# Correction Detection and Error Type Selection as an ESL Educational Aid

**Ben Swanson**
Brown University
chonger@cs.brown.edu

**Elif Yamangil**
Harvard University
elif@eecs.harvard.edu

## Abstract

We present a classifier that discriminates between types of corrections made by teachers of English in student essays. We define a set of linguistically motivated feature templates for a log-linear classification model, train this classifier on sentence pairs extracted from the Cambridge Learner Corpus, and achieve 89% accuracy improving upon a 33% baseline. Furthermore, we incorporate our classifier into a novel application that takes as input a set of corrected essays that have been sentence aligned with their originals and outputs the individual corrections classified by error type. We report the F-Score of our implementation on this task.

## 1 Introduction

In a typical foreign language education classroom setting, teachers are presented with student essays that are often fraught with errors. These errors can be grammatical, semantic, stylistic, simple spelling errors, etc. One task of the teacher is to isolate these errors and provide feedback to the student with corrections. In this body of work, we address the possibility of augmenting this process with NLP tools and techniques, in the spirit of Computer Assisted Language Learning (CALL).

We propose a step-wise approach in which a teacher first corrects an essay and then a computer program aligns their output with the original text and separates and classifies independent edits. With the program's analysis the teacher would be provided accurate information that could be used in effective lesson planning tailored to the students' strengths and weaknesses.

This suggests a novel NLP task with two components: The first isolates individual corrections made by the teacher, and the second classifies these corrections into error types that the teacher would find useful. A suitable corpus for developing this program is the Cambridge Learner Corpus (CLC) (Yannakoudakis et al., 2011). The CLC contains approximately 1200 essays with error corrections annotated in XML within sentences. Furthermore, these corrections are tagged with linguistically motivated error type codes.

To the best of our knowledge our proposed task is unexplored in previous work. However, there is a significant amount of related work in automated grammatical error correction (Fitzgerald et al., 2009; Gamon, 2011; West et al., 2011). The Helping Our Own (HOO) shared task (Dale and Kilgarriff, 2010) also explores this issue, with Rozovskaya et al. (2011) as the best performing system to date. While often addressing the problem of error type selection directly, previous work has dealt with the more obviously useful task of end to end error detection and correction. As such, their classification systems are crippled by poor recall of errors as well as the lack of information from the corrected sentence and yield very low accuracies for error detection and type selection, e.g. Gamon (2011).

Our task is fundamentally different as we assume the presence of both the original and corrected text. While the utility of such a system is not as obvious as full error correction, we note two possible applications of our technique. The first, mentioned

*2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 357–361,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics

above, is as an analytical tool for language teachers. The second is as a complementary tool for automated error correction systems themselves. Just as tools such as BLAST (Stymne, 2011) are useful in the development of machine translation systems, our system can produce accurate summaries of the corrections made by automated systems even if the systems themselves do not involve such fine grained error type analysis.

In the following, we describe our experimental methodology (Section 2) and then discuss the feature set we employ for classification (Section 3) and its performance. Next, we outline our application (Section 4), its heuristic correction detection strategy and empirical evaluation. We finish by discussing the implications for real world systems (Section 5) and avenues for improvement.

## 2 Methodology

Sentences in the CLC contain one or more error corrections, each of which is labeled with one of 75 error types (Nicholls, 2003). Error types include countability errors, verb tense errors, word order errors, etc. and are often predicated on the part of speech involved. For example, the category AG (agreement) is augmented to form AGN (agreement of a noun) to tag an error such as "here are some of my *opinion*". For ease of analysis and due to the high accuracy of state-of-the-art POS tagging, in addition to the full 75 class problem we also perform experiments using a compressed set of 15 classes. This compressed set removes the part of speech components of the error types as shown in Figure 1.

We create a dataset of corrections from the CLC by extracting sentence pairs $(x, y)$ where $x$ is the original (student's) sentence and $y$ is its corrected form by the teacher. We create multiple instances out of sentence pairs that contain multiple corrections. For example, consider the sentence "With this letter I would ask you if you wuld change it". This consists of two errors: "ask" should be replaced with "like to ask" and "wuld" is misspelled. These are marked separately in the CLC, and imply the corrected sentence "With this letter I would *like to ask* you if you *would* change it". Here we extract two instances consisting of "With this letter I would *ask* you if you *would* change it" and "With this letter I would *like to ask* if you *wuld* change it", each paired with the fully corrected sentence. As each correction in the CLC is tagged with an error type $t$, we then form a dataset of triples $(x, y, t)$. This yields 45080 such instances. We use these data in cross-validation experiments with the feature based MaxEnt classifier in the Mallet (McCallum, 2002) software package.

## 3 Feature Set

We use the minimum unweighted edit distance path between $x$ and $y$ as a source of features. The edit distance operations that compose the path are Delete, Insert, Substitute, and Equal. To illustrate, the operations we would get from the sentences above would be (Insert, "like"), (Insert, "to"), (Substitute, "wuld", "would"), and (Equal, $w$, $w$) for all other words $w$.

