bRol: The Parser of Syntactic and Semantic Dependencies for Basque

Haritz Salaberri, Olatz Arregi, Beñat Zapirain
IXA Group - Faculty of Computer Sciences
University of the Basque Country, Spain
haritz.salaverri@ehu.eus

Abstract

This paper presents *bRol*, the first fully automatic system to be developed for the parsing of syntactic and semantic dependencies in Basque. The parser has been built according to the settings established for the *CoNLL-2009* Shared Task (Hajič et al., 2009), therefore, *bRol* can be thought of as a standard parser with scores comparable to the ones reported in the shared task. A second-order graph-based MATE parser has been used as the syntactic dependency parser. The semantic model, on the other hand, uses the traditional four-stage SRL pipeline.

The system has a labeled attachment score of 80.51%, a labeled semantic F_1 of 75.10, and a labeled macro F_1 of 77.80.

1 Introduction

Since 1999 *The Conference on Natural Language Learning* (CoNLL) has been holding shared tasks focusing around different topics which concern human language processing. The CoNLL Shared Task aims to evaluate such applications in a standard setting, and to establish, as a result, the evaluation measures according to which these systems are evaluated and compared with one another.

In 2009 participants had to choose between two tasks: the joint parsing of syntactic and semantic dependencies or a SRL-only task. In both cases dependencies had to be parsed for propositions centered around verbal and, in some cases, nominal predicates in seven different languages (Catalan, Chinese, Czech, English, German, Japanese and Spanish). The representation used to perform and evaluate SRL was a dependency-based representation for both the syntactic and the semantic dependencies. Our focus is on the parsing of syntactic and semantic dependencies for Basque. In addition to describing our parser and presenting our results we also attempt to make a correct reading of these by taking into account the morphological and typological nature of Basque.

bRol is implemented as a sequence of five cascaded subtasks: Syntactic parsing (D), predicate identification (PI), predicate classification (PC), argument identification (AI) and argument classification (AC). Additionally, a post-process method is performed in order to relabel the duplicated role labels that may be assigned to predicate arguments in the AC subtask. Each of these subtasks is addressed by using a separate component with no backwards feedback between them.

Section 2 lists the resources used, section 3 and 4 describe the syntactic submodel and the semantic submodel, respectively. Results are shown in section 5 and section 6 presents our conclusions.

2 Resources

In order to develop *bRol*, the Basque corpus EPEC, also known as the *Basque PropBank*, is used (Aldezabal et al., 2010). EPEC is a corpus of text annotated with information about basic semantic propositions. Predicate-argument relations were added to the syntactic trees in the corpus using the *Basque Verb Index* (BVI) verb lexicon, also known as the *Basque VerbNet* (Aldezabal et al., 2013). Each entry in BVI is linked to the corresponding verb entry in well-known resources such as PropBank, VerbNet, WordNet and the Levin classes. A Basque NomBank, which has not been developed yet, is necessary in order to build a parser capable of labeling arguments for nominal predicates.

2.1 The EPEC corpus

One half of the text contained in EPEC was extracted from the *Statistical Corpus of 20th Century Basque*. The other half was extracted from newspaper extracts from the *Euskaldunon*

Egunkaria, the only daily newspaper written entirely in Basque.

Syntax is annotated following the dependency-based formalism used in the *Prague Dependency Treebank* and the syntactic tag set consists of 30 different labels. Regarding semantic arguments we distinguish A0, A1, A2, A3, A4 and AM, which corresponds to adjuncts. There are 12 different types of adjuncts. Some other features of the corpus are: (1) the number of different verbs is 1,242; (2) there are 10,379 sentences and 161,812 tokens; (3) the language variety is the standard variety of Basque; and (4) all preprocessing steps (e.g. lemmatization) and the annotations of linguistic features (PoS, syntax, SRL, etc) in the corpus are manual.

