Procesamiento del Lenguaje Natural, Revista nº 66, marzo de 2021, pp. 153-164 recibido 23-11-2020 revisado 26-01-2021 aceptado 01-02-2021

Classifying Spanish *se* constructions: from bag of words to language models

Clasificación de construcciones con se en español: de modelos de bolsa de palabras a modelos de lenguaje

Nuria Aldama García, Álvaro Barbero Jiménez
Universidad Autónoma de Madrid
Instituto de Ingeniería del Conocimiento
nuria.aldama@estudiante.uam.es, alvaro.barbero@iic.uam.es

Abstract: Spanish *se* constructions are a complex linguistic phenomenon that challenges Natural Language Processing (NLP) tasks such as part-of-speech or dependency relation tagging. *Se* is a high-frequency word that appears in nine different types of syntactic constructions and adds information of diverse nature depending on the context. Thus, to solve the problem Spanish *se* constructions poses in an efficient way, this study proposes a tagging system for *se* applied to a corpus composed of 2,140 sentences. This corpus is used in a classification experiment where 9 classifiers based on machine learning models and a dependency parser are tested. Results show that pre-trained language models based on transformers architecture reach the highest accuracy (0.83) and f-score (0.70) values.
Keywords: Spanish *se* constructions, multiclass classification, machine learning.

Resumen: Las construcciones con *se* en español son un complejo fenómeno lingüístico que desafía tareas de Procesamiento del Lenguaje Natural (PLN) como el etiquetado automático de categoría gramatical (*POS tagging*) o de relaciones de dependencias. *Se* es una forma de alta frecuencia que aparece en nueve tipos de construcciones sintácticas del español, aportando información de diferente naturaleza en función del contexto. Por ello, para tratar el problema de clasificación que plantean las construcciones con *se* de manera eficiente, este estudio propone un sistema de etiquetado de *se* aplicado a un corpus de 2.140 oraciones y probado con 9 clasificadores basados en modelos de aprendizaje automático y un parser de dependencias. Los resultados muestran que los modelos pre-entrenados basados en arquitectura de *transformers* alcanzan los valores más elevados de exactitud (0,83) y de F-score (0,70).
Palabras clave: Construcciones con *se*, clasificación multiclase, aprendizaje automático.

1 Introduction

Spanish *se* constructions are a well-known and complex linguistic topic within the study of Spanish. *Se* constructions challenge Natural Language Processing (NLP) tasks such as automatic part-of-speech-tagging (POS) and dependency parsing for three main reasons. First, *se* is a high-frequency Spanish word. According to CORPES XXI (Real Academia Española de la Lengua, 2020), *se* is in the eleventh position of the ranking of most common grammatical elements and most common lemmas in Spanish and it is placed in the ninth position in the ranking of most common Spanish forms. Second, *se* may appear in nine different syntactic constructions where it conveys diverse semantic meanings and bears several syntactic roles (if any). Third, the form *se* does not bear any specific morphosyntactic feature that helps disambiguating one type of *se* from another.

The main goal of this study is to evaluate the performance of different classification strategies that are intended to solve the task of *se* disambiguation based on an adaptation of the analysis of *se* presented by Moreno Cabrera (1997, 2002). To do so, a corpus containing 2,140 sentences, the *SE*-corpus, is built as a means of training and evaluating nine classifiers and a state-of-the-art parser. A secondary

ISSN 1989-7553. DOI 10.26342/2021-66-13

objective is to understand the kind of information (lexicon, colocations, semantics, syntax, any other contextual information) that is needed by a machine learning model to best disambiguate Spanish *se* constructions.

The paper is structured as follows. Section 2 summarizes Spanish *se* constructions. Section 3 describes the *SE*-corpus. Section 4 presents *se* tag distribution. Section 5 deals with corpus quality. Section 6 introduces the classification strategies used in this study. Section 7 shows experimental results. Conclusions and future work are drawn in section 8.

2 Spanish se constructions

Se may appear in nine different syntactic constructions where it conveys diverse semantic meanings and bears several syntactic roles (if any). This section makes a brief theoretical review of this kind of constructions based on (Sánchez, 2002), (Sánchez, 2015), (Mendikoetxea, 1999 a), (Mendikoetxea, 1999 b), (Campos, 1999), (Fernández-Montraveta and Vázquez, 2017), (Moreno 1997) and (Moreno, 2002).

Se constructions may be classified as paradigmatic (if the concrete construction can be built with all the pronominal forms of the paradigm) or non-paradigmatic (if the concrete construction can only be built with the form *se*). Within the class of paradigmatic constructions, *se* may appear in transitive constructions like (1), (2) and (3). *Se* functions as an indirect object in (1) and (3) and it has a benefactive or recipient semantic role. In (2), *se* is the internal argument of the main predicate *comb*, it is accusative case assigned and bears the semantic role commonly known as patient. (1) is a ditransitive construction, (2) is a transitive reflexive construction and (3) is a transitive reciprocal one.

(1) Se lo dije a Juan ayer .
Him-DAT it-ACC tell-PST.1SG to Juan yesterday.
'I told it to Juan yesterday.'

(2) Juan se peina .
Juan himself-ACC comb-PRS.3SG.
'Juan combs himself.'

(3) Ellos se envían cartas.
They them-DAT send-PRS.3PL letters.
'They send letters to each other.'

