They Can Help: Using Crowdsourcing to Improve the Evaluation of Grammatical Error Detection Systems

Nitin Madnani[a] **Joel Tetreault**[a] **Martin Chodorow**[b] **Alla Rozovskaya**[c]
[a]Educational Testing Service
Princeton, NJ
{nmadnani,jtetreault}@ets.org
[b]Hunter College of CUNY
martin.chodorow@hunter.cuny.edu
[c]University of Illinois at Urbana-Champaign
rozovska@illinois.edu

Abstract

Despite the rising interest in developing grammatical error detection systems for non-native speakers of English, progress in the field has been hampered by a lack of informative metrics and an inability to directly compare the performance of systems developed by different researchers. In this paper we address these problems by presenting two evaluation methodologies, both based on a novel use of crowdsourcing.

1 Motivation and Contributions

One of the fastest growing areas in need of NLP tools is the field of grammatical error detection for learners of English as a Second Language (ESL). According to Guo and Beckett (2007), "over a billion people speak English as their second or foreign language." This high demand has resulted in many NLP research papers on the topic, a Synthesis Series book (Leacock et al., 2010) and a recurring workshop (Tetreault et al., 2010a), all in the last five years. In this year's ACL conference, there are four long papers devoted to this topic.

Despite the growing interest, two major factors encumber the growth of this subfield. First, the lack of consistent and appropriate score reporting is an issue. Most work reports results in the form of precision and recall as measured against the judgment of a single human rater. This is problematic because most usage errors (such as those in article and preposition usage) are a matter of degree rather than simple rule violations such as number agreement. As a consequence, it is common for two native speakers to have different judgments of usage. Therefore, an appropriate evaluation should take this into account by not only enlisting multiple human judges but also aggregating these judgments in a graded manner. Second, systems are hardly ever compared to each other. In fact, to our knowledge, no two systems developed by different groups have been compared directly within the field primarily because there is no common corpus or shared task—both commonly found in other NLP areas such as machine translation.[1] For example, Tetreault and Chodorow (2008), Gamon et al. (2008) and Felice and Pulman (2008) developed preposition error detection systems, but evaluated on three *different* corpora using *different* evaluation measures.

The goal of this paper is to address the above issues by using crowdsourcing, which has been proven effective for collecting multiple, reliable judgments in other NLP tasks: machine translation (Callison-Burch, 2009; Zaidan and Callison-Burch, 2010), speech recognition (Evanini et al., 2010; Novotney and Callison-Burch, 2010), automated paraphrase generation (Madnani, 2010), anaphora resolution (Chamberlain et al., 2009), word sense disambiguation (Akkaya et al., 2010), lexicon construction for less commonly taught languages (Irvine and Klementiev, 2010), fact mining (Wang and Callison-Burch, 2010) and named entity recognition (Finin et al., 2010) among several others.

In particular, we make a significant contribution to the field by showing how to leverage crowdsourc-

[1] There has been a recent proposal for a related shared task (Dale and Kilgarriff, 2010) that shows promise.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 508–513,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

ing to both address the lack of appropriate evaluation metrics and to make system comparison easier. Our solution is general enough for, in the simplest case, intrinsically evaluating a single system on a single dataset and, more realistically, comparing two different systems (from same or different groups).

2 A Case Study: Extraneous Prepositions

We consider the problem of detecting an *extraneous preposition error*, i.e., incorrectly using a preposition where none is licensed. In the sentence "*They came to outside*", the preposition *to* is an extraneous error whereas in the sentence *"They arrived to the town"* the preposition *to* is a confusion error (cf. *arrived in the town*). Most work on automated correction of preposition errors, with the exception of Gamon (2010), addresses preposition confusion errors e.g., (Felice and Pulman, 2008; Tetreault and Chodorow, 2008; Rozovskaya and Roth, 2010b). One reason is that in addition to the standard context-based features used to detect confusion errors, identifying extraneous prepositions also requires actual knowledge of when a preposition can and cannot be used. Despite this lack of attention, extraneous prepositions account for a significant proportion—as much as 18% in essays by advanced English learners (Rozovskaya and Roth, 2010a)—of all preposition usage errors.

