Native Language Detection with Tree Substitution Grammars

Ben Swanson
Brown University
chonger@cs.brown.edu

Eugene Charniak
Brown University
ec@cs.brown.edu

Abstract

We investigate the potential of Tree Substitution Grammars as a source of features for native language detection, the task of inferring an author's native language from text in a different language. We compare two state of the art methods for Tree Substitution Grammar induction and show that features from both methods outperform previous state of the art results at native language detection. Furthermore, we contrast these two induction algorithms and show that the Bayesian approach produces superior classification results with a smaller feature set.

1 Introduction

The correlation between a person's native language (L1) and aspects of their writing in a second language (L2) can be exploited to predict L1 label given L2 text. The International Corpus of Learner English (Granger et al, 2002), or ICLE, is a large set of English student essays annotated with L1 labels that allows us to bring the power of supervised machine learning techniques to bear on this task. In this work we explore the possibility of automatically induced Tree Substitution Grammar (TSG) rules as features for a logistic regression model[1] trained to predict these L1 labels.

Automatic TSG induction is made difficult by the exponential number of possible TSG rules given a corpus. This is an active area of research with two distinct effective solutions. The first uses a nonparametric Bayesian model to handle the large number of rules (Cohn and Blunsom, 2010), while the second is inspired by tree kernel methods and extracts common subtrees from pairs of parse trees (Sangati and Zuidema, 2011). While both are effective, we show that the Bayesian method of TSG induction produces superior features and achieves a new best result at the task of native language detection.

[1] a.k.a. Maximum Entropy Model

2 Related Work

2.1 Native Language Detection

Work in automatic native language detection has been mainly associated with the ICLE, published in 2002. Koppel et al (2005) first constructed such a system with a feature set consisting of function words, POS bi-grams, and character n-grams. These features provide a strong baseline but cannot capture many linguistic phenomena.

More recently, Wong and Dras (2011a) considered syntactic features for this task, using logistic regression with features extracted from parse trees produced by a state of the art statistical parser. They investigated two classes of features: reranking features from the Charniak parser and CFG features. They showed that while reranking features capture long range dependencies in parse trees that CFG rules cannot, they do not produce classification performance superior to simple CFG rules. Their CFG feature approach represents the best performing model to date for the task of native language detection. Wong and Dras (2011b) also investigated the use of LDA topic modeling to produce a latent feature set of reduced dimensionality, but failed to outperform baseline systems with this approach.

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 193–197,
Jeju, Republic of Korea, 8-14 July 2012. ©2012 Association for Computational Linguistics

2.2 TSG induction

One inherent difficulty in the use of TSGs is controlling the size of grammars automatically induced from data, which with any reasonable corpus quickly becomes too large for modern workstations to handle. When automatically induced TSGs were first proposed by Bod (1991), the problem of grammar induction was tackled with random selection of fragments or weak constraints that led to massive grammars.

A more principled technique is to use a sparse nonparametric prior, as was recently presented by Cohn et al (2009) and Post and Gildea (2009). They provide a local Gibbs sampling algorithm, and Cohn and Blunsom (2010) later developed a block sampling algorithm with better convergence behavior. While this Bayesian method has yet to produce state of the art parsing results, it has achieved state of the art results for unsupervised grammar induction (Blunsom and Cohn, 2010) and has been extended to synchronous grammars for use in sentence compression (Yamangil and Shieber, 2010).

More recently, (Sangati and Zuidema, 2011) presented an elegantly simple heuristic inspired by tree kernels that they call DoubleDOP. They showed that manageable grammar sizes can be obtained from a corpus the size of the Penn Treebank by recording all fragments that occur at least twice, subject to a pairwise constraint of maximality. Using an additional heuristic to provide a distribution over fragments, DoubleDOP achieved the current state of the art for TSG parsing, competing closely with the absolute best results set by refinement based parsers.

