Conservatism and Over-conservatism in Grammatical Error Correction

Abstract

1 Introduction

Grammatical Error Correction (GEC) is a challenging research field, which interfaces with many other application areas of linguistics. The field 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 system evaluation (Tetreault and Chodorow, 2008; Madnani et al., 2011; Chodorow et al., 2012; Dahlmeier and Ng, 2012; Felice and Briscoe, 2015; Napoles et al., 2015), a notoriously difficult topic, in part due to the many valid corrections each source sentence may have.[1]

An important criterion in the evaluation of GEC systems (henceforth, *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, often when compiling gold standard corrections for the task, annotators are instructed to be conservative in their corrections (e.g., in the Treebank of Learner English (Nicholls, 2003)). A recent attempt to formally capture this precision/recall asymmetry has been the standardized use of $F_{0.5}$ over F_1, where Precision is emphasized over Recall (Dahlmeier and Ng, 2012).

However, penalizing over-correction more harshly than under-correction may lead to reluctance of correctors to make any changes (henceforth, *over-conservatism*). Using one or two reference corrections, which is 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.

We present results that indicate that current state of the art systems do suffer from over-conservatism. Evaluating the output of all 15 systems that participated in the recent CoNLL2014 shared task, we find that all of them substantially under-predict corrections relative to the gold standard (Section 2).

We pursue two approaches to decrease the undue penalization of valid corrections. First, we study the effect of increasing the amount of references, which we denote by M (Section 3). While previous evaluation explored the case of $M = 2$, no empirical assessment of its sufficiency or its added value over $M = 1$ has been carried out. We consider two representative measures 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 findings suggest that using $M = 1$ or $M = 2$ dramatically under-estimates the true performance of the systems, and commonly leads to cases where a valid correction is more likely to be penalized than to be deemed valid. We argue that this is a likely source of the observed over-conversatism (Section 3.6).

Second, we pursue a sematnic evaluation approach, which assesses the extent to which a correction faithfully represents the semantics of the source, by measuring the similarity of their semantic struc-

[1] OA: could you find a reference that alludes to the multiple reference explicitly?

tures (Section 4). Our experiments support the feasibility of this proposed approach, by showing that (1) semantic structural annotation can be consistently applied to learner's language (LL), and (2) that the proposed measure is less prone to undue penalization of valid corrections.

The two approaches address the insufficiency of using too few references from complementary angles. The first attempts to cover more of the probability mass of valid corrections by taking a larger M, while the second uses semantic instead of string similarity, in order to abstract away from some of the formal variation between different valid corrections.

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.

Experimental Setup. Our experiments are on the NUCLE dataset, a parallel corpus of LL essays and their corrected versions, which is the de facto standard in GEC. The corpus contains 1,414 essays in LL, each of about 500 words.

We evaluate all participating systems in the CoNLL 2014 shared task, in addition to the best performing system on this dataset (Rozovskaya and Roth, 2014). 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-Comte (UFC), University of Macau (UMC), and the best performing system by Rozovskaya and Roth (2016, RoRo). 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 prevalance 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 token of their own.

Measures of Conservatism. We consider three types of divergences between the source and the reference. 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 distance between the tokens.[2] 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 by computing 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 references and by each of 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, in the reference corrections there are 44 sentences which have 5 words corrections, where the most sentences with 5 word corrections by any corrector is 11.

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

3 Multi-Reference Measures

In this section we show that the common practice of using one or two references in reference-based GEC evaluation yields a substantial under-estimation of

[2] When breaking ties, we favor alignments where the place in the sentence is closer.

