

## THE MONTCLAIR ELECTRONIC LANGUAGE LEARNER DATABASE

E. FITZPATRICK AND M.S. SEEGMILLER

*Montclair State University, Upper Montclair, NJ 07043, USA*

*E-mail: {fitzpatr/seegmill}@sapir.montclair.edu*

The work described here aims to enable more efficient research and application design in the field of second language performance. We are doing this by expanding a corpus of error-annotated written English that we have built as a feasibility study [2]. The goal is to make the resulting corpus publicly available for applications in second language pedagogy, research in second language acquisition, and the design of online writing aids for second language learners.

### 1 Introduction

Research and development in the field of natural language engineering proceeds by building models of human language performance in an effort to duplicate that performance on a machine. Over the past 10 years, the paradigm for language modeling in natural language engineering has shifted. Models based on introspectively obtained rules have given way to models based on empirically observed patterns in archived data, or corpora. This shift followed the success of the empirical approach in speech recognition [4,5] and the increase in machine storage capacity that enables large amounts of data to be maintained and manipulated.

Since most language engineering applications serve the general user, most corpora are designed to model the language of the native speaker (NS). However, more recently, corpora that model the performance of non-native speakers (NNSs) of a language have begun to appear [3]. These corpora are designed primarily to enable the study of differences between NS usage and NNS usage, with the aim of understanding more about second language acquisition, and to enable tool development (spell checkers, grammar checkers, and other writing aids) for NNSs. This paper describes a particular type of NNS corpus of formal written English being developed at Montclair State.

For some applications, primary language data is sufficient for model building, but the value of the data is greatly increased when it is annotated with linguistic information like the part of speech of the words in a sentence or the syntactic structure of the sentence. After careful hand annotation of a representative subset of the language, subsequent annotation may be done automatically [1,7]. However, since the language of NNSs often varies greatly from the standard language, automatic annotation designed for the standard language performs poorly on NNS text. In this paper we describe a project at Montclair State that is hand annotating

the errors in NNS text in such a way that the text can subsequently be submitted for conventional automatic annotation for part of speech and syntactic structure.

Montclair is particularly well-suited to carry out this project. Its Center for Language Acquisition, Instruction, and Research (CLAIR) teaches a set of languages – though primarily English – to speakers of unusually diverse native language backgrounds. CLAIR also houses master teachers of English as a Second Language and linguists to annotate the text.

## 2 The Raw Corpus

The raw corpus currently consists of formal essays written by upper level students of English as a Second Language preparing for college work in the United States. A portion of the essays are timed essays written in class; the rest are untimed drafts written at home. The corpus is small at 25,000 words, but we have recently begun collecting the data systematically which will increase its size quickly.

Essays are either submitted electronically or transcribed from hand-written submissions. A record is kept as to how each essay was submitted.

Interested student authors sign a release form that entitles us to enter their written work into the corpus throughout the semester. These students also complete a background form on native language, other languages, schooling, and extent and type of schooling in the target language, currently only English.

## 3 The annotation.

Other corpora that are annotated for error, including the Hong Kong corpus [7] and the PELCRA corpus at the University of Lodz, Poland, use a predetermined tagset to mark the errors. While this approach guarantees a high degree of tagging consistency among the annotators, it limits the errors recognized to those in the tagset. Our concern in using a tagset was that we would skew the construction of a model of L2 writing by using a list that is essentially already a model of L2 errors. The use of a tagset also introduces the possibility that annotators will misclassify. Finally, we are concerned that the 'one size fits all' approach of a tagset would force us to apply the same standards to different written genres, e.g., email or postings to listserves.

In place of a tagset, we ask annotators to minimally reconstruct the error to yield an acceptable English sentence. Each error is followed by a slash and a minimal reconstruction of the error is written within curly brackets. Missing items and items to be deleted are represented by "0". Tags and reconstructions look like this:

school systems {is/are}  
since children {0/are} usually inspired  
becoming {a/0} good citizens

Reconstruction is faster than classification, there is no chance of misclassifying, and even less common errors are captured. Additionally, syntactic parsers and part-of-speech taggers often fail with ungrammatical input. A reconstructed text can be more easily parsed and tagged for part-of-speech information.

