Parser Features for Sentence Grammaticality Classification

Sze-Meng Jojo Wong
Centre for Language Technology
Macquarie University
Sydney, NSW, Australia
sze.wong@mq.edu.au

Mark Dras
Centre for Language Technology
Macquarie University
Sydney, NSW, Australia
mark.dras@mq.edu.au

Abstract

Automatically judging sentences for their grammaticality is potentially useful for several purposes — evaluating language technology systems, assessing language competence of second or foreign language learners, and so on. Previous work has examined parser 'byproducts', in particular parse probabilities, to distinguish grammatical sentences from ungrammatical ones. The aim of the present paper is to examine whether the primary output of a parser, which we characterise via CFG production rules embodied in a parse, contains useful information for sentence grammaticality classification; and also to examine which feature selection metrics are most useful in this task. Our results show that using gold standard production rules alone can improve over using parse probabilities alone. Combining parser-produced production rules with parse probabilities further produces an improvement of 1.6% on average in the overall classification accuracy.

1 Introduction

Automatically judging sentences for their grammaticality has been a long-standing research problem within the natural language processing community. The ability of distinguishing grammatical sentences from ungrammatical ones has many potential applications, which include evaluating language technology systems such as natural language generation (Mutton et al., 2007) and machine translation (Gamon et al., 2005), as well as assessing language competence of second language or foreign language learners (Brockett et al., 2006; Gamon et al., 2008; Han et al., 2010).

Various approaches have been proposed in the past to address this typical classification problem. A number of these existing studies attempt to exploit some form of 'parser byproduct' as classification features for machine learning: for instance, (log) probability of a parse tree, number of partial (incomplete) parse trees, parsing duration, and such (Mutton et al., 2007; Sun et al., 2007; Foster et al., 2008; Wagner et al., 2009). The aim of this paper is to examine whether the primary output of a parser contains useful information for this classification problem; we characterise this information by the CFG production rules embodied in a parse. The intuition is that particular production rules might be strongly characteristic of ungrammatical sentences, and that looking at individual rules might provide clues that are aggregated out in measures such as the parse tree probability.

We carry out experiments to test this intuition, using the Penn treebank and an artificially created ungrammatical version created by Foster et al. (2008). This allows a large amount of data to be used for classification, embodying a controlled ungrammaticality that is suitable for preliminary work in this direction; it is for similar reasons that construction of erroneous corpora has become a more prominent line of computational linguistic research lately (Foster and Andersen, 2009; Han et al., 2010; Dickinson, 2010).

The present study is carried out in two stages. In the first stage, following Foster et al. (2008), we induce three models from a probabilistic parser by re-training it with a (presumably) grammatically well-formed corpus, a grammatically ill-formed corpus, and a mixed corpus consisting of both grammatical and ungrammatical sentences. The model which outperforms the others is then used for all the subsequent parsing tasks. In the next stage, we utilise the outputs of the parser from the first stage and a parser trained on only grammatical text for sentence grammaticality classification, in which two classes of feature are to be examined — parse probabilities based on the parser outputs

Sze-Meng Jojo Wong and Mark Dras. 2010. Parser Features for Sentence Grammaticality Classification. In *Proceedings of Australasian Language Technology Association Workshop*, pages 67−75

and production rules based on both the gold standard and the parser outputs. A number of feature selection metrics are explored to obtain a set of discriminative parse rules for classifying English sentences based on their grammaticality.

The remainder of this paper is structured as follows. We review some related work in Section 2. In Section 3, we detail the experimental settings and the feature selection metrics. Sections 4.1 and 4.2 then present the parsing results and the classification results, respectively; followed by discussion in Section 5.

2 Related Work

In this section, we briefly review some of the related studies on judging sentence grammaticality. We also discuss some of the recent work concerning the construction of erroneous corpora for the purpose of grammatical error detection.

2.1 Sentence Grammaticality Judgement

In some ways, the present study is an extension of the work presented by Foster et al. (2008). Their intention is to improve the robustness of a probabilistic parser which might not be initially designed to handle ungrammatical sentences. Retraining on both grammatical and ungrammatical sentences enabled a parser to parse ungrammatical sentences at a relatively satisfactory level without compromising its initial performance on grammatical sentences. To attain an optimal parsing accuracy, the parser output of a sentence is chosen according to the highest parse probability of the most likely parse tree returned by the three induced models of the Charniak and Johnson reranking parser (Charniak and Johnson, 2005) trained across three different corpora — grammatical, ungrammatical, and a combination of both. Their experiments show that their parse probability-based classifier which can be considered as an integration of two parsers (one trained on grammatical data and the other trained on some ungrammatical data) is able to parse ungrammatical sentences better than the original parser trained exclusively on a grammatical corpus. The grammatical corpus used by Foster et al. (2008) is the Wall Street Journal (WSJ) treebank, and the ungrammatical version is one that they generated (see Section 2.2).

