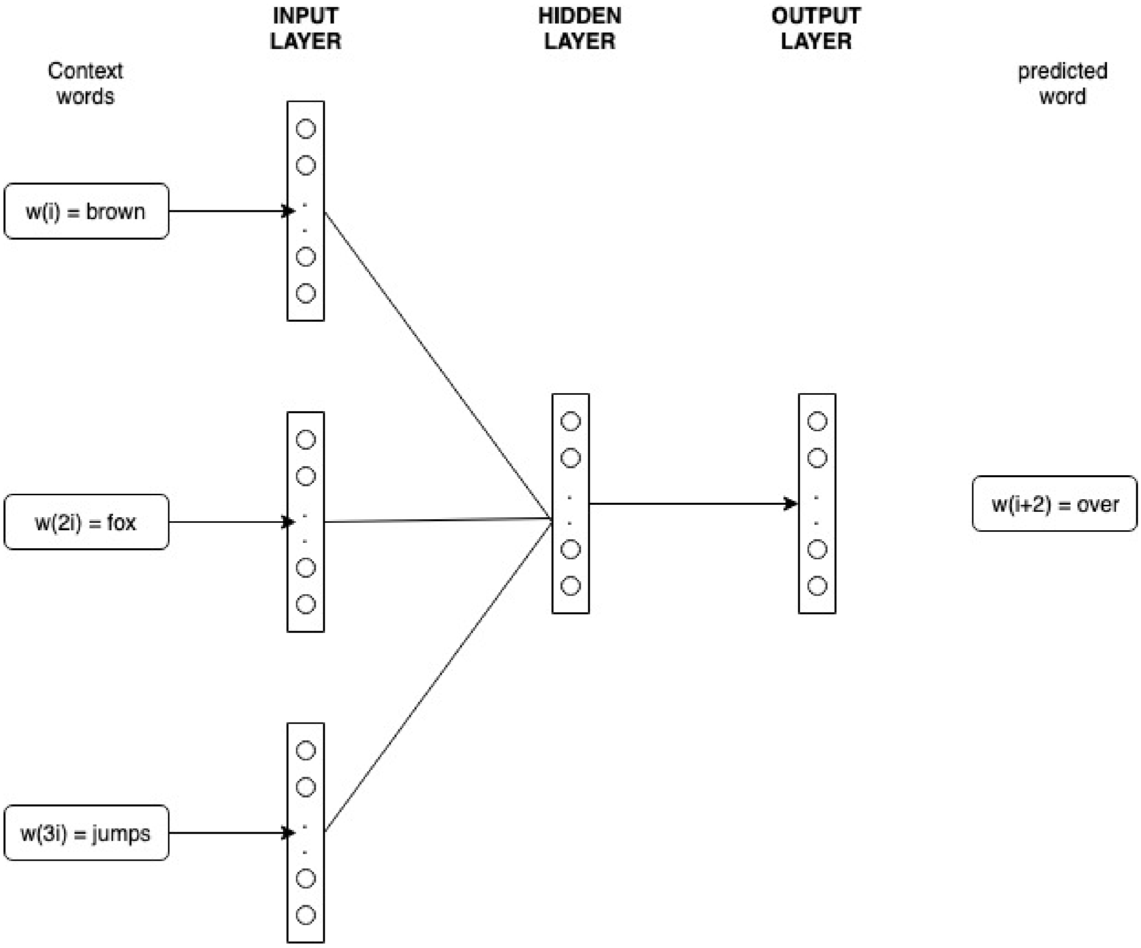


Figure 2: Diagram illustrating the CBOW model, inspired by Mikolov [2]

2.4.2 Relation Vectors

Unlike Word Embeddings that try to learn the most descriptive features for each word, Relation Vectors try to learn the relations between the words that appear in a context being considered a form of complementary representations to word vectors.

For this thesis, we used the SeVen relation vectors [19] that represent all the relationships between words in a graph form. Each node in the graph is a word, and each edge encodes a relation between two nodes. Moreover, edges add a label component representing the vector encoding of such relation.

Besides, in order to compute the relation vectors, the average of word vectors of a target input pair of words (a,b) in the sentence that appears are computed. Specifically, six different vectors for each pair in a sentence were created: the first one is