Statistics on our data can be seen and compared to the ones in the CoNLL-2009 Shared Task in tables 1, 2 and 3. These statistics reflect several key features of the addressed languages, such as the degree of inflectionality, as well as features related to the annotation specification and conventions used.

2.2 The BVI verb lexicon

The Basque Verb Index (BVI) was created manually. Initially, it contained the verbs in the Database for Basque Verbs (EADB) proposed in (Aldezabal, 2004), an in-depth study of 100 verbs selected from the 622 that occur in the *Statistical Corpus of 20th Century Basque*. When EPEC was built BVI was extended from the initial 100 verbs to 243 verbs. These verbs are the ones with a minimum of 30 occurrences in the corpus.

3 Syntactic dependency parsing

The two main approaches to dependency parsing are transition-based dependency parsing (Nivre, 2003) and Maximum Spanning Tree-based dependency parsing (McDonald and Pereira, 2006). Our system uses MATE (Bohnet, 2010), a Maximum Spanning Tree-based dependency parser (also known as graph-based or MST-based). In MST-based dependency parsing the directed graph $G_x = (V_x, E_x)$ is defined for each sentence x where

$$V_x = \{x_0 = root, x_1, ..., x_n\}$$
$$E_x = \{(i, j) : x_i = x_j, x_i \in V_x, x_j \in V_x - root\}$$

That is, G_x is a graph where all the words and the root symbol are vertices and there is a directed edge between every pair of words and from the root symbol to every word. Dependency trees for x and spanning trees for G_x coincide, since both kinds of trees are required to reach all the words in the sentence. Therefore, finding the dependency tree of highest score is equivalent to finding the maximum spanning tree in G_x rooted in the root (McDonald et al., 2006).

The MATE parser used in *bRol* consists of the second-order parsing algorithm described in (Carreras, 2007), the non-projective approximation algorithm in (McDonald and Pereira, 2006) used to handle non-projective dependency trees, the passive-aggressive *SVM* algorithm and a feature extraction component. The second-order algorithm has a complexity of $O(n^4)$.

3.1 Non-projectivity

The total number of syntactic links in the training set of EPEC is 108,003 and out of these 2224 (2.06%) are non-projective. The number of sentences that contain at least one non-projective link is 1078, which constitute 15.5% of the sentences in the training set. These values are higher than the values reported for non-projectivity in, for example, the training set of English for the CoNLL-2009 shared task (0.4% of non-projective links and 7.6% sentences with at least one non-projective link).

According to (Johansson and Nugues, 2008) non-projectivity cannot be handled by span-based dynamic programming algorithms. Normally, the Chu-Liu/Edmonds algorithm (Chu and Liu, 1965) is used to find the highest scoring non-projective spanning tree in directed graphs; nevertheless, this algorithm cannot be extended to the second order (McDonald et al., 2006) and for this reason MATE uses the Non-Projective Approximation Algorithm in (McDonald and Pereira, 2006).

3.2 Features

In order to select the features for the syntactic dependency parser we took into account that Basque, on the contrary to English, Chinese, Spanish and Catalan, is a morphologically rich language (MRL) that exhibits a high degree of inflectional and derivational morphology. It is stated in (Nilsson et al., 2007) that the use of state-of-the-art parsers for non-inflecting languages like English does not reach similar performance levels when labeling MRLs like Basque. To overcome this difference, morphological information is normally used as a feature for parsing languages.

Based on the results reported in (Goenaga et al., 2013) we selected the following features from the ones annotated in EPEC: (1) declension case; (2) number; (3) type of subordinate sentence.

4 Semantic dependency parsing

From the sequence of five cascaded subtasks mentioned in section 1, all but the first form the semantic dependency parsing module (PI+PC+AI+AC). In addition, a post-process method is used to relabel duplicate roles.