Example (4) corresponds to a *pure* pronominal construction (the pronoun is inherent to the predicate and its semantic meaning) where *se* does not bear a syntactic function. *Se* in (5) is an emphatic pronoun that is sometimes called emphatic or interest dative and that may be elided because it does not bear any semantic or syntactic function.

(4) Juan se desmayó de repente.
Juan him faint-PST.3SG suddenly.
'Juan suddenly fainted.'

(5) Juan (se) comió un bocadillo.
Juan him eat-PST.3SG a sandwich.
'Juan ate a sandwich.'

Se in examples in (6), (7), (8) and (9) does not behave as a pronoun bearing a syntactic, semantic, emphatic or discursive function like the ones in (1) - (5), but it works as a valency reduction mark signaling that the number of arguments of the main predicate is reduced. More concretely, the agentive external argument of the constructions in (6) – (9) is elided due to different linguistic strategies. (6) is an active construction where the agent is not present because it is the inchoative variant of the causative-inchoative alternation duplicity allowed by the predicate *romper*. (7) is a *media* voice construction where the agent *washer* is not present and where the property of *washing-well* is assigned to the shirt itself. (8) is a reflexive passive where the *looker* is not present for some reason. (9) is an impersonal construction where a general gone-without-saying subject is understood to perform the action of eating.

(6) El jarrón se rompió.
The vase - break-PST.3SG.
'The vase broke.'

(7) La camisa se lava muy bien.
The shirt - wash-PRS.3SG very well.
'The shirt washes very well.'

(8) Se buscan camareros.
- look.for-PRS.3PL bartenders.
'Bartenders are required.'

(9) Se come bien aquí.
- eat-PRS.3SG well here.
'It's a good place to eat in here.'

Moreno (1997, 2002) presents a unifying analysis that treats *se* constructions as a continuum of transitivity. Transitive constructions are placed at one end of the continuum where *se* can behave as an internal argument of the main predicate. All those *se* bearing the syntactic functions of direct and indirect objects are at this end of the continuum. Those *se* constructions that are traditionally considered paradigmatic (they belong to paradigmatic class) but where *se* does not bear any syntactic/semantic function, that is, *se* part of pure pronominal predicates and emphatic *se* are placed in the mid part of the continuum. Those *se* signaling main predicate valency reduction are placed at the other end of the continuum, that is, impersonal, passive *se* and those *se* that appear in *media* voice and inchoative constructions.

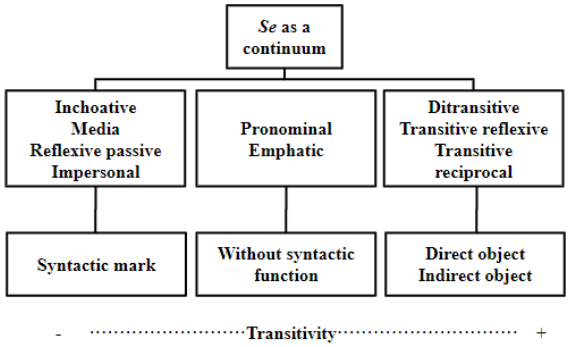

Figure 1: *Se* constructions as a continuum, of transitivity.

Following Moreno (1997, 2002) an annotation scheme for *se* constructions is proposed in the following section.

3 SE-corpus reduced version[1]

The *SE*-corpus reduced version (from now on *SE*-corpus) is composed of 2,140 sentences that come from CORPES XXI (*Real Academia Española de la Lengua, 2020*). The sentence selection procedure starts picking up, from the whole CORPES XXI, every sentence that contains the word *se* and belongs to the *news, leisure and daily life* domain in the European Spanish variant. From the output of this retrieval query, 3,000 sentences (complete SE-corpus) are randomly selected. Through the last filter, those sentences having more than one instance of *se* are eliminated.

Summing-up, the corpus used to carry out this research is composed of 2,140 sentences containing a single instance of *se*. The corpus is representative of the *news, leisure and daily life* domain in the European Spanish variant because it maintains real usage distribution of *se* constructions.

The annotation process is carried out following the next annotation criteria:

- *se-mark*: Cases of valency reduction (6) - (9).
- *expl*: Pure pronominal predicates or emphatic contexts (4) – (5).
- *iobj*: *Se* as indirect object of the main predicate (1) and (3).
- *obj*: *Se* as direct object of the main predicate (2).

4 Se tags distribution

The distribution of *se* tags presented in the corpus is quite unbalanced, as shown in table 1. The most prominent category (*se-mark*) is twelve times more frequent than the less prominent category (*obj*). Besides the intermediate categories, *expl* and *iobj* are quite extreme too: *expl* is close to the most frequent category (*se-mark*) whereas *iobj* is close in volume to the less frequent category *obj*. Thus, the corpus is unbalanced with two very frequent categories and two very infrequent ones. This distribution challenges the classification task.

Tag	Volume	%
se-mark	964	45.05
expl	946	44.21
iobj	154	7.2
obj	76	3.55
TOTAL	2,140	100

Table 1: *Se* tag distribution in the *SE-corpus.*

5 SE-corpus quality

The *SE*-corpus is annotated by a single annotator (annotator 1) due to human and time resources restrictions.[2] However, for the sake of

[1] The original *SE*-corpus is composed of 3,000 sentences that include one or more *se* per sentence. The reduced version of the *SE*-corpus is built from sentences that include a single instance of *se*.

