A Machine Learning Approach to Portuguese Clause Identification

Eraldo R. Fernandes[1,2], Cícero N. dos Santos[3], and Ruy L. Milidiú[1]

[1] Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio
Rio de Janeiro, Brazil

[2] Laboratório de Automação
Instituto Federal de Educação, Ciência e Tecnologia de Goiás – IFG
Jataí, Brazil

[3] Mestrado em Informática Aplicada – MIA
Universidade de Fortaleza – UNIFOR
Fortaleza, Brazil

{efernandes,milidiu}@inf.puc-rio.br, cnogueira@unifor.br

Abstract. In this work, we apply and evaluate a machine-learning-based system to Portuguese clause identification. To the best of our knowledge, this is the first machine-learning-based approach to this task. The proposed system is based on *Entropy Guided Transformation Learning*. In order to train and evaluate the proposed system, we derive a clause annotated corpus from the *Bosque* corpus of the *Floresta Sintá(c)tica Project* – an European and Brazilian Portuguese treebank. We include part-of-speech (POS) tags to the derived corpus by using an automatic state-of-the-art tagger. Additionally, we use a simple heuristic to derive a phrase-chunk-like (PCL) feature from phrases in the Bosque corpus. We train an extractor to this sub-task and use it to automatically include the PCL feature in the derived clause corpus. We use POS and PCL tags as input features in the proposed clause identifier. This system achieves a $F_{\beta=1}$ of 73.90, when using the golden values of the PCL feature. When the automatic values are used, the system obtains $F_{\beta=1} = 69.31$. These are promising results for a first machine learning approach to Portuguese clause identification. Moreover, these results are achieved using a very simple PCL feature, which is generated by a PCL extractor developed with very little modeling effort.

1 Introduction

Clause identification [1] is a natural-language-processing task consisting of splitting a sentence into clauses. A clause is defined as a word sequence that contains a subject and a predicate. Clause identification is a special kind of shallow

This work was partially funded by CNPq and FAPERJ grants 557.128/2009-9 and E-26/170028/2008. The first author was supported by a CNPq doctoral fellowship.

parsing, like phrase chunking [2]. Nevertheless, it is more difficult than phrase chunking, since some clauses also contain embedded clauses. Clause information is important for several more elaborated tasks such as full parsing and semantic role labeling.

The *PALAVRAS* parser [3] produces syntactic trees for Portuguese texts which include clause information. A manual-rule-based system to Portuguese clause identification is proposed in [4]. To the best of our knowledge, there is no machine-learning-based approach to Portuguese clause identification.

Conversely, for the English language, there are several such systems. The CoNLL'2001 shared task [1] is devoted to clause identification for English language texts. A corpus with clause annotations is provided and six systems have participated in the competition. The best system at CoNLL'2001 [5] shows a $F_{\beta=1} = 81.73$ and is based on boosted trees. After the competition, other systems were proposed and evaluated in the same corpus. A system based on *Entropy Guided Transformation Learning* (ETL) achieves a $F_{\beta=1} = 80.55$ with very little modeling effort [6]. The current state-of-the-art system [7] for this corpus achieves $F_{\beta=1} = 85.03$. This system is based on a modified perceptron algorithm specialized for phrase recognition.

In this work, we apply and evaluate an ETL system for Portuguese clause identification. ETL [8] is a machine learning strategy that generalizes *Transformation Based Learning* (TBL) [9] by automatically solving the TBL bottleneck: the construction of good template sets. ETL uses *entropy* in order to select the feature combinations that provide good template sets. First, ETL employs decision tree induction to perform an entropy guided template generation. Next, it applies the TBL algorithm to learn a set of transformation rules. ETL is an effective way to eliminate the need of a problem domain expert to build TBL templates.

Since our approach is based on a supervised machine learning method, we need a corpus annotated with clause boundaries in order to train our system. In this work, we derive the training corpus from the *Bosque* corpus of the *Floresta Sintá(c)tica Project* [10] – an European and Brazilian Portuguese treebank. We call this derived corpus the *clause corpus*. In our experiments, we randomly split it into three parts: train, development, and test.

