5 Experiments

This chapter presents the experimental setup, *corpora* statistics and performance results for the token classification approach to dependency parsing. First, the chosen *corpora* are described and analyzed, as well as their evaluation methods. Then, the application of Machine Learning algorithms is described and for each one of them, we present its corresponding parameter setting. Finally, the achieved performance with each particular combination of algorithms, modeling and parameters, as well as baseline classifiers is presented and analyzed.

5.1 Corpora

In 2006, 2007, 2008 and 2009, dependency parsing related tasks have been part of the Conference on Natural Language Learning Shared Task. For the first two years, the task was to solve the dependency parsing problem itself for a wide range of languages, while for the other two years a joint task of syntactic and semantic dependencies has been proposed. Since this work is focused on syntactic dependencies, it is mainly concerned with the tasks of those first two years.

In 2006, *corpora* for thirteen languages were made available, namely: Arabic, Chinese, Czech, Danish, Dutch, German, Japanese, Portuguese, Slovene, Spanish, Swedish, Turkish and Bulgarian [95]. In 2007, they made available *corpora* for ten languages, namely: Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian and Turkish [33]. However, only four of those *corpora* are still publicly available, namely, the Dutch, Danish, Portuguese and Swedish ones.

Regarding how these *corpora* were made available by the CoNLL 2006, only the Dutch and the Swedish come from actual dependency treebanks, respectively the Danish Dependency Treebank [96] and the Talbanken05 [97, 98, 99]. The Dutch and Portuguese *corpora* come respectively from the Alpino [100, 101] and Bosque [102] treebanks, two phrase structure treebanks that were properly converted into dependency treebanks by the occasion of the conference [95].

These *corpora* provide the following features: word form, token position, lemma of the word, coarse-grained part-of-speech, fine-grained part-of-speech and a list of set-valued syntactic and morphological features. Since the Swedish *corpus* provides no lemma for each word, only one type of part-of-speech and no list of set-valued features, thus greatly deviating from the other *corpora*, it is not used in our work. Table 5.1 provides statistical information about each chosen *corpus*, that is, Danish, Dutch and Portuguese.

	Danish	**Dutch**	**Portuguese**
Number of Tokens	100 238	200 654	212 545
Number of Sentences	5 512	13 735	9 359
Tokens per sentence	18.2	14.6	22.8
Number of different coarse *postags*	10	13	15
Number of different fine *postags*	25	302	21
Percentage of punctuation tokens	13.9	11.3	14.2
Percentage of non-projective relations[1]	1.0%	5.4%	1.3%
Percentage of sentences with at least one non-projective relation[1]	15.6%	36.4%	22.2%

Table 5.1: *Corpora* Statistics.

Information regarding the set of part-of-speech tags for coarse-grained, fine-grained features and possible values for the list of syntactic and morphological features for each language is presented in Appendix A. At last, these *corpora* had their sentences divided into a training and a test set by the occasion of the conference and this work follows the exact same division.

5.2 Evaluation Metrics

To evaluate dependency parsers the three most common metrics are the *labeled attachment score* (LAS), the *unlabeled attachment score* (UAS) and the *label accuracy* (LA). LAS is the percentage of tokens where the system correctly predicts both its head and the relation type that the token holds with its head. UAS is the percentage of tokens where the system correctly predicts its head, whereas LA is the percentage of tokens with correct relation type.

For both the CoNLL 2006 [95] and the CoNLL 2007 [33] shared tasks, LAS is used as the main evaluation metric. Nevertheless, systems' results are reported for all three metrics. In this work, we use UAS as our metric, since our

[1] Including non-scoring tokens

concerns are the prediction of correct token heads and how to model this as a token classification problem.

Finally, following the CoNLL 06 Shared Task, punctuation tokens are excluded from scoring.

5.3 Performance Results

To evaluate our model effectiveness we apply the ETL algorithm to the three described *corpora* and to evaluate our systems we use the evaluation script of the CoNLL 2006 Shared Task. All results shown with ETL were achieved using the *Template Evolution* option from ETL, given that every attempt without it is very time and memory consuming, sometimes requiring more than the available memory resources.

First, we present a baseline system for the dependency parsing using our special tagset. Then, the parameters used when applying ETL to solve it, as well as the results achieved are presented. We also present a model to solve the dependency parsing in three subtasks, its parameters and results. Finally, we use clause and chunk information provided by the Portuguese *corpus* as features and evaluate its impact in our models' accuracy.

5.3.1 Baseline Classifier

For the ETL baseline system we assign to each token the most frequently seen class for its part-of-speech in the training set.