2.3 Fragment Based Classification

The use of parse tree fragments for classification began with Collins and Duffy (2001). They used the number of common subtrees between two parse trees as a convolution kernel in a voted perceptron and applied it as a parse reranker. Since then, such tree kernels have been used to perform a variety of text classification tasks, such as semantic role labeling (Moschitti et al, 2008), authorship attribution (Kim et al, 2010), or the work of Suzuki and Isozaki (2006) that performs question classification, subjectivity detection, and polarity identification.

Syntactic features have also been used in non-kernelized classifiers, such as in the work of Wong and Dras (2011a) mentioned in Section 2.1. Additional examples include Raghavan et al (2010), which uses a CFG language model to perform authorship attribution, and Post (2011), which uses TSG features in a logistic regression model to perform grammaticality detection.

3 Tree Substitution Grammars

Tree Substitution Grammars are similar to Context Free Grammars, differing in that they allow rewrite rules of arbitrary parse tree structure with any number of nonterminal or terminal leaves. We adopt the common term *fragment*[2] to refer to these rules, as they are easily visualised as fragments of a complete parse tree.

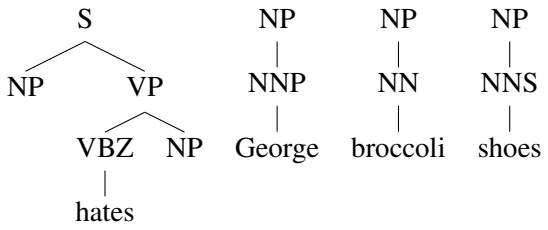

Figure 1: Fragments from a Tree Substitution Grammar capable of deriving the sentences "George hates broccoli" and "George hates shoes".

3.1 Bayesian Induction

Nonparametric Bayesian models can represent distributions of unbounded size with a dynamic parameter set that grows with the size of the training data. One method of TSG induction is to represent a probabilistic TSG with Dirichlet Process priors and sample derivations of a corpus using MCMC.

Under this model the posterior probability of a fragment e is given as

$$P(e|\mathbf{e}^-, \alpha, P_0) = \frac{\#_e + \alpha P_0}{\#_\bullet + \alpha} \qquad (1)$$

where $\mathbf{e}^-$ is the multiset of fragments in the current derivations excluding e, $\#_e$ is the count of the fragment e in $\mathbf{e}^-$, and $\#_\bullet$ is the total number of fragments in $\mathbf{e}^-$ with the same root node as e. P_0 is

[2] As opposed to *elementary tree*, often used in related work

a PCFG distribution over fragments with a bias towards small fragments. α is the concentration parameter of the DP, and can be used to roughly tune the number of fragments that appear in the sampled derivations.

With this posterior distribution the derivations of a corpus can be sampled tree by tree using the block sampling algorithm of Cohn and Blunsom (2010), converging eventually on a sample from the true posterior of all derivations.

3.2 DoubleDOP Induction

DoubleDOP uses a heuristic inspired by tree kernels, which are commonly used to measure similarity between two parse trees by counting the number of fragments they share. DoubleDOP uses the same underlying technique, but caches the shared fragments instead of simply counting them. This yields a set of fragments where each member is guaranteed to appear at least twice in the training set.

In order to avoid unmanageably large grammars only maximal fragments are retained in each pairwise extraction, which is to say that any shared fragment that occurs inside another shared fragment is discarded. The main disadvantage of this method is that the complexity scales quadratically with the training set size, as all pairs of sentences must be considered. It is fully parallelizable, however, which mediates this disadvantage to some extent.

4 Experiments

4.1 Methodology

Our data is drawn from the International Corpus of Learner English (Version 2), which consists of raw unsegmented English text tagged with L1 labels. Our experimental setup follows Wong and Dras (2011a) in analyzing Chinese, Russian, Bulgarian, Japanese, French, Czech, and Spanish L1 essays. As in their work we randomly sample 70 training and 25 test documents for each language. All reported results are averaged over 5 subsamplings of the full data set.