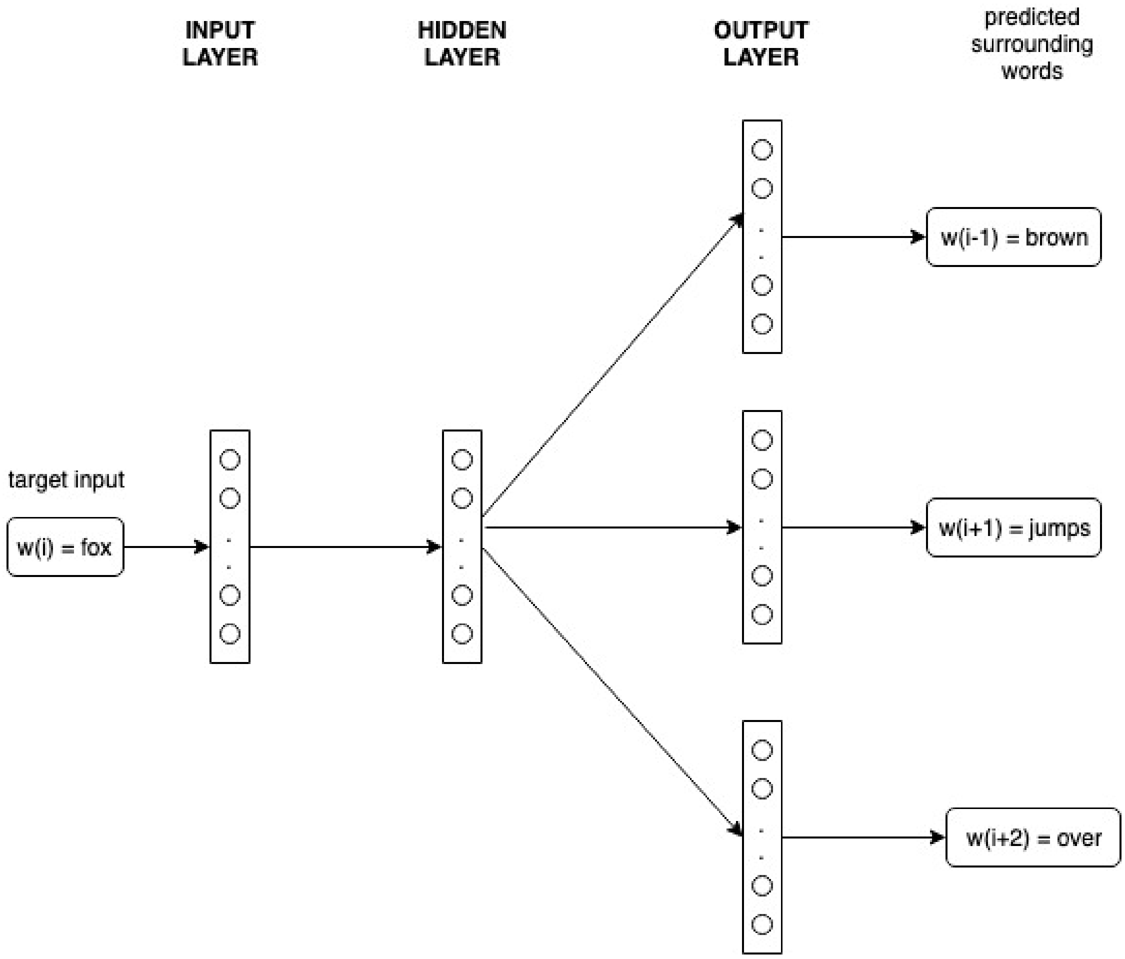


Figure 3: Diagram illustrating the Skip-Gram model, inspired by Mikolov [2]

formed by the words that appear before a; the second one, is formed by words that are situated between a and b and finally, the third vector is constructed by those words that appear after b. Next, the average of these three vectors to obtain the relation vector of the pair of words (a,b) were computed.

Chapter 3

State of the Art

The problem of collocation classification has been aimed for different authors over the years. For instance, (Wanner et al. 2015) [20]. tries to classify the collocate item of a collocation depending on its context for Spanish collocations with the syntactic pattern of Verb-Noun and Noun - Adjective. In order to that, they trained two classifiers, an SVM and Naïve Bayes, with the following semantic features related to the context of the collocation: lexical features, POS - features, morphological features, and syntactic dependency features. The authors conclude that the use of semantic features is not significant to raise the accuracy of the classifier; however, the use of lexical features improves the discrimination of verb-noun collocations. Besides, the performance of the SVM is better than the Naïve Bayes because the first one captures the correlation between feature while the second one treats the features as independent.

Following the same line of research, (Wanner et al. 2019) [5] compare the performance of four machine learning techniques (Nearest Neighbor technique, Naïve Bayesian network, Tree-Augmented Network Classification and ID3-algorithm for decision tree) aiming collocation classification. Besides, (Gelbukh et al. 2010) [21] the authors train and compare the performance of several supervised classifiers whose task is to classify new collocations according to their Lexical Function. The best classifier achieved the average F-measure of 74%.

Furthermore, the classification of collocations is not the only task that has been highlighted in research: the task of lexical/semantic relations classification has been equally important. Regarding this last task, (Girju et al. 2007) [22] created a benchmark data as an attempt to standardize the data used for the evaluation of different experiments for semantic relation classification. Following the same line we have, (Hendrickx et al. 2010) [23] where the authors perform the extraction and recognition of semantic relations between pairs of nominals from an English text with Supervised Methods, one binary classifier per semantic relation.

Besides, it is important for this thesis to introduce the concept of relation vectors, a complementary vector representation to word vectors that try to learn the relation between a pair of words. In recent years, some studies have been published about this topic, for instance (Espinosa et al. 2018) [19] presents the task of learning semantic relations between words (not necessarily collocations) using the current state of the art, a word vector representation in the continuous vector space. Specifically, the followed strategy to learn those relations is based on averaging the word vectors of the surrounding tenth words of the target pair in the sentences that these pair appear.

So, in the last few years, some works related to the collocation and relation classification tasks using Machine Learning techniques for predicting the inherent abstract meaning or Lexical Function of unseen collocations have been developed. Hence, following the same research line of those works mentioned above, we choose Supervised Learning Techniques such as SVM multiclass classifier combined with state-of-the-art word embedding vectors from Mikolov (2013) and relation vectors in order to improve the baseline results for these text classification tasks.

Chapter 4

Methods

The architecture of the system, the type of Machine Learning technique used for the classification task as well as the evaluation metrics used to evaluate the obtained results are presented.

4.1 Overview of the data process

Figure 4, shows the data processing flow diagram in which we can see the different phases related to the data transformation, its retrieval until its transformation to training sample.

4.2 Preprocessing

First of all, we downloaded the latest Wikipedia dump for English and Spanish languages from `https://dumps.wikimedia.org/backup-index.html`. Then, we proceeded to clean the corpus of HTML markups, tags and other symbols. In order to do that we used the python script from Wesley Baugh that can be found in `https://github.com/bwbaugh/wikipedia-extractor`.