2.1 Data and Systems

For the experiments in this paper, we chose a proprietary corpus of about 500,000 essays written by ESL students for Test of English as a Foreign Language (TOEFL®). Despite being common ESL errors, preposition errors are still infrequent overall, with over 90% of prepositions being used correctly (Leacock et al., 2010; Rozovskaya and Roth, 2010a). Given this fact about error sparsity, we needed an efficient method to extract a good number of error instances (for statistical reliability) from the large essay corpus. We found all trigrams in our essays containing prepositions as the middle word (e.g., *marry with her*) and then looked up the counts of each trigram and the corresponding bigram with the preposition removed (*marry her*) in the Google Web1T 5-gram Corpus. If the trigram was unattested or had a count much lower than expected based on the bigram count, then we manually inspected the trigram to see whether it was actually an error. If it was, we extracted a sentence from the large essay corpus containing this erroneous trigram. Once we had extracted 500 sentences containing extraneous preposition error instances, we added 500 sentences containing correct instances of preposition usage. This yielded a corpus of 1000 sentences with a 50% error rate.

These sentences, with the target preposition highlighted, were presented to 3 expert annotators who are native English speakers. They were asked to annotate the preposition usage instance as one of the following: extraneous (*Error*), not extraneous (*OK*) or too hard to decide (*Unknown*); the last category was needed for cases where the context was too messy to make a decision about the highlighted preposition. On average, the three experts had an agreement of 0.87 and a kappa of 0.75. For subsequent analysis, we only use the classes *Error* and *OK* since *Unknown* was used extremely rarely and never by all 3 experts for the same sentence.

We used two different error detection systems to illustrate our evaluation methodology:[2]

- **LM**: A 4-gram language model trained on the Google Web1T 5-gram Corpus with SRILM (Stolcke, 2002).
- **PERC**: An averaged Perceptron (Freund and Schapire, 1999) classifier— as implemented in the Learning by Java toolkit (Rizzolo and Roth, 2007)—trained on 7 million examples and using the same features employed by Tetreault and Chodorow (2008).

3 Crowdsourcing

Recently,we showed that Amazon Mechanical Turk (AMT) is a cheap and effective alternative to expert raters for annotating preposition errors (Tetreault et al., 2010b). In other current work, we have extended this pilot study to show that CrowdFlower, a crowdsourcing service that allows for stronger quality control on untrained human raters (henceforth, Turkers), is more reliable than AMT on three different error detection tasks (article errors, confused prepositions

[2] Any conclusions drawn in this paper pertain only to these specific instantiations of the two systems.

& extraneous prepositions). To impose such quality control, one has to provide "gold" instances, i.e., examples with known correct judgments that are then used to root out any Turkers with low performance on these instances. For all three tasks, we obtained 20 Turkers' judgments via CrowdFlower for each instance and found that, on average, only 3 Turkers were required to match the experts.

More specifically, for the extraneous preposition error task, we used 75 sentences as gold and obtained judgments for the remaining 923 non-gold sentences.[3] We found that if we used 3 Turker judgments in a majority vote, the agreement with any one of the three expert raters is, on average, 0.87 with a kappa of 0.76. This is on par with the inter-expert agreement and kappa found earlier (0.87 and 0.75 respectively).

The extraneous preposition annotation cost only $325 (923 judgments × 20 Turkers) and was completed in a single day. The only restriction on the Turkers was that they be physically located in the USA. For the analysis in subsequent sections, we use these 923 sentences and the respective 20 judgments obtained via CrowdFlower. The 3 expert judgments are *not* used any further in this analysis.

4 Revamping System Evaluation

In this section, we provide details on how crowdsourcing can help revamp the evaluation of error detection systems: (a) by providing more informative measures for the intrinsic evaluation of a single system (§ 4.1), and (b) by easily enabling system comparison (§ 4.2).

4.1 Crowd-informed Evaluation Measures

When evaluating the performance of grammatical error detection systems against human judgments, the judgments for each instance are generally reduced to the single most frequent category: *Error* or *OK*. This reduction is not an accurate reflection of a complex phenomenon. It discards valuable information about the acceptability of usage because it treats all "bad" uses as equal (and all good ones as equal), when they are not. Arguably, it would be fairer to use a continuous scale, such as the proportion of raters who judge an instance as correct or incorrect. For example, if 90% of raters agree on a rating of *Error* for an instance of preposition usage, then that is stronger evidence that the usage is an error than if 56% of Turkers classified it as *Error* and 44% classified it as *OK* (the sentence "*In addition classmates play <u>with</u> some game and enjoy*" is an example). The regular measures of precision and recall would be fairer if they reflected this reality. Besides fairness, another reason to use a continuous scale is that of stability, particularly with a small number of instances in the evaluation set (quite common in the field). By relying on majority judgments, precision and recall measures tend to be unstable (see below).