Our data preproccesing pipeline is as follows: First we perform sentence segmentation with OpenNLP and then parse each sentence with a 6 split grammar for the Berkeley Parser (Petrov et al, 2006). We then replace all terminal symbols which do not occur in a list of 598 function words[3] with a single UNK terminal. This aggressive removal of lexical items is standard in this task and mitigates the effect of other unwanted information sources such as topic and geographic location that are correlated with native language in the data.

We contrast three different TSG feature sets in our experiments. First, to provide a baseline, we simply read off the CFG rules from the data set (note that a CFG can be taken as a TSG with all fragments having depth one). Second, in the method we call BTSG, we use the Bayesian induction model with the Dirichlet process' concentration parameters tuned to 100 and run for 1000 iterations of sampling. We take as our resulting finite grammar the fragments that appear in the sampled derivations. Third, we run the parameterless DoubleDOP (2DOP) induction method.

Using the full 2DOP feature set produces over 400k features, which heavily taxes the resources of a single modern workstation. To balance the feature set sizes between 2DOP and BTSG we pass back over the training data and count the actual number of times each fragment recovered by 2DOP appears. We then limit the list to the n most common fragments, where n is the average number of fragments recovered by the BTSG method (around 7k). We refer to results using this trimmed feature set with the label 2DOP, using 2DOP(F) to refer to DoubleDOP with the full set of features.

Given each TSG, we create a binary feature function for each fragment e in the grammar such that the feature $f_e(d)$ is active for a document d if there exists a derivation of some tree $t \in d$ that uses e. Classification is performed with the Mallet package for logistic regression using the default initialized MaxEntTrainer.

5 Results

5.1 Predictive Power

The resulting classification accuracies are shown in Table 1. The BTSG feature set gives the highest performance, and both true TSG induction techniques outperform the CFG baseline.

[3] We use the stop word list distributed with the ROUGE summarization evaluation package.

Model	Accuracy (%)
CFG	72.6
2DOP	73.5
2DOP(F)	76.8
BTSG	78.4

Table 1: Classification accuracy

Model	VoteOne (%)	VoteAll (%)
CFG	69.6	74.7
2DOP	69.1	73.5
BTSG	72.5	76.5

Table 2: Sentence based classification accuracy

The CFG result represents the work of Wong and Dras (2011a), the previous best result for this task. While in their work they report 80% accuracy with the CFG model, this is for a single sampling of the full data set. We observed a large variance in classification accuracy over such samplings, which includes some values in their reported range but with a much lower mean. The numbers we report are from our own implementation of their CFG technique, and all results are averaged over 5 random samplings from the full corpus.

For 2DOP we limit the 2DOP(F) fragments by choosing the 7k with maximum frequency, but there may exist superior methods. Indeed, Wong and Dras (2011a) claims that Information Gain is a better criteria. However, this metric requires a probabilistic formulation of the grammar, which 2DOP does not supply. Instead of experimenting with different limiting metrics, we note that when all 400k rules are used, the averaged accuracy is only 76.8 percent, which still lags behind BTSG.

5.2 Robustness

We also investigated different classification strategies, as binary indicators of fragment occurrence over an entire document may lead to noisy results. Consider a single outlier sentence in a document with a single fragment that is indicative of the incorrect L1 label. Note that it is just as important in the eyes of the classifier as a fragment indicative of the correct label that appears many times. To investigate this phenomena we classified individual sentences, and used these results to vote for each document level label in the test set.

We employed two voting schemes. In the first, VoteOne, each sentence contributes one vote to its maximum probability label. In the second, VoteAll, the probability of each L1 label is contributed as a partial vote. Neither method increases performance for BTSG or 2DOP, but what is more interesting is that in both cases the CFG model outperforms 2DOP (with less than half of the features). The robust behavior of the BTSG method shows that it finds correctly discriminative features across several sentences in each document to a greater extent than other methods.