First, verbal predicates are identified (PI) by examining every word in a sentence. Then, a certain roleset-ID is assigned to the words that have been marked as predicates (PC). Next, target arguments are discovered for the predicate(s) in a sentence (AI). Finally, the words that have been targeted as arguments are assigned a semantic role label by the default classifier (AC). Duplicated roles are relabeled using an Integer Linear Programming-based method (ILP post-process).

Classifiers: The classifiers used in the four-stage SRL pipeline of *bRol* are *Support Vector Machine* classifiers implemented using the *SVM-light* and *SVM-multiclass* packages (Joachims, 1999). The *SVM-light* package is used for binary classification (e.g. PI); the *SVM-multiclass* package, on the other hand, is used for multi-class problems (e.g. PC). The type of kernel function used is linear and the trade-off between training error and margin is computed through the $avg(x * x)^{-1}$ formula.

For the argument classification a maximum entropy classifier implemented with the MEGA package (Daumé III, 2004) is used. The specified minimum change in perplexity for the classifier is -99999 and the precision of the Gaussian prior is 1. The reason not to use a *Support Vector Machine* classifier for argument classification is motivated by the fact that standard *SVM* classifiers do not produce the posterior probability values ($P(class|input)$) that are needed, in our case, for the ILP post-process method (Platt et al., 1999).

Feature Selection: In order to select useful features for semantic dependency parsing we initially studied the features that were used by the participants in the the CoNLL-2009 Shared Task. Additionally, we also took into account the features that we proved to be useful for the classification of arguments in Basque (Salaberri et al., 2014). We then followed a Leave-One-Out (LOO) procedure to determine the impact that each individual feature had in each semantic subtask. This procedure evaluated the value of each feature that had been initially considered by iteratively removing the information relative to that feature and by then training the classifier with the rest of features.

4.1 Predicate Identification (PI)

We have treated the predicate identification subtask as a binary classification problem. Every word in a sentence is viewed as a candidate to be a predicate (punctuation marks are previously excluded from the candidates list for obvious reasons). For each candidate word a set of features is extracted. The following is the list of the features used:

WORD_Lex, WORD_Lemma, WORD_PoS, WORD_SubPoS, WORD_DepRel, HEAD_Lex, HEAD_Lemma, HEAD_PoS, HEAD_SubPoS, CHILD_DependRel_Set, CHILD_Lemma_Set, CHILD_Lex_Set.

4.2 Predicate Classification (PC)

After identifying the predicates from the list of candidate words, a roleset-ID is assigned to these predicates. For this purpose a single multiclass classifier is trained for all the predicates that have multiple senses (roleset-IDs). From the 243 different predicates in our training set 80 have multiple senses and 163 have a single sense. The following list shows the features that have been used:

PRED_Lex, PRED_Lemma, PRED_PoS, PRED_SubPoS, PRED_DepRel, PRED_DecCas, HEAD_Lex, HEAD_Lemma, HEAD_PoS, HEAD_SubPoS, CHILD_DependRel_Set, CHILD_Lemma_Set, CHILD_Lex_Set.

4.2.1 Handling ”new” predicates

We stated in section 2 that predicate-argument relations were added to the syntactic trees in the EPEC corpus using the BVI verb lexicon. The number of different verbs that can be found in EPEC is 1,242 and the number of verbs in the BVI verb lexicon is 243 as stated in section 2. These values indicate that 999 verbs in the corpus have no manually labeled predicate-argument relations. As a result *bRol*, which uses EPEC as a training corpus, would only be capable of assigning a