We modify the measures of precision and recall to incorporate distributions of correctness, obtained via crowdsourcing, in order to make them fairer and more stable indicators of system performance. Given an error detection system that classifies a sentence containing a specific preposition as *Error* (class 1) if the preposition is extraneous and *OK* (class 0) otherwise, we propose the following weighted versions of hits (H_w), misses (M_w) and false positives (FP_w):

$$\mathrm{H}_w = \sum_{i}^{N} (c^i_{\text{sys}} * p^i_{\text{crowd}}) \tag{1}$$

$$\mathrm{M}_w = \sum_{i}^{N} ((1 - c^i_{\text{sys}}) * p^i_{\text{crowd}}) \tag{2}$$

$$\mathrm{FP}_w = \sum_{i}^{N} (c^i_{\text{sys}} * (1 - p^i_{\text{crowd}})) \tag{3}$$

In the above equations, N is the total number of instances, c^i_{sys} is the class (1 or 0) , and p^i_{crowd} indicates the proportion of the crowd that classified instance i as *Error*. Note that if we were to revert to the majority crowd judgment as the sole judgment for each instance, instead of proportions, p^i_{crowd} would always be either 1 or 0 and the above formulae would simply compute the normal hits, misses and false positives. Given these definitions, weighted precision can be defined as $\mathrm{Precision}_w = \mathrm{H}_w/(\mathrm{H}_w + \mathrm{FP}_w)$ and weighted recall as $\mathrm{Recall}_w = \mathrm{H}_w/(\mathrm{H}_w + \mathrm{M}_w)$.

[3] We found 2 duplicate sentences and removed them.

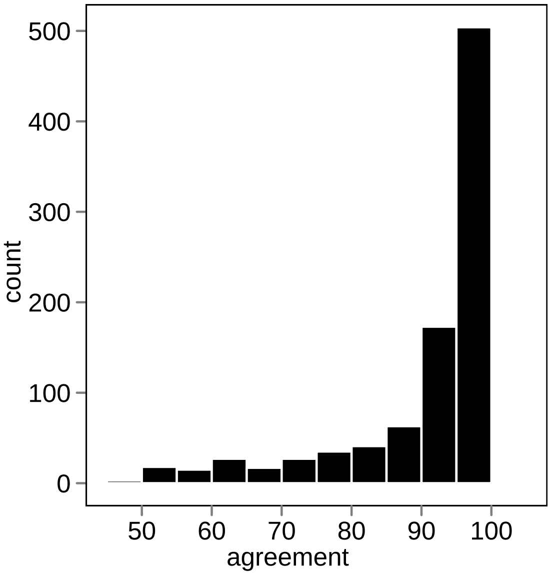

Figure 1: Histogram of Turker agreements for all 923 instances on whether a preposition is extraneous.

	Precision	Recall
Unweighted	0.957	0.384
Weighted	0.900	0.371

Table 1: Comparing commonly used (unweighted) and proposed (weighted) precision/recall measures for LM.

To illustrate the utility of these weighted measures, we evaluated the LM and PERC systems on the dataset containing 923 preposition instances, against all 20 Turker judgments. Figure 1 shows a histogram of the Turker agreement for the majority rating over the set. Table 1 shows both the unweighted (discrete majority judgment) and weighted (continuous Turker proportion) versions of precision and recall for this system.