Finally, using the python library Spacy we parsed the two corpora in CONLL format, reflects information about the syntactic dependencies of each sentence token. This process is shown in Figure 5. Specifically, the parser input is a single sentence from

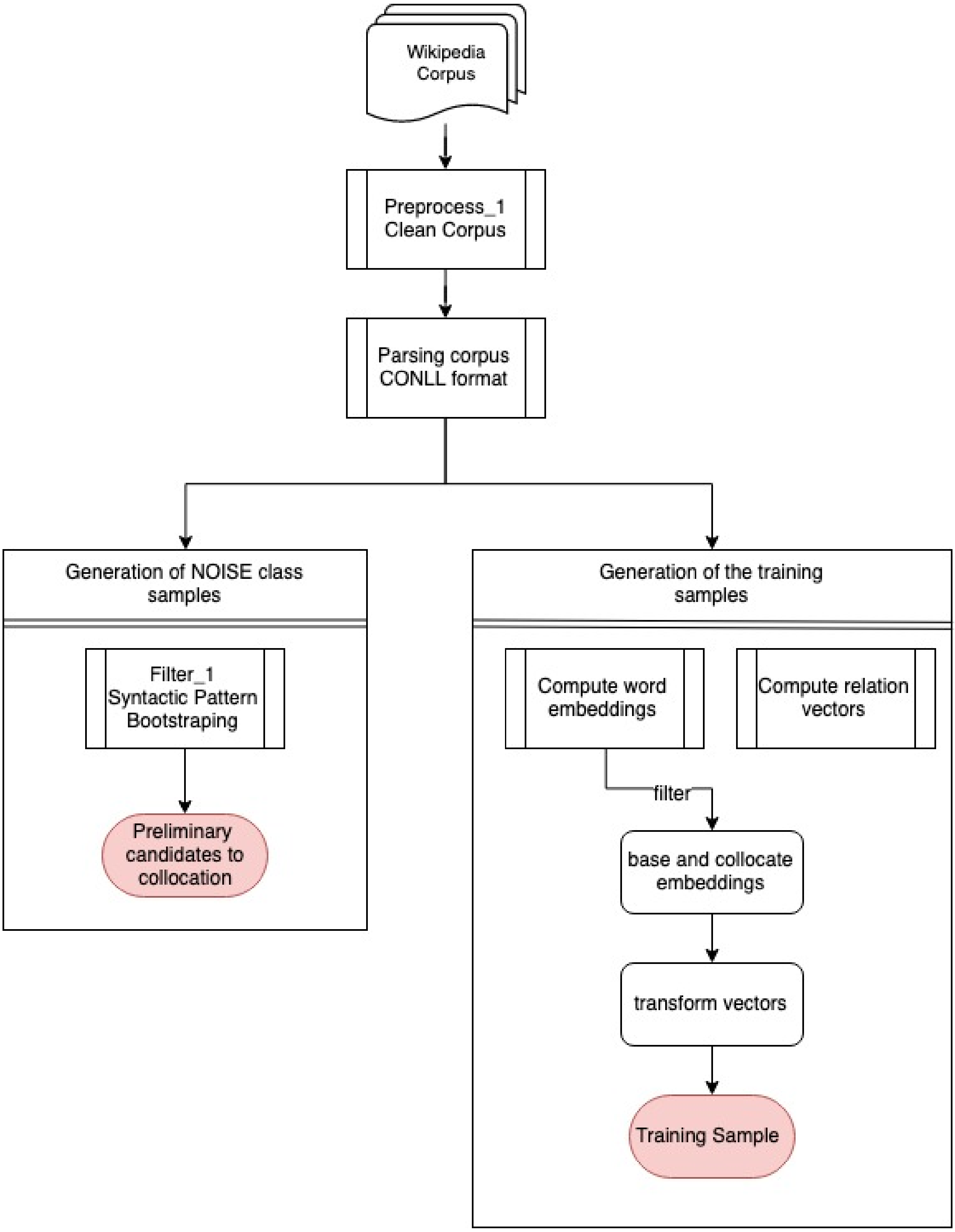

Wikipedia Corpus
Preprocess_1 Clean Corpus
Parsing corpus CONLL format
Generation of NOISE class samples
Filter_1 Syntactic Pattern Bootstraping
Preliminary candidates to collocation
Generation of the training samples
Compute word embeddings
Compute relation vectors
filter
base and collocate embeddings
transform vectors
Training Sample

Figure 4: Data Workflow diagram.

the corpus whilst the parser output is a sentence with morphological information as well as syntactic dependencies information of each word.

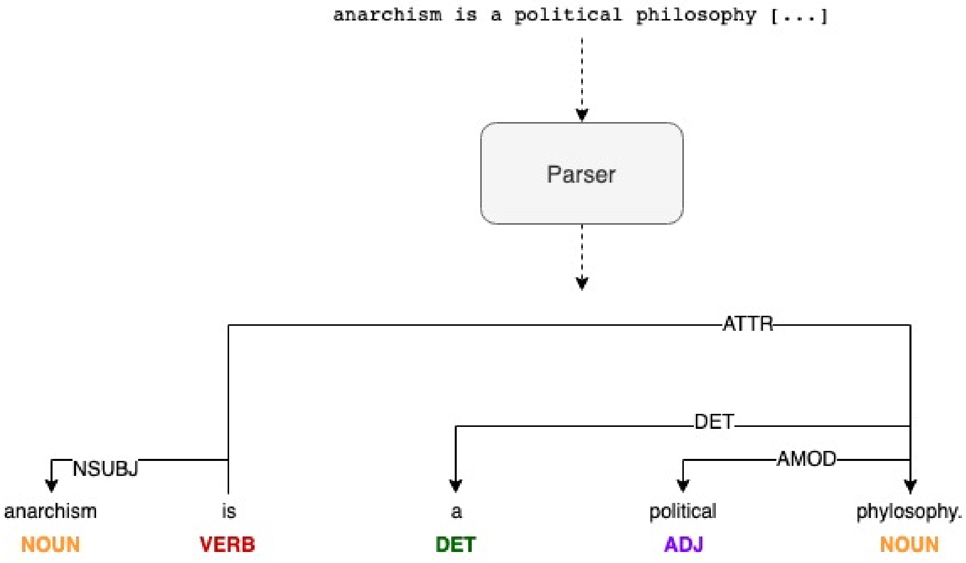


Figure 5: Spacy Parser inputs and outputs.