[2] Annotation processes take quite a long time. Besides, it is not easy to find annotators with a

consistency and annotation quality, 100 sentences are annotated by the main annotator and two experts in the field of theoretical study of Spanish *se* (annotator 2 and annotator 3). All the annotators had the same annotation information, followed the same annotation guidelines and were aware of the 9 *se* types this classification experiments are focused on. The average inter-annotator agreement value[3] is 76.90%. The f1-score obtained by an average expert annotator against the gold standard is 0.85.

Pair of annotators	Agreement (%)
Anno1-Anno2	75.71
Anno1-Anno3	83.57
Anno2-Anno3	71.43

Table 2: Inter-annotator agreement.

Having a look at table 2, it can be observed that annotation agreement experiments some variations. The agreement value between annotator 1 and annotator 3 is higher in nearly 8 points than the agreement value between annotator 1 and annotator 2. The agreement value between annotator 1 and annotator 3 is also higher in 12.14 points than the agreement value between annotator 2 and 3. However, it is important to mention that the agreement value between annotator 2 and annotator 3 differs in 7.1 points with the next lowest agreement value, meaning that there are no significant differences in annotation quality nor consistency among the three annotators. Main disagreement cases come from media constructions that are not always easy to tell apart from pronominal predicates.[4] It is important to say that neither pronominal nor media constructions are part of the under-represented categories. The less frequent categories are those where *se* displays argument functions, namely, *obj* and *iobj*. These differences and similarities in agreement values may point towards the complexity of classifying Spanish *se* constructions and the possible alternative interpretations that may arise despite consistent annotations.

certain level of knowledge of the object of study, annotation, and computer skills.

[3] Raw or observed agreement (Bayerl and Paul, 2011), (Artstein, 2017).

[4] Pronominal predicates also called 'pure' pronominal predicates or inherently pronominal predicates in the literature introduced in section 2.

6 Classification strategies

To test whether the annotation scheme is efficient and can be easily learnt, and, whether the *SE*-corpus is big enough to deal with this classification problem, the *SE*-corpus is automatically segmented in train (1,713 sentences) and test (427 sentences) corpora.[5] Except for the es-BERT and UD-Pipe models, all text processing, vectorization steps and classifiers were implemented using Scikit Learn (Pedregosa et al., 2020). The tags of both the train and test corpora are preprocessed and turned into numbers using *LabelEncoder*. The classification task is performed by eight different models and a state-of-the-art parser. Precision (10), recall (11) and F1-score (12) are calculated per tag. Macro average F-score (13) and Accuracy, that is, the percentage of correct answers, show overall performance.[6] Model hyperparameters are tuned using a standard grid search with 5 folds stratified cross-validation. Parameter ranges are detailed in Appendix B. Different strategies are carried out for each concrete model to reduce the effect of unbalanced tag distribution:

- No balancing: models are trained using the unaltered training dataset.
- Search scoring (SC): Grid Search is configured to optimize the f1 macro scoring function.
- Class weight balancing (CW): the models are configured to give more relevance during training to patterns belonging to underrepresented classes.[7]
- Oversampling (OS): synthetic samples from underrepresented classes are added to the training dataset until all classes are balanced. The new samples are duplicates of samples already present in the training data.

[5] The test corpus remains the same along the experimental procedure for the sake of comparison between the different models and parser. The train corpus is expanded up to 3,195 sentences to run oversampling experiments.

[6] Equations taken from (Shmueli, 2019) and (Shung, 2018).

[7] For all the lineal models, a combination of search scoring and class weight strategies is also tested with similar results to the search scoring and class weight strategies applied independently.

$$\frac{True\ positives}{True\ positives + False\ positives} \quad (10)$$

$$\frac{True\ positives}{True\ positives + False\ negatives} \quad (11)$$

$$2 * \frac{Precision * Recall}{Precision + Recall} \quad (12)$$

$$\frac{\sum F\text{-}Score}{Total\ number\ of\ tags} \quad (13)$$

6.1 Baseline

The baseline model is generated by annotating the whole test corpus with the most frequent tag *se-mark* and comparing it to the gold standard. The accuracy and macro average f-score raise up to 0.49 and 0.16 respectively.

Metric	expl	iobj	obj	Se-mark
Precision	0.00	0.00	0.00	0.49
Recall	0.00	0.00	0.00	1
F-score	0.00	0.00	0.00	0.66
Acc/MAF[8]	**0.49 / 0.16**			

Table 3: Baseline results.[9]

6.2 Non-linear bag of words model

The first bag of words model is built with *CountVectorizer* and a *Random Forest Classifier* (Breiman, 2001) model. Random Forest is an ensemble of classification trees that has been shown to perform well on a wide range of problem. The best grid search parameters include pentagrams of characters from text inside word boundaries. The highest accuracy value obtained goes up to 0.63 and the highest macro average f-score reaches 0.34 points.

Metric	expl	iobj	obj	Se-mark	BS[10]
Precision	0.55	0.00	0.00	0.68	OP
Recall	0.73	0.00	0.00	0.65	
F-score	0.63	0.00	0.00	0.67	
Acc/MAF	0.61 / 0.32				
Precision	0.55	0.00	0.00	0.68	SC
Recall	0.73	0.00	0.00	0.65	
F-score	0.63	0.00	0.00	0.67	
Acc/MAF	0.61 / 0.32				
Precision	0.58	1.00	0.00	0.66	CW
Recall	0.65	0.03	0.00	0.74	
F- score	0.61	0.06	0.00	0.70	
Acc/MAF	**0.63** / **0.34**				
Precision	0.57	0.50	0.00	0.65	OS
Recall	0.61	0.03	0.00	0.75	
F-score	0.59	0.06	0.00	0.70	
Acc/MAF	0.62 / **0.34**				

Table 4: Non-linear bag of words model results.