Characteristics	Basque	Catalan	Chinese	Czech	English	German	Japanese	Spanish
Training data size (sent.)	6941	13200	22277	38727	39279	36020	4393	14329
Training data size (tokens)	108003	390302	609060	652544	958167	648677	112555	427442
Avg. Sent length	15.56	29.6	27.3	16.8	24.4	18.0	25.6	29.8
Tokens with arguments (%)	10.75	9.6	16.9	63.5	18.7	2.7	22.8	10.3
DEPREL types	30	50	41	49	69	46	5	49
POS types	26	12	41	12	48	56	40	12
FEAT types	298	237	1	1811	1	267	302	264
FORM vocabulary size	20051	33890	40878	86332	39782	72084	36043	40964
LEMMA vocabulary size	9042	24143	40878	37580	28376	51993	30402	26926
Evaluation data size (sent)	3438	1862	2556	4213	2399	2000	500	1725
Evaluation data size (tokens)	53809	53355	73153	70348	57676	31622	13615	50630
Evaluation FORM OOV	12.41	5.40	3.92	7.98	1.58	7.93	6.07	5.63
Evaluation LEMMA OOV	6.38	4.14	3.92	3.03	1.08	5.83	5.21	3.69

Table 1: Elementary data statistics for the CoNLL-2009 Shared Task languages plus the statistics for Basque (EPEC). All evaluation data statistics are derived from the in-domain evaluation data.

DEPREL	Basque		Catalan		Chinese		Czech		English		German		Japanese		Spanish	
Labels	ncmod	0.26	sn	0.16	COMP	0.21	Atr	0.26	NMOD	0.27	NK	0.31	D	0.93	sn	0.16
	PUNC	0.15	spec	0.15	NMOD	0.14	Aux	0.10	P	0.11	PUNC	0.14	ROOT	0.04	spec	0.15
	lot	0.09	f	0.11	ADV	0.10	Adv	0.10	PMOD	0.10	MO	0.12	P	0.03	f	0.12
	auxmod	0.08	sp	0.09	UNK	0.09	Obj	0.07	SBJ	0.07	SB	0.07	A	0.00	sp	0.08
	ncsubj	0.07	suj	0.07	SBJ	0.08	Sb	0.06	OBJ	0.06	ROOT	0.06	I	0.00	suj	0.08
Total	0.65		0.58		0.62		0.59		0.61		0.70		1.00		0.59	

Table 2: Unigram probability is shown for the five most frequent DEPREL labels in the training data of the CoNLL-2009 Shared Task and in the training data from the EPEC corpus. Total is the probability mass covered by the five dependency labels shown.

APRED	Basque		Catalan		Chinese		Czech		English		German		Japanese		Spanish	
Labels	A1	0.21	arg1-pat	0.22	A1	0.30	RSTR	0.30	A1	0.37	A0	0.40	GA	0.33	arg1-pat	0.20
	A2	0.15	arg0-agt	0.18	A0	0.27	PAT	0.18	A0	0.25	A1	0.39	WO	0.15	arg0-agt	0.19
	A0	0.14	arg1-tem	0.15	ADV	0.20	ACT	0.17	A2	0.12	A2	0.12	NO	0.15	arg1-tem	0.15
	AM-TMP	0.08	argM-tmp	0.08	TMP	0.07	APP	0.06	AM-TMP	0.06	A3	0.06	NI	0.09	arg2-atr	0.08
	AM-MNR	0.07	arg2-atr	0.08	DIS	0.04	LOC	0.04	AM-MNR	0.03	A4	0.01	DE	0.06	argM-tmp	0.08
Total	0.65		0.71		0.91		0.75		0.83		0.97		0.78		0.70	
Avg.	1.97		2.25		2.26		0.88		2.20		1.97		1.71		2.26	

Table 3: Unigram probability is shown for the five most frequent APRED labels in the training data of the CoNLL-2009 Shared Task and in the training data from the EPEC corpus. Total is the probability mass covered by the five argument labels shown.

roleset-ID and consequently semantic role labels to instances of the 243 verbs in the lexicon.

We decided to add a translation component (TC) to the PC problem. The TC is used to assign a roleset-ID to instances of the 999 predicates, or any other new verb predicate, that is not mapped in BVI. By using this component we achieve an increase in number of predicate-argument relations that are labeled by *bRol*. The relations labeled as a result of the TC can not be compared to any manual annotation; therefore, the performance of the TC can not be evaluated. Nevertheless, we believe that the TC is able to correctly label many predicate-argument relations, since these relations correspond to predicates that have less than 30 occurrences in the corpus (usually, these infrequent verbs have only one roleset-ID in PropBank).