4.3 Classification model

The task of collocation identification and classification has been considered a machine learning task, the Support Vector Machine the preferred method for achieving it. According to Joachim [24] SVM is good for text classification tasks for several reasons:

- SVM can deal with high dimensional input space. In other words, it can handle large feature spaces.
- Few irrelevant features: in text categorization there are few features that are irrelevant and '*that a good classifier should combine many features and that aggressive feature selection may result in a loss of information*' [24].
- Document Vectors are sparse, meaning that only a few entries of the vector representing a document are not zero.
- Text categorization problems are linearly separable.

For all these reasons, a Support Vector Machine classifier has been chosen to carry out the experiments.

Specifically, an SVM multiclass classifier with linear kernel has been trained following the one-against-all strategy, where for each class in the dataset one binary classifier is trained, with the instances of the current class as positive examples and the instances of other classes as negative. To classify an unseen instance we choose that classifier whose probability estimation (eq. 4.1) is higher than that of others classifiers.

$$\hat{y} = argmax f_k(x) \tag{4.1}$$

where f corresponts to a list of classifiers trained with k samples.

4.4 Evaluation metrics

The following metrics have been chosen because besides commonly used in the evaluation of classification tasks, they also have good results discriminating the optimal classifier during the training phase [25].

The first metric is *accuracy*, also known as classification accuracy. This metric measures the ratio of correct predictions over the total number of samples (eq. 4.2).

$$Accuracy = \frac{Number\, of\, correct\, predictions}{Total\, number\, of\, predictions} \tag{4.2}$$

The second one is *precision*, also known as positive predictive value. This metric measures the ability of the classifier of detecting the important instances of the task. (eq. 4.3)

$$Precision = \frac{True\, positive}{True\, positive + False\, positive} \tag{4.3}$$

The third one is *recall*, also known as sensitivity. This metric measures the ability of the classifier of detecting all the important instances of the dataset. (eq. 4.4)

$$Recall = \frac{True\, positive}{True\, positive + False\, negative} \tag{4.4}$$

The last is the *balanced F1- Score* metric, interpreted as the weighted average of precision and recall. (eq. 4.5)

$$Balanced\,F1\,score = 2 \cdot \frac{precision \cdot recall}{Precision + recall} \tag{4.5}$$

Chapter 5

Results

The scope of this thesis, two independent experiments have been carried out, both of them with the same goal: the automatic identification and posterior classification of collocations by their corresponding semantic labels according to a previously defined topology. This task is considered a machine learning-based classification task hence, an SVM multiclass classifier has been trained for each one of the experiments.

In the following subsections the datasets, the experiments set up and the obtained results are presented.

5.1 Data

The data is composed of a group of collocations labeled by its corresponding lexical function. Specifically, each data sample consists of a pair of words where the first corresponds to the base and the second to the collocate. Because the different types of collocations are not equally distributed in the language, the data we are working with is imbalanced and thus each class has a different number of instances.

There are two different datasets for each language. The first one, called *Extended Lexical Functions dataset*, is composed of 9,464 instances for English and 2,233 for Spanish, distributed in 9 Lexical Functions plus an extra class called noise that contains the negative examples.

In the following Table the distribution of instances per each category and language is shown.

Table 2: Extended Lexical Functions dataset.

LF	**Instances in English**	**Instances in Spanish**
Antibon	168	70
Antimagn	239	70
Bon	142	94
Magn	2,491	213
Causfunc0	150	100
Liqufunc0	118	30
Oper1	1,037	418
Real1	316	111
Sing	72	16
Noise	4,731	1,111

The second dataset, called *Aggregated Lexical Functions dataset*, is composed of 8,508 total instances for English and 2,159 for Spanish. In this case, the samples are distributed in 5 categories, each of them representing the aggregation of similar lexical functions into a unique Generic LF [26].

The classes that form Generic LF are shown in Table 2, and the distribution of instances per class and language shown in Table 3.

Finally, the noise class contains random pairs of words that were generated following strategies:

1. Compute the cartesian product over random pairs of words from the original data [27].

2. Select random pairs of words extracted from the previous preprocessed Wikipedia Corpus with the following syntactic structure: verb + dobj and noun + amod.

Table 3: Proposed Generic LF Aggregation.

Generic LF	**Aggregated LF**
Cause	SCausAntiFact1, CausFact0, Magn++Caus2Func, Involv, SCausFunc1, SCausFunc2, CausLabor21, Caus1Fact0, Labreal12, Labor21, CausAntiFunc2, CausFunc1, CausPred, Caus3Func1, Caus1Func1, ScausAntiFunc0, CausFunc2, Caus4Func1, LiquFact0, Labor12, causreal
Experiment	Func2, Func0, CausAntiFunc1, CausAntiFunc0, SnonFunc0, SFunFunc0, SOper3, SOper1, SOper0, Oper0,Oper3, FincFunc0
Intensity	AntiMagn, Magn, Bon, Centr, AntiBon
Manifest	SFact0, Caus1Func3, Caus1Func0, Fact0, Fact1, SLiquFunc0, LiquFunc0, LiquFunc1, Manif, Liqu1Func1, AntiReal3, SAntiReal3, SReal3, IncepReal1, FinReal1, AntiReal2, Gener, SManif, Real3, Son, LiquReal
Phase	IncepOper2, SIncepFunc2, SIncepFunc0, PreparReal3, PreparAntiReal1, Degrad, PreparLiquFunc1, PreparAntiFact0, IncepFunc0, IncepFunc3, IncepFunc2, Magn+IncepOper1, PreparLiquFunc0, PreparFunc0, PreparOper3, PreparOper1, IncepPred, CausPreparFunc1, CausPreparFunc0, CausPreparFunc3, LiquPreparFunc0, SPreparAntiReal1, IncepFact0

Table 4: Aggregated Lexical Functions dataset.