6.3 Linear bag of words model

The second bag of words model is built with *CountVectorizer* and a Linear Support Vector Classification model (Fan et al. 2008). Such model has been widely used in text classification problems; however, it lacks the ability to deal with multiclass problems. Hence, an *OneVsRestClassifier w*rapper is applied to split the problem into 4 one-versus-rest binary problems. *GridSearch* best parameters include groups of n-grams from 5 to 7 characters within word boundaries. As shown in table 5, there is no result variation. The highest accuracy and macro average f-score values are 0.61 and 0.32, respectively.

Metric	expl	iobj	obj	Se-mark	BS
Precision	0.58	0.00	0.00	0.65	OP
Recall	0.65	0.00	0.00	0.71	
F-score	0.61	0.00	0.00	0.68	
Acc/MAF	**0.61** / **0.32**				
Precision	0.58	0.00	0.00	0.65	SC
Recall	0.65	0.00	0.00	0.71	
F-score	0.61	0.00	0.00	0.68	
Acc/MAF	**0.61** / **0.32**				
Precision	0.57	0.00	0.00	0.64	CW
Recall	0.64	0.00	0.00	0.71	
F-score	0.60	0.00	0.00	0.67	

[8] *Acc* stands for accuracy and *MAF* for macro average F-score.

[9] For baseline, *CountVectorizer*, *HashingVectorizer*, *TF-IDF* and *UD-Pipe* models, precision, recall, f-score and acc/MAF results are obtained from training and testing procedure using the *SE-corpus* and the original parameter configuration (OP); SC method results are obtained using the Search scoring strategy; CW results are obtained using the Class Weight balancing strategy; OS results are obtained using the Oversampling strategy.

[10] Balancing strategy.

Acc/MAF	**0.61 / 0.32**				
Precision	0.55	0.00	0.00	0.66	OS
Recall	0.68	0.00	0.00	0.66	
F-score	0.61	0.00	0.00	0.66	
Acc/MAF	0.60 / **0.32**				

Table 5: Linear bag of words model results.

6.4 Non-linear hashing model

As a variant of the non-linear bag of words model, a vectorization through the Hashing trick (Weinberger, 2009) was also explored. This vectorization is able to produce more space-efficient representations that can lead to better results. Table 6 shows the classification results of the first model composed of a *Hashing Vectorizer* and *Random Forest Classifier* algorithms. Using 100 classification trees, and a n-gram range of 5-7 characters from text inside word boundaries, the model achieves 0.62 accuracy points and 0.32 macro average points.

Metric	**expl**	**iobj**	**obj**	**Se-mark**	**BS**
Precision	0.55	0.00	0.00	0.66	OP
Recall	0.66	0.00	0.00	0.69	
F-score	0.60	0.00	0.00	0.67	
Acc/MAF	0.61 / **0.32**				
Precision	0.55	0.00	0.00	0.66	SC
Recall	0.66	0.00	0.00	0.69	
F-score	0.60	0.00	0.00	0.67	
Acc/MAF	0.61 / **0.32**				
Precision	0.54	0.00	0.00	0.63	CW
Recall	0.63	0.00	0.00	0.68	
F-score	0.58	0.00	0.00	0.65	
Acc/MAF	0.59 / 0.31				
Precision	0.6	0.00	0.00	0.64	OS
Recall	0.6	0.00	0.00	0.78	
F-score	0.6	0.00	0.00	0.70	
Acc/MAF	**0.62** / **0.32**				

Table 6: Non-linear *hashing* model results.

6.5 Linear hashing model

Similar to the previous model, the fourth model is a hashing version of the linear bag of words model. Again, a *GridSearch* algorithm extracts the best training parameters, that are 100 classification trees and a range of 5 to 7 n-grams of characters from text inside word boundaries. The accuracy goes up to 0.61 points and the macro average to 0.32.

Metric	**expl**	**iobj**	**obj**	**Se-mark**	**BS**
Precision	0.53	0.00	0.00	0.65	OC
Recall	0.68	0.00	0.00	0.64	
F-score	0.60	0.00	0.00	0.65	
Acc/MAF	0.59 / 0.31				
Precision	0.53	0.00	0.00	0.65	SC
Recall	0.68	0.00	0.00	0.64	
F-score	0.60	0.00	0.00	0.65	
Acc/ MAF	0.59 / 0.31				
Precision	0.56	0.00	0.00	0.66	CW
Recall	0.66	0.00	0.00	0.70	
F-score	0.61	0.00	0.00	0.68	
Acc/ MAF	**0.61** / **0.32**				
Precision	0.55	0.00	0.00	0.68	OS
Recall	0.69	0.00	0.00	0.68	
F-score	0.61	0.00	0.00	0.68	
Acc/MAF	**0.61** / **0.32**				

Table 7: Linear *hashing* model results.

6.6 Non-linear TF-IDF

The fifth model is formed by a combination of *TF-IDF* and *Random Forest Classifier*. *TF-IDF* is a weighed variant bag of words, that promotes words that are highly specific of the document under analysis. The best training parameters extracted by a *GridSearch* algorithm convey 100 classification trees and a range of 5 to 7 n-grams of characters from text inside word boundaries. The highest accuracy value goes up to 0.63 points and the macro average to 0.35.

