ETL Ensembles for Chunking, NER and SRL

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

[1] Mestrado em Informática Aplicada – MIA
Universidade de Fortaleza – UNIFOR
Fortaleza, Brazil
cnogueira@unifor.br
[2] Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio
Rio de Janeiro, Brazil
milidiu@inf.puc-rio.br
[3] Laboratório de Automação
Instituto Federal de Educação, Ciência e Tecnologia de Goiás – IFG
Jataí, Brazil
{ccrestana,efernandes}@inf.puc-rio.br

Abstract. We present a new ensemble method that uses Entropy Guided Transformation Learning (ETL) as the base learner. The proposed approach, ETL Committee, combines the main ideas of Bagging and Random Subspaces. We also propose a strategy to include redundancy in transformation-based models. To evaluate the effectiveness of the ensemble method, we apply it to three Natural Language Processing tasks: Text Chunking, Named Entity Recognition and Semantic Role Labeling. Our experimental findings indicate that ETL Committee significantly outperforms single ETL models, achieving state-of-the-art competitive results. Some positive characteristics of the proposed ensemble strategy are worth to mention. First, it improves the ETL effectiveness without any additional human effort. Second, it is particularly useful when dealing with very complex tasks that use large feature sets. And finally, the resulting training and classification processes are very easy to parallelize.

Keywords: entropy guided transformation learning, ensemble methods, text chunking, named entity recognition, semantic role labeling.

1 Introduction

Ensemble methods are learning algorithms that generate multiple individual classifiers and combine them to classify new samples. Usually, the final classification is done by taking a weighted or majority vote of the individual predictions. Such model combinations are known as *ensemble models* or *committees*. The main purpose of model combination is to reduce the generalization error of a classifier. Ensemble algorithms have received considerable attention in the last years [1,2].

A. Gelbukh (Ed.): CICLing 2010, LNCS 6008, pp. 100–112, 2010.

Transformation Based Learning (TBL) is a machine learning algorithm introduced by Brill [3]. TBL is a corpus-based error-driven approach that learns a set of ordered transformation rules which correct mistakes of a baseline classifier. It has been successfully used for several important NLP tasks. Nevertheless, it suffers from a serious drawback: the need of costly human expertise to build the required TBL rule templates. This is a bottleneck for wide spreading its application. Entropy Guided Transformation Learning (ETL) [4] eliminates the TBL bottleneck by providing an automatic mechanism to construct good rule templates. Hence, ETL allows the construction of ensemble models that use Transformation Learning.

In this work, we present an ensemble method that uses ETL as the base learner. The proposed approach, ETL Committee, combines the main ideas of Bagging [5] and Random Subspaces [6]. In order to evaluate the effectiveness of the ensemble method, we apply it to three Natural Language Processing tasks: Text Chunking (TCK), Named Entity Recognition (NER) and Semantic Role Labeling (SRL). Our experimental findings indicate that ETL Committee significantly outperforms single ETL models, achieving state-of-the-art competitive results for the three tasks. As far as we know, this is the first study that uses transformation rule learning as the base learner for an ensemble method.

The remainder of the paper is organized as follows. In section 2, we briefly describe the ETL strategy. In section 3, we detail the ETL Committee approach. In section 4, the experimental design and the corresponding results are reported. Finally, in section 5, we present our concluding remarks.

2 Entropy Guided Transformation Learning

Entropy Guided Transformation Learning [4] generalizes Transformation Based Learning by automatically generating rule templates. ETL employs an *entropy guided template generation* approach, which uses Information Gain (IG) in order to select the feature combinations that provide good template sets [7]. ETL has been successfully applied to part-of-speech (POS) tagging [8], phrase chunking [4], named entity recognition [7], clause identification [9] and dependency parsing [10], producing results at least as good as the ones of TBL with handcrafted templates. A detailed description of ETL can be found in [7]. In the next two subsections, we present two variations on the basic strategy. These variations are very useful when using ETL as a base learner for an ensemble method.