Reconstruction, however, has its own difficulties. Without a tagset, annotators can vary greatly in what they consider an error. One recurring example of this involves the use of articles in English. For instance, the sentence *The learning process may be slower for {the/0} students as well* is correct with or without the article before students. However, the use of *the* indicates that a particular group of students had been identified earlier in the essay, whereas the absence of *the* indicates that *students* refers to students in general. An additional difficulty is that different annotators may reconstruct an error differently. For example, *the student need help* can be reconstructed as *the {student/students} need help* or *the student {need/needs} help*.

We are performing several experiments to determine how much accuracy and efficiency we could achieve in tagging errors[2]. The first experiment was a baseline test to determine if it is possible to get any sort of tagging agreement without a predetermined tagset. In a set of 1549 words, we identified 152 errors. The annotators achieved an average precision rate of .85 and a recall rate of .81.<sup>1</sup> Encouraged by these results, we tested whether we could develop annotation guidelines that would improve tagging agreement. The authors independently tagged a set of essays and compared annotations. This comparison is shown as Test One in Table 1. We then discussed our annotations, agreed on guidelines, annotated a second set of essays and compared. The comparison after discussion and guidelines is shown as Test Two.

Test	Words	Errors	Recall	Precision
One	2476	241	.73	.84
Two	2418	193	.76	.90

Table 1. Experiment with annotator guidelines after Test One.

Given these results, we are now replicating this experiment with master teachers to develop careful guidelines and a model tagged data set for graduate student annotators to follow. We anticipate that we will not achieve a higher level of agreement between the annotators tagging independently and that we will continue to need two annotators to produce reliably tagged data.

#### 4 Annotation Tools

Currently, annotators are using a simple Linux text processor of their choice to annotate the text. We anticipate that we will be able to annotate common errors like subject-verb disagreement automatically and present 'cleaner' text to the annotators who will be left to deal with the more idiosyncratic errors. We intend to automate soon for a few high frequency errors and test whether this improves inter-annotator accuracy and/or efficiency in tagging.

Other annotation projects increase efficiency by using interactive annotation tools [1], which also help accuracy by reducing some of the tedium of the task. These tools are better suited to part-of-speech and syntactic tagging where either small windows of text or partial syntactic trees are shown to the annotator. Since our annotation sometimes requires a global judgment at the paragraph level (for instance, in the case of the referent of *students* in the example given in section 3), we have not used interactive tools.

Annotators compare their tags word by word with a Linux shell script using `sdiff` that lines up the text as shown below and also counts the number of shared tags and the number of tagging discrepancies. This enables the annotators to concentrate on the discrepancies efficiently

{this/it}	{this/it}
{will:would}	will
not	not
be	be
surprising	surprising

#### 5 Accessing the Data

Each essay is stored in a separate file keyed by a unique number. Each file contains the essay including the annotations. Where the two annotators disagreed about a tag, both annotations are saved. A Linux `sed` script enables a researcher examining the essays to see either the original, unannotated essay or the text with annotations. Corpus subdirectories divide the text data by course level and particular class and further subdivide it by essay type (timed or untimed).

Background information on the author of each essay is kept in a single data file linked to the essay by the key. A menu driven by a Perl script gives the researcher access to the background information for a particular essay. We are currently writing a script that will enable the researcher to accumulate background information for a particular kind of error.

## 6 Data Processing Tools

We are currently using only Linux tools to look for patterns in the data while we build the corpus. These include searching for particular kinds of error and calculating error frequency given corpus and essay size. An issue we have yet to address regarding the corpus user is the idiosyncrasy of the tags. Currently a user cannot search for a particular error type, for example number disagreement, since the tags do not indicate the error type. For high frequency errors, we plan to convert the tags automatically to a named error type for easy search. However, for less frequently occurring errors, the user will still have to peruse a list of tagged errors. Tests of how the user searches the corpus will inform our design of a tool to display less frequently occurring error types.