Metric	**expl**	**iobj**	**obj**	**Se-mark**	**BS**
Precision	0.55	0.00	0.00	0.67	OC
Recall	0.67	0.00	0.00	0.70	
F-score	0.61	0.00	0.00	0.68	
Acc/MAF	0.61 / 0.32				
Precision	0.55	0.00	0.00	0.67	SC
Recall	0.67	0.00	0.00	0.70	
F-score	0.61	0.00	0.00	0.68	
Acc/MAF	0.61 / 0.32				
Precision	0.58	1.00	0.00	0.67	CW
Recall	0.67	0.03	0.00	0.72	
F-score	0.62	0.06	0.00	0.70	
Acc/MAF	**0.63** / **0.35**				
Precision	0.59	0.50	0.00	0.64	OS
Recall	0.60	0.03	0.00	0.77	
F-score	0.59	0.06	0.00	0.7	
Acc/MAF	0.62 / 0.32				

Table 8: Non-linear *TF-IDF* model results.

6.7 Linear TF-IDF model

The second TF-IDF model is built up with *TF-IDF* and a linear SVC classifier. The

GridSearch algorithm yields that the best parameters include 100 classification trees and a n-gram range between 5 and 7 characters found within word boundaries. The accuracy reaches 0.64 points, and the macro average goes up to 0.34.

Metric	expl	iobj	obj	Se-mark	BS
Precision	0.57	0.00	0.00	0.72	OC
Recall	0.75	0.00	0.00	0.7	
F-score	0.65	0.00	0.00	0.71	
Acc/MAF	**0.64** / **0.34**				
Precision	0.57	0.00	0.00	0.72	SC
Recall	0.75	0.00	0.00	0.7	
F-score	0.65	0.00	0.00	0.7	
Acc/MAF	**0.64** / **0.34**				
Precision	0.57	0.00	0.00	0.72	CW
Recall	0.75	0.00	0.00	0.70	
F-score	0.65	0.00	0.00	0.71	
Acc/MAF	**0.64 / 0.34**				
Precision	0.55	0.00	0.00	0.69	OS
Recall	0.71	0.00	0.00	0.68	
F-score	0.62	0.00	0.00	0.69	
Acc/MAF	0.62 / 0.33				

Table 9: Linear *TF-IDF* model results.

6.8 Recurrent network with embeddings

All the models presented above confront the learning task with no prior knowledge of the Spanish language, a trait that might limit the performance on some applications. A common approach to inject some semantic and syntactic knowledge is to make use of word embeddings (Mikolov, 2013), whereby a numerical vector representative of each word is pre-trained with a large unannotated corpus, then used as inputs for the task at hand instead of the original words. In this work we use the Spanish embeddings provided by fasttext project (Bojanowski, 2016).

The simplest way to use word embeddings is to compute a document embedding as the average of embeddings the words in the document, then feed such sentence vector into a machine learning model (e.g. Random Forest). However, this approach turned out to produce very poor results for our task. Instead, we resort to implementing a small recurrent neural network with GRU layers (Cho, 2014) to obtain a better mixing of the embedding vectors.

The network is comprised of an Embedding layer, 1 to 3 GRU layers (the first one bidirectional), global average pooling and 1 to 3 dense layers with ReLU activations. Dropouts are added at the embeddings, GRU and dense levels to prevent overfitting. We do not fine-tune the embedding vectors. Since many parameters in the network design are susceptible to tuning, we run a Bayes Search optimization strategy, as implemented in scikit-learn. With this, we are able to attain an accuracy of 0.62 and macro average f1 of 0.41.

Metric	expl	iobj	obj	Se-mark
Precision	0.56	0.29	0.50	0.71
Recall	0.71	0.19	0.07	0.64
F-score	0.63	0.23	0.12	0.68
Acc/MAF	0.62 / 0.41			

Table 10: *Recurrent network with embeddings model* results.

6.9 es-BERT[11]

Recent advances in statistical NLP are mainly based on making use of a fully pre-trained deep neural network that models the conditional distribution of tokens in a specific language: a language model. In particular, the BERT model has proven very successful in many applications (Devlin, 2018). Such model is adapted to specific NLP tasks through a so-called fine-tuning procedure. The first BERT-based model for Spanish is *es-BERT* (Cañete, Chaperon and Fuentes, 2020). We used the Transformers library (Wolf et al, 2019) to train an es-BERT classifier. Following a similar approach to the previous model, to perform the hyperparameter tuning we follow a Bayes Search strategy. The resulting accuracy goes up to 0.83 points and the macro average raises to 0.70.

Metric	expl	iobj	obj	Se-mark
Precision	0.75	0.71	0.50	0.95
Recall	0.89	0.65	0.36	0.84
F-score	0.82	0.68	0.42	0.89
Acc/MAF	0.83 / 0.70			

Table 11: *es-BERT* results.

6.10 UD-Pipe

UD-Pipe (Straka and Straková, 2017) (Straka, Hajič and Straková, 2016) is a state-of-the-art, embedding-based,[12] dependency parsing tool,

[11] An adaptation of Barbero (2020) was used to train transformer-based models.