Language	**Coarse-grained Part-of-speech**	**Fine-grained Part-of-speech**
Danish	33.09%	34.87%
Dutch	41.24%	41.44%
Portuguese	51.31%	56.72%

Table 5.2: Baseline System Accuracy with different Tagsets.

Table 5.2 shows the baseline accuracy in the test set when using either the coarse-grained or the fine-grained part-of-speech to identify the token class, while Table 5.3 shows the accuracy achieved in each subtask by its baseline system and ETL model. For the subtasks, we use the coarse-grained part-of-speech, since it consistently gives better results.

Finally, Appendix B presents the complete baseline classifiers description for both *one task* and subtasks approaches for each one of the three languages.

System	**Danish** **(%)**	**Dutch** **(%)**	**Portuguese** **(%)**
Baseline for Head Side	73.91	66.36	82.35
Baseline for Head PoS	53.42	62.53	67.00
Baseline for Head Distance	93.49	91.36	93.27

Table 5.3: Baseline Accuracy in each Subtask.

5.3.2 Parameter Tuning

To find the best set of parameters to ETL and to verify the effectiveness of derived features, we create a development set. This set is created by randomly selecting 10% of the sentences of each training set. First, by testing a wide range of values the best set of initial parameters as window size, rule threshold and template evolution features window is found. Based on these results we use, further on every experiment, a window size of 7, a rule threshold of 4 and a template evolution with rules ranging from 2 to 5 features.

5.3.3 One Task Model

In this scenario, an ETL model that predicts the token's head according to our special tagset is trained and evaluated.

Even though the fine-grained baseline system initially achieves higher accuracy, further experiments show that after applying an ETL model the coarse-grained baseline system performs better.

At the extraction step, after the algorithm classifies each token, we use the attributed tag to identify its head. In case it is not possible to consistently identify its head, *e.g.* its tagged as the third verb to the left, but there are only two verbs, the token is simply classified as *root*.

In this work, we create and test a great number of derived features described in Chapter 3, however only few of them improve our results when tested on the development set: *the number of verbs before the token*, *the number of verbs after the token* and *the lemma of the nearest verb before the token*.

Language		**Number**	**of**	**features**	
	2	**3**	**4**	**5**	**Total**
Portuguese	18	43	179	314	554
Danish	29	46	133	201	409
Dutch	39	131	158	157	485

Table 5.4: Number of Generated Templates.

Table 5.4 shows how many templates are automatically generated for each language and how many features each one has when using derived features, while Table 5.5 shows the number of learned rules for each language and according to the number of features in each rule.

Language		**Number**	**of**	**features**	
	2	**3**	**4**	**5**	**Total**
Portuguese	696	708	647	279	2330
Danish	370	477	359	109	1315
Dutch	989	1043	442	245	2717

Table 5.5: Number of Learned Rules.

Table 5.6 shows the UAS results of applying an ETL model to correctly identify a token's head to the test set and the improvements gained when using the derived features mentioned before. For each language, we present the results of the baseline system, the ETL algorithm, the average score of the 18 systems that took part in the CoNLL 2006 shared task and the state-of-the-art by the occasion of the task.

System	**Danish (%)**	**Dutch (%)**	**Portuguese (%)**
State-of-the-art	90.58	83.57	91.36
ETL (derived features)	83.71	75.21	87.48
ETL	83.45	74.87	87.02
Average	84.52	75.07	86.46
Baseline	34.87	41.44	56.72

Table 5.6: UAS for one model ETL results.

In two of the three languages our system has an above average performance. Moreover, in the three cases our results are within a 10% *error-margin* of the state-of-the-art systems, what suggests that this is a promising approach.

5.3.4 Three Subtasks Model

Our tagging style allows for splitting the dependency parsing into three subtasks, namely: identifying if the head of the token comes before (left) or after (right) the token; identifying the part-of-speech of the token's head; find the distance from the token to its head counting tokens with the same part-of-speech as the head. In all these subtasks, there is a *root* class when the token is root of the dependency tree.

We apply the ETL algorithm to solve each of the three subtasks. Another ETL model is used to join these partial findings and solve the dependency parsing. Like in the *one task model*, we use a window size of 7, a rule threshold of 4, template evolution ranging from 2 to 5 features and the three derived features.

Table 5.7 shows the accuracy achieved in each subtask by its baseline system and application of an ETL model.

System	**Danish (%)**	**Dutch (%)**	**Portuguese (%)**
Baseline for Head Side	73.91	66.36	82.35
ETL for Head Side	**93.77**	**85.53**	**96.89**
Baseline for Head PoS	53.42	62.53	67.00
ETL for Head PoS	**89.47**	**86.82**	**92.39**
Baseline for Head Distance	93.49	91.36	93.27
ETL for Head Distance	**93.67**	**92.04**	**93.77**

Table 5.7: ETL Accuracy in each Subtask.

