Conservatism and Over-conservatism in Grammatical Error Correction

Abstract

Grammatical Error Correction systems (henceforth, *correctors*) aim to correct ungrammatical text, while changing it as little as possible. However, whereas such conservatism is a virtue for correctors, we find that state-of-the-art systems make substantially less changes to the source sentences than needed. Analyzing the distribution of possible corrections for a given sentence, we show that this over-conservatism likely stems from the inability of a handful of reference corrections to account for the full variation of valid corrections for a given sentence. This results in undue penalization of valid corrections, thus disincentivizing correctors to make changes. We also show that simply increasing the number of references is unlikely to resolve this problem, and conclude by presenting an alternative reference-less approach based on semantic similarity.

1 Introduction

Grammatical Error Correction (GEC) is receiving considerable interest recently, notably through the GEC-HOO (Dale and Kilgarriff, 2011; Dale et al., 2012) and CoNLL shared tasks (Kao et al., 2013; Ng et al., 2014). Within GEC, considerable effort has been placed on evaluation (Tetreault and Chodorow, 2008; Madnani et al., 2011; Felice and Briscoe, 2015; Napoles et al., 2015), a notoriously difficult challenge, in part due to the many valid corrections each learner's language (LL) sentence may have (Chodorow et al., 2012).

An important criterion in the evaluation of correctors is their ability to generate corrections that are faithful to meaning of the source. In fact, many would prefer a somewhat cumbersome or even an occasionally ungrammatical correction over a correction that alters the meaning of the source (Brockett et al., 2006). As a result, annotators are often instructed to be conservative in their corrections when compiling gold standard corrections for the task (e.g., in the Treebank of Learner English (Nicholls, 2003)). There were different attempts to formally capture this precision/recall asymmetry such as the standardized use of $F_{0.5}$ over F_1 (Dahlmeier and Ng, 2012) and the choices of weights in I-measure (Felice and Briscoe, 2015).

However, during development and training penalizing over-correction more harshly than under-correction may lead to reluctance of correctors to make any changes (henceforth, *over-conservatism*). Using only one or two reference corrections, a common practice in GEC, compounds this problem, as correctors are not only harshly penalized for making incorrect changes, but are often penalized for making **correct** changes not found in the reference.

Indeed, we show that current state of the art systems suffer from over-conservatism. Evaluating the output of 12 recent correctors, we find that all of them substantially under-predict corrections relative to the gold standard (§2).

We first assess whether the undue penalization of valid corrections can be resolved by increasing the number of references, which we denote with M (§3). We start by estimating the number and frequency distribution of the valid corrections per sentence, arriving at an estimate of over 1000 corrections for sentences of no more than 15 tokens. We then consider two representative reference-based measures (henceforth, *RBMs*) for assessing the validity of a proposed correction relative to a set of references, and characterize the distribution of their scores as a function of M. Our results show that both measures substantially under-estimate the true performance of the correctors. Moreover, they show that increasing M only partially addresses the incurred bias, as both RBMs approach saturation for M values of 10–20, indicating that a prohibitively large M may be required for reliable estimation.

Our findings echo the results of Bryant and Ng (2015), who study the effect of M on F-score, the most commonly used measure for GEC. Their work focused on obtaining a more reliable estimate of correctors' performance, and proposed to do so by normalizing corrector's estimated performance with the performance of a human corrector. However, while such normalization may yield more realistic performance estimates, it has no effect on the training and tuning of correctors.

We conclude by proposing an alternative reference-less semantic evaluation approach which assesses the extent to which a correction faithfully represents the semantics of the source, by measuring the similarity of their semantic structures (§4). This approach can be combined with a reference-less measure of grammaticality, based on automatic error detection, as proposed by Napoles et al. (2016). Our experiments support the feasibility of the proposed approach, by showing that (1) semantic structural annotation can be consistently and automatically applied to LL, (2) that the proposed measure is less prone to unduly penalize valid corrections and (3) that the measure penalizes corrections that change the semantic structure significantly.

2 Over-Conservativism in GEC Systems

We demonstrate that current correctors suffer from over-conservatism: they tend to make too few changes to the source.

2.1 Notation

We assume each source sentence x has a set of valid corrections $Correct_x$, and a discrete distribution $\mathcal{D}_x$ over them, where $P_{\mathcal{D}_x}(y)$ for $y \in Correct_x$ is the probability a human annotator would correct x as y.

Let X be the evaluated set of source LL sentences where X consists of the sentences $x_1 \dots x_N$, each independently sampled from some distribution $\mathcal{L}$ over LL sentences and denote $\mathcal{D}_i := \mathcal{D}_{x_i}$. Each x_i is paired with M corrections $Y_i = \{y_i^1, \dots, y_i^M\}$, which are independently sampled from $\mathcal{D}_i$.[1] We define the *coverage* of M references for a sentence x_i to be $P(y \in Y_i | y \in Correct_i)$ for Y_i of size M, and y sampled according to $\mathcal{D}_i$.

A corrector C is a function from LL sentences to proposed corrections (strings). An assessment measure is a function from X, Y and C to a real number. We use the term "true measure" to refer to the measure's output where the references include all possible corrections, i.e., $Y_i = Correct_i$ for every i.

Experimental Setup. We conduct all experiments on the NUCLE test dataset, a parallel corpus of LL essays and their corrected versions, which is the de facto standard in GEC. The corpus contains 1414 essays in LL and 50 test essays, each of about 500 words.

We evaluate all participating systems in the CoNLL 2014 shared task, in addition to two of the best performing systems on this dataset. The particiapting systems and their abbreviations are: Adam Mickiewicz University (AMU), University of Cambridge (CAMB), Columbia University and the University of Illinois at Urbana-Champaign (CUUI), Indian Institute of Technology, Bombay (IITB), Instituto Politecnico Nacional (IPN), National Tsing Hua University (NTHU), Peking University (PKU), Pohang University of Science and Technology (POST), Research Institute for Artificial Intelligence, Romanian Academy (RAC), Shanghai Jiao Tong University (SJTU), University of Franche Comté (UFC), University of Macau (UMC), Rozovskaya and Roth (2016, RoRo), Xie et al. (2016, char). All are trained and tested on the NUCLE corpus.

We compare the prevalence of changes made to the source by the correctors, relative to their prevalence in the NUCLE reference. In order to focus on the more substantial changes, we exclude from our evaluation all non-alphanumeric characters, both within tokens or as tokens of their own.

