Crowdsourcing for Grammatical Error Correction

Ellie Pavlick
University of Pennsylvania
epavlick@seas.upenn.edu

Rui Yan
University of Pennsylvania
rui.yan.peking@gmail.com

Chris Callison-Burch
University of Pennsylvania
ccb@cis.upenn.edu

Abstract
We discuss the problem of grammatical error correction, which has gained attention for its usefulness both in the development of tools for learners of foreign languages and as a component of statistical machine translation systems. We believe the task of suggesting grammar and style corrections in writing is well suited to a crowdsourcing solution but is currently hindered by the difficulty of automatic quality control. In this proposal, we motivate the problem of grammatical error correction and outline the challenges of ensuring quality in a setting where traditional methods of aggregation (e.g., majority vote) fail to produce the desired results. We then propose a design for quality control and present preliminary results indicating the potential of crowd workers to provide a scalable solution.

Author Keywords
ESL, Postediting, Crowdsourcing

Background and Motivation
Grammatical error correction has arisen as an important natural language processing task. The large and growing number of speakers of English as a second language has generated interest in developing tools to help students and professionals improve the grammaticality, fluency, and overall quality of their writing. The recent CoNLL shared
task on grammatical error correction has produced an
even greater wave of interest in developing systems to
correct non-native speaker errors [4]. In machine
translation, researchers have experimented with automatic
postediting to improve system output [2] and speed up
the work of human translators [3].

ESL error correction is naturally addressed by nonexperts,
since the task requires only fluency in English.
Crowdsourcing has been used successfully for proofreading
tasks, as in the Soylent word processor[1], but quality
control in these applications is ultimately performed
manually by the user, making it harder to scale.

Table 1: Multiple ways of producing a correct sentence for the
same input.

<table>
<thead>
<tr>
<th>Orig.</th>
<th>Gold</th>
<th>dist=33</th>
<th>dist=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>For serendipity discovery, the time taken is considered short</td>
<td>For serendipitous discovery, the time taken is considered short</td>
<td>Serendipitous discoveries do not take long</td>
<td>For serendipity discovery, the time taken is considered short</td>
</tr>
</tbody>
</table>

Table 2: Edit distance may favor lazy workers over workers
who make a conscientious effort.

Challenges of Quality Control
Effectively applying crowdsourcing to the problem of ESL
error detection requires creative methods for ensuring
worker quality and reliability. Unlike other crowdsourced
tasks, such as sentiment labeling, error annotations and
corrections require structured labels which can vary
considerably. We expect that workers will find multiple
ways to correctly edit a single sentence (see table 1), and
aggregating edits or measuring agreement is non-trivial.

Natural methods for automated quality control prove to be
problematic when applied to error correction. Using
simple consensus as an indicator of correctness is likely to
produce few or no edits, as the majority correction for any
given error span will likely be no correction.

Measuring performance on embedded gold standard data
also presents a challenge. Standard ways of comparing the
Turker’s corrected sentence to a reference corrected
sentence, such as string edit distance, are likely to favor
lazy workers. Table 2 shows how a simple edit distance
greatly prefers the unedited sentence to the sentence that
was corrected conscientiously.

Preliminary Results from Mechanical Turk
We asked Turkers to edit a subset of the sentences from
the training data released for the CoNLL 2013 shared
task. While their overall performance fell below the best
automatic systems participating in the shared task (figure
1), the performance of individual Turkers shows that there
are a considerable number of Turkers who are able to do
the task reliably (figure 2). The question is how to isolate
the reliable Turkers in order to obtain high-quality edits.
Figure 1: F1s (x100) of automated systems in CoNLL 2013 shared task (blue) and of Turkers (red). Turker performance is measured by taking the edits produced by the single highest-scoring Turker for each sentence. Turkers edited a subset of the training data, not the final test data.

Figure 2: F1 scores of each Turker vs. # of sentences corrected. Red lines show F1 scores of best CoNLL systems, black line is the average F1 of CoNLL systems. Omitted are 7 Turkers with >1000 sentences corrected. All had F1<0.15.

Proposed Designs
We propose a method for gathering workers’ edits which tracks atomic operations on single words and phrases. We designed an interface which constrains workers to make structured edits and stores those edits in a graph-based data structure (figure 3). This design allows us to isolate individual corrections in order to measure accuracy on reference translations, compute agreement between workers, and gather annotations on edits.

Figure 3 shows how we are able to use this data structure to compare edits across Turkers, even when both the manner in which the Turkers make their edits and the final version of the sentence vary greatly. It also shows how we can isolate phrases containing single edits in order to solicit edit-specific annotations in a second pass HIT.

Experimental Applications
To determine the types of errors we should anticipate in sentences written by non-native English speakers, we performed a survey of computer science and linguistics studies on ESL error correction. We compiled a list of error types mentioned in the literature, and organized the list into a hierarchy. The proposed taxonomy encompasses a much larger set of errors than that which is currently used by researchers working on automated correction.

We believe that Turkers will show high agreement when correcting errors from some of these categories (such as Noun Number) and less agreement on others (such as Run-on Sentence). In addition to crowdsourcing the editing itself, we plan to gather annotations based on our proposed taxonomy for each of the edits we receive from Turkers. We can use these annotations to measure agreement in order to settle on a set of error tags which results in high inner-annotator agreement.
Figure 3: Data structure allows us to measure agreement on a specific edit, even if final versions of the sentence vary considerably. Here, we are able to tell that the two workers agree that ‘into’ should be changed to ‘in’, even though they each perform the edit on a different version of the sentence.

As part of a complete pipeline for crowdsourcing error correction, we plan to provide feedback and error-analysis to the original translators. In a longitudinal study, we hope to show that, when provided with focused feedback, the number of errors that our translators make decreases over time. This design for crowdsourcing translation and postediting would both increase the quality of the translations and provide workers with an educational incentive to perform translation via crowdsourcing.

References

