# Predicting Grammaticality on an Ordinal Scale

**Michael Heilman** **Aoife Cahill** **Nitin Madnani** **Melissa Lopez** **Matthew Mulholland**
Educational Testing Service
Princeton, NJ, USA
{mheilman,acahill,nmadnani,mlopez002,mmulholland}@ets.org

**Joel Tetreault**
Yahoo! Research
New York, NY, USA
tetreaul@yahoo-inc.com

## Abstract

Automated methods for identifying whether sentences are grammatical have various potential applications (e.g., machine translation, automated essay scoring, computer-assisted language learning). In this work, we construct a statistical model of grammaticality using various linguistic features (e.g., misspelling counts, parser outputs, $n$-gram language model scores). We also present a new publicly available dataset of learner sentences judged for grammaticality on an ordinal scale. In evaluations, we compare our system to the one from Post (2011) and find that our approach yields state-of-the-art performance.

## 1 Introduction

In this paper, we develop a system for the task of predicting the grammaticality of sentences, and present a dataset of learner sentences rated for grammaticality. Such a system could be used, for example, to check or to rank outputs from systems for text summarization, natural language generation, or machine translation. It could also be used in educational applications such as essay scoring.

Much of the previous research on predicting grammaticality has focused on identifying (and possibly correcting) specific types of grammatical errors that are typically made by English language learners, such as prepositions (Tetreault and Chodorow, 2008), articles (Han et al., 2006), and collocations (Dahlmeier and Ng, 2011). While some applications (e.g., grammar checking) rely on such fine-grained predictions, others might be better addressed by sentence-level grammaticality judgments (e.g., machine translation evaluation).

Regarding sentence-level grammaticality, there has been much work on rating the grammaticality of machine translation outputs (Gamon et al., 2005; Parton et al., 2011), such as the MT Quality Estimation Shared Tasks (Bojar et al., 2013, §6), but relatively little on evaluating the grammaticality of naturally occurring text. Also, most other research on evaluating grammaticality involves *artificial* tasks or datasets (Sun et al., 2007; Lee et al., 2007; Wong and Dras, 2010; Post, 2011).

Here, we make the following contributions.

- We develop a state-of-the-art approach for predicting the grammaticality of sentences on an ordinal scale, adapting various techniques from the previous work described above.
- We create a dataset of grammatical and ungrammatical sentences written by English language learners, labeled on an ordinal scale for grammaticality. With this unique data set, which we will release to the research community, it is now possible to conduct realistic evaluations for predicting sentence-level grammaticality.

## 2 Dataset Description

We created a dataset consisting of 3,129 sentences randomly selected from essays written by non-native speakers of English as part of a test of English language proficiency. We oversampled lower-scoring essays to increase the chances of finding ungrammatical sentences. Two of the authors of this paper, both native speakers of English with linguistic training, annotated the data. We refer to these annotators as expert judges. When making judgments of the sentences, they saw the previous sentence from the same essay as context. These two authors were not directly involved in development of the system in §3.

Each sentence was annotated on a scale from 1 to 4 as described below, with 4 being the most



*Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 174–180,
Baltimore, Maryland, USA, June 23-25 2014. 

grammatical. We use an ordinal rather than binary scale, following previous work such as that of Clark et al. (2013) and Crocker and Keller (2005) who argue that the distinction between grammatical and ungrammatical is not simply binary. Also, for practical applications, we believe that it is useful to distinguish sentences with minor errors from those with major errors that may disrupt communication. Our annotation scheme was influenced by a translation rating scheme by Coughlin (2003).

Every sentence judged on the 1–4 scale must be a clause. There is an extra category ("Other") for sentences that do not fit this criterion. We exclude instances of "Other" in our experiments (see §4).

**4. Perfect** The sentence is native-sounding. It has no grammatical errors, but may contain very minor typographical and/or collocation errors, as in Example (1).

(1) For instance, i stayed in a dorm when i went to collge.

**3. Comprehensible** The sentence may contain one or more minor grammatical errors, including subject-verb agreement, determiner, and minor preposition errors that do not make the meaning unclear, as in Example (2).

(2) We know during Spring Festival, Chinese family will have a abundand family banquet with family memebers.

"Chinese family", which could be corrected to "Chinese families", "each Chinese family", etc., would be an example of a minor grammatical error involving determiners.

**2. Somewhat Comprehensible** The sentence may contain one or more serious grammatical errors, including missing subject, verb, object, etc., verb tense errors, and serious preposition errors. Due to these errors, the sentence may have multiple plausible interpretations, as in Example (3).

(3) I can gain the transportations such as buses and trains.

**1. Incomprehensible** The sentence contains so many errors that it would be difficult to correct, as in Example (4).

(4) Or you want to say he is only a little boy do not everything clearly?

