4
Machine Learning Algorithms

One of the most remarkable characteristics of Machine Learning algorithms is its use of data to improve its accuracy at some given task. In Supervised Machine Learning, labeled data is given as input, from which the Machine Learning model is generalized. After a model is built, it can be applied to previously unseen data to evaluate its accuracy and to effectively solve the proposed task.

Supervised Learning algorithms present some powerful properties that make them specially fit for NLP problems. One of this properties is language independence. Once an algorithm has been successfully applied to a problem, achieving good results, it can be easily applied to other languages given that there is an analogous labeled data as input also available. Another important property of Supervised Learning is its low dependency on domain knowledge. Once labeled data is provided it is possible to achieve results that were only possible to human experts.

For the past years, Entropy Guided Transformation Learning (ETL) has been successfully applied to part-of-speech tagging [77], phrase chunking [78, 79] named entity recognition [77, 80, 79] and semantic role labeling [79], producing results at least as good as the ones of TBL with handcrafted templates and competitive with state-of-the-art results. Also, several ETL-based language processors, for different languages, are freely available on the Web through the F-EXT-WS [1] service [81]. Finally, ETL has the advantage of an easy modeling along with a proper use of neighbor token's features, therefore, being a suitable algorithm for an NLP task as dependency parsing.

This chapter first presents Transformation Based Learning, an algorithm upon which ETL is built. It then describes ETL and how it solves TBL bottleneck in the creation and improvement of its models.

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

4.1 Transformation Based Learning

Eric Brill [3] proposes Transformation Based Learning (TBL). TBL is a corpus-based, error-driven approach in which a set of transformation rules is learned to correct the errors of a baseline classifier. TBL has been successfully used for several Natural Language Processing tasks, such as part-of-speech tagging [3, 82, 78], phrase chunking [76, 83, 84, 85], spelling correction [86], appositive extraction [87], named entity recognition [88, 89] and semantic role labeling [90]. As input, TBL requires *corpus* (the labeled data), a baseline classifier and a set of rule templates.

The learning step of TBL is defined as follows: the baseline classifier is applied to the given *corpus*. Based on the set of rule templates, transformation rules that correct the errors left by the baseline system are created. These rules are evaluated by a score function and the rule with the highest score is added to the TBL model. Then, this same rule is applied to the *corpus* and new rules are generated. Transformation rules are added to the model until a score threshold is achieved or no correcting transformation can be created, when the algorithm stops. Hence, the application of a TBL model to unseen data is to first apply the baseline classifier and then the learned transformation rules in the order they were learned. Figure 4.1 shows the TBL learning step.

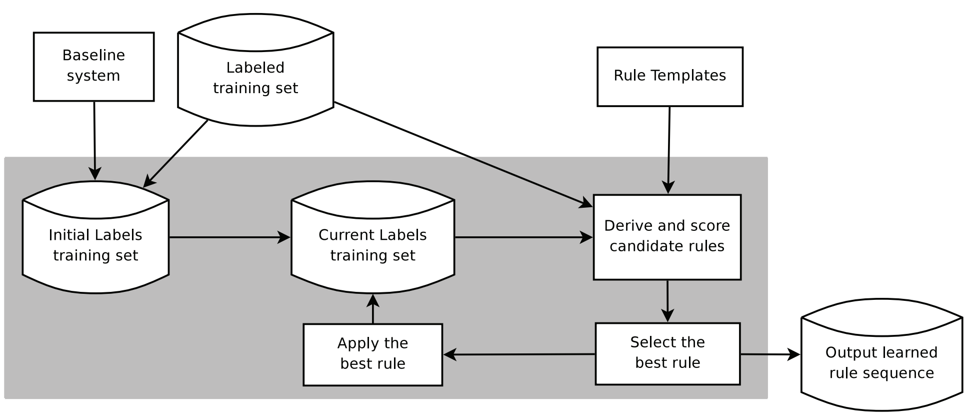

Figure 4.1: The Transformation Based Learning algorithm.

Generally, the score function used to evaluate rules is defined as the difference between the number of examples that are corrected by the rule and the number of errors that the rule created.

The set of rule templates defines what are the possible transformations that TBL can create to correct the baseline classifier errors. The creation of the transformation rules consists in the attribution of values to the features that compose a given template. For instance, the template in table 4.1 has the following attributes: the word of a given token (*word[0]*), its part-of-speech (*pos[0]*) and the part-of-speech of the previous token (*pos[-1]*) - the values inside brackets refer to position relative to the token.

word[0]	**pos[0]**	**pos[-1]**

Table 4.1: An Example of a Template.