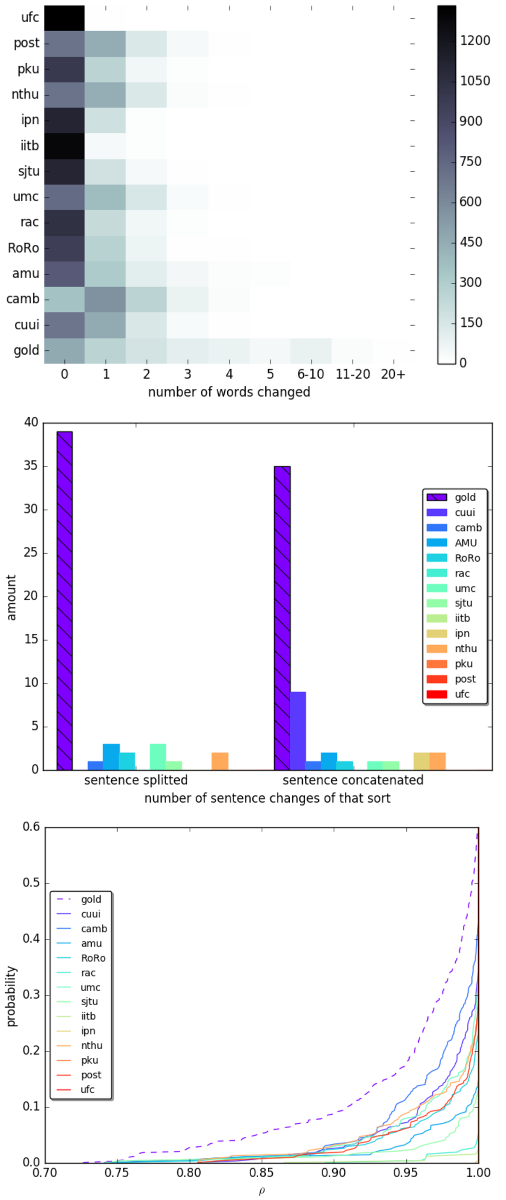

Figure 1: The prevalance of changes of different types in corrector's output and in the NUCLE references. The top figure presents the number of sentence pairs (heat) for each number of word changes (x-axis; measured by WORDCHANGE) for each of the different systems and the references (y-axis). The middle figure presents the percentage of sentence pairs (y-axis) where the Spearman ρ values do not exceed a certain threshold (x-axis). The bottom figure presents the number of source sentences (y-axis) split (right bars) or concatenated (left bars) in the references (striped column) and in the corrector's output (colored columns). See Section 2 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.

system's performance. We discuss the implications of under-estimation, specifically showing how designing correctors so as to optimize existing measures, may lead them to over-conservatism. Finally, we discuss what number of references may be more suitable for a reliable evaluation.

3.1 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 $y \in Correct_s$ 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 $x_1 \ldots x_N$, each independently sampled from some distribution $\mathcal{L}$ over LL sentences. For each x_i, denote with $\mathcal{D}_i$ its distribution of corrections. Each x_i is paired with M corrections $Y = \{y_i^1, \ldots, y_i^M\}$, which are indpendently sampled from $\mathcal{D}_i$.[3]

A corrector C is a function from LL sentences to proposed corrections (strings). A corrector's output is a set of proposed corrections $\{C(x_1), ..., C(x_N)\}$. 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.

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 52 sentences with a maximum length of 15 from the NUCLE test data. The length restriction was introduced to avoid introducing too many independent errors that may drastically increase the number of annotations (as every combination of corrections to these errors are possible), thus resulting in an unreliable estimation for $\mathcal{D}_x$. Sentences with less than 6 words (about a third of the overall source sentences) were discarded, as they were mostly a result of sentence segmentation errors.

Crowdsourcing has proven effective in GEC evaluation (Madnani et al., 2011; Napoles et al., 2015)

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

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 by Amazon Mechanical Turk workers.[4] 4 sentences did not need require any correction according to a large part of 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., 2015), 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 was originally developed for assessing how many variants a gene might have, including undiscovered ones, which is a similar estimation problem to the one tackled here. Manual tests of unseenEst with small artificially created frequencies show satisfying results.[5] [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. The estimated distributions tend to have steps, with many corrections with the same (low) frequency. 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, 8.72 corrections account for 58% of the total probability mass of the corrections, each occuring with a probability of 0.01 or higher.[7]

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 now turn to estimating the resulting bias, namely the under-estimation of reference-based similarity measures, for different values of M.