The numbers clearly show that in the unweighted case, the performance of the system is overestimated simply because the system is getting as much credit for each contentious case (low agreement) as for each clear one (high agreement). In the weighted measure we propose, the contentious cases are weighted lower and therefore their contribution to the overall performance is reduced. This is a fairer representation since the system should not be expected to perform as well on the less reliable instances as it does on the clear-cut instances. Essentially, if humans cannot consistently decide whether a case is an error then a system's output cannot be considered entirely right or entirely wrong.[4]

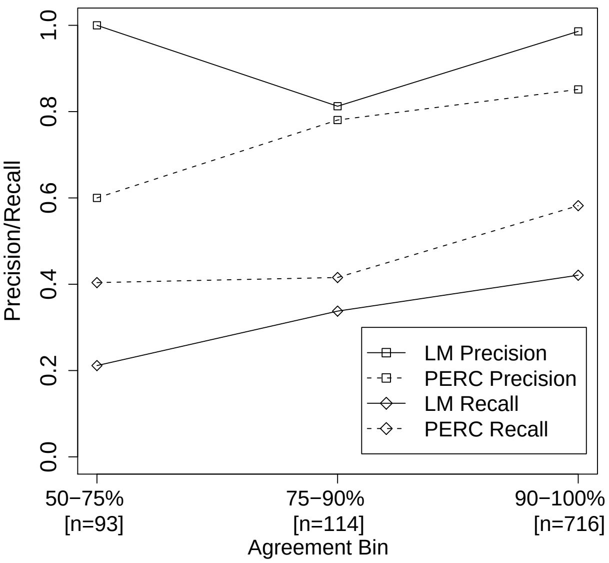

Figure 2: Unweighted precision/recall by agreement bins for LM & PERC.

As an added advantage, the weighted measures are more stable. Consider a contentious instance in a small dataset where 7 out of 15 Turkers (a minority) classified it as *Error*. However, it might easily have happened that 8 Turkers (a majority) classified it as *Error* instead of 7. In that case, the change in unweighted precision would have been much larger than is warranted by such a small change in the data. However, weighted precision is guaranteed to be more stable. Note that the instability decreases as the size of the dataset increases but still remains a problem.

4.2 Enabling System Comparison

In this section, we show how to easily compare different systems both on the same data (in the ideal case of a shared dataset being available) and, more realistically, on different datasets. Figure 2 shows (unweighted) precision and recall of LM and PERC (computed against the majority Turker judgment) for three *agreement bins*, where each bin is defined as containing only the instances with Turker agreement in a specific range. We chose the bins shown

[4] The difference between unweighted and weighted measures can vary depending on the distribution of agreement.

since they are sufficiently large and represent a reasonable stratification of the agreement space. Note that we are *not* weighting the precision and recall in this case since we have already used the agreement proportions to create the bins.

This curve enables us to compare the two systems easily on different levels of item contentiousness and, therefore, conveys much more information than what is usually reported (a single number for unweighted precision/recall over the whole corpus). For example, from this graph, PERC is seen to have similar performance as LM for the 75-90% agreement bin. In addition, even though LM precision is perfect (1.0) for the most contentious instances (the 50-75% bin), this turns out to be an artifact of the LM classifier's decision process. When it must decide between what it views as two equally likely possibilities, it defaults to *OK*. Therefore, even though LM has higher unweighted precision (0.957) than PERC (0.813), it is only really better on the most clear-cut cases (the 90-100% bin). If one were to report unweighted precision and recall without using any bins—as is the norm—this important qualification would have been harder to discover.

While this example uses the same dataset for evaluating two systems, the procedure is general enough to allow two systems to be compared on two *different* datasets by simply examining the two plots. However, two potential issues arise in that case. The first is that the bin sizes will likely vary across the two plots. However, this should not be a significant problem as long as the bins are sufficiently large. A second, more serious, issue is that the error rates (the proportion of instances that are actually erroneous) in each bin may be different across the two plots. To handle this, we recommend that a kappa-agreement plot be used instead of the precision-agreement plot shown here.

5 Conclusions

Our goal is to propose best practices to address the two primary problems in evaluating grammatical error detection systems and we do so by leveraging crowdsourcing. For system development, we recommend that rather than compressing multiple judgments down to the majority, it is better to use agreement proportions to weight precision and recall to yield fairer and more stable indicators of performance.

For system comparison, we argue that the best solution is to use a shared dataset and present the precision-agreement plot using a set of agreed-upon bins (possibly in conjunction with the weighted precision and recall measures) for a more informative comparison. However, we recognize that shared datasets are harder to create in this field (as most of the data is proprietary). Therefore, we also provide a way to compare multiple systems across *different* datasets by using kappa-agreement plots. As for agreement bins, we posit that the agreement values used to define them depend on the task and, therefore, should be determined by the community.