5.3 Concision

One possible explanation for the superior performance of BTSG is that DDOP is prone to yielding multiple fragments that represent the same linguistic phenomena, leading to sets of highly correlated features. While correlated features are not crippling to a logistic regression model, they add computational complexity without contributing to higher classification accuracy.

To address this hypothesis empirically, we considered pairs of fragments e_A and e_B and calculated the pointwise mutual information (PMI) between events signifying their occurrence in a sentence. For BTSG, the average pointwise mutual information over all pairs (e_A, e_B) is $-.14$, while for 2DOP it is $-.01$. As increasingly negative values of PMI indicate exclusivity, this supports the claim that DDOP's comparative weakness is to some extent due to feature redundancy.

6 Conclusion

In this work we investigate automatically induced TSG fragments as classification features for native language detection. We compare Bayesian and DoubleDOP induced features and find that the former represents the data with less redundancy, is more robust to classification strategy, and gives higher classification accuracy. Additionally, the Bayesian TSG features give a new best result for the task of native language detection.

References

Mohit Bansal and Dan Klein 2010. Simple, accurate parsing with an all-fragments grammar. *Association for Computational Linguistics*.

Phil Blunsom and Trevor Cohn 2010. Unsupervised Induction of Tree Substitution Grammars for Dependency Parsing. *Empirical Methods in Natural Language Processing*.

Rens Bod 1991. A Computational Model of Language Performance: Data Oriented Parsing. *Computational Linguistics in the Netherlands*.

Trevor Cohn, Sharon Goldwater, and Phil Blunsom. 2009. Inducing Compact but Accurate Tree-Substitution Grammars. In *Proceedings NAACL*.

Trevor Cohn, and Phil Blunsom 2010. Blocked inference in Bayesian tree substitution grammars. *Association for Computational Linguistics*.

Michael Collins, Nigel Duffy 2001. Convolution Kernels for Natural Language. *Advances in Neural Information Processing Systems*.

Joshua Goodman 2003. Efficient parsing of DOP with PCFG-reductions. *In Bod et al. chapter 8.*.

S. Granger, E. Dagneaux and F. Meunier. 2002. *International Corpus of Learner English*, (ICLE).

Sangkyum Kim, Hyungsul Kim, Tim Weninger, and Jiawei Han 2010. Authorship classification: a syntactic tree mining approach. *Proceedings of the ACM SIGKDD Workshop on Useful Patterns*.

Koppel, Moshe and Schler, Jonathan and Zigdon, Kfir. 2005. Determining an author's native language by mining a text for errors. *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*.

Alessandro Moschitti, Daniele Pighin and Roberto Basili 2008. Tree Kernels for Semantic Role Labeling. *Computational Linguistics*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. *Association for Computational Linguistics*.

Matt Post and Daniel Gildea. 2009. Bayesian Learning of a Tree Substitution Grammar. *Association for Computational Linguistics*.

Matt Post. 2011. Judging Grammaticality with Tree Substitution Grammar Derivations. *Association for Computational Linguistics*.

Sindhu Raghavan, Adriana Kovashka and Raymond Mooney 2010. Authorship attribution using probabilistic context-free grammars. *Association for Computational Linguistics*.

Sangati, Federico and Zuidema, Willem 2011. Accurate Parsing with Compact Tree-Substitution Grammars: Double-DOP. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Jun Suzuki and Hideki Isozaki 2006. Sequence and tree kernels with statistical feature mining. *Advances in Neural Information Processing Systems*.

Sze-Meng Jojo Wong and Mark Dras 2011. Exploiting Parse Structures for Native Language Identification. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Sze-Meng Jojo Wong and Mark Dras 2011. Topic Modeling for Native Language Identification. *Proceedings of the Australasian Language Technology Association Workshop*.

Elif Yamangil, Stuart M. Shieber 2010. Bayesian Synchronous Tree-Substitution Grammar Induction and Its Application to Sentence Compression.. *Association for Computational Linguistics*.