The most effective systems to clause identification in English texts make use of part-of-speech tags and phrase chunks. We include POS tags in the clause corpus using a state-of-the-art tagger [11], which is also based on ETL. To the best of our knowledge, there is no phrase chunking definition for Portuguese language. Hence, using a simple heuristic, we derive a *phrase-chunk-like* (PCL) feature from phrases in the Bosque corpus. We train an ETL-based PCL extractor and use it to automatically generate this information in the clause corpus.

The proposed system achieves a $F_{\beta=1}$ of 73.90, when using the golden values of the PCL feature. When the automatic values are used, the system obtains $F_{\beta=1} = 69.31$. Using automatic values for the PCL feature yields more realistic estimates of the expected system performance for new texts. This sensitivity analysis indicates the potential impact of improvements on the PCL extractor.

The remainder of this paper is structured as follows. In Section 2, we describe the corpus derivation process. The general ETL method is briefly described in Section 3. In Section 4, we present the ETL modeling for Portuguese clause identification. Experimental results are reported and discussed in Section 5. Finally, in Section 6, we present our concluding remarks.

2 Corpus

Our approach is based on a supervised machine learning method. Therefore, we use a corpus annotated with clause boundaries in order to train our system. Here, we derive this training corpus from the *Bosque* corpus, which is a subset of the *Floresta Sintá(c)tica* [10] corpus. The *Floresta Sintá(c)tica Project*'s corpus consists in a treebank of European and Brazilian Portuguese texts. The syntactic trees has been automatically generated by the *PALAVRAS* parser [3]. However, the *Bosque* part of the corpus has been manually revised by linguists. Clause boundaries are one kind of syntactic information, among several others, that is available in the Bosque treebank. An example of a sentence from this corpus, broken into clauses by parentheses, is presented in Figure 1.

```
( Ninguém percebe ( que ele quer ( impor sua presença ) ) . )
```

Fig. 1. A clause annotated Bosque sentence

We call *clause corpus* the corpus annotated with clause boundaries that we derive from the Bosque. This corpus format is the same as the one provided in the CoNLL'2001 shared task [1]. Each line in the corpus contains a token along its corresponding features. As proposed in the CoNLL'2001 shared task, we tackle the clause identification task in three steps: (i) clause *start* identification; (ii) clause *end* identification; and (iii) complete *clause* identification. For each step, the clause corpus has one output feature. The format of the three output features is depicted in Table 1, based on the sentence in Figure 1. The *Start* column contains a binary feature that indicates the tokens where (at least) one clause *starts* (*S* tag). The *End* column contains a binary feature that indicates the tokens where (at least) one clause *ends* (*E* tag). Finally, the *Clause* feature codifies the complete clause set by using parentheses.

In the Bosque corpus, a clause is classified among three types: finite (*fcl*), non-finite (*icl*), and averbal (*acl*). In this work, we ignore the *averbal* clauses due to their unusual structure: they do not contain a verb. Additionally, we are not interested in classifying clauses according to their types. We just want to identify the clause boundaries. The corpus sizes are depicted in Table 2.

Table 1. Clause corpus format

Input			***Output***		
Word	*POS*	*PCL*	*Start*	*End*	*Clause*
Ninguém	pron-indp	B-NP	S	X	(S*
percebe	v-fin	B-VP	X	X	*
que	conj-s	B-PP	S	X	(S*
ele	pron-pers	B-NP	X	X	*
quer	v-fin	B-VP	X	X	*
impor	v-inf	B-VP	S	X	(S*
sua	pron-det	B-NP	X	X	*
presença	n	I-NP	X	E	*S)S)
.	.	O	X	E	*S)

Table 2. Clause corpus sizes

Part	*#Sentences*	*#Tokens*	*#Clauses*
Train	6,557	158,819	14,767
Development	1,405	34,596	3,180
Test	1,405	35,256	3,157

2.1 Input Features

The most effective systems to clause identification in English texts make use of POS tags and phrase chunks. We include POS tags in the clause corpus using a state-of-the-art tagger, also based on ETL. This tagger was proposed in [11] and its reported accuracy – evaluated on two Portuguese corpora – is over 96%.