In a related work (Wagner et al., 2009), a number of parser outputs are utilised for classifying a sentence as to whether it is grammatical or ungrammatical. In addition to the widely used part-of-speech n-grams, they made use of two types of parsers, each based on a different grammar — the precision grammar parser (XLE parser) and the probabilistic parser (Charniak and Johnson parser). Features extracted from the probabilistic parser, which include the differences in log probabilities of parse trees and the structural differences between parse trees, are better discriminants as compared to both the n-gram features and the parser statistics outputs obtained from the precision-grammar-based parser. The overall accuracy achieved is within the range of 65-75% by using the combination of all the feature sets.

A similar idea had been used by Mutton et al. (2007), who discovered that parser outputs can be used as metrics for assessing generated sentence fluency. The underlying idea is that a poorer performance of the parser on one sentence relative to another might indicate that there is some degree of ungrammaticality or disfluency in the former. Outputs from multiple parsers, such as log probability of the most likely parse, number of partial parse trees, and number of invalid parses were investigated. The combination of multiple parser outputs outperforms individual parser metrics.

Parse probability was also used by Sun et al. (2007) for machine learning based classification. There, the type of feature they term 'labelled sequential patterns' like non-contiguous n-grams, proves more important for sentence grammaticality classification with an accuracy rate of over 80%. To provide useful feedback to learners of English as a Second Language (ESL), two English learner corpora are used — Japanese and Chinese.

The techniques of phrase-based SMT have been adapted for grammaticality judgement on ESL sentences as well. Brockett et al. (2006) treat error correction as a translation task, and solve it by using the noisy channel model. They made use of the Chinese Learner Error Corpus as a template for training data creation; but also needed large sets of parallel corpora.

2.2 Erroneous Corpora Construction

Large-scale ungrammatical corpora are crucial for research concerning grammaticality judgement, in particular for classification training. Recently, a number of pieces of corpus-based research have been undertaken to collect authentic errors as well as to generate synthetic errors for this purpose.

The thesis work of Foster (2005) involved an extensive analysis of grammar error types across a 20K word corpus consisting of newspaper articles, emails, Internet forum postings, and academic papers. This led to the development of an ungrammatical version of the WSJ treebank according to a model derived from this analysis (Foster, 2007); this also included a procedure for constructing trees for the ungrammatical sentences. Ungrammatical sentences are constructed by introducing errors into the original (grammatical) WSJ sentences through the operations of word insertion, substitution, and deletion. Each ungrammatical sentence is then tagged with the gold standard parse tree, a transformation of the original parse tree of its grammatical counterpart with the *intended* meaning remained intact. The types of errors introduced include missing word, extra word, real-word spelling, agreement, and verb form: according to Foster, these comprise 72% of the analysed errors. Subsequently, Foster and Anderson (2009) developed an automated error generation tool that can be applied to any text.

Okanohara and Tsujii (2007) attempt to produce grammatically ill-formed sentences termed as *pseudo-negative* examples which are not representative of authentic errors but more like machine translation outputs. Han et al. (2010), construct an error-annotated English corpus comprised of texts written by Korean learners of English and demonstrate that classifiers trained on error-annotated data outperform those that trained exclusively on well-formed data produced by native English speakers. Dickinson (2010), in other recent corpus-based research aiming to address morphological errors found in highly inflecting languages, creates learner-like morphological errors from a segmented lexicon.

3 Experimental Setup

We first describe the data used, and then the consequent re-training of the parser in the first stage of the experiments. We follow that with a description of the feature selection metrics for the classification experiments in the second stage.

3.1 Grammatical and Ungrammatical Corpora

Given that the goal of the present study is to distinguish between grammatical and ungrammatical sentences, two corpora are needed. For the grammatical sentences, we take the WSJ treebank by making the assumption that they are grammatically well-formed. On the other hand, the ungrammatical sentences are obtained from noisy (distorted) versions of WSJ created by Foster (2007) and used in Foster et al. (2008). As mentioned earlier, the grammatically ill-formed WSJ sentences were generated by introducing errors to the initially well-form WSJ sentences through the operations of insertion, deletion, and substitution.