The phrase "do not everything" makes the sentence practically incomprehensible since the subject of "do" is not clear.

**O. Other/Incomplete** This sentence is incomplete. These sentences, such as Example (5), appear in our corpus due to the nature of timed tests.

(5) The police officer handed the

This sentence is cut off and does not at least include one clause.

We measured interannotator agreement on a subset of 442 sentences that were independently annotated by both expert annotators. Exact agreement was 71.3%, unweighted $\kappa = 0.574$, and Pearson's $r = 0.759$.[1] For our experiments, one expert annotator was arbitrarily selected, and for the doubly-annotated sentences, only the judgments from that annotator were retained.

The labels from the expert annotators are distributed as follows: 72 sentences are labeled 1; 538 are 2; 1,431 are 3; 978 are 4; and 110 are "O".

We also gathered 5 additional judgments using Crowdflower.[2] For this, we excluded the "Other" category and any sentences that had been marked as such by the expert annotators. We used 100 (3.2%) of the judged sentences as "gold" data in Crowdflower to block contributors who were not following the annotation guidelines. For those sentences, only disagreements within 1 point of the expert annotator judgment were accepted. In preliminary experiments, averaging the six judgments (1 expert, 5 crowdsourced) for each item led to higher human-machine agreement. For all experiments reported later, we used this average of six judgments as our gold standard.

For our experiments (§4), we randomly split the data into training (50%), development (25%), and testing (25%) sets. We also excluded all instances labeled "Other". These are relatively uncommon and less interesting to this study. Also, we believe that simpler, heuristic approaches could be used to identify such sentences.

We use "GUG" ("Grammatical" versus "UnGrammatical") to refer to this dataset. The dataset is available for research at `https://github.com/EducationalTestingService/gug-data`.

[1] The reported agreement values assume that "Other" maps to 0. For the sentences where both labels were in the 1–4 range ($n = 424$), Pearson's $r = 0.767$.

[2] `http://www.crowdflower.com`

## 3 System Description

This section describes the statistical model (§3.1) and features (§3.2) used by our system.

### 3.1 Statistical Model

We use $\ell_2$-regularized linear regression (i.e., ridge regression) to learn a model of sentence grammaticality from a variety of linguistic features.[3][4]

To tune the $\ell_2$-regularization hyperparameter $\alpha$, the system performs 5-fold cross-validation on the data used for training. The system evaluates $\alpha \in 10^{\{-4,\ldots,4\}}$ and selects the one that achieves the highest cross-validation correlation $r$.

### 3.2 Features

Next, we describe the four types of features.

#### 3.2.1 Spelling Features

Given a sentence with with $n$ word tokens, the model filters out tokens containing nonalphabetic characters and then computes the number of misspelled words $n_{miss}$ (later referred to as `num_misspelled`), the proportion of misspelled words $\frac{n_{miss}}{n}$, and $\log(n_{miss}+1)$ as features. To identify misspellings, we use a freely available spelling dictionary for U.S. English.[5]

#### 3.2.2 $n$-gram Count and Language Model Features

Given each sentence, the model obtains the counts of $n$-grams ($n = 1 \ldots 3$) from English Gigaword and computes the following features:[6]

- $\sum_{s \in S_n} \frac{\log(\text{count}(s)+1)}{\|S_n\|}$
- $\max_{s \in S_n} \log(\text{count}(s)+1)$
- $\min_{s \in S_n} \log(\text{count}(s)+1)$

where $S_n$ represents the $n$-grams of order $n$ from the given sentence. The model computes the following features from a 5-gram language model trained on the same three sections of English Gigaword using the SRILM toolkit (Stolcke, 2002):

- the average log-probability of the given sentence (referred to as `gigaword_avglogprob` later)
- the number of out-of-vocabulary words in the sentence

Finally, the system computes the average log-probability and number of out-of-vocabulary words from a language model trained on a collection of essays written by non-native English speakers[7] ("non-native LM").

#### 3.2.3 Precision Grammar Features

Following Wagner et al. (2007) and Wagner et al. (2009), we use features extracted from precision grammar parsers. These grammars have been hand-crafted and designed to only provide complete syntactic analyses for grammatically correct sentences. This is in contrast to treebank-trained grammars, which will generally provide *some* analysis regardless of grammaticality. Here, we use (1) the Link Grammar Parser[8] and (2) the HPSG English Resource Grammar (Copestake and Flickinger, 2000) and PET parser.[9]

We use a binary feature, `complete_link`, from the Link grammar that indicates whether at least one complete linkage can be found for a sentence. We also extract several features from the HPSG analyses.[10] They mostly reflect information about unification success or failure and the associated costs. In each instance, we use the logarithm of one plus the frequency.