2.1 Template Sampling

There are cases where learning the largest rule set is necessary. For instance, when training an ensemble of classifiers using different training data sets, overfitting can be beneficial. This is because, in this specific case, overfitting can introduce diversity among the ensemble members. As an example, some DT ensemble learning methods do not use pruning [11,12,6].

However, the larger the rule set the longer it takes to be learned. Therefore, in our ETL implementation, we also include the *template sampling* functionality, which consists in training the ETL model using only a randomly chosen fraction of the generated templates. Besides being simple, this strategy provides a speed up control that is very useful when multiple ETL models are to be learned.

2.2 Redundant Transformation Rules

As previously noticed by Florian [13], the TBL learning strategy shows a total lack of redundancy in modeling the training data. Only the rule that has the largest score is selected at each learning iteration. All alternative rules that may correct the same errors, or a subset of the errors, are ignored. This greedy behavior is not a problem when the feature values tested in the alternative rules and the ones tested in the selected rule always co-occur. Unfortunately, this is not always the case when dealing with sparse data.

Florian includes redundancy in his TBL implementation by adding to the list of rules, after the training phase has completed, all the rules that do not introduce error. Florian shows that these additional rules improve the TBL performance for tasks were a word classification is independent of the surrounding word classifications.

In our ETL implementation, we also include redundancy in the TBL step, but in a different way. At each iteration, when the best rule b is learned, the algorithm also learns all the rules that do not include errors and correct exactly the same examples corrected by b. These redundant rules do not alter the error-driven learning strategy, since they do not provide any change in the training data. This kind of redundancy is more effective for low scored rules, since they are more likely to use sparse feature values and their selection is supported by just a few examples.

Redundant rules increase the model overfitting since more information from the training set is included in the learned model. Therefore, redundant rules does not improve the performance of single ETL classifiers. However, the inclusion of redundancy improves the classification quality when several classifiers are combined, since overfitting can be beneficial to generate more diverse classifiers in an ensemble strategy.

3 ETL Committee

According to Dietterich [14], a necessary and sufficient condition for an ensemble of classifiers to have a lower generalization error than any of its individual members is that the classifiers are accurate and diverse. A classifier is considered to be accurate if its error rate on new data is lower than just guessing. Two classifiers are diverse if they make different errors on new data.

In this section, we present ETL Committee, an ensemble method that uses ETL as a base learner. The ETL Committee strategy relies on the use of training data manipulation to create an ensemble of ETL classifiers. ETL Committee combines the main ideas of Bagging [5] and Random Subspaces [6]. From Bagging, we borrow the bootstrap sampling method. From Random Subspaces, we use the feature sampling idea. In the ETL Committee training, we use ETL with template sampling, which provides an additional randomization step.

3.1 ETL Committee Training Phase

Given a labeled training set $\mathcal{T}$, the ETL Committee algorithm generates L ETL classifiers using different versions of $\mathcal{T}$. In Figure 1, we detail the ETL Committee training phase. The creation of each classifier is independent from the others. Therefore, the committee training process can be easily parallelized. In the creation of a classifier c, the first step consists in using *bootstrap sampling* to produce a bootstrap replicate $\mathcal{T}^{'}$ of the training set $\mathcal{T}$. Next, *feature sampling* is applied to $\mathcal{T}^{'}$, generating the training set $\mathcal{T}^{''}$. Finally, in the *ETL training* step, a rule set is learned using $\mathcal{T}^{''}$ as a training set. In Section 4.5, we show some experimental results that highlight the contribution of each one of these steps to the committee behavior. These steps are detailed in the following subsections.