It should be noted that there are two noisy versions of WSJ. The first is a complete parallel of the original WSJ which consists of 24 sections (from Section 0 to Section 23) and the second set is a much smaller one covering only 6 sections (including Section 0, Section 2-5, and Section 23). The latter is considered noisier data since the sentences were generated by applying the error generation procedures to the first set of ungrammatical WSJ sentences. Hencefore, we denote the three sets of WSJ treebank as follows: *PureWSJ* — the original WSJ; *NoisyWSJ* — the first set of less noisy WSJ; and *NoisierWSJ* — the second set of more noisy WSJ.

In Figure 1 we give examples of sentences with trees generated by insertion and deletion, and their grammatical counterparts.

3.2 Re-training of Parsers

In order to enable a parser to be able to parse ungrammatical sentences, we re-train a probabilistic parser on both grammatical and ungrammatical corpora. This idea is adopted from Foster et al. (2008). By and large, we replicate the experiments conducted in Foster et al. (2008) with the exception that the parser used in our study is the Stanford Parser (Klein and Manning, 2003), chosen for ease of re-training.

In this first stage, we conduct five experiments to re-train the Stanford Parser to induce a more robust parser capable of parsing both grammatical and ungrammatical sentences. In the first three experiments, three models of parser are induced by training on three different sets of corpora — first on the original WSJ (*PureWSJ*); second on the noisy WSJ (*NoisyWSJ*); and third on both the original and noisy WSJ (*PureWSJ* plus *NoisyWSJ*). We denote these three parser models as *PureParser*, *NoisyParser*, and *MixedParser*. In order to gauge its ability of parsing both grammatical and ungrammatical sentences, each of these models is

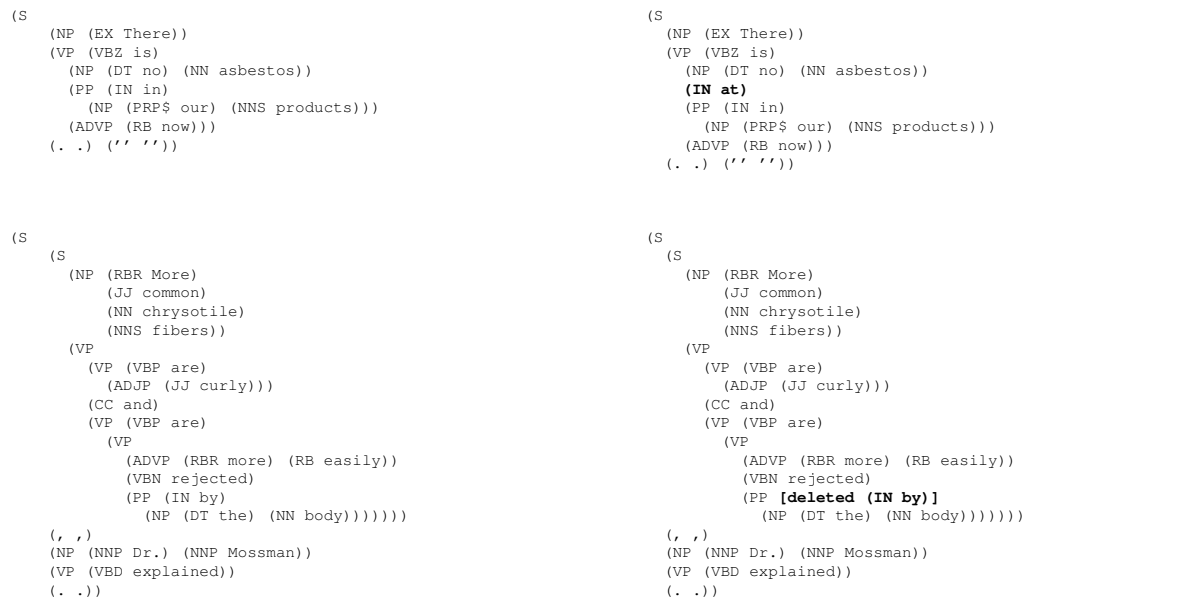

Figure 1: Grammatical (left) and ungrammatical (right) versions of sentences, illustrating insertion errors (top) and deletion errors (bottom)

then evaluated against the three sets of WSJ (i.e. *PureWSJ*, *NoisyWSJ*, and *NoisierWSJ*) using the labelled f-score measure.