After all subtasks are executed, we apply a final ETL model. For this last ETL baseline system, the subtasks results are simply joined and, if in any of those a token was classified as *root*, his initial class will also be *root*. Table 5.8 presents the UAS achieved with this initiative. For better comparison, it also reports the state-of-the-art, the average score of the 18 systems that took part in the CoNLL 2006 shared task and the results achieved with only one ETL model.

System	**Danish (%)**	**Dutch (%)**	**Portuguese (%)**
State-of-the-art	90.58	83.57	91.36
ETL joining Subtasks	84.87	79.19	87.98
One model ETL with derived features	83.97	75.21	87.48
Average	84.52	75.07	86.46
Baseline joining Subtasks	82.16	75.79	86.58
Baseline	34.87	41.44	56.72

Table 5.8: UAS for ETL joining Subtasks Results.

Dividing the problem in three subtasks consistently improves our results in the three languages, with a error decrease of 6% (Danish), 16% (Dutch) and 4% (Portuguese). placing our results above the average in all three languages.

5.3.5 Clause and Phrase Chunk Impact

The Portuguese *corpus*, Bosque [102], also provides information about clause boundaries i.e., where clauses start and end in a sentence. This information follows a similar definition and format as the ones used in the CoNLL 2001 shared task of clause identification [5]. Additionally, [103] presents an heuristic that uses the syntactic information provided in Bosque to derive *phrase chunking* information, similarly to the CoNLL 2000 shared task [4].

Word	*Chunking*	*Start*	*End*	*Clause*
Ninguém	B-NP	S	X	(S*
percebe	B-VP	X	X	*
que	B-PP	S	X	(S*
ele	B-NP	X	X	*
quer	B-VP	X	X	*
impor	B-VP	S	X	(S*
sua	B-NP	X	X	*
presença	I-NP	X	E	*S)S)
.	O	X	E	*S)

Table 5.9: Example of Clause and Phrase Chunk.

To evaluate the impact of these two features in the dependency parsing, we add both features to the original CoNLL 2006 *corpus* as follows. Clause information generates three features: *a feature that identifies whenever a clause starts*, *a feature that identifies whenever a clause ends* and *a feature that identifies all clauses with a parentheses notation*. The phrase chunking information is represented in only one feature, according to the IOB2 tagging style. Table 5.9 shows an example with the clause and chunking features.

System	**Head**	**Head**	**Head**
	Side(%)	**Part-of-speech(%)**	**Distance(%)**
ETL (chunk + clause)	97.45	94.35	93.21
ETL	96.89	92.39	93.77

Table 5.10: Subtasks Results with Clause and Phrase Chunk.

Following the same parameters and modeling of previous experiments, ETL models were trained to solve the dependency parsing, as well as the subtasks proposed in this work. Table 5.10 presents the results of adding those features to solve the subtasks, while Table 5.11 shows the results when solving the full dependency parsing. The results obtained without the chunking and clause information are also presented for comparison.

The chunk and clause information improve the head side and head part-of-speech subtasks, with only a minor decrease in the head distance accuracy. When comparing the results for the dependency parsing, the chunk and clause information improve the UAS both with the subtasks approach and the one ETL approach, decreasing its error in 14% for the first one and 11% for the second one.

System	Portuguese (%)
State-of-the-art	91.36
ETL joining Subtasks + Chunk and Clause	89.74
ETL (derived features) + Chunk and Clause	88.86
ETL joining Subtasks	87.98
ETL (derived features)	87.48
Baseline for joining Subtasks	86.58

Table 5.11: UAS for ETL with Clause and Phrase Chunk.

5.3.6 Error Analysis

When analyzing the most common errors of our models, we identify that the most misclassified tags are those where the head of the token is either a verb or a noun. Table 5.12 shows the percentage of errors that corresponds to the tags where the head is the first verb, first noun, second verb or second noun, either to the left or to the right of the token.

Head of the Token	Danish (%)	Dutch (%)	Portuguese (%)
First verb	24.8	28.5	24.0
First noun	12.1	9.9	14.5
Second verb	6.5	12.6	13.4
Second noun	5.4	4.0	8.8
Total	**48.7**	**55.0**	**60.7**

Table 5.12: Most Common Errors.

These results suggest a deeper investigation of noun and verb heads, that can lead to new derived features to improve our models accuracy, since this error pattern is present in all three languages.