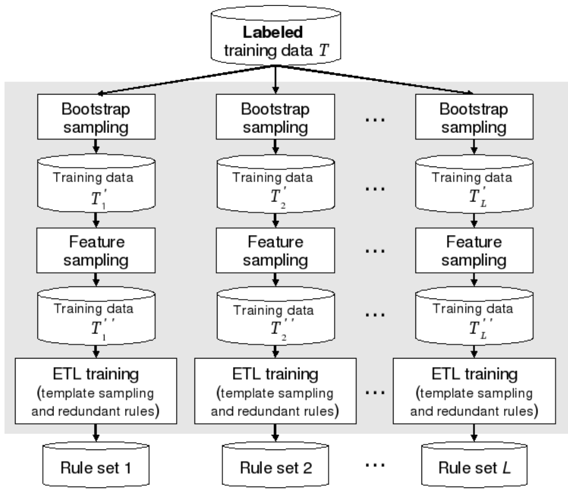

Fig. 1. ETL Committee training phase

Bootstrap sampling. In the *bootstrap sampling* step, a new version of the training set is generated using bootstrapping. *Bootstrapping* consists of sampling at random with replacement from the training set to generate an artificial training set of the same size as the original one. Hence, given a training set $\mathcal{T}$ consisting of n examples, a bootstrap replicate $\mathcal{T}^{'}$ is constructed by sampling n examples at random, with replacement, from $\mathcal{T}$. Bootstrapping is the central idea of Bagging, where it is used to provide diversity among the ensemble members.

According to Breiman [5], an ensemble of classifiers trained on different bootstrap replicates can be effective if the base learner is *unstable*. An unstable classifier is the one where small changes in the training set result in large changes in its predictions. Due to the greedy nature of the TBL learning process, rule selection is very sensitive to the occurrence of just a few examples. Usually, the rules in the tail of the learned rule set are selected based on just one or two error corrections. Therefore, we believe that small changes in the training set are able to significantly change the learned rule set. Moreover, since ETL uses DT to obtain templates and DT is an unstable learner [5], there are variability between the template sets generated from different bootstrap replicates. The use of different template sets has the potential to increase the ensemble diversity. The number of bootstrap replicates is called the *ensemble size*.

Feature sampling. In this step, a new version of the training set is generated by randomly selecting a subset of the available features. The manipulation of the input feature set is a general technique for generating multiple classifiers. As each classifier is generated using a randomly drawn feature subset, the diversity among the ensemble members tends to increase. Feature sampling is the main idea used in the Random Subspaces ensemble method. This strategy is particularly useful when a large set of features is available. The percentage of input features to be included in the subset is a parameter of ETL Committee.

ETL training. In the *ETL training* step, a set of transformation rules is learned using the training set resulted from the two previous steps. Here, *template sampling* and *redundant transformation rules* are used. We use template sampling for two reasons: (1) it provides more diversity among the ensemble members, since it increases the chance of each classifier to be trained with a very different template set; (2) it speeds up the training process, since less templates are used, enabling the learning of larger rule sets in a reasonable time. Note that by sampling templates we are sampling feature combinations. Hence, the template sampling can be seen as a kind of feature sampling at the base learner level. The number of templates to be sampled is also a parameter of ETL Committee.

We use redundant rules since it increases the overfitting, and more information from the training set is included in the learned model. Overfitting is another way to introduce diversity among the ensemble members [6,12,11].

3.2 ETL Committee Classification Phase

When classifying new data, each transformation rule set is independently applied to the input data. For each data point, each ETL model gives a classification, and we say the model “votes” for that class. The final data point classification is computed by majority voting. A drawback of ETL Committee, as well as the other ensemble methods, is that it increases the classification time. However, this process can be easily parallelized, since the application of each rule set is independent from the others.

3.3 Related Work

Breiman [12] presents an ensemble model called *Random Forest*, which uses bootstrapping and feature sampling. In the Random Forest learning process, first, bootstrap sampling is employed to generate multiple replicates of the training set. Then, a decision tree is grown for each training set replicate. When growing a tree, a subset of the available features is randomly selected at each node, the best split available within those features is selected for that node. Each tree is grown to the largest extent possible, and there is no pruning. Random Forest is specific for decision trees, since the feature sampling step occurs at the base learner level. ETL Committee differs from Random Forest in three main aspects: the base learner, where ETL is used; the feature sampling, which is done outside of the base learner; and the template sampling, which is a feature combination sampling method employed at the base learner level.