The last two experiments can be viewed as the use of an integrated parser, in which each test sentence is parsed by two types of parser — one trained exclusively on grammatical data (i.e. *PureParser*) and the other trained on some ungrammatical data (i.e. either *NoisyParser* or *MixedParser*). The best parse is selected by choosing the one with the higher parse probability. Hence, *PureParser* is integrated with *NoisyParser* for the fourth experiment and with *MixedParser* for the last experiment. (It should be noted that all trainings are performed on Section 2 to Section 21 while all testings are on Section 0.)

3.3 Sentence Classification

This second stage is the core of the present study where we experiment with production rules as features for sentence grammaticality classification. Apart from the parse probabilities returned together with the parse trees, we extract the individual production rules (from either the gold standard or the parse trees) and their corresponding rule probabilities (from parse trees) as classification features. The use of the gold standard is a kind of oracle, to assess the impact of parser inaccuracies. An example with a grammatical-ungrammatical pair is given in Figure 2. We explore various feature selection metrics to obtain a set of production rules for classifying grammatical and ungrammatical sentences.

Parse probability features For the feature class of parse probabilities, we perform similar procedures as in the last two experiments in the first stage. As before, each sentence (be it for training or testing) is parsed with two types of parser — *PureParser* and either *NoisyParser* or *MixedParser*. The parse probability returned by each parser type is used as a classification feature. Therefore, there are only two feature values for this feature class — the parse probability from *PureParser* and the parse probability from either *NoisyParser* or *MixedParser*. A classifier consisting only of these two features is our baseline.

Production rule features We first parse the sentences (for both training and testing) by using the best performing parser induced from the five experiments in the first stage. Production rules are then extracted automatically from both the gold standard and the parser outputs. Various feature selection metrics are used to select a set of discriminative parse rules as classification features.[1] The metrics we use are as follows (with r representing a production rule and c a class, i.e. gram-

[1]There were approximately 26K unique production rules drawn from the training data that could possibly be used as classification features. However, our machine learner described below could not handle this large set of features; but in any case, further experiments showed that a larger feature set resulted in a monotonically lower accuracy.

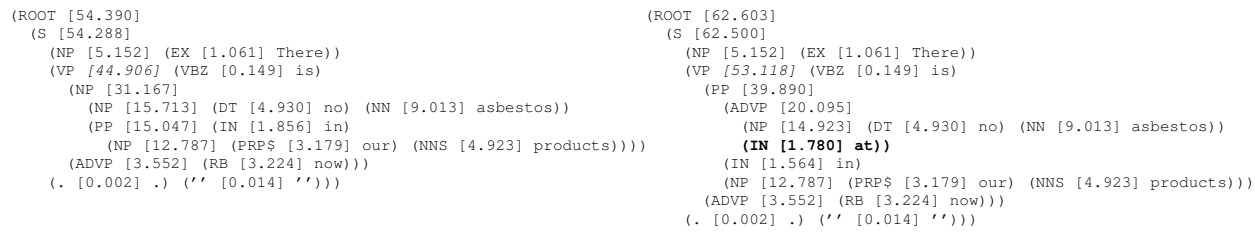

Figure 2: Parser outputs for original (left) and insertion-error (right) variants, annotated with log probabilities for each production rule

matical or ungrammatical):

- *Frequency* (FREQ): We take the n most frequently occurring parse rules within the grammatical corpus and the ungrammatical corpus, where $n \in \{50, 100, 2500\}$. Feature values are the relative frequency of each parse rule within a sentence and also the binary value of their presence or absence.

- *Ratio* (RATIO): We take the ratio of the number of occurrences of a parse rule in the grammatical corpus to the number of occurrences of that rule in the ungrammatical corpus. We pick the 50 parse rules with the highest ratio and another 50 parse rules with the lowest ratio as features. Feature values are of binary type.