[12] UD-Pipe is a neural network parser based on embeddings. Form embeddings are adjusted from

capable of analyzing different linguistic aspects (lemma, PoS, morphological features, dependency relations) of each token of a sentence encoded in CoNLL-U (Universal Dependencies, 2020) format.[13] The model used in text classification is *Spanish-gsd-ud-2.5-191206.udpipe* (Ballesteros et al., 2019).[14] To predict the tags assigned to each instance of *se* the whole architecture (tokenizer, tagger and parser) is re-trained using the default parameter configuration. The results achieved go up to 0.62 points of accuracy and 0.45 points of macro average F-score.

Metric	**expl**	**iobj**	**obj**	**Se-mark**	**BS**
Precision	0.56	0.64	0.00	0.70	OC
Recall	0.72	0.29	0.00	0.62	
F-score	0.63	0.40	0.00	0.66	
Acc/MAF	**0.62** / 0.42				
Precision	0.61	0.35	0.10	0.70	OS
Recall	0.55	0.39	0.21	0.70	
F-score	0.58	0.37	0.14	0.70	
Acc/MAF	0.60 / **0.45**				

Table 12: *UD-Pipe* results.

7 Results

Table 13 shows the highest accuracy and macro average F-score values obtained for each of the models and the parser in the different training experiments. The highest accuracy value is reached by *es-BERT* model (0.83). The highest macro average f-score is also achieved by *es-BERT* model (0.70).

It is important to mention that the value accuracy reached for most models doubles the macro average F-score. However, in the case of models that make use of some kind of transfer learning (recurrent network with embeddings, BERT and *Spanish-gsd-ud-2.5-191206.udpipe)* the difference between accuracy and macro average F-score values is around 0.13-0.20 points. This might mean that, whereas classical classification models always pay more attention to the most frequent tags, models making use of prior knowledge seem to take more into consideration the whole tag set distribution. This hypothesis is supported by the precision, recall and f-score values obtained for the less frequent tags *iobj* and *obj*: whereas BERT-like and UD-Pipe model learn to discriminate the four categories, the non-linear bag of words and non-linear TF-IDF models learn to discriminate the three most frequent categories *se-mark, expl* and *iobj,* but ignore the category *obj*. Linear bag of words, hashing and linear TF-IDF models together with non-linear hashing model only learn to discriminate the two most frequent categories *se-mark* and *expl*, paying no attention to *iobj or obj* cases. Besides, it is important to mention that the best performing models make use of transfer learning: they use and adjust already learnt information whereas classic models need to learn to disambiguate from scratch without any other additional information. Furthermore, the very best results are obtained by BERT, showing that doing transfer learning of not just the word representations but also the mixing layers contributes positively to this task. Our hypothesis is that syntactic knowledge of the Spanish language is required to perform *se* classification correctly, and so the pre-trained Transformer layers are providing critical contextual information to expose such syntactic elements. It is also remarkable how the performance of BERT is close in accuracy to that of an expert human annotator, though a gap still exists in f1-score due to misclassifications in minority classes.

Having a look at the confusion matrix obtained from the best classification model, es-Bert, it can be seen that class frequency is directly related to the higher accuracy results: the model learns better to classify the most frequent classes *expl* and *se-mark*. On the contrary, the model gets worse results for the less represented classes *iobj* and *obj*.

It is important to mention that the model never predicts the tag *iobj* in front of direct object *se* or valency reduction values of *se*. Besides, the model rarely predicts the tag *se-mark* for argumental *se cases.* However, the model gets confused and sometimes assigns the tag *expl* to argumental uses of *se* (14)-(15) and the other way round (16).

Spanish word2vec embeddings. The rest of embedding layers are randomly started and adjusted along the training procedure. See appendix C for more information on UD-pipe architecture.

[13] There are other state of the art parsing tools such as FreeLing (Padró & Stanilovsky, 2012), Ixa-pipes (Agerri, Bermudez & Rigau, 2014), Stanza (Qi et al., 2020) or Spacy (Honnibal & Montani, 2017). Usability and training ease have been key aspects for the selection of UD-Pipe.

[14] See Straka and Straková (2017) and Straka and Straková (2019) for a detailed description of the training and hyperparameter adjustment procedure.

Model	Accuracy	Macro Avg
Baseline model	0.49	0.16
CountVectorizer + RandomForestClassifier + GridSearchCV	0.61	0.33
CountVectorizer + OneVsRestClassifier + LinearSVC	0.61	0.32
HashingVectorizer + RandomForestClassifier + GridSearchCV	0.62	0.33
HashingVectorizer + OneVsRestClassifier + LinearSVC	0.61	0.32
TF-IDF + RandomForestClassifier + GridSearchCV	0.65	0.34
TF-IDF + OneVsRestClassifier + LinearSVC	0.65	0.34
Recurrent network with embeddings	0.62	0.41
es-BERT	**0.83**	**0.70**
Spanish-gsd-ud-2.5-191206.udpipe	0.62	0.45
Expert human annotator (average)	0.88	0.85

Table 13: Summary of best results.

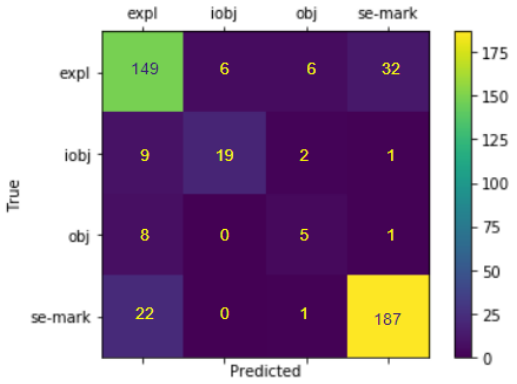

Figure 2: es-BERT confusion matrix.