Panov & Dzeroski [1] describe an ensemble method that also combines Bagging and Random Subspaces. Their intention is to achieve an algorithm whose behavior is similar to the one of Random Forests, but with the advantage of being applicable to any base learner. Their method uses bootstrap sampling followed by feature sampling to generate different training set. They show that, when using DT as a base learner, their approach has a comparable performance to that of random forests. The ETL Committee method is similar to the one of Panov & Dzeroski in terms of training set manipulation. On the other hand, ETL Committee differs from the Panov & Dzeroski approach because it includes template sampling, which is a randomization at the base learner level.

4 Experiments

This section presents the experimental setup and results of the application of ETL Committee to three tasks: Text Chunking (TCK), Named Entity Recognition (NER) and Semantic Role Labeling (SRL). ETL Committee results are compared with the results of ETL and the state-of-the-art system for each corpus.

4.1 Machine Learning Modeling

The three tasks are modeled as token classification problems. Which means that, given a text, the learned system must predict a class label for each token.

We use the following ETL and ETL Committee common parameter setting in our experiments with the three tasks. The parameters are empirically tuned using the training and development sets available for the NER and SRL tasks.

ETL: we use a context window of size seven. We use templates which combine at most six features. Therefore, when extracting templates from DTs, the extraction process examines only the six first DT levels. We let the ETL algorithm learn rules whose score is at least two.

ETL$_{CMT}$**:** for the ETL Committee, in the *bootstrap sampling* step, we use sentences as sampling units for bootstrapping. We set the ensemble size to 100. In the *feature sampling* step, we randomly sample 90% of the features for each classifier. In the *ETL training* step, we let the ETL algorithm to learn the largest rule set possible. We use 50 as the default number of templates to be sampled in the creation of each classifier. However, we use 100 templates for the SRL task. This is because SRL involves a large number of features, which produces a larger number of templates.

BLS: For the TCK task, the initial classifier, or baseline system (BLS), assigns to each word the *chunk tag* that was most frequently associated with the part-of-speech of that word in the training set. For the NER task, the BLS assigns to each word the *named entity tag* that was most frequently associated with that word in the training set. If capitalized, an unknown word is tagged as a person, otherwise it is tagged as non entity. *Unknown words*, are the words that do not appear in the training set. For the SRL task, we use the same BLS proposed for the CoNLL-2004 shared task [15], which is based on six heuristic rules that make use of POS and phrase chunks.

4.2 Text Chunking

Text chunking consists in dividing a text into syntactically correlated parts of words [16]. It provides a key feature that helps on more elaborated NLP tasks such as NER and SRL.

The data used in the Text Chunking experiments is the CoNLL-2000 corpus, which is described in [16]. This corpus contains sections 15-18 and section 20 of the Penn Treebank, and is pre-divided into 8936-sentence training set and a 2012-sentence test set. This corpus is tagged with both POS and chunk tags. The *chunk tags* feature provides the phrase chunking annotation. We use the `IOB2` tagging style, where: `O`, means that the word is not a phrase; `B-X`, means that the word is the first one of a phrase type `X` and `I-X`, means that the word is inside of a phrase type `X`.

In [17], the authors present an SVM-based system with state-of-the-art performance for the CoNLL-2000 Corpus. Therefore, for this Corpus, we also list the SVM system performance reported by Wu et al.

In Table 1, we summarize the system performance results. The ETL system reduces the BLS $F_{\beta=1}$ error by 66%, from 22.93 to 7.72. The ETL_{CMT} system significantly reduces the $F_{\beta=1}$ error by 13% when compared to the single ETL. The ETL_{CMT} performance is competitive with the one of the SVM system.