- *Mutual information* (MI): We calculate the mutual information between a parse rule and each class (i.e grammatical and ungrammatical). The 100 parse rules with the highest mutual information are selected as features with binary-typed values. We adopt the formula from Yang and Pedersen (1997):

$$MI(r, c) = \log \frac{\Pr(r \wedge c)}{\Pr(r)\Pr(c)} \quad (1)$$

- *Information gain, version 1* (IG-FREQ): We pick the 100 and 500 rules with the highest information gain as features. The formula is again adopted from Yang and Pederson (1997), with $m = 2$.

$$\begin{aligned} IG(r) = & -\sum_{i=1}^{m} \Pr(c_i) \log \Pr(c_i) \\ & + \Pr(r) \sum_{i=1}^{m} \Pr(c_i|r) \log \Pr(c_i|r) \\ & + \Pr(\bar{r}) \sum_{i=1}^{m} \Pr(c_i|\bar{r}) \log \Pr(c_i|\bar{r}) \end{aligned} \quad (2)$$

- *Information gain, version 2* (IG-PROB): In addition, we attempt a different way to calculate the information gain of a parse rule, where the probability of each parse rule $\Pr(r)$ is estimated based on its rule probabilities extracted from the parse trees instead of its occurrence in the corpora. Hence, $\Pr(r)$ is the sum of all the parse probabilities of a parse rule divided by the sum of the parse probability of all the parse trees. All feature values are of binary type. The intuition is that it might not be particular production rules that are characteristic of grammaticality, but their probability: for example, ungrammatical parses might have more unlikely rules. As an illustration, the grammatical tree in Figure 2 (left) has log prob 44.906 at the highest VP node, while the ungrammatical tree (right) has log prob 53.118; notwithstanding the contribution of 1.780 from the insertion of the lexical item *at*, there are some unlikely production rules in this subtree of the ungrammatical tree.

- *Bi-normal separation* (BNS): Forman (2003) suggested that this feature selection metric can be competitive with information gain. The metric is defined as below, where *F(x)* = cumulative probability function of a normal distribution:

$$BNS(r, c) = \left| F^{-1}(\Pr(r|c)) - F^{-1}(\Pr(r|\bar{c})) \right| \quad (3)$$

Similarly, the 100 and 500 rules with the highest BNS scores are selected as classification features with binary-typed values.

Besides investigating these five feature selection methods individually, we also explore the effects of their combinations as well as the combination with parse probabilities.

Training set The training set is a balanced set of grammatical and ungrammatical sentences. As mentioned in Section 3.1, the grammatical sentences are adopted from the *PureWSJ*, while the ungrammatical sentences are from the *NoisyWSJ*; both are based on Section 2 to Section 21. There are 79664 sentences in total for training.

Testing set The testing set is also a balanced set of grammatical and ungrammatical sentences. However, we have two sets of testing data. The first set is formed from *PureWSJ* and *NoisyWSJ*,

Exp	Parser	PureWSJ	NoisyWSJ	NoisierWSJ
1	PureParser	85.61	78.42	72.64
2	NoisyParser	84.31	80.32	76.19
3	MixedParser	82.63	78.69	74.25
4	Pure-NoisyParser	85.39	80.43	76.40
5	Pure-MixedParser	85.49	80.04	75.53

Table 1: Parsing results (labelled f-score %) of five experiments on three versions of WSJ Section 0

and the second set is from *PureWSJ* and *NoisierWSJ*; all are based on Section 0. The latter set is used to testify whether the degree of noisiness in the data would have any effects on the classification performance. There are 3840 sentences in total for testing.

Classifiers A support vector machine (SVM) is used for all the classification tasks. We use the online SVM tool *LIBSVM* (Version 2.89), implemented by Chang and Lin (2001). All the classifications are first conducted under the default settings where the radial basic function (RBF) kernel is used. The kernel is further tuned to find the best pair of parameters (C, γ) for an optimal classification model. [2] In addition to SVM, another machine learner — logistic regression — is also examined to study its effects on classification. Here, we use the logistic regression classifier with ridge regularization from WEKA (Version 3.6.1) (Witten and Frank, 2005).

4 Results

4.1 Parsers

In Table 1, we present the parsing results of the five experiments conducted in the first stage where the intention is to induce a more robust parser that can handle ungrammatical sentences without compromising its performance on grammatical ones.

The integrated parser in Experiment 4 — *Pure Parser* integrated with *Noisy Parser* — is able to attain a relatively good parsing performance for ungrammatical data while at the same time maintaining its performance for grammatical data. This parser is therefore the one that was used for all the parsing tasks in the second stage.