LF	**Instances in English**	**Instances in Spanish**
Cause	150	165
Experiment	1,037	424
Intensity	2,792	125
Manifest	316	132
Phase	118	234
Noise	4,095	1,079

5.2 Word vector approach

5.2.1 Experimental setup

This experiment, compare the performance of different classifiers in relation with the basic vector operations[1] computed over the word vectors.

[1] The basis vector operations are: multiplication, subtraction and addition. For the aim of these experiments, we considered concatenation as a vector operation.

First of all, we compute the word vectors (word2vec) of the corpus using the skip gram model where each vector has a size of 100 features. Then, we select vectors corresponding to each pair of words that are present on the dataset.

Next, we trained one classifier for each one of the basic vector operations. The training samples are generated by performing the corresponding operation over the base and the collocate vectors. For example, for the word vectors of the instance *prediction*-b *make*-c, where *b* stands for base and *c* for collocate, from the category experimenter, the following representations of the training sample are generated:

- multiplication operation: [$\vec{b}$ * $\vec{c}$] Label
- subtraction operation: [$\vec{b}$ - $\vec{c}$] Label
- addition operation: [$\vec{b} + \vec{c}$] Label
- concatenation operation: [($\vec{b}$, $\vec{c}$)] Label

where $\vec{b}$, $\vec{c}$ are vector representation of a word and LF is the label corresponding to one of the Lexical Function in the dataset, in case that the input is a collocation, or the noise label otherwise.

The obtained results are compared in section 5.4.

5.2.2 Results

In Tables 5 and 6, the main result obtained of the experiment performed over Extended Lexical Functions dataset are presented. In the same way, Tables 7 and 8 present the obtained results for Aggregated Lexical Functions dataset.

Table 5: Extended Lexical Functions Dataset Results for English.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.75**	**0.77**	**0.75**	**0.72**
Subtraction	0.71	0.71	0.71	0.67
Addition	0.71	0.71	0.71	0.67
Multiplication	0.57	0.60	0.57	0.48

Further results can be found in Appendix Results.

Table 6: Extended Lexical Functions Dataset Results for Spanish.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.62**	**0.61**	**0.62**	**0.54**
Subtraction	0.59	0.56	0.59	0.50
Addition	0.57	0.56	0.57	0.46
Multiplication	0.49	0.26	0.49	0.32

Table 7: Aggregated Lexical Functions Dataset Results for English.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.77**	**0.74**	**0.77**	**0.75**
Subtraction	0.74	0.73	0.74	0.72
Addition	0.73	0.71	0.73	0.71
Multiplication	0.59	0.62	0.59	0.52

Table 8: Aggregated Lexical Functions Dataset Results for Spanish.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.61**	0.65	**0.61**	**0.55**
Subtraction	0.60	**0.68**	0.60	0.52
Addition	0.57	0.57	0.57	0.48
Multiplication	0.50	0.25	0.50	0.33

5.3 Relation vector approach

5.3.1 Experimental setup

This experiment is based on the previous experiment, with the only difference being the representation of the input training samples with which the classifier are trained.

In this case, a semantic relation vector of the current collocation is concatenated to the resulting vector of the applied operation. This vector provides information about the context of the collocation that hypothetically allows learning relation between base and collocate that will help to identify which type of collocation is it. So for example, for the word vectors of the instance *prediction*-b *make*-c from the category *experimenter* we have the following training sample for each operation:

- multiplication operation: [$\vec{b}$ * $\vec{c}$, SeVen[b,c]] Label
- subtraction operation: [$\vec{b}$ - $\vec{c}$, SeVen[b,c]] Label

- addition operation: [$\vec{b} + \vec{c}$, SeVen[b,c]] Label
- concatenation operation: [($\vec{b}$, $\vec{c}$), SeVen[b,c]] Label

where $\vec{b}$,$\vec{c}$ are vector representation of a word, SeVen[b,c] [19] is the relation vector of the base and the collocate and LF is the label corresponding to one of the Lexical Function on the dataset, in case that the input is a collocation, or the noise label otherwise.

As in the previous experiment, this is performed twice per language and dataset.

5.3.2 Results

In Tables 9 and 10, the main results obtained for the experiment performed over Extended Lexical Functions dataset are presented. Alike, Tables 11 and 12 present the obtained results for Aggregated Lexical Functions dataset. Further results can be found in Appendix Result.

Table 9: Extended Lexical Functions Dataset Results for English.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.89**	**0.88**	**0.89**	**0.87**
Subtraction	0.87	0.84	0.87	0.83
Addition	0.88	0.87	0.88	0.85
Multiplication	0.79	0.74	0.79	0.74

Table 10: Extended Lexical Functions Dataset Results for Spanish.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.75**	**0.73**	**0.75**	**0.69**
Subtraction	0.73	0.70	0.73	0.66
Addition	0.73	0.70	0.73	0.66
Multiplication	0.60	0.53	0.60	0.49

5.4 Results discussion

The obtained results of the different performed experiments can be analysed based on three main factors that influence the performance of the classifiers. Firstly, the

Table 11: Aggregated Lexical Functions Dataset Results for English.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.90**	**0.89**	**0.90**	**0.89**
Subtraction	0.89	0.86	0.89	0.87
Addition	0.90	0.88	0.90	0.88
Multiplication	0.82	0.77	0.82	0.79

Table 12: Aggregated Lexical Functions Dataset Results for Spanish.

Operation	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Concatenate	**0.80**	**0.82**	**0.80**	**0.77**
Subtraction	0.78	0.80	0.78	0.75
Addition	0.79	0.80	0.80	0.76
Multiplication	0.65	0.70	0.65	0.57

dataset which we are training the classifier; secondly, the arithmetical operations computed over the base - collocate vector embedding to create the training data and finally, the use of relation vector for the same purpose that the last one.