Measures of Conservatism. We consider three types of divergences between the source and the ref-

[1]Our analysis assumes M is fixed across source sentences. Generalizing the analysis to sentence-dependent M values is straightforward.

erence. First, we measure to what extent *words* were changed: altered, deleted or added. To do so, we compute word alignment between the source and the reference, casting it as a weighted bipartite matching problem, between the source's words and the correction's. Edge weights are assigned to be the edit distances between the tokens. We note that aligning words in GEC is much simpler than in machine translation, as most of the words are kept unchanged, deleted fully, added, or changed slightly. Following word alignment, we define the WORDCHANGE measure as the number of unaligned words and aligned words that were changed in any way.

Second, we quantify word *order* differences using Spearman's ρ between the order of the words in the source sentence, and the order of their corresponding words in the correction according to the word alignment. $\rho = 0$ where the word order is uncorrelated, and $\rho = 1$ where the orders exactly match. We report the average ρ over all source sentences pairs.

Third, we report how many source sentences were split and how many concatenated by the reference and the correctors.

Results. Figure 1 presents the outcome of the three measures. Results show that the reference corrections make changes to considerably more source sentences than any of the correctors, and within each changed sentence changes more words and makes more word order changes, often an order of magnitude more. For example, the reference has 36 sentences with 6 word changes, where the most sentences with 6 word changes by any corrector is 5.

For completeness, we measured the prevalence of changes in another corpus, the TreeBank of Learner English (Yannakoudakis et al., 2011), and obtained similar results.

3 Multi-Reference Measures

In this section we argue that the observed over-conservatism of correctors likely stems from them being developed to optimize RBMs that suffer from low-coverage. We begin with a motivating analysis of the relation between low-coverage and over-conservatism (§3.1).We then continue with an empirical assessment of the distribution of corrections for a given sentence (§3.3) and the effect of M on commonly used RBMs (§3.4). We discuss the im-

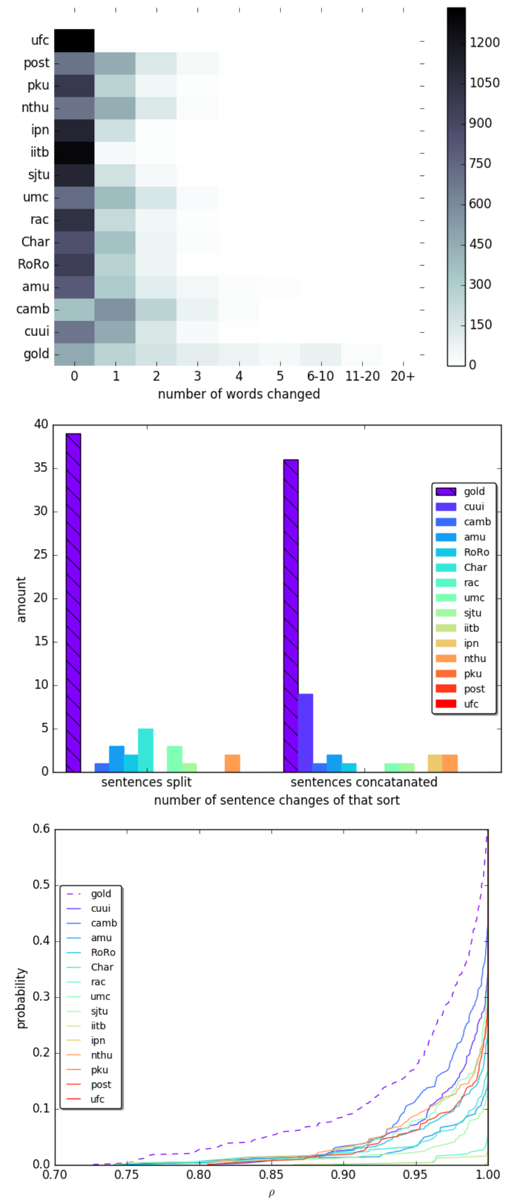

Figure 1: The prevalence of changes of different types in correctors' output and in the NUCLE references. The top figure presents the number of sentence pairs (heat) for each number of word changes (x-axis; measured by WORDCHANGE) for each of the different systems and the references (y-axis). The middle figure presents the number of source sentences (y-axis) concatenated (right bars) or split (left bars) in the references (striped column) and in the correctors' output (colored columns). The bottom figure presents the percentage of sentence pairs (y-axis) where the Spearman ρ values do not exceed a certain threshold (x-axis). See §2.1 for a legend of the correctors. The three figures show that under all measures, the gold standard references make substantially more changes to the source sentences than any of the correctors, in some cases an order of magnitude more.

plications of our results, concluding that RBMs may only partially address over-conservatism (§3.6).

3.1 Motivating Analysis

The relation between coverage and over-conservatism requires some explanation. We abstract away from the details of the training procedure, and assume that correctors attempt to maximize an objective function, over some training or development data, and assume for simplicity of the argument that training proceeds by iterating over the samples, as with the Perceptron algorithm.

Assume the corrector is faced with a phrase which it predicts to be ungrammatical. Assume p_{detect} is the probability that this prediction is correct. Assume $p_{correct}$ is the probability it is able to predict a valid correction for this phrase (including correctly identifying it as erroneous). Finally, assume that the corrector is evaluated against M references for which the coverage of the phrase is $p_{coverage}$, namely the probability that a valid correction will be found among M randomly sampled references.

We will now assume that the corrector may either choose to correct with the correction it finds the most likely or not at all. If it chose not to correct, its probability of being rewarded (i.e., its output is in the reference set Y) is $(1 - p_{detect})$. Otherwise, its probability of being rewarded is $p_{correct} \cdot p_{coverage}$. In cases where

$$p_{correct} \cdot p_{coverage} < 1 - p_{detect} \tag{1}$$

a corrector is disincentivized from altering the phrase. We expect Condition 1 to frequently hold in cases that require non-trivial changes, which are characterized both by low $p_{coverage}$ (as non-trivial changes can often be made in numerous ways), and by lower expected performance by the corrector.

Moreover, asymmetric measures (e.g., $F_{0.5}$) penalize invalidly correcting more harshly than not correcting an ungrammatical sentence. In these cases, Condition 1 should be rephrased as

$$p_{correct} \cdot p_{coverage} - (1 - p_{correct} p_{coverage})\,\alpha < 1 - p_{detect}$$

where α is the ratio between the penalty for introducing a wrong correction and the reward for a valid correction. Condition 1 is much more likely to hold in these cases.

In order to validate this analysis empirically, we conduct an experiment for determining whether increasing the number of references available for training indeed reduces conservatism. As there is no multiple-reference corpus available which is large enough for re-training a corrector, we take an oracle reranking approach as a simulation, and test whether the availability of increasingly more references to train on reduces its conservativeness.