[2] As there is no significant difference between the classification results prior to and after tuning, we only report the prior ones. In addition, no other kernels demonstrated better results than the RBF, so we omit these.

Feature	PureWSJ-NoisyWSJ	PureWSJ-NoisierWSJ
Parse Prob	65.42	74.19

Table 2: SVM results (accuracy %) with parse probabilities as features on both NoisyWSJ and NoisierWSJ

Feature (Metrics)	Gold Standard	Parser Output
FREQ	64.35	53.28
RATIO	50.08	50.0
MI	50.0	n/a
IG-FREQ	67.65**	60.67
IG-PROB	n/a	54.22
BNS	63.75	57.58

Table 3: SVM results (%) with parse rules as features on NoisyWSJ — based on top 100 rules from both gold standard and parser outputs

4.2 Classification

4.2.1 Parse Probabilities

For classification, by using just parse probabilities alone as features, we can see that a reasonably good accuracy is achievable (see Table 2). As expected, for more noisy data, their ability to distingush grammatical sentences from ungrammatical sentences is even more prominent — comparing the classification accuracy of 65.42% (*NoisyWSJ*) with 74.19% (*NoisierWSJ*). This classifier is our baseline for the rest of the sentence grammaticality classifications utilising production rules.

4.2.2 Production Rules

As mentioned in Section 3.3, we first examined the production rules extracted from both the gold standard and the parser outputs with five different feature selection metrics. The classification accuracies achieved by using the top 100 rules for the testing of the less noisy ungrammatical data — *NoisyWSJ* — are shown in Table 3.

It appears that standard information gain (IG-FREQ) outperforms the rest of the selection metrics and it is the only one that performs better than parse probabilities if the gold standard parse trees were available (with this result being statistically significant with 95% confidence). [3] It is, however, worth noting that information gain which utilises rule probabilities (IG-PROB) does not turn out to be a better discriminant as compared to information gain (IG-FREQ). Bi-normal separation and frequency are the next potential candidates; but the former is a better choice in the absence of the gold standard. Ratio and mutual information perform no better than chance.

[3] All significance tests are based on the McNemar's test.

Feature (Metrics)	n = 200	n = 500	n = 2500
FREQ	54.84	n/a	54.04
MI	n/a	50.0	n/a
IG-FREQ	n/a	58.72	n/a
BNS	n/a	61.93	n/a

Table 4: SVM results (%) with parse rules as features on NoisyWSJ — based on larger numbers of rules from gold standard

Feature (Metrics)	Gold Standard	Parser Output
IG-FREQ	75.65*	63.44
BNS	71.2	61.51

Table 5: SVM results (%) with parse rules as features on NoisierWSJ — based on top 100 rules from both gold standard and parser outputs

Table 4 presents results showing the impact of using more production rules as selected by the various metrics; and the results were all poorer than using just 100 rules. In view of this poorer result, we subsequently made use of only the top 100 rules for all the subsequent classifications.

Next, we performed testing on more noisy data — *NoisierWSJ* — to see whether the degree of noisiness in data would have any effects on the classification. Not surprisingly, the more noisy data appears to be easier to be distingushed from the grammatically well-formed data (see Table 5). Similarly, standard information gain (IG-FREQ) may perform better than parse probabilities if the gold standard were available (although this is only marginally statistically significant at 90% confidence.) We only examined two metrics here — IG-FREQ and BNS — as these are the two most competitive ones.

4.2.3 Combinations of Features

From the tables above, it is observed that using production rules by itself for sentence grammaticality classification is generally not better than using parse probabilities alone. We therefore attempted to combine the various metrics for parse rules as well as with the parse probabilities. Again, we use only IG-FREQ and BNS.

Table 6 shows that combining various metrics for production rules does not lead to any significant improvement in classification accuracy

Features	IG-FREQ+BNS	IG-FREQ+FREQ	BNS+FREQ
NoisyWSJ	64.76	66.38	63.98

Table 6: SVM results (%) with the combinations of metrics as features on NoisyWSJ — based on top 100 rules from gold standard

Features	NoisyWSJ	NoisierWSJ
IG-FREQ (gold standard) + Parse probabilities	66.59***	77.58***
IG-FREQ (parser output) + Parse probabilities	65.6	75.31***
BNS (gold standard) + Parse probabilities	66.85***	77.66***
BNS (parser output) + Parse probabilities	66.02*	75.6***