Note that both of these practices can also be implemented by using 20 experts instead of 20 Turkers. However, we show that crowdsourcing yields judgments that are as good but without the cost. To facilitate the adoption of these practices, we make all our evaluation code and data available to the community.[5]

Acknowledgments

We would first like to thank our expert annotators Sarah Ohls and Waverely VanWinkle for their hours of hard work. We would also like to acknowledge Lei Chen, Keelan Evanini, Jennifer Foster, Derrick Higgins and the three anonymous reviewers for their helpful comments and feedback.

References

Cem Akkaya, Alexander Conrad, Janyce Wiebe, and Rada Mihalcea. 2010. Amazon Mechanical Turk for Subjectivity Word Sense Disambiguation. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 195–203.

Chris Callison-Burch. 2009. Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazon's Mechanical Turk. In *Proceedings of EMNLP*, pages 286–295.

Jon Chamberlain, Massimo Poesio, and Udo Kruschwitz. 2009. A Demonstration of Human Computation Using the Phrase Detectives Annotation Game. In *ACM SIGKDD Workshop on Human Computation*, pages 23–24.

[5] `http://bit.ly/crowdgrammar`

Robert Dale and Adam Kilgarriff. 2010. Helping Our Own: Text Massaging for Computational Linguistics as a New Shared Task. In *Proceedings of INLG*.

Keelan Evanini, Derrick Higgins, and Klaus Zechner. 2010. Using Amazon Mechanical Turk for Transcription of Non-Native Speech. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 53–56.

Rachele De Felice and Stephen Pulman. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of COLING*, pages 169–176.

Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. 2010. Annotating Named Entities in Twitter Data with Crowdsourcing. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 80–88.

Yoav Freund and Robert E. Schapire. 1999. Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, 37(3):277–296.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alexander Klementiev, William Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using Contextual Speller Techniques and Language Modeling for ESL Error Correction. In *Proceedings of IJCNLP*.

Michael Gamon. 2010. Using Mostly Native Data to Correct Errors in Learners' Writing. In *Proceedings of NAACL*, pages 163–171.

Y. Guo and Gulbahar Beckett. 2007. The Hegemony of English as a Global Language: Reclaiming Local Knowledge and Culture in China. *Convergence: International Journal of Adult Education*, 1.

Ann Irvine and Alexandre Klementiev. 2010. Using Mechanical Turk to Annotate Lexicons for Less Commonly Used Languages. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 108–113.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Synthesis Lectures on Human Language Technologies. Morgan Claypool.

Nitin Madnani. 2010. *The Circle of Meaning: From Translation to Paraphrasing and Back*. Ph.D. thesis, Department of Computer Science, University of Maryland College Park.

Scott Novotney and Chris Callison-Burch. 2010. Cheap, Fast and Good Enough: Automatic Speech Recognition with Non-Expert Transcription. In *Proceedings of NAACL*, pages 207–215.

Nicholas Rizzolo and Dan Roth. 2007. Modeling Discriminative Global Inference. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC)*, pages 597–604, Irvine, California, September.

Alla Rozovskaya and D. Roth. 2010a. Annotating ESL errors: Challenges and rewards. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

Alla Rozovskaya and D. Roth. 2010b. Generating Confusion Sets for Context-Sensitive Error Correction. In *Proceedings of EMNLP*.

Andreas Stolcke. 2002. SRILM: An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 257–286.

Joel Tetreault and Martin Chodorow. 2008. The Ups and Downs of Preposition Error Detection in ESL Writing. In *Proceedings of COLING*, pages 865–872.

Joel Tetreault, Jill Burstein, and Claudia Leacock, editors. 2010a. *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

Joel Tetreault, Elena Filatova, and Martin Chodorow. 2010b. Rethinking Grammatical Error Annotation and Evaluation with the Amazon Mechanical Turk. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*, pages 45–48.

Rui Wang and Chris Callison-Burch. 2010. Cheap Facts and Counter-Facts. In *Proceedings of the NAACL Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 163–167.

Omar F. Zaidan and Chris Callison-Burch. 2010. Predicting Human-Targeted Translation Edit Rate via Untrained Human Annotators. In *Proceedings of NAACL*, pages 369–372.