4.3 Named Entity Recognition

Named Entity Recognition (NER) is the problem of finding all proper nouns in a text and to classify them among several given categories of interest. Usually, there are three given categories: Person, Organization and Location.

For the NER experiment, we use the Spanish CoNLL-2002 Corpus [18]. This corpus is annotated with four named entity categories: Person, Organization, Location and Miscellaneous. This corpus is pre-divided into training and test sets. It also includes a development set which have characteristics similar to the test corpora. This corpus is annotated with *POS* and *named entity (NE) tags*. We use the `IOB1` tagging style, where: `O`, means that the word is not a NE; `I-X`, means that the word is part of a NE type `X` and `B-X` is used for the leftmost word of a NE beginning immediately after another NE of the same type.

Table 1. System performances for the CoNLL-2000 Corpus

System	**Precision (%)**	**Recall (%)**	**$F_{\beta=1}$**
SVM	**94.12**	**94.13**	**94.12**
ETL_{CMT}	93.11	93.42	93.27
ETL	92.24	92.32	92.28
BLS	72.58	82.14	77.07

We generate three derived features: *Capitalization Information*, which classify the words according to their capitalization: First Uppercase, All Uppercase, Lowercase, Number or Punc.; *Dictionary Membership*, which assumes one of the following categorical values: Upper, Lower, Both or None; and *Word Length*, which classify the words according to their lengths: 1, 2, 3, 4, 5, 6-8 or >8.

Our named entity recognition approach follows the two stages strategy proposed in [3] for POS tagging. The first stage, the *morphological*, classifies the unknown words using morphological information. The second stage, the *contextual*, classifies the known and unknown words using contextual information. We use ETL and ETL Committee for the contextual stage only, since the morphological stage uses trivial templates.

In [19], the authors present an AdaBoost system with state-of-the-art performance for the Spanish CoNLL-2002 Corpus. Their AdaBoost system uses decision trees as a base learner. Therefore, for this Corpus, we also list the AdaBoost system performance reported by Carreras et al.

Table 2. System performances for the Spanish CoNLL-2002 Corpus

System	**Precision (%)**	**Recall (%)**	**$F_{\beta=1}$**
AdaBoost	**79.27**	**79.29**	**79.28**
ETL_{CMT}	76.99	77.94	77.46
ETL	75.50	77.07	76.28
BLS	49.59	63.02	55.51

In Table 2, we summarize the system performance results for the test set. The ETL system reduces the BLS $F_{\beta=1}$ error by 47%, from 44.49 to 23.72. The ETL_{CMT} system reduces the $F_{\beta=1}$ error by 5% when compared to the single ETL system. The ETL_{CMT} performance is very competitive with the one of the AdaBoost system. Moreover, for the Spanish CoNLL-2002, the ETL Committee system is in top three when compared with the 12 CoNLL-2002 contestant systems.

4.4 Semantic Role Labeling

Semantic Role Labeling (SRL) is the process of detecting basic event structures such as *who* did *what* to *whom*, *when* and *where* [20]. More specifically, for each predicate of a clause, whose head is typically a verb, all the constituents in the sentence which fill a semantic role of the verb have to be recognized. A verb and its set of semantic roles (arguments) form a *proposition* in the sentence. SRL provides a key knowledge that helps to build more elaborated document management and information extraction applications.

Since our purpose is to examine the ETL Committee performance for a complex task, we do not use the full parsing information in our SRL experiments. Therefore, we evaluate the performance of ETL Committee over the CoNLL-2004 Corpus [15]. This corpus was used in the CoNLL-2004 shared task, which consisted in resolving SRL without full parsing. It is a subset of the Proposition Bank (PropBank), an approximately one-million-word corpus annotated with predicate-argument structures. The PropBank annotates the Wall Street Journal part of the Penn TreeBank with verb argument structure. The CoNLL-2004 Corpus uses Penn TreeBank sections 15-18 for training and section 21 for test. Section 20 is used as a development set.