Concretely, given a set of sentences, each paired with $\mathcal{M}$ references, and given a k-best list produced by a corrector, we define an oracle re-ranker that selects the highest scoring correction of the k-best list, according to a given evaluation measure. As a test case, we use the RoRo system, with k=100, and apply it to the largest available LL corpus which is paired with a substantial amount of references, namely the NUCLE-test corpus, which has 12 references (Bryant and Ng, 2015). We use the common F-score as an evaluation measure.

We examine the conservativeness of the oracle reranker for different M values, averaging over 1312 samples of M references from the available set of $\mathcal{M} = 12$. Our results show a consistent decrease in conservatism in word changes with the increase in coverage (Figure 2), and no significant change in word order,[2] indicating that conservatism is indeed related to the number of references available to the learner.[3][4]

3.2 Data

Our analysis assumes that we have a reliable estimate for the distribution of corrections $\mathcal{D}_x$ for the source sentences we evaluate. Our experiments in the following section are run on a random sample of 52 sentences with a maximum length of 15 from the NUCLE test data. Through the length restriction we avoid introducing too many independent errors that may drastically increase the number of annotations variants (as every combination of corrections for these errors is possible), thus resulting in an unreliable estimation for $\mathcal{D}_x$. Sentences with less than 6 words were discarded, as they were mostly a result of sentence segmentation errors.

Crowdsourcing has proven effective in GEC eval-

[2] As we only rerank individual sentences, there is clearly no change in the number of sentences split or concatenated.

[3] OA: do we need to add a discussion as to why this is not trivial?

[4] LC: I am not sure it won't only overcomplicate things.

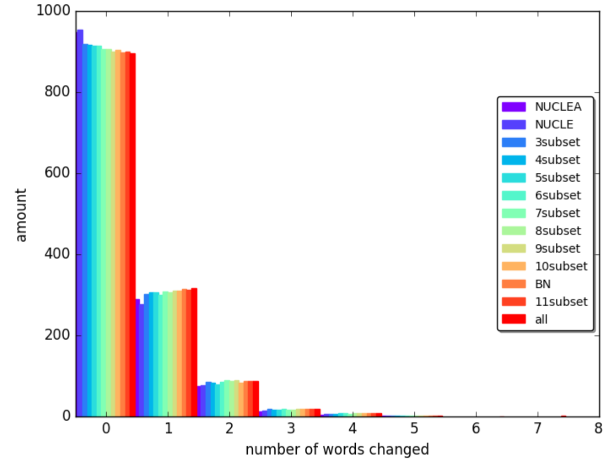

Figure 2: The amount of sentences (y-axis) with a given number of words changes (x-axis), following oracle reranking with different M values (column colors).

uation (Madnani et al., 2011; Napoles et al., 2015) and in related tasks such as machine translation (Zaidan and Callison-Burch, 2011; Post et al., 2012). We thus use crowdsourcing for obtaining a sample from $\mathcal{D}_x$. Specifically, for each of the 52 source sentences, we elicited 50 corrections from Amazon Mechanical Turk workers. Aiming to judge grammaticality rather than fluency, we asked the workers to correct only when necessary, not for styling. 4 sentences did not require any correction according to almost half the workers and were hence discarded.

3.3 Estimating the Distribution of Corrections

We begin by estimating $\mathcal{D}_x$ for each sentence, using the crowdsourced corrections. We use UNSEENEST (Zou et al., 2016), a non-parametric algorithm that estimates a multinomial distribution, in which the individual values do not matter, only the distribution of probabilities across values. UNSEENEST aims to minimize earthmover distance between the estimated histogram and the histogram of the distribution and proven empirically and mathematically to be very exact. [5] UNSEENEST was originally developed for assessing how many variants a gene might have, including undiscovered ones, and their relative frequencies. This is a similar setting to the one tackled here. Our manual tests of UNSEENEST with small artificially created frequencies showed satisfactory results.[6]

By the estimates from UNSEENEST, most source sentences have a large number of corrections with low probability accounting for the bulk of the probability mass and a rather small number of frequent corrections. Table 1 presents the mean numbers of different corrections with frequency at least γ (for different values of γ), and their total probability mass. For instance, 74.34 corrections account for 75% of the total probability mass of the corrections, each occurring with a frequency of 0.1% or higher.

	Frequency Threshold (γ)			
	0	0.001	0.01	0.1
Variants	1351.24	74.34	8.72	1.35
Mass	1	0.75	0.58	0.37

Table 1: Estimating the distribution of corrections $\mathcal{D}_x$. The table presents the mean number of corrections per sentence with probability of more than γ (top row), as well as their total probability mass (bottom row).

The overwhelming number of rare corrections raises the question whether these can be regarded as noise. To test this we conducted another crowdsourcing experiment, where 3 annotators were asked to judge whether a correction produced in the first experiment, is indeed a valid correction. Results of the mean number of annotators who judged a correction that was proposed a given number of times in the training data, is given in Figure 4. Results show that the original frequency of the correction has little effect on how often it was deemed valid, where even the rarest corrections were judged valid 78% of the times.

3.4 Under-estimation as a function of M

In the previous section we presented empirical assessment of the distribution of corrections to a sentence. We turn to estimate the resulting bias, i.e., the under-estimation of RBMs, for different M values.

We discuss two similarity measures. One is the sentence-level accuracy (also called "Exact Match") and the other is the GEC F-score.

Sentence-level Accuracy. Sentence-level accuracy (also "Exact Match") is the percentage of cor-

[5] <to be disclosed upon publication>UnseenEst implementation

[6] All data we collected, along with the estimated distributions can be found in <to be disclosed upon publication>

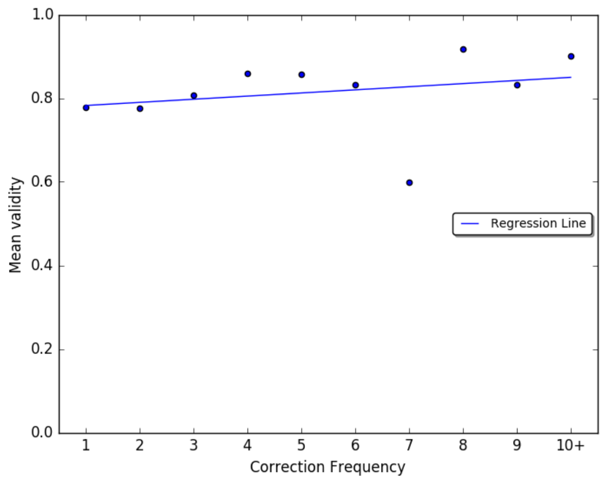

Figure 3: The mean frequency (y-axis) in which a correction that was produced a given number of times (x-axis), was judged to be valid.