(14) El maestro José Fernández
The maestro José Fernández
se ha propuesto redescubrir
him-DAT have-PRS.3SG propuesto redescubrir […].
propose-PTCP rediscover-INF […].
'Maestro José Fernández has proposed himself to rediscover […].'

(15) Aquí cerca , el joven Tomás
Here nearby, the young Tomás
Rodaja […] , **se**
Rodaja […] , **him-ACC**
ofrecía como
offer-PST.IMPV.3SG as
criado […] .
servant […] .

(16) […] atraca su velero […] , se
[…] docks his boat […] , him
alquila una villa
rent-PRS.3SG a vaccation-house
o dos y juega con […].
or two and play-PRS.3SG with […].

8 *Conclusions and further work*

Se constructions constitute a complex linguistic phenomenon that challenges annotation criteria creation, annotation and automatic classification tasks. Transformer-based models entail exceptional advantages for complex classification problems like the one posed by *se* constructions, obtaining the highest accuracy and f-score classification values. Corpus unbalance is an important factor affecting the results, which prevents attaining automated annotations on par with those of an expert human annotator. Thus, future work needs to be done into the following research lines: first, enlarging the existing *SE*-corpus while maintaining the real distribution of *se*-constructions, and second, evaluating whether this enlarged version of the *SE-corpus* may palliate category unbalance improving classification results. Another open research line is to study how to integrate a *se* construction classifier as an extra module into a NLP pipeline to turn it into a general use tool.

Acknowledgements

We would like to thank Cristina Sánchez López and Amaya Mendikoetxea Pelayo for their help and dedication in the development of this study. The authors acknowledge financial support from PID2019-106827GB-I00 / AEI / 10.13039/501100011033 and from the European Regional Development Fund and from the Spanish Ministry of Economy, Industry, and Competitiveness - State Research Agency, project TIN2016-76406-P (AEI/FEDER, UE).

References

Agerri, R., J. Bermudez and G. Rigau. 2014. IXA pipeline: Efficient and Ready to Use

Multilingual NLP tools. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC2014), May, 2014, Reykjavik, Iceland.*

Artstein, R. 2017. Inter-annotator Agreement. In Ide, N. and J. Pustejovsky (Eds). *Handbook of Linguistic Annotation*. Springer: Dordrecht, 297-314.

Bayerl, P. and K. Paul. 2011. What determines inter-coder agreement in manual annotations? Ametaanalytic investigation. *Computational Linguistics*. 37(4): 699-725.

Ballesteros, M., H. Martínez, R. McDonald, E. Pascual, N. Silveira, D. Zeman and J. Nivre. 2019. Spanish-gsd-ud-2.5-191206.udpipe https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3105. Accessed date: 23/09/2020.

Barbero, A. 2020. Training transformer models. https://github.com/Spain-AI/transformers. Accessed date: 26/09/2020.

Bojanowski, P., E. Grave, A. Joulin and T. Mikolov. 2016. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

Breiman, L. 2001. Random Forests. *Machine Learning 45*, 5-32. https://doi.org/10.1023/A:1010933404324.

Campos, H. 1999. Transitividad e intransitividad. In Bosque, I. and V. Demonte (Eds.) *Gramática descriptiva de la Lengua Española*. Madrid: Espasa. V2, 1519-1574.

Cañete, J., G. Chaperon and R. Fuentes. 2020. Spanish pre-trained BERT model and evaluation data. To appear in *ICLR 2020 workshop*. https://users.dcc.uchile.cl/~jperez/papers/pml4dc2020.pdf. Accessed date: 22/09/2020.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Devlin, J., M. Chang, K. Lee and K. Toutanova. 2018. BERT: Pre-trained of Deep Bidirectional Transformers for Language Understanding. *Computer Science*. https://arxiv.org/abs/1810.04805. Accessed date: 22/09/2020.

Fan, R., K. Chang, C. Hsieh, X. Wang and C. Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9, 1871-1874.

Fernández Montraveta, A., and G. Vázquez. 2017. *Las construcciones con se en español (Cuadernos de lengua española 130).* Madrid: Arco/Libros-La Muralla.

Honnibal, M. and I. Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Mendikoetxea, A. 1999 a. Construcciones con se: Medias, pasivas e impersonales. In Bosque, I. and V. Demonte (Eds.) *Gramática descriptiva de la Lengua Española*. Madrid: Espasa. V2, 1631-1722.

Mendikoetxea, A. 1999 b. Construcciones inacusativas y pasivas. In Bosque, I. and V. Demonte (Eds.) *Gramática descriptiva de la Lengua Española*. Madrid: Espasa. V2, 1575-1630.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Moreno, J.C. 1997. *Introducción a la lingüística general: un enfoque tipológico y universalista.* Madrid: Editorial Síntesis.

Moreno, J.C. 2002. *Curso Universitario de Lingüística General*. Madrid: Editorial Síntesis.