The Bosque corpus includes phrase information, but *phrase chunking* information is not included. Although phrases and phrase chunks are closely related, there are important differences between them. For instance, phrases can contain another phrases, that is, phrases can be embedded. On the other hand, phrase chunks are flat and are never embedded. Therefore, phrase chunks are simpler than phrases and, consequently, more suitable for machine learning methods.

The idea of breaking a sentence into phrase chunks, for the English language, was firstly proposed by Abney [12]. Phrase chunking is a kind of shallow parsing, yet powerful. It is related to prosodic aspects of the sentence. Unfortunately, as far as we know, there is no equivalent proposal for Portuguese language. There are works related to nominal chunks (base noun phrases) [13]. However, prepositional and verbal chunks also provide valuable information.

In this work, we propose a simple heuristic in order to derive a *phrase-chunk-like* feature from the phrases in the Bosque corpus. We define as chunk all *consecutive* tokens within the same deepest-level phrase. We consider three types of phrase chunks: verbal, nominal, and prepositional. In order to codify this feature, we use the IOB2 tagging style, as in the English-language corpus provided in the CoNLL'2000 shared task [14]. In Figure 2, we show a sentence along with its phrase tree and the resulting PCL feature. Although this heuristic is very simple,

the resulting feature conveys relevant information for the clause identification task, as indicated by some experiments reported in Section 5.

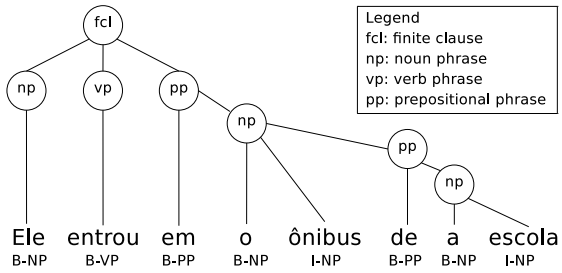

Fig. 2. Sentence along with its phrase tree and the corresponding PCL feature

We train an ETL-based PCL extractor and use it to automatically include this feature in the clause corpus. Therefore, we have two versions of the PCL feature: (i) the golden values derived from the Bosque treebank by the heuristic and (ii) the automatic values given by the trained PCL extractor. It is important to notice that this extractor has been developed with little modeling effort.

3 Entropy Guided Transformation Learning

Entropy Guided Transformation Learning [2] generalizes Transformation Based Learning (TBL) [9] by automatically generating rule templates. ETL employs an *entropy guided template generation* approach, which uses the *information gain* measure in order to select feature combinations that provide good template sets. ETL has been successfully applied to part-of-speech tagging [11], phrase chunking, named entity recognition [8, 15], and dependency parsing [16] – producing results at least as good as the ones of TBL with handcrafted templates. In Figure 3, we present a concise description of the ETL algorithm. A detailed description of ETL can be found in [2, 8]. Several ETL-based multi-language processors are freely available on the Web through the *F-EXT*[4] service [17].

4 ETL Modeling

In this section, we show our ETL modeling for the Portuguese clause identification task. This modeling is strongly based on the one proposed in [6] to English clause identification. We approach the clause identification problem in three steps: (i) clause start identification; (ii) clause end identification; and (iii) complete clause identification. We solve these three sub-tasks sequentially. Therefore, we use the information produced in previous steps as input to the next ones.

[4] http://www.learn.inf.puc-rio.br/

1. Applies the baseline system to the training corpus.
2. Generates the rule templates by using an entropy-guided approach.
3. *Repeat*:
   (a) Generates, for each classification error in the current version of the training corpus, correcting rules by instantiating the templates.
   (b) Computes rule scores. The rule score is defined as the difference between the total number of repaired errors and the total number of generated errors.
   (c) *Stop*, if there is no rule with a score above a given threshold.
   (d) Applies the best-scoring rule to the training corpus.
   (e) Adds the best-scoring rule to the sequence of learned rules.
4. Returns the sequence of learned rules.