	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 with a probability of more than γ (top row), as well as their total probability mass (bottom row).

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 is the number of sentences whose corrections exactly matches one of the references (also called “Exact Match”). While this measure, in its raw form, is not commonly used as an evaluation measure in GEC, it is a basic, interpretable measure, and extentions of which have been recently proposed for GEC evaluation (Felice and Briscoe, 2015). It is also closely related to the 0-1 loss function used for training statistical GEC systems (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 proposed corrections $\{C(x_1), ..., C(x_N)\}$, we define C's accuracy to be

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

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:

$$TrueAcc\left(C\right) = P_{x\sim L}\left(C\left(x\right) \in Correct_x\right). \qquad (2)$$

The bias of $Acc\left(C; X, Y\right)$ for a sample of N sentences, each paired with M references is then[8]

[4] OA: say a bit more on the correction procedure. How long did it take? Did you apply any filtering? Anything interesting about the protocol you used?

[5] OA: do you mean you conducted these exps? or did the authors?

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

[7] OA: maybe the graph to show is the total probability mass of the k most frequent variants, where $k = 1, 2, 3, 4...$?

[8] OA: could you align it to the left?

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

It is easy to see that the bias is not affected by N, only by M. As M grows, Y becomes a better approximation of $Correct_x$, and b_M tends to 0.

In order to gain insight about 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 (Equation 6) can be re-written as

$$P(C(x) \in Correct_x) \cdot (1 - P_{\mathcal{D}_x}(y \in Y)). \qquad (7)$$

We will from now on assume that C is perfect (i.e., its true accuracy $Pr(C(x) \in Correct_x$ is 1), and denote its bias with b_M. It is easy to see that assuming any other value for C's true accuracy would simply scale the bias by that accuracy. Similarly, assuming only a percentage p of the sentences require correction will scale the bias by p.

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

$$\hat{b}_M = 1 - \frac{1}{N}\sum_{i=1}^{N} P_{Y\sim\mathcal{D}_i^M, y\sim\mathcal{D}_i}\left(y \in Y\right). \qquad (8)$$

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 computation 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.[9]

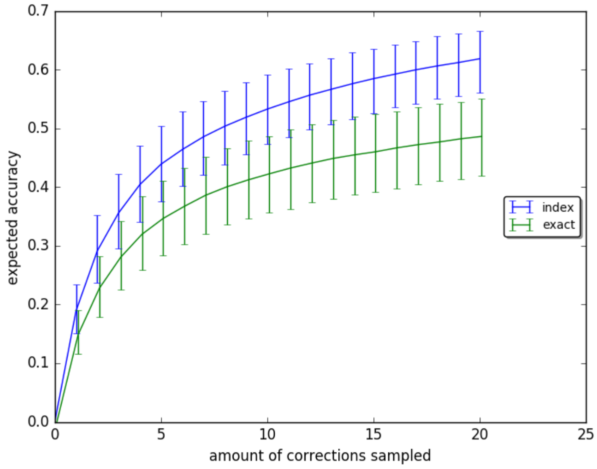

Figure 2: Mean Accuracy and Exact Index Match values for a perfect corrector (y-axis) as a function of the number of references M (x-axis). Each data point is paired with a confidence interval ($p = .95$).

Figure 2 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 of about 0.5.

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 proposed corrections c and c' over a source sentence x match if their word alignments with the source (computed as above) $a : \{1, ..., |x|\} \rightarrow \{1, ..., |c|, Null\}$ and $a' : \{1, ..., |x|\} \rightarrow \{1, ..., |c'|, Null\}$, it holds that $c_{a(i)} \neq x_i$ iff $c'_{a'(i)} \neq x_i$ (y_{NULL} and y'_{NULL} are empty strings). Figure 2 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 $M = 20$ it is no more than 0.65.