The translation component is implemented using the Basque-to-English *Elhuyar Hiztegia* dictionary and PropBank. When a word that has been targeted as a predicate at the PI stage is handed over to the PC stage, *bRol* checks whether or not this predicate is present in the lexicon. If the predicate can not be found, it is delivered to the TC.

The translation component operates in the following way: first the predicate is translated into English; then the translation is looked for in PropBank (PB). If the translation can be found as a PropBank frame, the first roleset-ID mapped for this frame is assigned to the original predicate. If

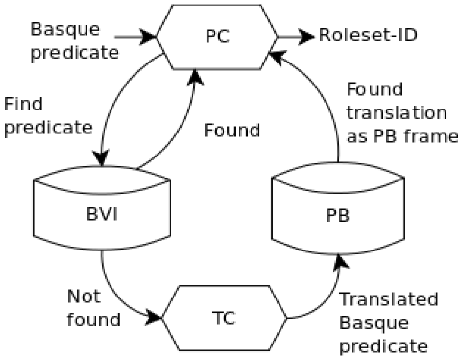

Figure 1: PC pipeline.

not, the original predicate will not be assigned a roleset-ID and consequently its arguments will not be labeled. Figure 1 illustrates the PC pipeline.

4.3 Argument Identification (AI)

After the PC subtask is completed and predicates are assigned a roleset-ID sentences are handed to the AI module. *bRol* performs argument identification based on a high precision heuristic. Every word in a sentence is treated as a candidate to be an argument for each semantic predicate (in the sentence).

Our heuristic uses information such as the predicted PoS tag, the syntactic head (HEAD) and dependency relation to the head (DEPREL) in order to determine if word w_i is an argument for predicate P_j. More precisely, if word w_i's head is predicate P_j and the dependency relation is not labeled as *auxmod* (auxiliary), *haos* (component of a multiword lexical unit), *postos* (component of a multiword postposition), *entios* (component of a multiword entity) or *PUNC* (punctuation), then, w_i is considered to be an argument of P_j but only if P_j's PoS tag is not ADK (phrasal verb). We came up with the optimal argument identification heuristic after several train-test runs.

We performed several experiments in order to determine which approach, the machine learning-based or the heuristic-based, would prove to be the best for AI. We concluded the heuristic-based approach to be the best; in addition to a slightly higher performance, the running time is reduced thanks to the fact that there is no need for a feature extraction component (these are usually the most time-consuming components in ML-based systems).

4.4 Argument Classification (AC)

When predicate argument identification by the AI component has been completed the arguments that have been identified are handed over to the AC component. Our system treats argument classification as a multi-class classification problem; the machine-learning method used in this stage is maximum entropy. The model gives every argument a probability to take each semantic role and the one with the highest value is assigned to the argument. The features used are shown in the following list:

PRED_Roleset, PRED_Lemma, ARG_Lemma, ARG_PoS, ARG_SubPoS, ARG_DependRel, ARG_DecCas.

4.5 The post-process method

Before the final semantic role labeling result is generated, a post-process similar to the one described in (Che et al., 2008) is performed. The arguments corresponding to the same predicate which have been labeled with the same core argument label by the AC component are relabeled through a Integer Linear Programming-based method (ILP).

In some languages, as for example English, the possibility to have duplicated roles exists. Statistics show that most roles usually appear only once for a predicate; nevertheless, some rare cases exist. Before starting with the development of *bRol* we examined the verbs in our lexicon one by one; we did not find any duplicated roles.