[3] We use ridge regression from the `scikit-learn` toolkit (Pedregosa et al., 2011) v0.23.1 and the SciKit-Learn Laboratory (`http://github.com/EducationalTestingService/skll`).

[4] Regression models typically produce conservative predictions with lower variance than the original training data. So that predictions better match the distribution of labels in the training data, the system rescales its predictions. It saves the mean and standard deviation of the training data gold standard ($M_{gold}$ and $SD_{gold}$, respectively) and of its own predictions on the training data ($M_{pred}$ and $SD_{pred}$, respectively). During cross-validation, this is done for each fold. From an initial prediction $\hat{y}$, it produces the final prediction: $\hat{y}' = \frac{\hat{y}-M_{pred}}{SD_{pred}} * SD_{gold} + M_{gold}$. This transformation does not affect Pearson's $r$ correlations or rankings, but it would affect binarized predictions.

[5] `http://pythonhosted.org/pyenchant/`

[6] We use the New York Times (nyt), the Los Angeles Times-Washington Post (ltw), and the Washington Post-Bloomberg News (wpb) sections from the fifth edition of English Gigaword (LDC2011T07).

[7] This did not overlap with the data described in §2 and was a subset of the data released by Blanchard et al. (2013).

[8] `http://www.link.cs.cmu.edu/link/`

[9] `http://moin.delph-in.net/PetTop`

[10] The complete list of relevant statistics used as features is: `trees`, `unify_cost_succ`, `unify_cost_fail`, `unifications_succ`, `unifications_fail`, `subsumptions_succ`, `subsumptions_fail`, `words`, `words_pruned`, `aedges`, `pedges`, `upedges`, `raedges`, `rpedges`, `medges`. During development, we observed that some of these features vary for some inputs, probably due to parsing search timeouts. On 10 preliminary runs with the development set, this variance had minimal effects on correlations with human judgments (less than 0.00001 in terms of $r$).

| | $r$ |
|---|---|
| our system | 0.668 |
| – non-native LM (§3.2.2) | 0.665 |
| – HPSG parse (§3.2.3) | 0.664 |
| – PCFG parse (§3.2.4) | 0.662 |
| – spelling (§3.2.1) | 0.643 |
| – gigaword LM (§3.2.2) | 0.638 |
| – link parse (§3.2.3) | 0.632 |
| – gigaword count (§3.2.2) | 0.630 |

Table 1: Pearson's $r$ on the development set, for our full system and variations excluding each feature type. "$-X$" indicates the full model without the "$X$" features.

#### 3.2.4 PCFG Parsing Features

We find phrase structure trees and basic dependencies with the Stanford Parser's English PCFG model (Klein and Manning, 2003; de Marneffe et al., 2006).[11] We then compute the following:

- the parse score as provided by the Stanford PCFG Parser, normalized for sentence length, later referred to as `parse_prob`
- a binary feature that captures whether the top node of the tree is sentential or not (i.e. the assumption is that if the top node is non-sentential, then the sentence is a fragment)
- features binning the number of `dep` relations returned by the dependency conversion. These `dep` relations are underspecified for function and indicate that the parser was unable to find a standard relation such as `subj`, possibly indicating a grammatical error.

## 4 Experiments

Next, we present evaluations on the GUG dataset.

### 4.1 Feature Ablation

We conducted a feature ablation study to identify the contributions of the different types of features described in §3.2. We compared the performance of the full model with all of the features to models with all but one type of feature. For this experiment, all models were estimated from the training set and evaluated on the development set. We report performance in terms of Pearson's $r$ between the averaged 1–4 human labels and unrounded system predictions.

The results are shown in Table 1. From these results, the most useful features appear to be the $n$-gram frequencies from Gigaword and whether the link parser can fully parse the sentence.

### 4.2 Test Set Results

In this section, we present results on the held-out test set for the full model and various baselines, summarized in Table 2. For test set evaluations, we trained on the combination of the training and development sets (§2), to maximize the amount of training data for the final experiments.

We also trained and evaluated on binarized versions of the ordinal GUG labels: a sentence was labeled 1 if the average judgment was at least 3.5 (i.e., would round to 4), and 0 otherwise. Evaluating on a binary scale allows us to measure how well the system distinguishes grammatical sentences from ungrammatical ones. For some applications, this two-way distinction may be more relevant than the more fine-grained 1–4 scale. To train our system on binarized data, we replaced the $\ell_2$-regularized linear regression model with an $\ell_2$-regularized logistic regression and used Kendall's $\tau$ rank correlation between the predicted probabilities of the positive class and the binary gold standard labels as the grid search metric (§3.1) instead of Pearson's $r$.

For the ordinal task, we report Pearson's $r$ between the averaged human judgments and each system. For the binary task, we report percentage accuracy. Since the predictions from the binary and ordinal systems are on different scales, we include the nonparametric statistic Kendall's $\tau$ as a secondary evaluation metric for both tasks.