Finally, Figures 2 include confidence intervals ($p = .95$), which are computed analytically. Specifically, we note that the accuracy measure is a Poisson Binominal distribution (i.e., a sum of independent Bernoulli variables with different success probabil-

[9] OA: how much faster?

ities), whose success probabilities are $P_{y,Y\sim\mathcal{D}_i}(y \in Y)$, which we compute, as before, using UNSEENEST's estimate for $\mathcal{D}_i$. We use the method of Hong (2013) to efficiently compute the intervals.

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.

Computing F-score for GEC is not at all straightforward. The score is computed in terms of *edit* matches between the correction and the reference, where edits are sub-strings of the source that are replaced in the correction/reference. Since correctors do not normally produce edits, F-score is defined optimistically, maximizing over all possible ways to annotate the source with edits, so as to end up with the correction.[10] The resulting optimization problem is NP-hard, but designated scorers have been developed to estimate it, notably the M^2 scorer (Dahlmeier and Ng, 2012).

The complexity of the measure prohibits an analytic analysis, 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 samples its corrections from 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). In order to obtain comparable results to those obtained when experimenting on the NUCLE corpus (see below), we take $N = 1312$, as in NUCLE's test set, and further assume that 136 of these sentences require no correction. These sentences reduce the overall bias by their frequency in the corpus (in our case by 10.4%), and are thus important to include for obtaining comparable 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.[11]

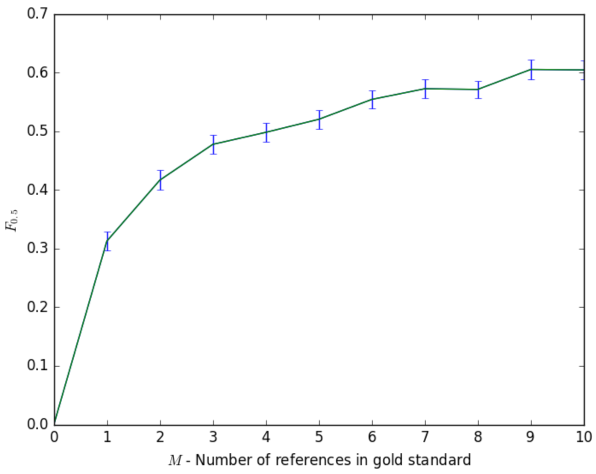

Figure 3: Mean $F_{0.5}$ values for a perfect corrector (y-axis) as a function of the number of references M (x-axis). Each data point is paired with a confidence interval ($p = .95$).

Figure 3 presents the results of this procedure, which provide further indication for the insufficiency of M values of 1 or 2 in 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 $M = 2$.

While out experiments focus on the accuracy and F-score measures, but we expect that our results would generalize to other reference-based measures, such as the I-measure () and GLEU (), which are adaptations of the F-score and accuracy measures, respectively, that address orthogonal shortcomings from the ones presented here.[12]

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 corrector's performance reported in the literature. We report results with the bootstrapping protocol (Section 3.4) to compute the confidence interval of different correctors with the current NUCLE test data ($M = 2$).

Figure 4 shows our results, which present a mixed picture: some of the differences between previously

[10] Since the corrections in our experimental corpus do not include an explicit annotation of edits, we produce edits heuristically.

[11] We use the standard Python scikits.bootstrap implementation of this method.

[12] OA: please verify!