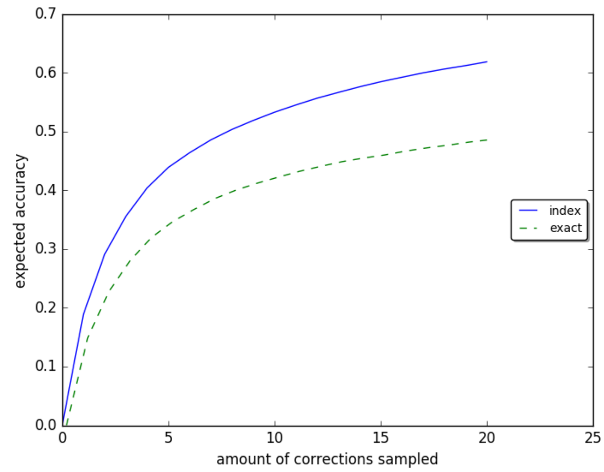

Figure 4: Accuracy and Exact Index Match values for a perfect corrector (y-axis) as a function of the number of references M (x-axis).

rections that exactly match one of the references. Accuracy is a basic, interpretable measure, used in GEC by, e.g. Rozovskaya and Roth (2010). It is also closely related to the 0-1 loss function commonly used for training statistical correctors (Chodorow et al., 2012; Rozovskaya and Roth, 2013).

Formally, given test sentences $X = \{x_1, \ldots, x_N\}$, their references $Y_1, \ldots, Y_N$, and a corrector C, we define C's accuracy to be

$$Acc\left(C; X, Y\right) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{C(x_i)\in Y_i}. \quad (2)$$

Note that C's accuracy is in fact an estimate of C's probability to produce a valid correction for a sentence, or C's *true accuracy*. Formally:

$$TrueAcc\left(C\right) = P_{x\sim L}\left(C\left(x\right) \in Correct_x\right).$$

The bias of $Acc\left(C; X, Y\right)$ for a sample of N sentences, each paired with M references is then

$$TrueAcc\left(C\right) - \mathbb{E}_{X,Y}\left[Acc\left(C; X, Y\right)\right] = \quad (3)$$
$$TrueAcc\left(C\right) - P\left(C\left(x\right) \in Y\right) = \quad (4)$$
$$Pr\left(C\left(x\right) \in Correct_x\right) \cdot \quad (5)$$
$$\left(1 - Pr\left(C\left(x\right) \in Y | C\left(x\right) \in Correct_x\right)\right) \quad (6)$$

We observe that the bias, denoted b_M, is not affected by N, only by M. As M grows, Y approximates $Correct_x$ better, and b_M tends to 0.

In order to gain insight into the evaluation measure and the GEC task (and not the idiosyncrasies of specific systems), we consider an idealized learner, which, when correct, produces a valid correction with the same distribution as a human annotator (i.e., according to $\mathcal{D}_x$). Formally, we assume that, if $C(x) \in Correct_x$ then $C(x) \sim \mathcal{D}_x$. Hence the bias b_M (Equation 6) can be re-written as

$$P(C(x) \in Correct_x) \cdot (1 - P_{Y\sim\mathcal{D}_i^M, y\sim\mathcal{D}_x}(y \in Y)).$$

We will henceforth assume that C is perfect (i.e., its true accuracy $Pr\left(C(x) \in Correct_x\right)$ is 1). Note that assuming any other value for C's true accuracy would simply scale b_M by that accuracy. Similarly, assuming only a fraction p of the sentences require correction scales b_M by p.

We estimate b_M empirically using its empirical mean on our experimental corpus:

$$\hat{b}_M = 1 - \frac{1}{N}\sum_{i=1}^{N} P_{Y\sim\mathcal{D}_i^M, y\sim\mathcal{D}_i}\left(y \in Y\right).$$

Using the UNSEENEST estimations of $\mathcal{D}_i$, we can compute $\hat{b}_M$ for any size of Y_i (value of M). However, as this is highly computationally demanding, we estimate it using sampling. Specifically, for every $M = 1, ..., 20$ and x_i, we sample Y_i 1000 times (with replacement), and estimate $P\left(y \in Y_i\right)$ as the covered probability mass $P_{\mathcal{D}_i}\{y : y \in Y_i\}$.

We repeated all our experiments where Y_i is sampled without replacement, in order to simulate a case where reference corrections are collected by a single annotator, and are thus not repeated. We find similar trends with faster increase in accuracy reaching over 0.47 with $M = 10$.

Figure 4 presents the expected accuracy values for our perfect corrector (i.e., 1-$\hat{b}_M$) for different values of M. Results show that even for values of M which are much larger than those considered in the GEC literature (e.g., $M = 20$), the expected accuracy is only about 0.5. As M increases, the contribution of each additional correction gets smaller to the point it contributes little to the accuracy (the slope is about 0.004 around $M = 20$).

We also experiment with a more relaxed measure, *Exact Index Match*, which is only sensitive to the identity of the changed words and not to what they were changed to. Formally, two corrections c and c' over a source sentence x match if for their word alignments with the source (computed as above) $a : \{1, ..., |x|\} \to \{1, ..., |c|, Null\}$ and $a' : \{1, ..., |x|\} \to \{1, ..., |c'|, Null\}$, it holds that $c_{a(i)} \neq x_i$ iff $c'_{a'(i)} \neq x_i$, where $c_{Null} = c'_{Null}$.

Figure 4 also presents the expected accuracy in this case for different values of M, which indicate that while scores of a perfect corrector are somewhat higher, still with $M = 10$, it is 0.54. As Exact Index Match can be interpreted as an accuracy measure for error detection (rather than correction), our results indicate that error detection systems may suffer from similar difficulties.

The analytic tools we have developed support the computation of the entire distribution of the accuracy, and not only its expected values. From Equation 2 we see that Accuracy has a Poisson Binominal distribution (i.e., it is a sum of independent Bernoulli variables with different success probabilities), whose success probabilities are $P_{y,Y \sim \mathcal{D}_i}(y \in Y)$, which can be computed, as before, using UNSEENEST's estimate for $\mathcal{D}_i$. Estimating the density function allows for the straightforward definition of significance tests for the measure, and can be performed efficiently (Hong, 2013).[7]

F-Score. While accuracy is commonly used as a loss function for training GEC systems, the F_α score is standard when reporting system performance (and consequently in hyper-parameter tuning).

Computing F-score for GEC is not at all straightforward. The score is computed in terms of *edit* matches between a correction and the references, where edits are sub-strings of the source that are replaced in the correction/reference. The HOO shared task used an earlier version of F-score, which required that the proposed corrections include edits explicitly. Later on, relieving correctors from the need to produce edits, F-score was redefined optimistically, maximizing over all possible annotations that generate the correction from the source.[8] M^2 (Dahlmeier and Ng, 2012) was designated to compute this F score and is the standard evaluation for GEC.