The CoNLL-2004 Corpus is annotated with four basic input features: *POS tags*, *phrase chunks*, *clauses* and *named entities*. The Corpus also includes two other features: the *target verbs* feature, which indicates the verbs whose arguments must be labeled; and *srl tags*, which provides the semantic labeling. The *srl tags* used in the PropBank annotation numbers the arguments of each predicate from `A0` to `A5`. Adjunctive arguments are referred to as `AM-T`, where `T` is the type of the adjunct. Argument references share the same label with the actual argument prefixed with `R-`. References are typically pronominal.

Using the input features, we produce the following thirteen derived features. *Token Position*: indicates if the token comes before or after the target verb. *Temporal*: indicates if the word is or not a temporal keyword. *Path*: the sequence of chunk tags between the chunk and the target verb. *Pathlex*: the same as the *path* feature with the exception that here we use the preposition itself instead of the `PP` chunk tag. *Distance*: the number of chunks between the chunk and the target verb. *VP Distance*: distance, in number of `VP` chunks, between the token and the verb. *Clause Path*: the clause bracket chain between the token and the target verb. *Clause Position*: indicates if the token is inside or outside of the clause which contains the target verb. *Number of Predicates*: number of target verbs in the sentence. *Voice*: indicates the target verb voice. *Target Verb POS*: POS tag of the target verb. *Predicate POS Context*: the POS tags of the words that immediately precede and follow the predicate. *Predicate Argument Patterns*: for each predicate, we identify the most frequent left and right patterns of the core arguments (`A0` through `A5`) in the training set. All these features were previously used in other SRL systems [21].

SRL Preprocessing. Our system classifies chunks instead of words. Therefore, here, a token represents a complete text chunk. In the preprocessing step, the original word-based tokens are collapsed in order to generate the new representation. In the collapsing process, only the feature values of the phrase chunk headwords are retained. The chunk headword is defined as its rightmost word. This preprocessing speeds up the training step, since the number of tokens to be annotated are reduced. Moreover, larger sentence segments are covered with smaller context window sizes.

We treat propositions independently. Therefore, for each target verb we generate a separate sequence of tokens to be annotated. In general, all the arguments of a proposition are inside the target verb clause. Hence, we do not include tokens that are outside of the target verb clause. The only exception is when we have a nested clause that begins with a target verb. Here, we must also include the external clause.

SRL Results. Hacioglu et al. [21] present a SVM system with state-of-the-art performance for the CoNLL-2004 Corpus. Therefore, we also list the SVM system performance reported by Hacioglu et al.

In Table 3, we summarize the system performance results for the test set. The ETL system reduces the BLS $F_{\beta=1}$ error by 40%, from 60.55 to 36.63. The ETL_{CMT} system reduces the $F_{\beta=1}$ error by 11% when compared to the single ETL system. The ETL_{CMT} performance is very competitive with the SVM system. Nevertheless, the ETL Committee system is in top two when compared with the 10 CoNLL-2004 contestant systems. Moreover, the precision of the ETL_{CMT} system is better than the one of the SVM system, and a reasonable recall is maintained. We obtain similar results in the development set.

Table 3. System performances for the CoNLL-2004 Corpus

System	**Precision (%)**	**Recall (%)**	$\mathbf{F}_{\beta=1}$
SVM	72.43	**66.77**	**69.49**
ETL_{CMT}	**76.44**	60.25	67.39
ETL	70.60	57.48	63.37
BLS	55.57	30.58	39.45

4.5 ETL Committee Behavior

In this section, we present some results on the behavior of the ETL Committee learning strategy. Our intention is three-fold: to analyze the importance of redundant rules; to investigate how the ensemble performance behaves as the ensemble size increases and; to analyze the ETL Committee performance sensitivity to the percentage of sampled features. We use the SRL CoNLL-2004 development set to assess the system performances.