Our system uses the probabilities given by the maximum entropy model in the AC component in order to perform the relabeling process. For every set of arguments which have been assigned a label that is duplicated for the predicate we maximize the objective function

$$f = \sum log(p_{ir}.v_{ir})$$

where v_{ir} is a binary variable indicating whether the argument indexed i (token ID) is assigned role $r \in R$ or not (where R is the set of role labels). p_{ir}, on the other hand, denotes the probability of the argument indexed i to be labeled as label r. We establish a No Duplicated Roles constraint and when the process is finished we obtain the optimal labeling for each predicate from the assignments to v_{ir}.

Measures	Basque	Catalan	Chinese	Czech	English	German	Japanese	Spanish
Labeled Attachment Score	80.51	87.86 (2)	79.17 (5)	80.38 (2)	89.88 (1)	87.48 (1)	92.57 (3)	87.64 (2)
Semantic Labeled F_1	75.10	80.10 (4)	77.15 (3)	86.51 (3)	86.15 (4)	78.61 (3)	78.26 (3)	80.29 (4)
Macro F_1 Score	77.80	83.01 (4)	76.38 (3)	83.27 (3)	87.69 (4)	82.44 (3)	85.65 (3)	83.31 (4)

Table 4: Official results of the Joint task (in-domain, closed challenge) reported by the teams that participated in the CoNLL-2009 Shared Task plus the results of *bRol*. The results shown correspond to the systems with the best performance. Teams are denoted by the last name of the author who registered for the evaluation data [(1):Bohnet, (2):Merlo, (3):Che, (4):Chen, (5):Ren].

5 Results and Discussion

In order to evaluate the performance of *bRol* we have run the scorer function from the CoNLL-2009 Shared Task (*eval09.pl*) on our test set. As of today there is no other Basque corpus than EPEC manually annotated with syntactic and semantic dependencies. For this reason the only available test set that can be used for evaluation is the one extracted from this corpus; thus, the only evaluation that can be made is an in-domain evaluation.

Table 4 shows the results obtained by our parser and the results reported by the participants in the Joint Task of the CoNLL-2009 Shared Task (in-domain, closed challenge). The results in the table correspond to the systems that, according to the language, performed best with respect to the official evaluation measures.

5.1 Syntactic Dependency Parsing

The Labeled Attachment Score (LAS) is defined as the percentage of tokens for which a parser has predicted the correct syntactic head and dependency relation. Our parser has a LAS of 80.51 points. If we compare our score with the ones reported for the other seven languages in table 4, our LAS is more than one point better than the score reported for Chinese (79.17) and 0.13 points better than the score reported for Czech (80.38). On the opposite site, our LAS is almost twelve points lower than the score reported for Japanese (92.57), nine points lower than the score reported for English (89.88) and almost seven points lower than the scores reported for Catalan (87.86), Spanish (87.64) and German (87.48).

We believe that several linguistic and data-related factors need to be addressed in order to correctly interpret this result. Linguistically speaking, we must bear in mind that, in general, the syntactic parsing results reported for morphologically rich languages (MRL) like Basque, despite the use of morphological features, do not reach the performance levels of languages like English. In the CoNLL-2009 Shared Task, for instance (see table 4), Czech and German, which are MRLs, get worse results than English, Spanish and Catalan, which are not MRLs. In our opinion the outstanding LAS score obtained by Japanese (92.57), which has an agglutinating morphology, is the result of having a DEPREL set of just five different labels (see tables 1 and 2). Chinese, on the other hand, which has a poor morphology, presents the worst labeled attachment score (79.17); we believe this score to be a result of the typological nature of Chinese; namely, that Chinese presents an isolating morphology, e.g. that each morpheme corresponds to an independent word or semantic unit and that therefore there is hardly any overt morphology. In fact, according to (Seddah et al., 2013) languages which are typologically farthest from English, such as Semitic and Asian languages, are still among the hardest to parse, regardless of the parsing method used.

In addition to the previously mentioned, another key factor in order to correctly interpret the LAS obtained by *bRol* is the free word order displayed by Basque syntax in combination with its rich morphology. As a matter of fact, (Donelaicio et al., 2013) state that it has been observed that richly inflected languages, which often also exhibit relatively free word order, obtain lower parsing accuracy, especially compared to English.