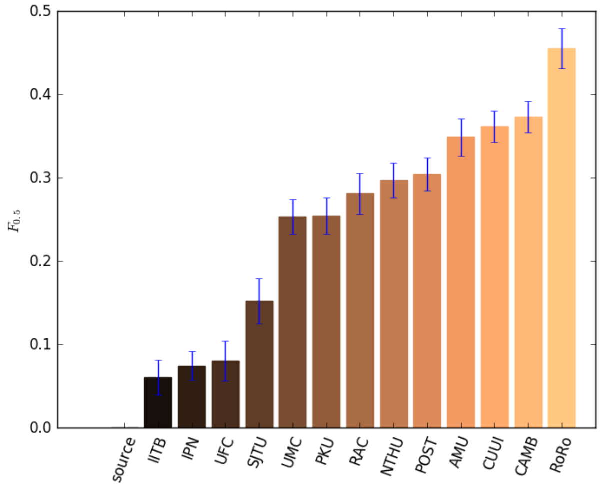

Figure 4: $F_{0.5}$ values for different correctors, including confidence interval ($p = .95$). The left-most column ("source") presents the F-score of a corrector that doesn't make any changes to the source sentences. See Section 2 for a legend of the correctors.

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 a reliable reference-based estimation is considerably greater than was previously expected. On average, two references cover only about a quarter of the probability mass of valid corrections (Section 3.4). While taking a large enough value of N may yield statistically significant results for comparing the different systems (Section 3.5), low values of M, and the low coverage they entail, lead to other problems that cannot be mended in the same way.

Importantly, low coverage of the reference set, in conjunction with asymmetric evaluation measures that penalize over-correction more severely than under-correction, result in evaluation measures that incentivize correctors not to correct, even if they are able to predict a valid correction with certainty. Assume that for a given source sentence x, references $Y \sim \mathcal{D}_x$, and proposed correction c, a given evaluation measure increases by α if $c \in Y$, and decreases by $\beta > \alpha$ if $c \notin Y$. Assume that the corrector is perfect, i.e., it invariably produces $y \in Correct_x$. Still, if the covered probability mass $P(y \in Y)$ for Y of size M and y sampled from $\mathcal{D}_x$ is less than 0.5, the expected utility of the corrector from producing y is $P(y \in Y) \cdot \alpha - P(y \notin Y) \cdot \beta < 0$. For a non-perfect corrector, the utility of predicting a correction is further decreased, due to the probability of producing $y \notin Correct_x$.

As we have seen above, the expected coverage (which is equal to the accuracy; see Figure 2) is often lower than 0.5 for $M = 2$, which suggests that such incentivization for over-conservatism is likely to be pervasive. Considering the F-score of the best performing systems in Figure 4, 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 surpass a perfect corrector's F-score. While it is possible that such systems 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 in Section 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.

Finally, we turn to discussing what values of M yield a reliable estimate of the performance of GEC systems. While there is no one answer to this question, and coverage can be low even with M values of 20, both measures we considered begin to plateau (in the case of F-score) or present linear increases after an initial concave behavior (in the case of accuracy) at M values of 8–10. This may indicate that such values are a reasonable balance between cost and bias.

4 Semantic Faithfulness Measure

In this section we explore a semantic approach to overcoming the bias introduced by the multitude of different valid corrections for an LL sentence. The measure is based on the semantic similarity of the source and the proposed correction, measured as graph similarity between their semantic representations. We note that such a measure needs to be complemented with an additional measure of fluency (cf.

(Sakaguchi et al., 2016)), as it only captures the faithfulness dimension, namely the extent to which the meaning of the source is preserved in the correction, and not the correction's grammaticality.[13]

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

We conduct two experiments that support the feasibility of our approach. First, we show that semantic annotation can be consistently applied to LL, through inter-annotator agreement (IAA) experiments. Second, we show 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 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.

Existing approaches to syntactic representation of LL in NLP differ in how they annotate syntactic errors. Berzak et al. (2016) and Ragheb and Dickinson (2012) annotate syntactic structures 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 have considerably different structures. Nagata and Sakaguchi (2016) take an opposite approach, and attempt to be faithful to the syntax intended by the learner. 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 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 a semantic annotation scheme, influenced by typological and cognitive linguistic theories. The scheme aims to provide a coarse-grained, cross-linguistically applicable representation by directly reflecting the major semantic phenomena represented in the text and abstracting away from specific syntactic forms.