Our feature set consists of three main categories and a global category (See Figure 2). For each edit distance operation other than Equal we use an indicator feature, as well as word+operation indicators, for example "the word $w$ was inserted" or "the word $w_1$ was substituted with $w_2$". The *POS Context* features encode the part of speech context of the edit, recording the parts of speech immediately preceding and following the edit in the corrected sentence. For all POS based features we use only tags from the corrected sentence $y$, as our tags are obtained automatically.

For a substitution of $w_2$ for $w_1$ we use several targeted features. Many of these are self explanatory and can be calculated easily without outside libraries. The *In Dictionary?* feature is indexed by two binary values corresponding to the presence of the words in the WordNet dictionary. For the *Same Stem?* feature we use the stemmer provided in the freely downloadable JWI (Java Wordnet Interface) library. If the two words have the same stem then we also trigger the *Suffixes* feature, which is indexed by the two suffix strings after the stem has been removed. For global features, we record the total number of non-Equal edits as well as a feature which fires if one sentence is a word-reordering of the other.

| Description (Code) | Sample and Correction | Total # | % Accuracy |
|---|---|---|---|
| Unnecessary (U) | July is the *period of* time that suits me best<br>July is the time that suits me best | 5237 | 94.0 |
| Incorrect verb tense (TV) | She gave me autographs and *talk* really nicely.<br>She gave me autographs and *talked* really nicely. | 2752 | 85.2 |
| Countability error (C) | Please help them put away their *stuffs*.<br>Please help them put away their *stuff*. | 273 | 65.2 |
| Incorrect word order (W) | I would like to know what kind of clothes *should I* bring.<br>I would like to know what kind of clothes *I should* bring. | 1410 | 76.0 |
| Incorrect negative (X) | We recommend you *not to* go with your friends.<br>We recommend you *don't* go with your friends. | 124 | 18.5 |
| Spelling error (S) | Our music lessons are *speccial*.<br>Our music lessons are *special*. | 4429 | 90.0 |
| Wrong form used (F) | In spite of *think* I did well, I had to reapply.<br>In spite of *thinking* I did well, I had to reapply. | 2480 | 82.0 |
| Agreement error (AG) | I would like to take some *picture* of beautiful scenery.<br>I would like to take some *pictures* of beautiful scenery. | 1743 | 77.9 |
| Replace (R) | The idea *about* going to Maine is common.<br>The idea *of* going to Maine is common. | 14290 | 94.6 |
| Missing (M) | Sometimes you surprised when you check the balance.<br>Sometimes you *are* surprised when you check the balance. | 9470 | 97.6 |
| Incorrect argument structure (AS) | How much do I have to bring the money?<br>How much money do I have to bring? | 191 | 19.4 |
| Wrong Derivation (D) | The *arrive* of every student is a new chance.<br>The *arrival* of every student is a new chance. | 1643 | 58.6 |
| Wrong inflection (I) | I *enjoyded* it a lot.<br>I *enjoyed* it a lot. | 590 | 58.6 |
| Inappropriate register (L) | The *girls'd* rather play table tennis or badminton.<br>The *girls would* rather play table tennis or badminton. | 135 | 23.0 |
| Idiomatic error (ID) | The *level of life* in the USA is similar to the UK.<br>The *cost of living* in the USA is similar to the UK. | 313 | 15.7 |

Figure 1: Error types in the collapsed 15 class set.

### 3.1 Evaluation

We perform five-fold cross-validation and achieve a classification accuracy of $88.9\%$ for the 15 class problem and $83.8\%$ for the full 75 class problem. The accuracies of the most common class baselines are $33.3\%$ and $7.8\%$ respectively. The most common confusion in the 15 class case is between D (Derivation), R (Replacement) and S (Spelling). These are mainly due to context-sensitive spelling corrections falling into the Replace category or noise in the mark-up of derivation errors. For the 75 class case the most common confusion is between agreement of noun (AGN) and form of noun (FN). This is unsurprising as we do not incorporate long distance features which would encode agreement.

To check against over-fitting we performed an experiment where we take away the strongly lexicalized features (such as "word $w$ is inserted") and observed a reduction from $88.9\%$ to $82.4\%$ for 15 class classification accuracy. The lack of a dramatic reduction demonstrates the generalization power of our feature templates.

## 4 An Educational Application

As mentioned earlier, we incorporate our classifier in an educational software tool. The input to this tool is a group of aligned sentence pairs from original and teacher edited versions of a set of essays. This tool has two components devoted to (1) isolation of individual corrections in a sentence pair, and (2) classification of these corrections. This software could be easily integrated in real world curriculum as it is natural for the teacher to produce corrected versions of student essays without stopping to label and analyze distribution of correction types.