Regarding the influence of different datasets in the performance of the classification task, the obtained results between those classifiers trained with the Extended Lexical Function Dataset and those trained with the Aggregated Lexical Function Dataset are not significant. However, the same does not apply if we compare the results of the classifiers for the same datasets but different language where the classifiers trained to discriminate Spanish Lexical Functions have worse results than those trained for English. One possible explanation for this phenomenon is that the distribution of the collocations in a corpus is uneven for each language and, the composition of the datasets is *unique* and imbalanced in its own way.

As to the difference between the obtained results for each arithmetical operation used to create the input trained data show that the better classification results are for those classifiers trained with data created from the concatenation of the base and collocate embeddings. On the other hand, the classifiers trained with the base and collocate vectors multiplication obtain the worse results.

Finally, we compared the results obtained by the use of relation vectors for the LF classification task. Specifically, we can see that despite the results for those classifiers that have been trained without relation vectors are pretty high, those classifiers trained with input training data formed by word embeddings plus relation vectors present far better results. The most feasible explanation is that when we use relation vectors for training the classifier we are adding specific contextual information for each instance of the training set allowing the classifier to have a better possibility of LF discrimination.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, we applied a Supervised Learning Machine technique in order to identify collocations from a corpus and classify them by its corresponding Lexical Function. After carrying out different experiments, where several SVM multiclass classifiers have been trained using different configurations for different languages and with different datasets, we can conclude that first of all, the SVM method is able to classify LF with very good results as well as indirectly identify the collocations present in a text. In fact, those lexical functions that are the best classified, for both English and Spanish are *oper1, sing, and magn.* Secondly, regardless of the dataset used to train the classifiers, the improvement of the performance is insignificant so, it would be advisable for future work to use the extended dataset that would allow analyzing in depth each LF separately.

The classifiers that provide the best results are those that are trained with samples formed by the concatenation of the word vectors of a pair of words corresponding to the base and the collocate of a collocation. On the contrary, the worst results are obtained with those training sets whose samples have been formed by multiplying the vectors of the base and the collocate. Finally, there is a significant difference in the classification capacity of LF of those classifiers that incorporated relation vectors in the training stage in comparison to those that only have used word vectors.

6.2 Future work

This thesis opens the door to several future lines of work. The first one, the data augmentation for the creation of a balanced dataset, where all the Lexical Functions that compose the dataset have the same number of instances. In this way, when we analyze the classification results for each LF, we can hypothetically ensure that one LF is better classified than another not because the classifier has been trained with more examples of the former than of the latter, but because the features that present the first are more significant than the second.

Furthermore, it would be interesting to investigate the differences between the LF samples correctly classified as for instance samples of the class *Magn* and those that the classifier is not able to correctly label as *Antimagn*, with the aim of extracting some pattern from the data that facilitates the task of the classifier for those cases with a fatal result.

Bibliography

[1] Karan, M., Šnajder, J. & Bačic, B. D. Evaluation of classification algorithms and features for collocation extraction in croatian (2012).

[2] Mikolov, T., Chen, K., Corrado, G. S. & Dean, J. Efficient estimation of word representations in vector space (2013). URL http://arxiv.org/abs/1301.3781.

[3] Barnbrook, G., Mason, O. & Krishnamurthy, R. *Collocation: Applications and implications* (Palgrave MacMillan, 2013).

[4] Rodríguez Fernández, S. *Collocation and collocation error processing in the context of second language learning*. Ph.D. thesis, Universitat Pompeu Fabra. Departament de Tecnologies de la Informacio i les Comunicacions (2018). URL http://hdl.handle.net/10803/463080.

[5] Leo Wanner, M. G., Bernd Bohnet. What is beyond collocations? insights from machine learning experiments. *Proceedings of the 12th EURALEX International Congress* 1071–1087 (2006).

[6] Garcia, M., Garcia Salido, M. & Alonso-Ramos, M. Using bilingual word-embeddings for multilingual collocation extraction (2017).

[7] Mel'čuk, I. Collocations and Lexical Functions. *Phraseology. Theory, Analysis, and Applications* 23–53 (1998).

[8] Gelbukh, A. & Kolesnikova, O. Supervised learning algorithms evaluation on recognizing semantic types of spanish verb-noun collocations. 297–308 (Cen-

tro de Investigacion en Computacion (CIC) del Instituto Politecnico Nacional (IPN), 2012).

[9] *Researching Collocations in Another Language - Multiple Interpretations* (Palgrave Macmillan, 2009).

[10] Firth, R. Modes of meaning. 190–215 (1957).

[11] Halliday, M. A. K. Lexis as a linguistic level. 148–62 (1966).

[12] Sinclair, J. Beginning the study of lexis. 410–30 (1957).

[13] Hausmann, F. J. Kollokationen im deutschen wörterbuch. ein beitrag zur theorie des lexikographischen beispiels. In Henning Bergenholtz Joachim Mugdan (Eds.), Lexikographie und Grammatik. Akten des Essener Kolloquiums zur Grammatik im Wörterbuc 118–129 (1985).

[14] Mel'čuk, I. Lexical functions: A tool for the description of lexical relations in a lexicon. *Lexical Functions in Lexicography and Natural Language Proecessing* 23–54 (1996).

[15] Gelbukh, A. & Bolshakov, I. Lexical Functions in Spanish. 383–395 (Centro de Investigacion en Computacion (CIC) del Instituto Politecnico Nacional (IPN), Méjico, 1998).

[16] Jurafsky, D. & Martin, J. A. *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (Draft, 2018).

[17] Mikolov, T. *et al.* Exploiting similarities among languages for machine translation (2013).

[18] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. Distributed representations of words and phrases and their compositionality. 3111–3119 (Curran Associates, Inc., 2013).

[19] Espinosa-Anke, L. & Schockaert, S. Seven: Augmenting word embeddings with unsupervised relation vectors (2018).