Figure 2 demonstrates the relationship of the $F_{\beta=1}$ for a given number of ETL classifiers in the ensemble. We can see that the ensemble performance increases rapidly until approximately 40 classifiers are included. Then, the $F_{\beta=1}$ increases slowly until it gets stable with around 100 classifiers. Note that using just 50 models we have a $F_{\beta=1}$ of 68.7. ETL Committee has a similar behavior in the other two tasks: TCK and NER.

In Table 4, we show the ETL Committee performance for different values of the feature sampling parameter. For this experiment, we create ensembles of 50

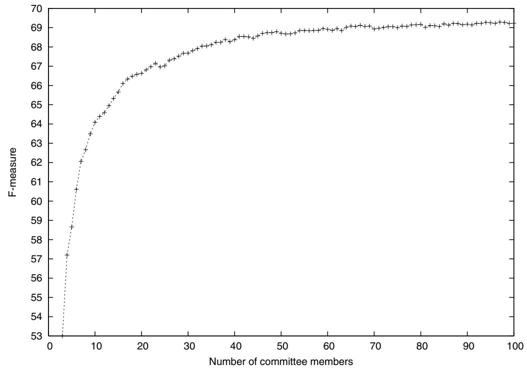

Fig. 2. $F_{\beta=1}$ x Number of committee members curve

classifiers. The best performance occurs when 70% of the features are randomly sampled for each classifier. In this case, the $F_{\beta=1}$ increases by about 0.7 when compared to the result in the first table line, where all features are used. In Table 4, we can see that even using only 50% of the features, the performance does not degrade. However, using less than 70% of the features can lead to poor results for tasks with a few number of features such as TCK.

Table 4. ETL Committee performance sensitivity to the percentage of sampled features

Percentage of sampled features	**Precision (%)**	**Recall (%)**	$\mathbf{F_{\beta=1}}$
100%	75.43	61.95	68.03
90%	75.97	62.21	68.40
70%	76.44	**62.40**	**68.71**
50%	**76.64**	61.50	68.24

In Table 5, we show the ETL Committee performance when redundant rules are used or not used. For this experiment, we also create ensembles of 50 classifiers. The result in the first table line corresponds to the default ETL Committee method, which uses redundant rules. The second table line presents the ensemble performance when redundant rules are not used. In this case, the $F_{\beta=1}$ drops by about two points. This indicates that the overfitting provided by redundant rules is very important to the construction of more diverse ETL classifiers.

Table 5. Importance of redundant rules for the ETL Committee performance

Redundant rules	**Precision (%)**	**Recall (%)**	$\mathbf{F_{\beta=1}}$
YES	76.44	62.40	68.71
NO	76.63	59.10	66.73

5 Conclusions

Entropy Guided Transformation Learning is a machine learning algorithm that generalizes TBL. In this work, we present ETL Committee, a new ensemble method that uses ETL as the base learner. It combines the main ideas of Bagging and Random Subspaces. We also propose a strategy to include redundancy in transformation-based models. To evaluate the effectiveness of the ensemble method, we apply it to three NLP tasks: TCK, NER and SRL.

Our experimental results indicate that ETL Committee significantly outperforms single ETL models. We also find out that redundant rules have a significant impact in the ensemble result. This finding indicates that the overfitting provided by redundant rules helps the construction of more diverse ETL classifiers. Some positive characteristics of the proposed ensemble strategy are worth to mention. First, it improves the ETL effectiveness without any additional human effort. Second, it is particularly useful when dealing with very complex tasks that use large feature sets. This is the case of the SRL task, where ETL Committee provides a significant $F_{\beta=1}$ improvement. And finally, the resulting training and classification processes are very easy to parallelize, since each classifier is independent from the others. The main drawback of ETL Committee is the increasing of the classification time. A possible way to overcome this issue is to convert transformation rules into deterministic finite-state transducers, as proposed by [22].