Fig. 3. Entropy Guided Transformation Learning

First, we use *Start* tags as input for the end classifier. Next, we use both *Start* and *End* tags to identify the complete clauses.

4.1 Baseline System

We adopt the simple baseline system proposed in the CoNLL'2001 shared task. This system just assigns one clause for the whole sentence. This baseline system is used in the three steps.

4.2 Clause Boundary Candidates

The first and second steps consist in identifying the clause boundary candidates, that is, start and end tokens. These steps identify the tokens that are good candidates to clause boundaries, without any concern to consistence among them. We model these two sub-tasks as token-classification problems. In Table 1, we illustrate the corpus format through an example. The *Start* and *End* columns in the table respectively indicate the *start* and *end* classifications. In the first step, if a token *starts* one or more clauses, it must be classified as S, otherwise, it must be classified as X. Similarly, in the second step, if a token *ends* one or more clauses, it must be classified as E, otherwise as X.

4.3 Complete Clause Identification

The last and most difficult step consists in splitting a given sentence into clauses. In the clause corpus, the complete clauses within a sentence are encoded through a unique token feature using the following tags: *(S** – indicating that the token starts a clause; **S)* – indicating that the token ends a clause; * – representing a token that neither starts nor ends a clause; and any combinations of the previous to represent tokens that start or end more than one clause. The *Clause* column in Table 1 contains the tags that encode the clauses within the sentence illustrated in Figure 1.

For this last sub-task, we present two modeling approaches: *ETL-Token* and *ETL-Pair*. The *ETL-Token* consists of a token classification approach. In this approach, we apply ETL in a straightforward manner. We train an ETL model to classify each token as *, (S*, *S), or any tag combination appearing in the *Clause* column of the training corpus. This approach is very simple but also limited. We observe that many clauses are tokenwise long. For instance, in the training corpus, the fraction of clauses with length longer than 14 tokens is greater than 40%. For such cases, even using a window of 27 tokens (the current token plus the thirteen tokens on each side), one clause boundary is not included when classifying the other one. We observe that this window size is computationally prohibitive for the ETL algorithm.

In order to capture a broader context, we try a second modeling approach to the third step: *ETL-Pair*. This approach uses the output of the *Start* and *End* classifiers to create a new corpus. For each start-end pair of tokens from a given sentence in the original corpus, we generate one example in the new corpus. We attach to this new example all the original input features of both start and end tokens. Next, we train a binary ETL model that learns to classify which examples (pairs of tokens) correspond to correct clause boundaries.

4.4 Derived Features

We use the three input features in the clause corpus – word, POS, and PCL – plus some derived features. We derive these additional features in the same fashion as in [18], although we use just a small subset of the features proposed by these authors.

The derived features inform about the occurrence of relevant elements within a specific sentence fragment. The following elements are the relevant ones: *pronouns*, *conjunctions*, *verbal chunks*, *start tokens*, and *end tokens*. We call *verbal chunks* the ones with chunk tag with value `verb`. We generate two features for each relevant element and sentence fragment: a flag indicating the occurrence within the fragment and the number of occurrences within the fragment.

For the token classifiers (*Start*, *End*, and *ETL-Token*) we use the same sentence fragmentation scheme. For each token we derive twenty features: ten for the sentence fragment before the token and ten for the sentence fragment after it. For the *ETL-Pair* classifier we use a different scheme. For each start-end pair of tokens we derive thirty features: ten for the sentence fragment before the start token; ten for the sentence fragment after the end token; and ten for the sentence fragment between the start and end tokens. Observe that a derived feature is only used when its required information is available.

5 Experiments

We use the development corpus in order to tune the ETL parameters. For the three token classifiers (*Start*, *End*, and *ETL-Token*), we set the context window

size parameter to 7. Whereas for the *ETL-Pair* classifier we set the window size to 9. For all approaches, we set the rule score threshold to 2.