5.2 Semantic Dependency Parsing

bRol has a Semantic Labeled F_1 score of 75.10 points. The exact definition of how the Semantic Labeled F_1 score is computed can be seen in (Hajič et al., 2009) (section 2). As may be noticed in table 4, our result is two points lower than the result reported for Chinese (77.15), which is the language with the lowest Semantic Labeled F_1 score among the ones in CoNLL-2009.

We believe that the distribution of the APRED labels in our training data (see table 3) and other characteristics such as the number of PoS types or the number of FEAT types (see table 1) do not constitute any added difficulty when compared to the distribution and the characteristics in the other languages. In our opinion the only reason for this result in Basque, which compared to the results for the other seven languages can be understood as low or at least not average, is that the size of our training set is very reduced. In fact, the number of sentences in our training set is 6,941 and the number of tokens is 108,003. If we compare these to the average sentence and token number in the rest of the training sets (24,032 sentences and 542,678 tokens) we find that the number of sentences is 71.1% smaller and the number of tokens is 80.1% smaller in our training set.

Next we present the results for *bRol* through the four-stage SRL pipeline (see table 5). For this purpose we have used standard precision, recall and F1 score metrics.

Subtask	Precision	Recall	F_1
Predicate Identification (PI)	87.00	88.00	87.50
Predicate Classification (PC)	79.41	81.29	79.82
Argument Identification (AI)	72.70	86.10	78.80
Argument Classification (AC)	77.60	77.80	77.50

Table 5: Results for the semantic subtasks

The semantic subtask with the best F_1 score is predicate identification (87.50), followed by predicate classification (79.82) and argument identification (78.80). Argument classification, on the other hand, gets the lowest F_1 score (77.50). In our opinion, these results are highly dependent on the complexity of the subtask itself. In fact, PI is a binary classification problem, whereas PC and AC are multiclass classification problems.

Another way to approach the PC subtask would be by training a separate classifier for each predicate with multiple senses, as in (Che et al., 2008). Nevertheless, we decided not to implement *bRol* using separate PC classifiers for two reasons: (1) The size of our training set is too limited for this approach to be effective: we have 11,740 predicate instances and 8,166 correspond to the 80 verbs with multiple senses (69.55%). Thus, the average number of instances available for training each separate classifier is 102. We consider this amount to be too small. (2) We consider that the *PRED_Lemma* feature used to train our single PC classifier is given enough weight by the learning algorithm when training the classifier. We understand that operations where roleset-ID A_i of predicate A is assigned to predicate B are avoided.

5.3 Overall result

In order to compute the overall result of our parser, the syntactic and semantic measures (LAS and Semantic Labeled F_1 score) are combined into one global measure using Macro Averaging. The exact way in which this is achieved can be found in (Hajič et al., 2009) (section 2). The Macro F_1 score of *bRol* is 77.80. If we compare our score to the Macro F_1 scores reported in CoNLL-2009 (see table 4), we find out that our parser performs 1.42 points better than the result reported for Chinese. As opposed to this, *bRol* has a performance of about five Macro F_1 points lower than the results reported for Catalan, Spanish, Czech and German; eight points lower than the results reported for Japanese, and ten points lower than the results reported for English. Although the performance that our parser would have in an out-of-domain setup can not be evaluated, we believe that our results would drop in approximately 10 labeled macro F_1 points, as in the results reported for CoNLL-2005 (Carreras and Màrquez, 2005) and CoNLL-2009.

It is worth mentioning that before running *bRol* over the test set we deactivated the translation component, since the predicates and their corresponding arguments that would have been labeled as a consequence of the TC are not manually annotated in the test set. As a result, all of these would have been computed as fails although some might be correctly labeled by *bRol*.