Padró, L. and E. Stanilovsky. 2012. FreeLing 3.0: Towards Wider Multilinguality. In *Proceedings of the Language Resources and Evaluation Conference (LREC 2012) ELRA May, 2012. Istanbul, Turkey*.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. https://scikit-learn.org. Accessed date: 22/09/2020

Qi, P., Y. Zhang, Y. Zhang, J.Bolton, and C.D. Manning. 2020. Stanza: A Python Natural

Language Processing Toolkit for Many Human Languages. In *Association for Computational Linguistics (ACL) System Demonstrations*, 101-108.

Real Academia Española de la Lengua. 2020. Banco de datos (CORPES XXI) [online]. Corpus del Español del Siglo XXI (CORPES) https://www.rae.es/recursos/banco-de-datos/corpes-xxi. Accessed date: 20/09/2020

Sánchez, C. 2015. Se y sus valores. In Gutiérrez Rexach, J. (Ed.) *Enciclopedia de Lingüística Hispánica*, Vol. 2, 1-12.

Sánchez, C. 2002. Las construcciones con se (Gramática del Español, 8). Madrid: Visor Libros.

Scikit optimize. https://scikit-optimize.github.io/stable/. Accessed date: 15/11/2020.

Shmueli, B. 2019. Multi-Class Metrics Made Simple, Part II. *Towards Data Science*. https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1. Accessed date: 20/09/2020.

Shung, K. 2018. Accuracy, Precision, Recall or F1? *Towards Data Science*. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. Accessed date: 20/09/2020.

Straka, M. and J. Straková. 2019. Universal Dependencies 2.5 Models for UDPipe (2019-12-06). https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3131. Accessed date: 21/11/2020

Straka, M. and J. Straková. 2017. Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, Vancouver, Canada, August.

Straka M., J. Hajič and J. Straková. 2016. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, May.

Universal dependencies. 2020. *CoNLL-U format*. https://universaldependencies.org/format.html. Accessed date: 23/09/2020.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998-6008.

Weinberger, K., A. Dasgupta, J. Langford, A. Smola, and J. Attenberg. 2009. Feature hashing for large scale multitask learning. In *Proceedings of the 26th annual international conference on machine learning*, 1113-1120.

Wolf, T. L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest and A. M. Rush. 2019. Transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.

A Supplementary material

To aid in the reproducibility of the results presented in this manuscript, both the SE-corpus and a Jupyter notebook with the experimental procedure are available as supplementary material at the following link: https://github.com/albarji/sepln-spanish-se-constructions.

B Hyperparameter ranges

The following hyperparameter ranges were used for searching for optimal model parameters.

Model	**Parameter**	**Values**
Count Vectorizer / Hashing Vectorizer	Analyzer	char_wb
	Binary counts	[False, True]
	N-gram range	[(1,1), (1,2), (1,3), (1,4), (1,5), (2,2), (3,3), (3,5), (5,5), (5,7), (7,7), (10,10)
Random Forest	Number of estimators	[1, 10, 100]
LinearSVC	C	[1e-4, 1e-3, …, 1e4]
Recurrent network with embeddings	Spatial dropout	[0.0, …, 0.9]
	GRU layers	[1, 2, 3]
	GRU units	[16, 32, …, 1024]
	GRU dropout	[0.0, …, 0.9]
	Dense layers	[1, 2, 3]
	Dense units	[16, 32, …, 1024]
	Dense dropout	[0.0, …, 0.9]

	Training epochs	[50, …, 200]
es-BERT	Model casing	[cased, uncased]
	Learning rate	[10^{-6}, …, 10^{-4}]
	Training epochs	[1, …, 10]
	Batch size	[4, 8, 16, 32, 64]
	Attention dropout	[0.0, …, 0.9]
	Hidden dropout	[0.0, …, 0.9]

Table 14: Hyperparameter ranges.

C Spanish-gsd-ud-2.5-191206.udpipe params

The following parameters are the ones used along the training procedure of Spanish-gsd-ud-2.5-191206.udpipe. The same params are used to perform the experiments in 6.9.

Module	**Parameter**	**Values**
Tokenizer	Dimension	24
	Epochs	100
	Segment_size	200
	Initialization_range	0.1
	Batch_size	50
	Learning_rate	0.005
	Learning_rate_final	0
	dropout	0.2
	early_stopping	1
Tagger	es_gsd models	2
	templates_1	tagger
	guesser_suffix_rules_1	10
	guesser_enrich_ dictionary_1	6
	guesser_prefixes_ max_1	0
	use_lemma_1	1
	use_xpostag_1	1
	use_feats_1	1
	provide_lemma_1	0
	provide_xpostag_1	1
	provide_feats_1	1
	prune_features_1	0
	templates_2	lemmatizer
	guesser_suffix_rules_2	4
	guesser_enrich_ dictionary_2	4
	guesser_prefixes_ max_2	4
	use_lemma_2	1
	use_xpostag_2	1
	use_feats_2	1
	provide_lemma_2	1
	provide_xpostag_2	0
	provide_feats_2	0
	prune_features_2	0
Parser	es_gsd iterations	30
	embedding_upostag	20
	embedding_feats=20	20
	embedding_xpostag	0
	embedding_form	50
	embedding_form_file	=../ud-2.5-embeddings/ es_gsd.skip. forms.50. vectors
	embedding_lemma	0
	embedding_deprel	20
	learning_rate	0.01
	learning_rate_final	0.001
	l2	0.5
	hidden_layer	200
	batch_size	10
	transition_system	Link2
	transition_oracle	Static
	structured_interval	8

Table 15: Spanish-gsd-ud-2.5-191206.udpipe params.