[20] Wanner, L., Ferraro, G. & Moreno, P. Towards distributional semantics-based classification of collocations for collocation dictionaries. *International Journal of Lexicography* **30**, ecw002 (2016).

[21] Gelbukh, A. & Kolesnikova, O. Supervised learning for semantic classification of spanish collocations. 362–371 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010).

[22] Girju, R. *et al.* Semeval-2007 task 04: Classification of semantic relations between nominals. SemEval '07, 13–18 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2007). URL `http://dl.acm.org/citation.cfm?id=1621474.1621477`.

[23] Hendrickx, I. *et al.* Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. SemEval '10, 33–38 (Association for Computational Linguistics, Stroudsburg, PA, USA, 2010). URL `http://dl.acm.org/citation.cfm?id=1859664.1859670`.

[24] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. ECML'98, 137–142 (Springer-Verlag, Berlin, Heidelberg, 1998). URL `https://doi.org/10.1007/BFb0026683`.

[25] Hossin, M. & M.N, S. A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining Knowledge Management Process* **5**, 01–11 (2015).

[26] Moreno, P., Ferraro, G. & Wanner, L. Can we determine the semantics of collocations without using semantics (2013).

[27] Vylomova, E., Rimell, L., Cohn, T. & Baldwin, T. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. *CoRR* **abs/1509.01692** (2015). URL `http://arxiv.org/abs/1509.01692`. `1509.01692`.

Appendix A

Word2Vec Approach: Extended Lexical Function Dataset Results

LF	Language	Operation	Precision	Recall	F1-score
Bon	English	Concatenation	1	0.12	0.21
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0
	Spanish	Concatenation	1	0.06	0.11
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Antibon	English	Concatenation	0.75	0.10	0.18
		Subtraction	1	0.07	0.13
		Addition	1	0.07	0.13
		Multiplication	**0**	**0**	**0**
	Spanish	Concatenation	0	0	0
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	1	0.12	0.22

LF	Language	Operation	Precision	Recall	F1-score
Magn	English	Concatenation	0.74	0.84	0.79
		Subtraction	0.75	0.74	0.74
		Addition	0.76	0.73	0.74
		Multiplication	0.68	0.26	0.38
	Spanish	Concatenation	0.75	0.36	0.48
		Subtraction	0.78	0.33	0.47
		Addition	1	0.17	0.29
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Antimagn	English	Concatenation	0.89	0.19	0.31
		Subtraction	0.67	0.05	0.09
		Addition	0.67	0.10	0.17
		Multiplication	1	0.02	0.05
	Spanish	Concatenation	0	0	0
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Causfunc0	English	Concatenation	1	0.15	0.27
		Subtraction	1	0.04	0.07
		Addition	1	0.12	0.21
		Multiplication	0	0	0
	Spanish	Concatenation	0	0	0
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Liqfunc0	English	Concatenation	1	0.11	0.19
		Subtraction	1	0.11	0.19
		Addition	1	0.05	0.10
		Multiplication	0	0	0
	Spanish	Concatenation	0	0	0
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Oper1	English	Concatenation	0.75	0.86	0.80
		Subtraction	0.70	0.90	0.79
		Addition	0.69	0.90	0.78
		Multiplication	0.56	0.97	0.71
	Spanish	Concatenation	0.58	0.97	0.72
		Subtraction	0.56	0.98	0.71
		Addition	0.53	0.97	0.68
		Multiplication	0.49	1	0.66

LF	Language	Operation	Precision	Recall	F1-score
Real1	English	Concatenation	0.72	0.66	0.69
		Subtraction	0.72	0.51	0.59
		Addition	0.73	0.52	0.60
		Multiplication	0.78	0.07	0.13
	Spanish	Concatenation	0.80	0.56	0.66
		Subtraction	0.79	0.39	0.52
		Addition	0.74	0.34	0.46
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Sing	English	Concatenation	0.93	0.24	0.38
		Subtraction	0.89	0.15	0.25
		Addition	0.89	0.15	0.25
		Multiplication	1	0.07	0.14
	Spanish	Concatenation	1	0.05	0.09
		Subtraction	1	0.05	0.09
		Addition	1	0.05	0.09
		Multiplication	0	0	0

Appendix B

Word2Vec Approach: Aggregated Lexical Function Dataset Results

LF	Language	Operation	Precision	Recall	F1-score
Cause	English	Concatenation	0.67	0.16	0.26
		Subtraction	1	0.04	0.08
		Addition	0.50	0.04	0.07
		Multiplication	0	0	0
	Spanish	Concatenation	0.75	0.10	0.18
		Subtraction	1	0.03	0.07
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Experimenter	English	Concatenation	0.78	0.72	0.75
		Subtraction	0.87	0.58	0.69
		Addition	0.80	0.58	0.67
		Multiplication	0.85	0.06	0.11
	Spanish	Concatenation	0.69	0.47	0.56
		Subtraction	0.68	0.37	0.48
		Addition	0.62	0.32	0.42
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Intensity	English	Concatenation	0.80	0.85	0.82
		Subtraction	0.80	0.75	0.77
		Addition	0.81	0.72	0.76
		Multiplication	0.77	0.35	0.48
	Spanish	Concatenation	0.86	0.57	0.69
		Subtraction	1	0.48	0.65
		Addition	0.89	0.38	0.53
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Manifest	English	Concatenation	0.10	0.06	0.07
		Subtraction	0.10	0.06	0.07
		Addition	0.67	0.09	0.15
		Multiplication	0.75	0.13	0.22
	Spanish	Concatenation	0.86	0.57	0.69
		Subtraction	0.75	0.13	0.22
		Addition	1	0.04	0.08
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Phase	English	Concatenation	0.15	0.08	0.10
		Subtraction	0.16	0.08	0.10
		Addition	0.08	0.04	0.05
		Multiplication	0	0	0
	Spanish	Concatenation	0.71	0.11	0.19
		Subtraction	0.75	0.07	0.12
		Addition	0.50	0.04	0.08
		Multiplication	0	0	0