We also evaluated the binary system for the ordinal task by computing correlations between its estimated probabilities and the averaged human scores, and we evaluated the ordinal system for the binary task by binarizing its predictions.[12]

We compare our work to a modified version of the publicly available[13] system from Post (2011), which performed very well on an artificial dataset. To our knowledge, it is the only publicly available system for grammaticality prediction. It is very

[11] We use the Nov. 12, 2013 version of the Stanford Parser.

[12] We selected a threshold for binarization from a grid of 1001 points from 1 to 4 that maximized the accuracy of binarized predictions from a model trained on the training set and evaluated on the binarized development set. For evaluating the three single-feature baselines discussed below, we used the same approach except with grid ranging from the minimum development set feature value to the maximum plus 0.1% of the range.

[13] The Post (2011) system is available at `https://github.com/mjpost/post2011judging`.

| | Ordinal Task | | | Binary Task | | |
|---|---|---|---|---|---|---|
| | $r$ | $Sig._r$ | $\tau$ | % Acc. | $Sig._{\%Acc.}$ | $\tau$ |
| our system | **0.644** | | 0.479 | 79.3 | | 0.419 |
| our system$_{logistic}$ | 0.616 | * | **0.484** | **80.7** | | **0.428** |
| Post | 0.321 | * | 0.225 | 75.5 | * | 0.195 |
| Post$_{logistic}$ | 0.259 | * | 0.181 | 74.4 | * | 0.181 |
| `complete_link` | 0.386 | * | 0.335 | 74.8 | * | 0.302 |
| `gigaword_avglogprob` | 0.414 | * | 0.290 | 76.7 | * | 0.280 |
| `num_misspelled` | -0.462 | * | -0.370 | 74.8 | * | -0.335 |

Table 2: Human-machine agreement statistics for our system, the system from Post (2011), and simple baselines, computed from the averages of human ratings in the testing set (§2). "*" in a Sig. column indicates a statistically significant difference from "our system" ($p < .05$, see text for details). A majority baseline for the binary task achieves 74.8% accuracy. The best results for each metric are in bold.

different from our system since it relies on partial tree-substitution grammar derivations as features. We use the feature computation components of that system but replace its statistical model. The system was designed for use with a dataset consisting of 50% grammatical and 50% ungrammatical sentences, rather than data with ordinal or continuous labels. Additionally, its classifier implementation does not output scores or probabilities. Therefore, we used the same learning algorithms as for our system (i.e., ridge regression for the ordinal task and logistic regression for the binary task).[14]

To create further baselines for comparison, we selected the following features that represent ways one might approximate grammaticality if a comprehensive model was unavailable: whether the link parser can fully parse the sentence (`complete_link`), the Gigaword language model score (`gigaword_avglogprob`), and the number of misspelled tokens (`num_misspelled`). Note that we expect the number of misspelled tokens to be negatively correlated with grammaticality. We flipped the sign of the misspelling feature when computing accuracy for the binary task.

To identify whether the differences in performance for the ordinal task between our system and each of the baselines are statistically significant, we used the BC$_a$ Bootstrap (Efron and Tibshirani, 1993) with 10,000 replications to compute 95% confidence intervals for the absolute value of $r$ for our system minus the absolute value of $r$ for each of the alternative methods. For the binary task, we used the sign test to test for significant differences in accuracy. The results are in Table 2.

[14] In preliminary experiments, we observed little difference in performance between logistic regression and the original support vector classifier used by the system from Post (2011).

## 5 Discussion and Conclusions

In this paper, we developed a system for predicting grammaticality on an ordinal scale and created a labeled dataset that we have released publicly (§2) to enable more realistic evaluations in future research. Our system outperformed an existing state-of-the-art system (Post, 2011) in evaluations on binary and ordinal scales. This is the most realistic evaluation of methods for predicting sentence-level grammaticality to date.

Surprisingly, the system from Post (2011) performed quite poorly on the GUG dataset. We speculate that this is due to the fact that the Post system relies heavily on features extracted from automatic syntactic parses. While Post found that such a system can effectively distinguish grammatical news text sentences from sentences generated by a language model, measuring the grammaticality of real sentences from language learners seems to require a wider variety of features, including $n$-gram counts, language model scores, etc. Of course, our findings do not indicate that syntactic features such as those from Post (2011) are without value. In future work, it may be possible to improve grammaticality measurement by integrating such features into a larger system.

## Acknowledgements

We thank Beata Beigman Klebanov, Yoko Futagi, Su-Youn Yoon, and the anonymous reviewers for their helpful comments. We also thank Jennifer Foster for discussions about this work and Matt Post for making his system publicly available.