The complexity of the measure prohibits an analytic approach, and we instead use a bootstrapping approach to estimate the bias incurred by not being able to exhaustively enumerate the set of valid corrections. As with accuracy, in order to avoid confounding our results with system-specific biases, we assume the evaluated corrector is perfect and sample its corrections from the human distribution of corrections $\mathcal{D}_x$.

Concretely, given a value for M and for N, we uniformly sample from our experimental corpus source sentences $x_1, ..., x_N$, and M corrections for each $Y_1, ..., Y_N$ (with replacement). Setting a realistic value for N in our experiments is important for obtaining comparable results to those obtained on the NUCLE corpus (see below), as the expected value of F-score may depend on N (unlike Accuracy, it is not additive). In accordance with the NUCLE's test set, we set $N = 1312$ and assume that 136 of the sentences require no correction. The latter reduces the overall bias by their frequency in the corpus, and are thus important to include for obtaining realistic results.

The bootstrapping procedure is carried out by the accelerated bootstrap procedure (Efron, 1987), with 1000 iterations. We also report confidence intervals ($p = .95$), computed using the same procedure.[9]

Figure 5 presents the results of this procedure, which further indicate the insufficiency of commonly used M values for training and development (1 or 2) for obtaining a reliable estimation of a corrector's performance. For instance, the $F_{0.5}$ score for our perfect corrector, whose true F-score is 1, is only 0.42 with $M = 2$. Moreover, the saturation effect observed for accuracy is even more pronounced with our experiments on F-score.

[7] An implementation of this method and the estimated density functions will be released upon publication.

[8] Since our crowdsourced corrections do not include an explicit annotation of edits, we produce edits heuristically.

[9] We use the Python scikits.bootstrap implementation.

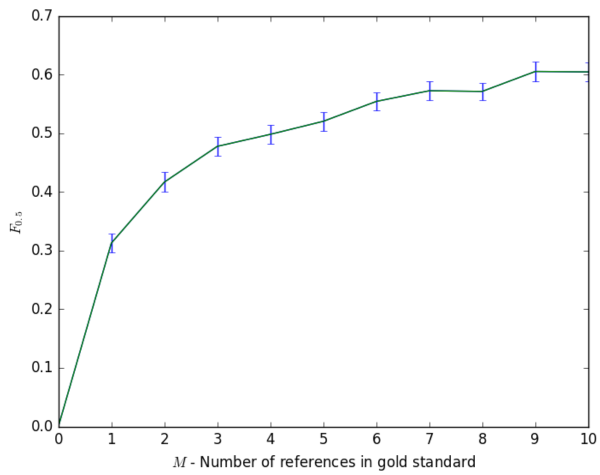

Figure 5: $F_{0.5}$ values for a perfect corrector (y-axis) as a function of the number of references M (x-axis). Each data point is paired with a confidence interval ($p = .95$).

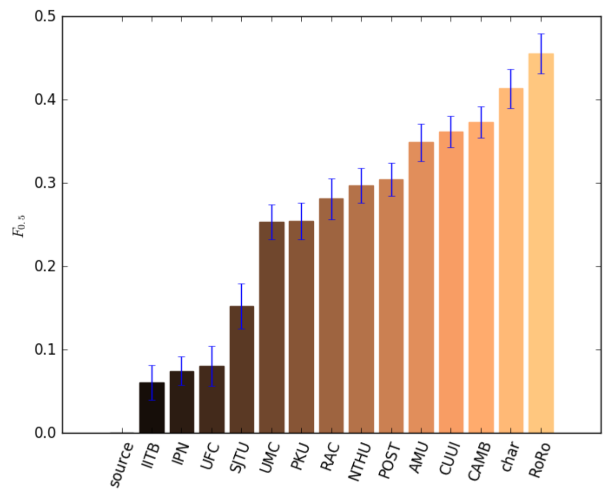

Figure 6: $F_{0.5}$ values for different correctors, including confidence interval ($p = .95$). The left-most column ("source") presents the F-score of a corrector that doesn't make any changes to the source sentences. See §2.1 for a legend of the correctors.

The F-score coverage experiment is very similar to that of Bryant and Ng (2015), who also compared the F-score of a human correction against an increasing number of references, and indeed produced similar results Bryant and Ng (2015). They differ from the experiments reported in this section, in that they did not attempt to estimate the distribution of corrections, and focused exclusively on the F-score measure.

3.5 Significance of Real-World Correctors

The bootstrapping method for computing the significance of the F-score can also be useful for assessing the significance of the differences in correctors' performance reported in the literature. We report results with the bootstrapping protocol (§3.4) to compute the confidence interval of different correctors with the current NUCLE test data ($M = 2$).

Our results (Figure 6) present a mixed picture: some of the differences between previously reported F-scores are indeed significant and some are not. For example, the best performing corrector is significantly better than the second, but the latter is not significantly better than the third and fourth.

3.6 Discussion

Our empirical results show that the number of corrections needed for reliable reference-based measures may be prohibitively large in practice. Results suggest that there are hundreds of valid corrections with low probability, whose total probability mass is substantial. RBMs such as accuracy and F-score thus show diminishing returns from increasing the value of M over values of about 10.

Returning to condition (1) (§3.1), we find that the coverage (which is equal to the accuracy depicted in Figure 4) is lower than 0.5 for $M = 2$ on average (for short sentences). For cases of non-trivial changes, we expect it might be even lower, suggesting that condition (1) often holds in practice, incentivizing over-conservatism.

Considering the F-score of the best performing systems in Figure 6, and comparing them to the F-score of a perfect corrector with $M = 2$, we find that their scores are comparable, where RoRo in fact surpasses a perfect corrector's F-score. While it is possible that these correctors outperform the perfect corrector by learning how to correct a sentence in the same way as one of the NUCLE annotators did, we view this possibility as unlikely as our results (§2) show that the output of these systems considerably diverges from NUCLE's references. A more likely possibility is that these systems' high performance relative to a perfect corrector's is due to these correctors having learned to predict when not to correct.

Two recent RBMS have been proposed. One is I-MEASURE (Felice and Briscoe, 2015) which introduces novel features to GEC evaluation, such as distinguishing different quality levels of ungrammatical corrections (e.g., some improve the quality of the source, while others degrade it), and restricting edits to only consist of single words, rather than phrases. The other is GLEU (Napoles et al., 2015), an adaptation of BLEU that was shown to correlate well with human rankings. We expect our findings, that RBMs substantially under-estimate the performance of correctors, to generalize to these RBMs, as they all apply string similarity measures relative to a fairly small number of references. These measures thus address orthogonal gaps in GEC evaluation from the ones presented here. Following the proposal of Sakaguchi et al. (2016), to emphasize fluency over grammaticality in reference corrections, only compounds this problem, as it results in a larger number of valid corrections.