We are also building a tool that will give the corpus user statistics on error occurrence, including error type plotted against background information. This is particularly useful in second language acquisition research which seeks to discriminate second language errors attributable to the native language background from errors attributable to the learning process in general, or to some universal features of language.

## 7 Applications

We plan to make the MELD corpus publicly available. The design of the corpus allows it to be used for several applications in second language pedagogy and research, as well as in the building of editing tools for second language learners. Here we give examples of possible applications of the corpus.

### 7.1 *Second Language Pedagogy*

Frequency of error by level or native language background gives a teacher information as to what writing problems s/he should concentrate on. By comparing word usage and syntactic usage against a comparable NS corpus, the teacher or textbook writer can discover gaps in the NNS use of the language and develop materials accordingly. The corpus can also be used for testing purposes since it allows testing to be targeted to specific levels and language backgrounds.

Several types of corpus-based exercises for students have been developed [6] though they are not widely available. A publicly available corpus will enable more exercises of this type. In addition, MELD's reconstruction of the text enables students to use portions of the corpus for proofreading exercise. Certain types of error can be 'turned off' so that the student sees only the type of usage s/he needs to master. The student can then compare corrections with those of the annotator.

### 7.2 *Second Language Acquisition Research*

As mentioned above, research in second language acquisition is heavily oriented to investigating the origin of errors either in the NNS's transfer of first language attributes or in use of an interlanguage that the NNS creates as ever closer approximations to the second language. The corpus will enable the researcher to statistically analyze the distribution of errors by native language background, level of study, gender, age, mastery of other languages, and spoken and written exposure to the target language.

The corpus can also be used by lexicographers to study how the NNS word usage diverges from that of native speakers.

### 7.3 *Editing Tools*

Spell checkers and grammar checkers are typically based on frequency of error distribution. They do not work well for the writing of NNSs because their errors show statistically different distributions. For example, error of complement type (*I need {of/0} somebody*) are rare in NS writing, but very common in the MELD corpus. The corpus provides the statistical base required to develop these tools.

## **8 Conclusion**

The corpus being collected and annotated should provide a wealth of empirical data to assist in second language pedagogy, research, and tool building. We plan to make the corpus and tools publicly available on the web.

## **9 Acknowledgements**

We thank the master teachers Jacqueline Cassidy, Norma Pravec, and Lenore Rosenbluth, who contributed careful labor and thoughtful discussion in providing a tagged data set and tagging guidelines and the graduate student annotators Jennifer Higgins and Donna Samko.

## **References**

1. Bredenkamp, A, B. Crysmann, and J. Klein. Annotation of error types for German news corpus. in Journées ATALA sur les Corpus Annotés pour la Syntaxe Treebanks Workshop (1999) Paris. 18-19 juin pp. 77-84.
2. Fitzpatrick, E. and Seegmiller, S. Experimenting with Error Tagging. The Second North American Symposium on Corpus Linguistics and Language

- Teaching. Northern Arizona University, Flagstaff, AZ, (March 31-April 2 2000).
3. Granger, S. (ed). *Learner English on Computer*. (1998) Addison-Wesley Longman.
  4. Jelinek, F. Self-organized language modeling for speech recognition. IBM T.J. Watson Research Center, Continuous Speech Recognition Group, Yorktown Heights, NY (1985).
  5. Jelinek, F. Markov source modeling of text generation. In *The Impact of Processing Techniques on Communications*, ed. by J.K. Skwirzinski (Nijhoff, Dordrecht, 1985).
  6. Milton, J. Exploiting L1 and interlanguage corpora in the design of an electronic language learning and production environment. In Granger, S (ed).
  7. Milton, J. and N. Chowdhury. Tagging the interlanguage of Chinese learners of English. In *Entering Text*, ed. by L. Flowerdew and A.K.K. Tong. Language Centre, The Hong Kong University of Science and Technology (1994).

---

<sup>1</sup> Precision is the measure of errors identified by both annotators divided by the errors identified by the 'non-expert'. (How many tags were correct out of all the errors s:he tagged?) Recall is the measure of errors identified by the non-expert divided by the errors identified by the expert. (Out of all the errors identified, how many did the non-expert get?) Precision and recall show the distance in tagging between the two annotators.