In order to evaluate the potential and real impact of the PCL feature in the proposed system performance, we train and evaluate three independent versions of the system: (i) using no information of the PCL feature; (ii) using the automatic values of the PCL feature; and (iii) using the golden values of the PCL feature. Only in version (i), where no PCL information is used, we consider the verbal tokens (POS tag equal to verb) as relevant elements when generating the derived features. In (ii), the PCL values are provided by the automatic PCL extractor. In (iii), the PCL values are obtained directly from the Bosque treebank by the PCL derivation heuristic.

The resulting performances are presented in Table 3. One can observe that the $F_{\beta=1}$ for the version that uses the PCL golden values is almost seven points greater than the one that uses no PCL information. The sensitivity of the system performance to this feature clearly indicates its potential positive impact. Using automatic values for the PCL feature yields more realistic estimates of the expected system performance for new texts. The system performance using the automatic values of this feature also indicates the positive impact of improvements on the PCL extractor.

Table 3. PCL impact on *ETL-Pair* performance

PCL	*Precision*	*Recall*	$F_{\beta=1}$
No	75.18	60.34	66.95
Automatic	78.14	62.27	69.31
Golden	83.78	66.11	73.90

In Table 4, we present the performances for the four proposed classifiers – *Start*, *End*, *ETL-Token*, and *ETL-Pair* – on the test corpus. These results are divided into two groups: golden and automatic values of the PCL feature. The $F_{\beta=1}$ of the *ETL-Pair* system is over two points greater than the one of the *ETL-Token* system. We believe that this improvement is due to the stronger contextual information used by the *ETL-Pair* approach.

6 Conclusions

In this paper, we apply and evaluate a machine-learning-based system to Portuguese clause identification. The system is based on the machine learning technique called *Entropy Guided Transformation Learning*. In order to train and evaluate our system, we derive a clause annotated corpus from the *Bosque* treebank of the *Floresta Sintá(c)tica Project*. We include POS tags in the clause corpus by using a state-of-the-art tagger, also based on ETL.

Table 4. Test corpus performances

Task/Strategy	*Golden*			*Automatic*		
	Precision	*Recall*	$F_{\beta=1}$	*Precision*	*Recall*	$F_{\beta=1}$
Start	93.37	87.25	90.20	90.63	84.25	87.32
End	85.64	79.89	82.67	84.29	74.78	79.25
ETL-Token	74.59	**68.45**	71.39	69.85	**64.65**	67.15
ETL-Pair	**83.78**	66.11	**73.90**	**78.14**	62.27	**69.31**
BLS	82.06	36.52	50.55	82.06	36.52	50.55

Phrase chunking is a very important feature for several Natural Language Processing tasks, including clause identification. However, to the best of our knowledge, there is no phrase chunking definition to Portuguese language. So, we propose a simple heuristic to derive a phrase-chunk-like feature from phrases in the Bosque treebank. We train an ETL extractor to this sub-task and use it to include this information in the derived clause corpus.

The clause identification modeling used in this work is based on the approach proposed in [6] to English language. The problem is divided into three steps: (i) clause start identification; (ii) clause end identification; and (iii) complete clause identification. We propose one system for the first step, another for the second step, and two systems for the third step.

We report the performance of the four systems on the derived clause corpus. The impact of the PCL feature and the automatic PCL extractor on the system performance is also evaluated. We report the system performance on three scenarios: using no PCL information, using the automatic values, and using the golden values of the PCL feature. These results indicate that the PCL feature is informative to the clause identification task and the ETL-based PCL extractor is effective to improve the performance of the proposed clause identifier.

We believe that using a better phrase chunking information we can improve our result in this task. We are working on a better heuristic to extract phrase chunks from Bosque. Additionally, the ETL-based PCL extractor can be substantially improved, since it has been developed with very little modeling effort.