Finally, note that addressing under-estimation by comparing to a human expected score (in our terms, a perfect corrector) with the same M (Bryant and Ng, 2015), does not address over-conservatism, as it only scales the original measure. Moreover, as seen above, a human correction's score is not necessarily an upper bound, as an over-conservative corrector may surpass a perfect corrector in performance.

4 Semantic Faithfulness Measure

In this section we propose a measure that eschews the use of reference corrections, instead measuring the semantic faithfulness of the proposed correction to the source. Concretely, we propose to measure the semantic similarity of the source and the proposed correction through the graph similarity of their representations. Such a measure has to be complemented with an error detection procedure, as it only captures faithfulness, the extent to which the meaning of the source is preserved in the correction, and not its grammaticality. See Napoles et al. (2016) for a proposal of a complementary measure based on automatic error detection.

A similar approach was also taken in the field of machine translation, after showing multiple references are better for evaluation, but costly (Albrecht and Hwa, 2008; Turian et al., 2006). Several ways to capture semantics (adequacy) to overcome this were proposed (Snover et al., 2009). More specifically, reference-less machine translation evaluation methods were proposed and shown at the time to be at least as reliable as the reference-based ones (Reeder, 2006; Albrecht and Hwa, 2007; Specia et al., 2009; Specia et al., 2010; Lo and Wu, 2011). Perhaps most similar is the work of Banchs et al. (2015), combining semantic and grammatical measures. Despite their similarity, all those methods did not use current semantic annotation schemes or had other specific reasons not to be used in our case.

As a test case, we use the UCCA scheme to define semantic structures (Abend and Rappoport, 2013), motivated by its recent use in semantic machine translation evaluation (Birch et al., 2016).

We conduct two experiments supporting the feasibility of our approach. We show that semantic annotation can be consistently applied to LL, through IAA experiments and that a perfect corrector scores high on this measure.

4.1 Structural Representation in LL

While linguistic theories propose that each learner makes consistent use of syntax (Huebner, 1985; Tarone, 1983), this use may not conform to the syntax of the learned language, or of any other known language. This entails difficulties in defining syntactic annotation for LL, as, on the face of it, the language of each learner has to be annotated in its own terms.

LL resources annotate syntactic errors in different ways. Berzak et al. (2016) and Ragheb and Dickinson (2012) annotate according to the syntax used by the learner, even if this use is not grammatical. Such annotation may be unreliable as a source of semantic information, as semantically similar sentences, formulated by different learners, may use considerably different structures. Nagata and Sakaguchi (2016) take the opposite approach, and try to be faithful to the syntax intended by the learner, this is also the case in works on robustness of parsers assuming grammar should convey meaning and stay robust to errors (Bigert et al., 2005; Foster, 2004). However, such an approach faces difficulties due to the multitude of different syntactic structures that can be used to express a similar meaning.

In this section, we use semantic annotation to structurally represent LL text. Semantic structures are faithful to the intended meaning of the sentence, and not to its formal realization, and thus face less conflicts where the syntactic structure used diverges from the one intended. We are not aware of any previous attempts to semantically annotate LL text.

UCCA. UCCA is a semantic annotation scheme that builds on typological and cognitive linguistic theories. The scheme's aims are to provide a coarse-grained, cross-linguistically applicable representation. Importantly, UCCA's categories directly reflect semantic, rather than distributional distinctions. For instance, UCCA is not sensitive to POS distinctions: a Scene's main relation can be a verb but also an adjective ("He is **thin**") or a noun ("John's **decision**"). Indeed, Sulem et al. (2015) have found that UCCA structures are preserved remarkably well across English-French translations.

UCCA structures are directed acyclic graphs, where the words in the text correspond to (a subset of) their leaves. The nodes of the graphs, called *units*, are either leaves or several elements jointly viewed as a single entity according to some semantic or cognitive consideration. The edges bear one or more categories, indicating the role of the sub-unit in the relation that the parent represents.

UCCA views the text as a collection of *Scenes* and relations between them. A Scene, the most basic notion of this layer, describes a movement, an action or a state which is persistent in time. Every Scene contains one main relation, zero or more *Participants*, which are interpreted in a broad sense, and include locations, destinations and complement clauses, and *Adverbials*, such as temporal descriptions.

4.2 Experimental Setup

We employ two annotators, and train them by annotating both LL and standard English passages, until a high enough agreement has been reached (a total of 6 hours of training). Training passages are excluded from the evaluation. We use UCCA's annotation guidelines[10] without any adaptations.

We experiment on 7 essays and their corrections, comprising each of about 500 words. In order to measure IAA, we assigned 4 of these essays to both annotators and compute their agreement. In order to measure the faithfulness score for a perfect corrector, we annotate both the source and the corrected version for 6 essays, some of which were annotated by both annotators.

[10] http://www.cs.huji.ac.il/~oabend/ucca.html

4.3 Semantic Similarity Measures

IAA Measure. We define a similarity measure over UCCA annotations G_1 and G_2 over the same set of leaves (tokens) W. For a node v in either graph, define its yield $yield(v) \subseteq W$ as its set of leaf descendants. Define a pair of edges $(v_1, u_1) \in G_1$ and $(v_2, u_2) \in G_2$ to be matching if $yield(u_1) = yield(u_2)$ and they have the same label. Labeled precision and recall are defined by dividing the number of matching edges in G_1 and G_2 by $|E_1|$ and $|E_2|$, respectively, and the *DAG F-score* is their harmonic mean. We note that the measure collapses to the common parsing F-score if G_1 and G_2 are trees.

Semantic Faithfulness Measure. Computing a faithfulness measure is slightly more involved, as the source sentence graph G_s and its correction G_c do not share the same set of leaves.

We assume a (possibly partial, possibly many-to-1) alignment between G_s and G_c, $A \subset V_s \times V_c$. An edge $(v_1, v_2) \in E_c$ is said to match an edge $(u_1, u_2) \in E_s$ if they have the same label and $(v_2, u_2) \in A$. Recall (Precision) is defined as the ratio of edges in E_s (E_c) that have a match in E_c (E_s) respectively, and F-score is their harmonic mean. We note that this measure indeed collapses to the DAG F-score discussed above where A includes all pairs of nodes in E_s and E_c that have the same yield.