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 terminals 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 manner or temporal descriptions.

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) has found that UCCA structures are preserved remarkably well across translations.

We use UCCA's standard annotation guidelines, without introducing any adaptations.

4.2 Experimental Setup

We employ two annotators, trained by annotating both LL and standard English passages, until a high enough agreement was reached (6 hours of training in total). Training passages were excluded from the evaluation.

We experiment on 7 essays and their corrections from NUCLE, each of about 500 tokens. In order to measure IAA, we assigned 4 of these essays to both annotators and compute their agreement (Section **??**). In order to measure the faithfulness score for a perfect corrector, we annotate both the source and corrected version for 6 essays, some of which

[13] LC: The reduction to semantic and error detection is another related way to convert this into an evaluation measure, one without references

were annotated by both annotators.

4.3 Semantic Similarity Measures

Inter-annotator Agreement 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 $|E_2|$. Their *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. [14]

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 Section 2. Denote the results leaf alignment with $A_l \subset Leaves_s \times Leaves_c$.

We cast the problem of finding a node alignment as a maximum weighted bipartite matching problem between the set of nodes V_s and the set V_c, where the weight of an edge $(u, v) \in V_s \times V_c$ is defined to be $w(u, v) = \frac{|A_l \cap (yield(u) \times yield(v))|}{|yield(u)|}$. A is defined to be the maximum weighted matching, excluding edges with weight 0.[15] The resulting F-score measure, using this computation of A is called UCCASIM.

We note that UCCASIM is somewhat more relaxed than DAG F-score defined above, as it aligns two nodes not only if their yields are in perfect alignment with one another, but it may also match nodes whose yields are in partial alignment with one another. This relaxation is necessary given that a correction often adds or removes nodes, thus eliminating the possibility of a perfect alignment between the sentences.

While generalizing DAG F-score, this approach has the shortcoming of being asymmetrical. We therefore report both the resulting F-score when aligning from the source to the correction and the other way around.[16]

For completeness, we also replicated the protocol of Sulem et al. (2015), which we call DISTSIM. For a given UCCA label l, denote $c_i(l)$ the number of l-labled UCCA nodes in the i-th source sentence, and $d_i(l)$ the number of l-labled UCCA nodes 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 annotated 3 corrected NUCLE passages, obtaining a similar agreement rate.

These results suggest that annotating LL with UCCA does not lead to any degradation of IAA, and can be applied as consistently to LL text as to non-LL texts.

Table 2 on its left presents the UCCASIM scores obtained by comparing the NUCLE gold standard and the source sentences, or equivalently the UCCASIM score of a perfect corrector. In order to control for differences between the annotators, we explore both a setting where the two annotations were annotated by the same annotator, and settings where they were annotated by different annotators. In order to obtain an upper bound on the UCCASIM of a perfect corrector, the table also presents UCCASIM scores between source sentences annotated by different annotators.

[14] The use of graph kernels as a similarity measure between UCCA structures is unsuitable here due to the small size of a UCCA DAG (Kashima et al., 2003).

[15] OA: why do we use many-to-1? **this doesn't make sense because it favors nodes up the tree. better to normalize**

[16] OA: this paragraph will not be true if we do 1-1 alignment

Our results indicate that a perfect corrector receives a faithfulness score comparable to the IAA, which is an indication that UCCASim is indeed insensitive to the surface divergence between a source sentence and its valid correction. However, more work is required to establish whether the converse holds, namely that the measure is sensitive to unfaithfulness of a proposed correction. This, of course, depends on the scope of distinctions covered by the semantic annotation. In UCCA's case, it is argument structure, the connections between scenes etc.