Acknowledgments

We thank Bernardo A. Pires and Guilherme De Napoli for their efforts to code the scripts used to derive the clause corpus. We also thank Maria Cláudia de Freitas for the important clarifications about Bosque treebank.

References

1. Sang, E.F.T.K., Déjean, H.: Introduction to the CoNLL-2001 shared task: Clause identification. In: Proceedings of Fifth Conference on Computational Natural Language Learning, Toulouse, France (2001)

2. Milidiú, R.L., dos Santos, C.N., Duarte, J.C.: Phrase chunking using entropy guided transformation learning. In: Proceedings of ACL-08: HLT, Columbus, USA, Association for Computational Linguistics (2008) 647–655
3. Bick, E.: The Parsing System Palavras: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework. PhD thesis, Aarhus University, Aarhus, Denmark (November 2000) Aarhus University Press.
4. Leffa, V.J.: Clause processing in complex sentences. In: Proceedings of the First International Conference on Language Resources and Evaluation. Volume 2., Granada, Espanha (1998) 937–943
5. Carreras, X., Màrquez, L.: Boosting trees for clause splitting. In: Proceedings of Fifth Conference on Computational Natural Language Learning, Toulouse, France (2001)
6. Fernandes, E.R., Pires, B.A., dos Santos, C.N., Milidiú, R.L.: Clause identification using entropy guided transformation learning. In: Proceedings of the 7th Brazilian Symposium in Information and Human Language Technology (STIL'2009), São Carlos, Brazil (2009)
7. Carreras, X., Màrquez, L., Castro, J.: Filtering-ranking perceptron learning for partial parsing. Machine Learning **60**(1–3) (2005) 41–71
8. dos Santos, C.N., Milidiú, R.L.: Entropy Guided Transformation Learning. In: Foundations of Computational Intelligence, Volume 1: Learning and Approximation. Volume 201 of Studies in Computational Intelligence. Springer (2009) 159–184
9. Brill, E.: Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. Computational Linguistics **21**(4) (1995) 543–565
10. Freitas, C., Rocha, P., Bick, E.: Floresta Sintá(c)tica: Bigger, thicker and easier. In Teixeira, A., de Lima, V.L.S., de Oliveira, L.C., Quaresma, P., eds.: Computational Processing of the Portuguese Language. Volume 5190 of Lecture Notes in Computer Science., Springer (2008) 216–219
11. dos Santos, C.N., Milidiú, R.L., Renteria, R.P.: Portuguese part-of-speech tagging using entropy guided transformation learning. In: Proceedings of PROPOR 2008, Aveiro, Portugal (2008)
12. Abney, S.: Parsing by Chunks. In: Principle-Based Parsing. Kluwer Academic Publishers, Dordrecht (1991)
13. Freitas, M.C., Garrao, M., Oliveira, C., dos Santos, C.N., Silveira, M.: A anotação de um corpus para o aprendizado supervisionado de um modelo de sn. In: Proceedings of the III TIL / XXV Congresso da SBC, São Leopoldo - RS - Brasil (2005)
14. Sang, E.F.T.K.: Text chunking by system combination. In: Proceedings of Conference on Computational Natural Language Learning, Lisbon, Portugal (2000)
15. Milidiú, R.L., dos Santos, C.N., Duarte, J.C.: Portuguese corpus-based learning using ETL. Journal of the Brazilian Computer Society **14**(4) (2008)
16. Milidiú, R.L., dos Santos, C.N., Crestana, C.E.M.: A token classification approach to dependency parsing. In: Proceedings of the 7th Brazilian Symposium in Information and Human Language Technology (STIL'2009), São Carlos, Brazil (2009)
17. Fernandes, E.R., dos Santos, C.N., Milidiú, R.L.: Portuguese language processing service. In: Proceedings of the Web in Ibero-America Alternate Track of the 18th World Wide Web Conference, Madrid (2009)
18. Carreras, X., Màrquez, L., Punyakanok, V., Roth, D.: Learning and inference for clause identification. In: Proceedings of the Thirteenth European Conference on Machine Learning. (2002) 35–47