Table 7: SVM results (%) with the combinations of parse rules (IG-FREQ and BNS) and parse probabilities as features

Feature (Metrics)	NoisyWSJ	NoisierWSJ
IG-FREQ (gold standard)	67.65	75.65
IG-FREQ (parser output)	60.83	63.41
BNS (gold standard)	63.83	71.28
BNS (parser output)	57.97	62.79

Table 8: Logistic regression results (%) with parse rules as features — based on top 100 rules from both gold standard and parser outputs

(i.e. their combinations still do not perform better than using parse probabilities alone). However, combining parse rules with parse probabilities as shown in Table 7 does demonstrate some modest improvement of 1.6% on average in the overall classification accuracy. With either gold standard or parser-derived production rules, combinations on more noisy data (*NoisierWSJ*) are statistically better than just using parse probabilities alone (all marked with *** are significant at 99% confidence level). This is also true on the less noisy data (*NoisyWSJ*), but only for gold standard production rules.

4.2.4 Effects of Classifiers

As mentioned in Section 3.3, we also examined the effects of using a different classifier — logistic regression. It appears that logistic regression performs on par with SVM as seen in some of the results for logistic regression presented in Table 8.

5 Discussion

Classification accuracy The overall classification accuracies are broadly in line with the published literature (approximately 65% to 80%), although direct comparisons are not possible because of the use of different data sets. Our classification accuracy may have been affected by the choice of parser. Our parser (Stanford) turns out to perform at a somewhat lower level compared to the one used in Foster et al. (2008) (Charniak and Johnson): on the original (grammatical) WSJ, the f-scores are around 85% vs 90%, while there is

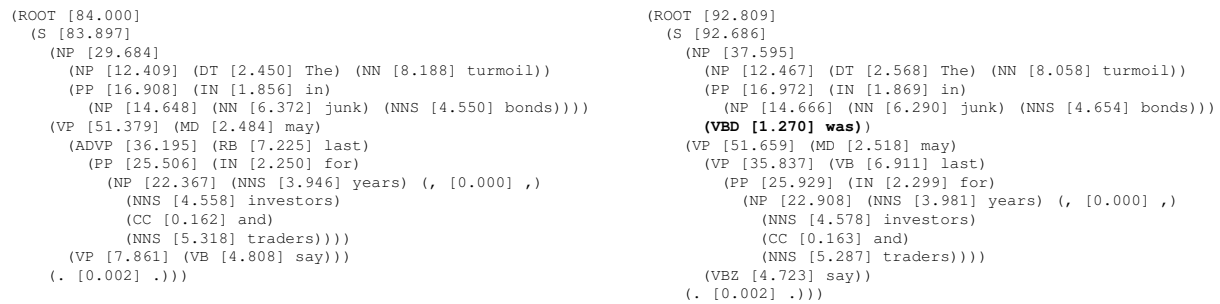

Figure 3: Example of rule selected by IG-FREQ

Feature (Metrics)	Prod Rule	Gram	Ungram
Information Gain (IG-FREQ)	NP → DT DT JJ NN	2	225
	VP → TO TO VP	0	89
	PP → IN IN S	0	73
	PP → NN IN NP	0	70
	NP → NP PP VBD	0	54
Bi-normal Separation (BNS)	PP → IN IN NP	105	1858
	NP → NP IN PP	6	275
	VP → VBZ VBZ NP	0	157
	NP → DT DT NN	2	531
	S → NP VBD VP .	0	242
Ratio	NP → NP , NP , VBD	0	48
	NP → VBP DT JJ CD	0	15
	PP → CD IN NP	0	9
	VP → VB VP PRP	0	9
	S → CC CC NP VP	0	48

Table 9: Examples of parse rules chosen by various metrics (IG-FREQ, BNS, and RATIO)

a slightly bigger difference on the noisy data set, with f-scores of 78–80% vs 85–90%.

Analysis of features We admit to some surprise that looking in detail at production rules did not perform better in general. We examined some of the chosen features under each metric, and these do appear to be strongly characteristic of ungrammatical parses; in particular, there are several instances where probabilities used in IG-PROB appear in our inspection to differ quite noticeably between grammatical and ungrammatical alternatives. We present the top 5 for each of IG-FREQ, BNS and RATIO in Table 9, along with the number of counts in the grammatical versus ungrammatical training corpora. Figure 3 shows an example of one of these rules in a corpus instance.