This template can generate the transformation rule in table 4.2. The first part of the rule defines where it can be applied. In this example, that corrects a part-of-speech tagger for Portuguese, the rule is applied when the given word is *estudo*, currently classified as *verb* with a *article* as previous token. The second part of the rule indicates that when these conditions are met, the token is classified as *noun*.

word[0]=*estudo* **pos[0]**=*verb* **pos[-1]**=*article* $\Longrightarrow$ **pos[0]**=*noun*

Table 4.2: An Example of a Transformation Rule.

The set of templates in TBL is meant to capture relevant feature combinations. Also, the set of templates is task specific and their quality depends on some domain knowledge. However, the set of templates for the TBL must be manually generated, implying in a bottleneck in the creation and improvement of TBL models.

4.2
Entropy Guided Transformation Learning

Entropy Guided Transformation Learning solves the TBL bottleneck by providing an automatic way of generating a set of templates [91]. In order to find good feature combinations for templates, ETL uses Information Gain (IG), which is based in data entropy and is a key measure for many feature selection strategies. Since the most popular Decision Tree (DT) algorithms [92, 93] use IG, they provide a quick way of obtaining entropy guided feature selection.

After the baseline classifier is applied, ETL applies a DT algorithm to the *corpus*. Then, each path from the root to a leaf turns into a template, where its attributes are the features of each node in the path. Figure 4.2 shows an example of template generation from a given DT.

After the template generation step is finished, ETL proceeds exactly as TBL in the creation and learning of transformation rules, as shown in figure 4.2.

ETL also implements a evolutionary template approach [94] to reduce the required training time. When using the *template evolution* approach, ETL first

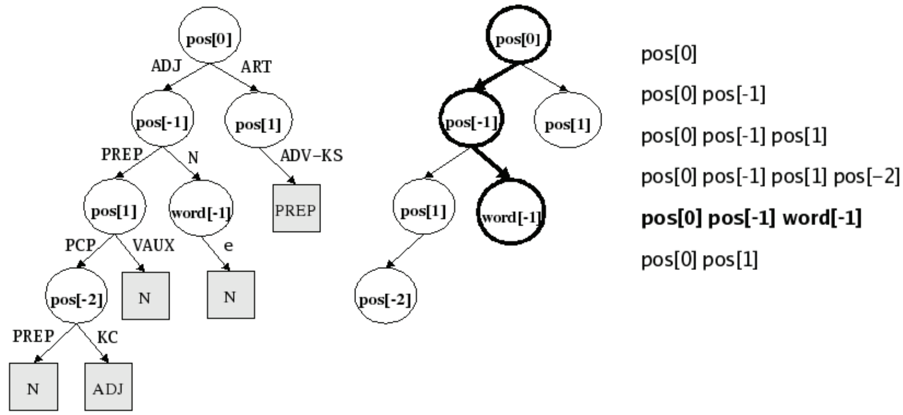

Figure 4.2: Example of Template Generation.

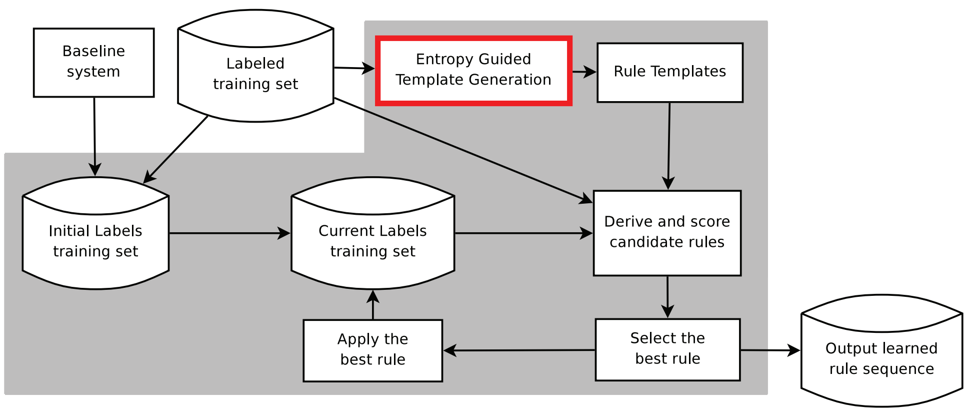

Figure 4.3: Entropy Guided Transformation Learning.

trains a TBL model using only templates composed by one feature. After this model is trained, it proceeds by training other models, incrementing the number of features allowed in each template by one each time. The final ETL model is simply the concatenation of all rules in the order they were learned.