In order to define the alignment between V_s and V_c, we begin by aligning the leaves (tokens) in V_s and V_c using the same method detailed in §2. Denote the resulting leaf alignment with $A_l \subset Leaves_s \times Leaves_c$. We now extend A_l to define the node alignment A, aligning each non-leaf $v \in V_s$ with the node $u \in V_c$ that maximizes

$$w(v, u) = \frac{|A_l \cap (yield(u) \times yield(v))|}{|yield(u)|}.$$

We exclude from A pairs v, u such that $w(v, u) = 0$. The resulting F-score measure, using the resulting A is called UCCA Similarity (UCCASIM). As the resulting alignment may differ when aligning nodes from V_c to V_s and the other way around, we report the resulting F-score in both directions.

Note that UCCASIM is somewhat more relaxed than DAG F-score defined above, as it also aligns nodes whose yields are not in perfect alignment with one another, unlike DAG F-score which requires a perfect match. While this relaxation is necessary, given that corrections often add or remove nodes, thus eliminating the possibility of a perfect alignment, in order to obtain comparable IAA scores, we report IAA using UCCASIM as well.

For completeness, we replicate the protocol used by Sulem et al. (2015) for comparing the UCCA annotations of English-French translations, which we call Distributional Similarity (DISTSIM). For a given UCCA label l, $c_i(l)$ is the number of l-labeled UCCA edges in the i-th source sentence, and $d_i(l)$ is the number of l-labeled UCCA edges in its corresponding correction. We define DISTSIM(l) between these sentences to be $\frac{1}{N}\sum_{i=1}^{N}|c_i(l) - d_i(l)|$, where N is the total number of sentence pairs.

4.4 The Faithfulness of a Perfect Corrector

We obtain an IAA DAG F-score of 0.845 (Precision 0.834, Recall 0.857), which is comparable to the IAA reported for English Wikipedia texts by (Abend and Rappoport, 2013). As another point of comparison, we doubly annotate 3 corrected NUCLE passages, obtaining a similar IAA.

These results suggest that annotating LL with UCCA does not degrade IAA, and can be applied as consistently to LL as to standard English.

Table 2 (left) presents the UCCASIM scores obtained by comparing the NUCLE references and the source sentences, or equivalently the score of a perfect corrector. To control for differences between the annotators, we explore both a setting where both sides were annotated by the same annotator, and a setting where they were annotated by different ones. As an upper bound on the score of a perfect corrector (using different annotators), we also report the IAA on source sentences, computed using UCCASIM.

Our results indicate that a perfect corrector obtains a score comparable to the IAA, which indicates that UCCASIM is indeed insensitive to the surface divergence between a source sentence and its valid corrections.

Finally, the right-hand side of Table 2 presents DISTSIM between the source and reference sentences. Our results are similar to the ones obtained by Sulem et al. (2015), who compared standard English sentences and their French translations.

	UCCASIM			DISTSIM	
	s→r	r→s	Avg	A+D	Scene
Different	0.85	0.83	0.84	0.96	0.93
Same	0.92	0.91	0.92	0.97	0.96
IAA	0.85	0.81	0.83	-	-
SAR15	-	-	-	0.95	0.96

Table 2: The faithfulness of a perfect corrector. The left-hand side presents UCCASIM where the alignment is computed from the source to the reference (s→r), the opposite direction (r→s), and their average (Avg). The right-hand side presents DISTSIM for the UCCA categories Participants and Adverbials together (A+D), and Scene (Scene), as reported by Sulem et al. (2015). The rows indicate whether the same annotator annotated the source and reference or not. As an upper bound, we report IAA computed using UCCASIM (IAA row). Results show that the perfect corrector's faithfulness is comparable with IAA. The bottom row presents the results reported by Sulem et al. (SAR15) on English-French translations, which are comparable to ours.

4.5 Automatic UCCASim

We turn to experimenting with an automatic variant of UCCASim, where the UCCA structures are parsed automatically. We use the TUPA UCCA parser (Hershcovich et al., 2017) to generate the UCCA structures, instead of the human annotators, replicating all other details of the experimental setup. TUPA is used in its biLSTM model, trained on the UCCA English Wikipedia corpus.

We obtain a UCCASim measure of 0.7 between the automatically generated parses of the reference correction and the LL source. This similarity is comparable to the parser's reported performance (0.73 in-domain, 0.68 out-of-domain), despite not performing any domain adaptation to LL. That is, the UCCA parses of the source and correction are roughly as similar to each other, as they are to their gold standard parse, which indicates that semantic parsing technology may already be sufficiently mature to use automatic faithfulness measures, such UCCASim. These results also suggest that an improvement in parsing performance will further improve these scores.

4.6 UCCASim's Sensitivity to Errors

Our results have hitherto shown that UCCA is insensitive to the differences between a source sentence and its valid correction. This sub-section presents an evaluation of the sensitivity of UCCASim to pro-

Source	the good student must know how to understand and work hard to get the iede .
Reference	A good student must be able to understand and work hard to get the idea .
Corrector	The good student must know how to understand and work hard to get on .

posed corrections, which diverge semantically from the source.

From a theoretical standpoint, a semantic measure is, by its definition, sensitive to variation in the semantic distinctions which it encodes. In UCCA's case, these distinctions are focused on predicate-argument structures, the inter-relations between them, as well as the semantic heads of complex arguments. These distinctions are widely accepted in NLP and in the liguistic literature, as fundamental.

Seeking empirical validation to this claim, we present an experiment which shows that corrections of a fairly low quality, indeed receive a much lower UCCASim faithfulness score. Noting that as the systems we experiment on in §2 and 3 are over-conservative, they are unlikely to produce semantically unfaithful sentences, we instead experiment on 5 partially trained correctors, provided by Sakaguchi et al. (2017), on the JFLEG corpus (Napoles et al., 2017).

UCCASIM was calculated automatically for each system's output on 754 sentences. We expected to have low results as indeed these outputs include major changes, sometimes deleting full phrases from the output or changing every other word in the sentence, they also receive a low score by GLUE. As a sanity check we ran UCCASIM over 4 references. The references had UCCASIM scores of 0.72-0.78, the corrector predicted (by GLUE score) to be worse got 0.19 and the rest 0.32-0.39. Showing that bad corrections, that change the meaning as represented by UCCA, are penalized. In the example **??** one could see the general trend, with the source getting a UCCASIM of 1 the reference 0.71 and the corrector 0.33. Moreover, even though the chosen reference changes more words, getting the verb structure wrong is enough for an overall lower score for the corrector's sentence.

5 Conclusion

This paper addresses the shortcomings of existing RBMs in GEC. We present evidence that state of the art correctors suffer from over-conservatism and argue that this over-conservatism results from training them against measures that not only more harshly penalize over-correction than under-correction, but also often penalize correctors for proposing perfectly valid corrections. In fact, our results indicate that systems are often more likely to be penalized for a valid correction than to receive credit for it, due to the small number of references taken into account.