Our attempts to conduct experiments to determine UCCASim's sensitivity to erroneous corrections introduced by the correctors described in Section 2 were unsuccessful, as there are only very few cases of structural (rather than word-level) corrections which these systems make (see Figure 1). We expect that once the over-conservatism of existing correctors is resolved, such an evaluation will be made possible.

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: Similarity measures for different mapping, their average and token analysis. For UCCASim the baseline is the score over two LL annotations and for DistSim the translation between English and French. For mapping the baseline one annotator was consistently considered the reference and the other the source. The correction similarity introduces less noise than the different annotators and annotating LL changes even less than translating does.

[17]

5 Conclusion

We presented results that indicate that state of the art systems in GEC suffer from over-conservatism. We hypothesize that this over-conservatism results from the use of scoring functions that not only tend to more severely penalize introducing grammatical errors, than to overlooking existing ones, but also to often penalize systems for generating valid corrections, if these are not included in the set of references. In fact, in many cases systems are more likely to be penalized for the generation of a valid correction than to receive credit for it, due to the small number of references taken into account.

We explored two approaches for tackling this caveat in current measures, one by increasing the number of references, and the other by employing a semantic similarity measure for estimating the semantic faithfulness of the correction to the source. We compute the scores a perfect corrector would receive with both approaches, showing that both increasing the number of references and using semantics are less prone to severely penalize valid corrections, and thus constitute valid solutions for addressing this caveat.

In many of the uses for correction, the user does know his grammar is not perfect and would accept a change in grammar when needed. Because of this approval we also hypothesis, and it may call for a user study to prove or disprove this hypothesis, that users might accept a correct text unit of theirs being corrected to another correct text unit with the same meaning. This would be an example of being faithful but not conservative. Maybe even more importantly, we aim to have as many correct sentences as possible, but as neither the grammar is fully correct in the first place, nor is the user's understanding of it, failing to correct grammar is acceptable. Changing meaning will be totally unacceptable, and also surely detectable by the user. In other words, the users do expect the corrector to be active and not too conservative, but only as long as it is faithful.

Moreover, as correctors are based on statistics, they might even just correct to a more common way of saying the same thing. Such unnecessary correction is not conservative, and at GEC maybe be unwanted, but not strictly unwanted as overall it is still faithful. Additionally, some may even consider

[17] OA: fix caption

such correction a needed one because it has a better grammar considering Fuzzy Grammar(Lakoff, 1973; Madnani et al., 2011) or a more fluent way to say the exact same thing. The latter was recently suggested as a necessary shift in the goals of GEC(Sakaguchi et al., 2016). Considering all this, we propose that next generation correctors and evaluation will be focused on faithfulness when possible rather than on conservatism. Of course that with the evaluation comes the development and it all suggests that it might be beneficial to incorporate faithfulness not only for assessment but also as a feature for correctors.

References

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

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. *arXiv preprint arXiv:1605.04278*.

Alexandra Birch, Omri Abend, Ondrej Bojar, and Barry Haddow. 2016. Hume: Human ucca-based evaluation of machine translation. *arXiv preprint arXiv:1607.00030*.

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.

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.

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.

Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. 2003. Marginalized kernels between labeled graphs. In *ICML*, volume 3, pages 321–328.

George Lakoff. 1973. Fuzzy grammar and the performance/competence terminology game. In *Papers from the Ninth Regional Meeting of the Chicago Linguistic Society*.

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.

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.

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

Alla Rozovskaya and Dan Roth. 2014. Building a state-of-the-art grammatical error correction system. *Transactions of the Association for Computational Linguistics*, 2:419–434.

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.

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.

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, Mokol Lek, Shamil Sunyaev, Mark Daly, Daniel MacArthur, et al. 2015. Quantifying the unobserved protein-coding variants in human populations provides a roadmap for large-scale sequencing projects. *bioRxiv*, page 030841.