The problem may be due to feature vector sparsity; looking at other types of cross-sections of parse trees, not only horizontal production rules, (as is done in the parse reranking approach of Charniak and Johnson (2005)), may help with this.

Substitution rules Inspecting the features above, it appears to be the case that substitution cases are hard to detect because the parser is too robust. The way that the Stanford parser handles cases of substitution, even where there is a significant change of part of speech (e.g. *if* for *is*, an example generated in the ungrammatical corpus), results in a parse that is identical to the original grammatical one: the parser is not troubled at all by the ungrammaticality. Supplementing production rules and parser probabilities by n-grams is likely to improve this.

Feature selection metrics It was not entirely surprising that mutual information performed poorly: it tends to select rare instances (Manning and Schutze, 1999) and often does poorly in classification tasks (Forman, 2003). Also as per Forman (2003), IG and BNS performed well. Interestingly, IG perform better in every case with rules alone, while BNS performed better in every combination of rules with parse probabilities, which was overall better than rules alone.

6 Conclusion

The present study has confirmed that parse probabilities are good discriminators for judging the grammaticality of sentences. The idea of exploiting details of the parses in the form of production rules, combined with the parse probabilities, leads to some modest improvement to the overall classification performance.

There are a number of ways in which we might develop further. One would be to use a wider range of features, as in the parser reranking approach noted in Section 5, to avoid sparsity problems. An alternative would be to adopt the noisy channel model: in an alternative to Brockett et al. (2006), ungrammatical trees would be considered noisy versions of their grammatical counterparts. Applying the approach to real ESL data might have different results, with the kinds of errors being less constrained and hence perhaps leading to more significant, and detectable, parse tree changes.

Acknowledgments

The authors would like to acknowledge the support of ARC Linkage Grant LP0776267. Much gratitude is due to Jennifer Foster for the erroneous version of WSJ.

References

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180, Ann Arbor, Michigan.

Markus Dickinson. 2010. Generating learner-like morphological errors in Russian. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, pages 259–267, Beijing, China.

George Forman. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305.

Jennifer Foster and Oistein Andersen. 2009. GenERRate: Generating errors for use in grammatical error detection. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 82–90, Boulder, Colorado.

Jennifer Foster, Joachim Wagner, and Josef van Genabith. 2008. Adapting a WSJ-trained parser to grammatically noisy text. In *Proceedings of ACL-08: HLT, Short Papers*, pages 221–224, Columbus, Ohio.

Jennifer Foster. 2005. *Good Reasons for Noting Bad Grammar: Empirical Investigations into the Parsing of Ungrammatical Written English*. Ph.D. thesis, Department of Computer Science, Trinity College, University of Dublin.

Jennifer Foster. 2007. Treebanks gone bad: Parser evaluation and retraining using a treebank of ungrammatical sentences. *International Journal of Document Analysis and Recognition*, 10(3–4):129–145.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level MT evaluation without reference translations: Beyond language modeling. In *European Association for Machine Translation (EAMT'05)*, pages 103–111, Budapest, Hungary.

Michael Gamon, Jianfeng Gao, Chris Brockett, and Re Klementiev. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP'08)*, pages 449–456, Hyderabad, India.

Na-Rae Han, Joel Tetreault, Soo-Hwa Lee, and Jin-Young Ha. 2010. Using an error-annotated learner corpus to develop an ESL/EFL error correction system. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*, pages 763–770, Valletta, Malta.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan.

Christopher Manning and Hinrich Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Andrew Mutton, Mark Dras, Stephen Wan, and Robert Dale. 2007. GLEU: Automatic evaluation of sentence-level fluency. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 344–351, Prague, Czech Republic.

Daisuke Okanohara and Jun'ichi Tsujii. 2007. A discriminative language model with pseudo-negative samples. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 73–80, Prague, Czech Republic.

Guihua Sun, Xiaohua Liu, Gao Cong, Ming Zhou, Zhongyang Xiong, John Lee, and Chin-Yew Lin. 2007. Detecting erroneous sentences using automatically mined sequential patterns. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 81–88, Prague, Czech Republic.

Joachim Wagner, Jennifer Foster, and Josef van Genabith. 2009. Judging grammaticality: Experiments in sentence classification. *CALICO Journal*, 26(3):474–490.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2 edition.

Yiming Yang and Jan O. Pedersen. 1997. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 412–420.