Estimating the distribution of valid corrections for a sentence, we find that increasing the number of references is beneficial only up to a point, after which the heavy tail of the corrections distribution entails only minor improvements to the coverage with every increase in M.

We thus propose a measure for the semantic faithfulness of the correction to the source, thereby avoiding the pitfalls of reference-based evaluation. We argue that using reference-less measures in conjunction with reference-based measures in the training and development of GEC systems will better address the challenge of over-conservatism.

Future work will assess the relative importance, ascribed by users of GEC systems, to different evaluation criteria of the output. We believe that in terms of conservatism, end users will be tolerant to changes in the sentence structure, i.e., violation of conservatism, but much less tolerant to changes in the sentence's meaning, i.e., violation of faithfulness. A better understanding of how these interact may lead to improved semantic evaluation, that will alleviate the need for a high number of references.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *ACL (1)*, pages 228–238.

Joshua Albrecht and Rebecca Hwa. 2007. Regression for sentence-level mt evaluation with pseudo references. In *annual meeting-association for computational linguistics*, volume 45, page 296.

Joshua Albrecht and Rebecca Hwa. 2008. The role of pseudo references in MT evaluation. In *Proceedings of the Third Workshop on Statistical Machine Transla-*

tion, pages 187–190, Columbus, Ohio, June. Association for Computational Linguistics.

Rafael E Banchs, Luis F D'Haro, and Haizhou Li. 2015. Adequacy-fluency metrics: Evaluating mt in the continuous space model framework. *IEEE Transactions on Audio, Speech, and Language Processing*, 23(3):472–482.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal dependencies for learner english. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746. Association for Computational Linguistics.

Johnny Bigert, Jonas Sjöbergh, Ola Knutsson, and Magnus Sahlgren. 2005. Unsupervised evaluation of parser robustness. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 142–154. Springer.

Alexandra Birch, Omri Abend, Ondřej Bojar, and Barry Haddow. 2016. Hume: Human ucca-based evaluation of machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1264–1274.

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. Association for Computational Linguistics.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *ACL (1)*, pages 697–707.

Martin Chodorow, Markus Dickinson, Ross Israel, and Joel R Tetreault. 2012. Problems in evaluating grammatical error detection systems. In *COLING*, pages 611–628. Citeseer.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249. Association for Computational Linguistics.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62. Association for Computational Linguistics.

Bradley Efron. 1987. Better bootstrap confidence intervals. *Journal of the American statistical Association*, 82(397):171–185.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *HLT-NAACL*, pages 578–587.

Jennifer Foster. 2004. Parsing ungrammatical input: an evaluation procedure. In *LREC*. Citeseer.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for ucca. *arXiv preprint arXiv:1704.00552*.

Yili Hong. 2013. On computing the distribution function for the poisson binomial distribution. *Computational Statistics & Data Analysis*, 59:41–51.

Thorn Huebner. 1985. System and variability in interlanguage syntax. *Language Learning*, 35(2):141–163.

Ting-Hui Kao, Yu-Wei Chang, Hsun-Wen Chiu, Tzu-Hsi Yen, J Boisson, J-c Wu, and JS Chang. 2013. Conll-2013 shared task: Grammatical error correction nthu system description. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 20–25.

Chi-kiu Lo and Dekai Wu. 2011. Meant: an inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility via semantic frames. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 220–229. Association for Computational Linguistics.

Nitin Madnani, Joel Tetreault, Martin Chodorow, and Alla Rozovskaya. 2011. They can help: Using crowdsourcing to improve the evaluation of grammatical error detection systems. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 508–513. Association for Computational Linguistics.

Ryo Nagata and Keisuke Sakaguchi. 2016. Phrase structure annotation and parsing for learner english.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, volume 2, pages 588–593.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2016. There's no comparison: Reference-less evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on*

Empirical Methods in Natural Language Processing, pages 2109–2115. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. Jfleg: A fluency corpus and benchmark for grammatical error correction. *arXiv preprint arXiv:1702.04066*.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *CoNLL Shared Task*, pages 1–14.

Diane Nicholls. 2003. The cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*, volume 16, pages 572–581.

Matt Post, Chris Callison-Burch, and Miles Osborne. 2012. Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409. Association for Computational Linguistics.

Marwa Ragheb and Markus Dickinson. 2012. Defining syntax for learner language annotation. In *COLING (Posters)*, pages 965–974.

Florence Reeder. 2006. Measuring mt adequacy using latent semantic analysis. In *Proceedings of the 7th Conference of the Association for Machine Translation of the Americas. Cambridge, Massachusetts*, pages 176–184.

Alla Rozovskaya and Dan Roth. 2010. Annotating esl errors: Challenges and rewards. In *Proceedings of the NAACL HLT 2010 fifth workshop on innovative use of NLP for building educational applications*, pages 28–36. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2013. Joint learning and inference for grammatical error correction. *Urbana*, 51:61801.

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proc. of ACL*, pages 2205–2215.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2017. Grammatical error correction with neural reinforcement learning. *arXiv preprint arXiv:1707.00299*.

Matthew Snover, Nitin Madnani, Bonnie J Dorr, and Richard Schwartz. 2009. Fluency, adequacy, or hter?: exploring different human judgments with a tunable mt metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268. Association for Computational Linguistics.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Lucia Specia, Dhwaj Raj, and Marco Turchi. 2010. Machine translation evaluation versus quality estimation. *Machine translation*, 24(1):39–50.

Elior Sulem, Omri Abend, and Ari Rappoport. 2015. Conceptual annotations preserve structure across translations: A french-english case study. *Proceedings of S2MT 2015*, page 11.

Elaine Tarone. 1983. On the variability of interlanguage systems. *Applied linguistics*, 4(2):142–164.

Joel R Tetreault and Martin Chodorow. 2008. Native judgments of non-native usage: Experiments in preposition error detection. In *Proceedings of the Workshop on Human Judgements in Computational Linguistics*, pages 24–32. Association for Computational Linguistics.

Joseph P Turian, Luke Shea, and I Dan Melamed. 2006. Evaluation of machine translation and its evaluation. Technical report, DTIC Document.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727*.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 180–189. Association for Computational Linguistics.

Omar F Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1220–1229. Association for Computational Linguistics.

James Zou, Gregory Valiant, Paul Valiant, Konrad Karczewski, Siu On Chan, Kaitlin Samocha, Monkol Lek, Shamil Sunyaev, Mark Daly, and Daniel G MacArthur. 2016. Quantifying unobserved protein-coding variants in human populations provides a roadmap for large-scale sequencing projects. *Nature Communications*, 7.