6 Conclusions

We have presented the first fully automatic system to be developed for the parsing of syntactic and semantic dependencies in Basque. The evaluation measures we have used to evaluate our parser are the ones used in the CoNLL-2009 Shared Task, as we understand these to be the standard metrics used in order to evaluate these kind of applications. In addition, we have established a performance baseline for Basque and compared our results to the results reported for languages of different morphological and typological natures.

Acknowledgments

Haritz Salaberri holds a PhD grant from the University of the Basque Country (UPV/EHU). In addition, this work has been supported by the FP7 NewsReader project (Grant No. 316404) and IXA Group, research group of type A (2010-2015)(IT34410).

References

Izaskun Aldezabal. 2004. *Aditz-azpikategorizazioaren azterketa sintaxi partzialetik sintaxi osorako bidean: 100 aditzen azterketa, Levin-en lana oinarri hartuta eta metodo automatikoak baliatuz*. PhD thesis, UPV-EHU, Donostia.

Izaskun Aldezabal, Maxux Aranzabe, Arantza D. Ilarraza, and Ainara Estarrona. 2010. Building the basque propbank. In *LREC*.

Izaskun Aldezabal, Maxux Aranzabe, Arantza D. Ilarraza, and Ainara Estarrona. 2013. A methodology for the semiautomatic annotation of epec-rolsem, a basque corpus labeled at predicative level following the propbank-verb net model. *UPV/EHU/LSI/TR; 01-2013*.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97. Association for Computational Linguistics.

Xavier Carreras. 2007. Experiments with a higher-order projective dependency parser. In *EMNLP-CoNLL*, pages 957–961.

Xavier Carreras and Lluis Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164. Association for Computational Linguistics.

Wanxiang Che, Zhenghua Li, Yuxuan Hu, Yongqiang Li, Bing Qin, Ting Liu, and Sheng Li. 2008. A cascaded syntactic and semantic dependency parsing system. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 238–242. Association for Computational Linguistics.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On shortest arborescence of a directed graph. *Scientia Sinica*, 14(10):1396.

Hal Daumé III. 2004. Notes on cg and lm-bfgs optimization of logistic regression. *Paper available at http://pub. hal3. name# daume04cg-bfgs, implementation available at http://hal3. name/megam*, 198:282.

Ka Donelaicio, Joakim Nivre, and Algis Krupavicius. 2013. Lithuanian dependency parsing with rich morphological features. In *Fourth Workshop on Statistical Parsing of Morphologically Rich Languages*, page 12.

Iakes Goenaga, Koldo Gojenola, and Nerea Ezeiza. 2013. Exploiting the contribution of morphological information to parsing: the basque team system in the sprml2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 61–67.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, and others. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–18. Association for Computational Linguistics.

Thorsten Joachims. 1999. Making large scale svm learning practical. *tu-dortmund.de*.

Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic-semantic analysis with propbank and nombank. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 183–187. Association for Computational Linguistics.

Ryan McDonald, Koby Crammer, and Fernando CN. Pereira. 2006. Spanning tree methods for discriminative training of dependency parsers. Technical Report MS-CIS-06-11, University of Pennsylvania, Pennsylvania.

Ryan McDonald and Fernando CN. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *EACL*.

Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. In *Proceedings of the CoNLL shared task session of EMNLP-CoNLL*, pages 915–932. sn.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*. Citeseer.

John Platt and others. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In Advances in large margin classifiers, volume 10, number 3, pages 61–74. Cambridge, MA.

Haritz Salaberri, Olatz Arregi, and Beñat Zapirain. 2014. First approach toward Semantic Role Labeling for Basque. In *Proceedings of the 9th edition of the Language Resources Evaluation Conference (LREC)*, pages 1387–1393.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, and others. 2013. Overview of the spmrl 2013 shared task: cross-framework evaluation of parsing morphologically rich languages. In *Fourth Workshop on Statistical Parsing of Morphologically Rich Languages*. Association for Computational Linguistics.