Appendix C

Relational Vector Approach: Extended Lexical Function Dataset Results

LF	Language	Operation	Precision	Recall	F1-score
Bon	English	Concatenation	0.83	0.23	0.36
		Subtraction	1	0.08	0.14
		Addition	1	0.23	0.38
		Multiplication	0	0	0
	Spanish	Concatenation	1	0.24	0.38
		Subtraction	0	0	0
		Addition	1	0.24	0.38
		Multiplication	1	0.06	0.11

LF	Language	Operation	Precision	Recall	F1-score
Antibon	English	Concatenation	0.75	0.10	0.18
		Subtraction	0	0	0
		Addition	0.33	0.03	0.06
		Multiplication	0	0	0
	Spanish	Concatenation	0	0	0
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Magn	English	Concatenation	0.84	0.94	0.89
		Subtraction	0.78	0.94	0.85
		Addition	0.82	0.94	0.88
		Multiplication	0.77	0.87	0.82
	Spanish	Concatenation	0.80	0.48	0.60
		Subtraction	0.71	0.40	0.52
		Addition	0.67	0.33	0.44
		Multiplication	0.67	0.05	0.09

LF	Language	Operation	Precision	Recall	F1-score
Antimagn	English	Concatenation	0.891	0.31	0.45
		Subtraction	1	0.05	0.09
		Addition	1	0.14	0.24
		Multiplication	0.50	0.02	0.05
	Spanish	Concatenation	0	0	0
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Causfunc0	English	Concatenation	1	0.50	0.67
		Subtraction	0	0	0
		Addition	0.85	0.42	0.56
		Multiplication	0	0	0
	Spanish	Concatenation	1	0.33	0.50
		Subtraction	0.71	0.28	0.40
		Addition	1	0.33	0.50
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Liqfunc0	English	Concatenation	1	0.42	0.59
		Subtraction	1	0.32	0.58
		Addition	0.78	0.37	0.50
		Multiplication	1	0-05	0.10
	Spanish	Concatenation	0	0	0
		Subtraction	0	0	0
		Addition	0	0	0
		Multiplication	0	0	0

LF	Language	Operation	Precision	Recall	F1-score
Oper1	English	Concatenation	0.94	0.99	0.96
		Subtraction	0.94	0.99	0.97
		Addition	0.94	0.99	0.96
		Multiplication	0.82	0.99	0.90
	Spanish	Concatenation	0.74	1	0.85
		Subtraction	0.72	0.99	0.83
		Addition	0.72	0.99	0.83
		Multiplication	0.60	0.99	0.74

LF	Language	Operation	Precision	Recall	F1-score
Real1	English	Concatenation	0.82	0.94	0.88
		Subtraction	0.76	0.94	0.88
		Addition	0.80	0.94	0.86
		Multiplication	0.70	0.55	0.62
	Spanish	Concatenation	0.73	0.90	0.81
		Subtraction	0.74	0.88	0.80
		Addition	0.73	0.90	0.80
		Multiplication	0.61	0.57	0.59

LF	Language	Operation	Precision	Recall	F1-score
Sing	English	Concatenation	0.93	0.24	0.38
		Subtraction	0.74	0.52	0.61
		Addition	0.75	0.39	0.51
		Multiplication	0.65	0.44	0.53
	Spanish	Concatenation	0.90	0.17	0.28
		Subtraction	1	0.24	0.38
		Addition	1	0.05	0.09
		Multiplication	0	0	0

Appendix D

Relational Vector Approach: Aggregated Lexical Function Dataset Results

LF	Language	Operation	Precision	Recall	F1-score
Cause	English	Concatenation	0.67	0.16	0.26
		Subtraction	0.25	0.04	0.07
		Addition	0.83	0.20	0.32
		Multiplication	0	0	0
	Spanish	Concatenation	1	0.24	0.39
		Subtraction	1	0.21	0.34
		Addition	1	0.24	0.39
		Multiplication	1	0.07	0.13

LF	Language	Operation	Precision	Recall	F1-score
Experimenter	English	Concatenation	0.79	0.91	0.84
		Subtraction	0.75	0.89	0.81
		Addition	0.80	0.91	0.85
		Multiplication	0.71	0.55	0.62
	Spanish	Concatenation	0.66	0.85	0.74
		Subtraction	0.64	0.87	0.74
		Addition	0.65	0.84	0.73
		Multiplication	0.48	0.59	0.53

LF	Language	Operation	Precision	Recall	F1-score
Intensity	English	Concatenation	0.95	0.94	0.94
		Subtraction	0.93	0.94	0.93
		Addition	0.93	0.92	0.93
		Multiplication	0.86	0.87	0.86
	Spanish	Concatenation	1	0.62	0.76
		Subtraction	1	0.57	0.73
		Addition	1	0.62	0.76
		Multiplication	1	0.10	0.17

LF	Language	Operation	Precision	Recall	F1-score
Manifest	English	Concatenation	0.34	0.28	0.31
		Subtraction	0.24	0.15	0.18
		Addition	0.34	0.26	0.29
		Multiplication	0.12	0.06	0.08
	Spanish	Concatenation	1	0.22	0.36
		Subtraction	0.80	0.17	0.29
		Addition	0.71	0.22	0.33
		Multiplication	1	0.04	0.08

LF	Language	Operation	Precision	Recall	F1-score
Phase	English	Concatenation	0.33	0.17	0.22
		Subtraction	0.24	0.11	0.15
		Addition	0.30	0.15	0.20
		Multiplication	0	0	0
	Spanish	Concatenation	0.74	0.58	0.65
		Subtraction	0.69	0.49	0.57
		Addition	0.73	0.53	0.62
		Multiplication	0.62	0.18	0.28