References

1. Panov, P., Džeroski, S.: Combining bagging and random subspaces to create better ensembles. In: Berthold, M.R., Shawe-Taylor, J., Lavrač, N. (eds.) IDA 2007. LNCS, vol. 4723, pp. 118–129. Springer, Heidelberg (2007)
2. García-Pedrajas, N., Ortiz-Boyer, D.: Boosting random subspace method. Neural Networks 21(9), 1344–1362 (2008)
3. Brill, E.: Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Comp. Linguistics 21(4), 543–565 (1995)
4. Milidiú, R.L., dos Santos, C.N., Duarte, J.C.: Phrase chunking using entropy guided transformation learning. In: Proceedings of ACL 2008, Columbus, Ohio (2008)
5. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
6. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. 20(8), 832–844 (1998)
7. dos Santos, C.N., Milidiú, R.L.: Entropy Guided Transformation Learning. In: Foundations of Computational Intelligence. Learning and Approximation, vol. 1. Studies in Computational Intelligence, vol. 201, pp. 159–184. Springer, Heidelberg (2009)
8. dos Santos, C.N., Milidiú, R.L., Rentería, R.P.: Portuguese part-of-speech tagging using entropy guided transformation learning. In: Teixeira, A., de Lima, V.L.S., de Oliveira, L.C., Quaresma, P. (eds.) PROPOR 2008. LNCS (LNAI), vol. 5190, pp. 143–152. Springer, Heidelberg (2008)
9. Fernandes, E.R., Pires, B.A., dos Santos, C.N., Milidiú, R.L.: Clause identification using entropy guided transformation learning. In: Proceedings of STIL 2009 (2009)
10. Milidiú, R.L., Crestana, C.E.M., dos Santos, C.N.: A token classification approach to dependency parsing. In: Proceedings of STIL 2009 (2009)

11. Banfield, R.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P.: Ensemble diversity measures and their application to thinning. Information Fusion 6(1), 49–62 (2005)
12. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
13. Florian, R.: Transformation Based Learning and Data-Driven Lexical Disambiguation: Syntactic and Semantic Ambiguity Resolution. PhD thesis, The Johns Hopkins University (2002)
14. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
15. Carreras, X., Màrquez, L.: Introduction to the conll-2004 shared task: Semantic role labeling. In: Ng, H.T., Riloff, E. (eds.) HLT-NAACL, Workshop CoNLL 2004, Boston, USA, May 2004, pp. 89–97. ACL (2004)
16. Sang, E.F.T.K., Buchholz, S.: Introduction to the conll-2000 shared task: chunking. In: Proceedings of the 2nd workshop on Learning language in logic and the 4th CONLL, Morristown, NJ, USA, pp. 127–132. ACL (2000)
17. Wu, Y.C., Chang, C.H., Lee, Y.S.: A general and multi-lingual phrase chunking model based on masking method. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 144–155. Springer, Heidelberg (2006)
18. Tjong Kim Sang, E.F.: Introduction to the conll-2002 shared task: Language-independent named entity recognition. In: Proceedings of CoNLL 2002, Taipei, Taiwan, pp. 155–158 (2002)
19. Carreras, X., Màrques, L., Padró, L.: Named entity extraction using adaboost. In: Proceedings of CoNLL 2002, Taipei, Taiwan, pp. 167–170 (2002)
20. Surdeanu, M., Màrquez, L., Carreras, X., Comas, P.: Combination strategies for semantic role labeling. JAIR 29, 105–151 (2007)
21. Hacioglu, K., Pradhan, S.S., Ward, W.H., Martin, J.H., Jurafsky, D.: Semantic role labeling by tagging syntactic chunks. In: Ng, H.T., Riloff, E. (eds.) HLT-NAACL, Workshop CoNLL 2004, Boston, USA. ACL (May 2004)
22. Roche, E., Schabes, Y.: Deterministic part-of-speech tagging with finite-state transducers. Comput. Linguist. 21(2), 227–253 (1995)