We devise a family of heuristic strategies to separate independent corrections from one another. Heuristic $h_i$ allows at most $i$ consecutive Equal edit distance operations in a single correction. This implies that $h_{n+1}$ would tend to merge more non-Equal edits than $h_n$. We experimented with $i \in \{0, 1, 2, 3, 4\}$. For comparison we also implemented

- Insert
    - Insert
    - Insert($w$)
    - POS Context
- Delete
    - Delete
    - Delete($w$)
    - POS Context
- Substitution
    - Substitution
    - Substitution($w_1$,$w_2$)
    - Character Edit Distance
    - Common Prefix Length
    - In Dictionary?
    - Previous Word
    - POS of Substitution
    - Same Stem?
    - Suffixes
- Global
    - Same Words?
    - Number Of Edits

Figure 2: List of features used in our classifier.

a heuristic $h^*$ that treats every non-Equal edit as an individual correction. This is different than $h_0$, which would merge edits that do not have an intervening Equal operation. F-scores (using 5 fold cross-validation) obtained by different heuristics are reported in Figure 3 for the 15 and 75 class problems. For these F-scores we attempt to predict both the boundaries and the labels of the corrections. The unlabeled F-score (shown as a line) evaluates the heuristic itself and provides an upper bound for the labeled F-score of the overall application. We see that the best upper bound and F-scores are achieved with heuristic $h_0$ which merges consecutive non-Equal edits.

## 5 Future Work

There are several directions in which this work could be extended. The most obvious is to replace the correction detection heuristic with a more robust algorithm. Our log-linear classifier is perhaps better suited for this task than other discriminative classifiers as it can be extended in a larger framework which maximizes the joint probability of all corrections. Our work shows that $h_0$ will provide a strong baseline for such experiments.

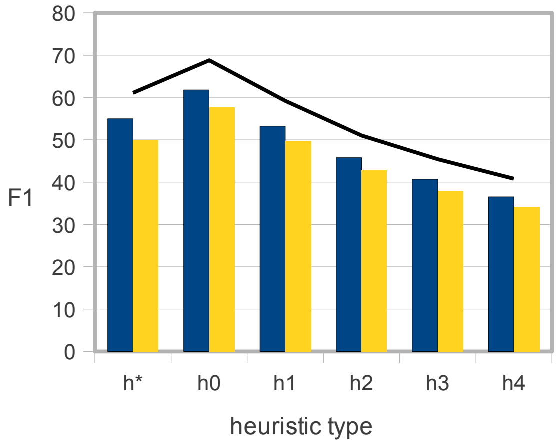

Figure 3: Application F-score against different correction detection strategies. The left and right bars show the 15 and 75 class cases respectively. The line shows the unlabeled F-score upper bound.

While our classification accuracies are quite good, error analysis reveals that we lack the ability to capture long range lexical dependencies necessary to recognize many agreement errors. Incorporating such syntactic information through the use of synchronous grammars such as those used by Yamangil and Shieber (2010) would likely lead to improved performance. Furthermore, while in this work we focus on the ESL motivation, our system could also be used to aid development of automated correction systems, as was suggested by BLAST (Stymne, 2011) for machine translation.

Finally, there would be much to be gained by testing our application in real classroom settings. Every day, teachers of English correct essays and could possibly provide us with feedback. Our main concern from such testing would be the determination of a label set which is appropriate for the teachers' concerns. We expect that the 15 class case is too coarse and the 75 class case too fine grained to provide an effective analysis.

## References

Robert Dale and Adam Kilgarriff. 2010. Helping our own: text massaging for computational linguistics as a new shared task. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG '10, pages 263–267, Stroudsburg, PA, USA. Association for Computational Linguistics.

Erin Fitzgerald, Frederick Jelinek, and Keith Hall. 2009. Integrating sentence- and word-level error identification for disfluency correction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 765–774, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Gamon. 2011. High-order sequence modeling for language learner error detection. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, IUNLPBEA '11, pages 180–189, Stroudsburg, PA, USA. Association for Computational Linguistics.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://www.cs.umass.edu/ mccallum/mallet.

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

A. Rozovskaya, M. Sammons, J. Gioja, and D. Roth. 2011. University of illinois system in hoo text correction shared task.

Sara Stymne. 2011. Blast: A tool for error analysis of machine translation output. In *ACL (System Demonstrations)*, pages 56–61.

Randy West, Y. Albert Park, and Roger Levy. 2011. Bilingual random walk models for automated grammar correction of esl author-produced text. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, IUNLPBEA '11, pages 170–179, Stroudsburg, PA, USA. Association for Computational Linguistics.

Elif Yamangil and Stuart M. Shieber. 2010. Bayesian synchronous tree-substitution grammar induction and its application to sentence compression. In *ACL*, pages 937–947.

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*, HLT '11, pages 180–189, Stroudsburg, PA, USA. Association for Computational Linguistics.