Error-Centric Annotation of Learner Corpora

Magdalena Leshtanska Aleksandar Dimitrov

March 30, 2009

1 Introduction

Literature overview. Bla Bla.

All of the annotation schemes mentioned above focus on the **text** itself, and will often even advise the annotator to *modify* the underlying text with the annotation. We believe that a more error-centric annotation of the data can be beneficial.

Firstly, every annotation scheme that operates directly on the text or spans over a piece of text will run into two kinds of problems:

- *Interleaving annotations* occur when an error doesn't end before another one begins. Given the tokens $a_1a_2a_3$, and two errors e_1 and e_2 ranging over tokens a_1, a_2 and a_2 a_3 respectively, the resulting markup will be confusing or outright impossible to read: $(e_1)a_1(e_2)a_2(/e_1)a_3(/e_2)$. This is particularly a problem with XML, since the specification[1]explicitly disallows interleaving markup.

- *Greedy annotation* covers tokens entirely uninvolved in the *"production"* of an error. If of the tokens $a_1a_2a_3$ only a_1 and a_3 are involved in the production of an error (say, an agreement error) marking the entire token sequence as erroneous would involve blaming the otherwise completely innocent token a_2. This can significantly increase the noise in a corpus.

There is little data about the primary use cases for learner corpora known to us, but it does not seem too far fetched to assume most linguists interested

[1]Located at `http://www.w3.org/TR/REC-xml/`

in such data are not likely to browse through the corpora in order to find an amusingly written piece of text. Instead, it seems to be safe to assume that most would be interested in the **kinds of errors** that can occur in such a corpus, and their particular properties. The error data should thus contain as little noise as possible and be maximally specific while retaining a large amount of generality. It should furthermore be easily accessible, without having to actually read the texts within the corpus itself.

Based on these assumptions, we decided to decouple the *error markup* from the *corpus data*. Specifically, to our annotation scheme, the errors and the text are two entirely different data structures. Every particular error can reference tokens within the corpus using a **key**, similar to the way modern Relational Data Bases reference their data. This makes the index more accessible and easier to maintain. Every marked up error contains exactly the amount of information it needs in order to justify its presence, viz. by defining what triggered its occurrence, and sufficient classification within the taxonomy presented in 2.1.

2 Error Taxonomy and Annotation Scheme

Learner errors are not easily classified due to their wide variety of appearance, kind, and ambiguities involved. Typically, an annotation scheme will strive to cover as large a scope it possibly can unambiguously classify, yet there are many practical limitations to the potential coverage one can achieve.

One of the most prevalent restrictions is the need to devise the annotation manual with great attention to detail, and to perform a meaningful assessment of the quality of the annotated data after the annotation process. Even a most meticulously well-defined manual is still going to be read and applied by humans, who all have their personal intuition and character. It is easy to see why this inherent individuality poses problems to the consistency of any kind of annotated data. A well-written annotation manual can limit the effect this natural variance can have on the data, but applying measures of consistency is strictly necessary. The assessment of our data will be discussed in section 4 while the remainder of this section will explain the rationale behind our annotation manual, which is in turn given in section 3.

2.1 Calling Errors by Name

In order to identify errors occurring in learner language, one has to first classify them. However, even then, identifying which class a particular error belongs to is in no way an easy job. Devising an exhaustive taxonomy of errors that can appear in natural language seems a daunting task, since science has so far failed to regularize what *is* a valid utterance of a language. Therefore, our taxonomy will try to do some things only, and do them well. In particular, we will not cover punctuation mistakes, since their possible corrections can have cascading effects on the rest of the sentences and suddenly give rise to all kinds of new errors.

Furthermore, our taxonomy will have a base case, called **gibberish** for every error that cannot otherwise be classified, necessitated by its deliberately established incompleteness. Sections delineated as such a case should, however, be few in number, and as local as possible.

There are three other kinds of errors we want to term:

1. *Spelling errors* are a primitive form of error. They contain no metadata or any kind of further specification except maybe a target word they should have ended up as.

2. *Form errors* are otherwise grammatical sentences which are, however, semantically incorrect given the current context.

3. *Grammar errors* simply do not belong to the target language because of their erroneous morphology or syntax.

2.2 Representing Errors in Annotations

3 Annotation Manual

3.1 Spelling Errors

A spelling error is an error that *cannot* be explained by mistakes in morphology, e.g. derivation or inflection. It will typically comprise only one word. If it comprises more than one word, it is seen as a *compound* spelling error, i.e. two tokens separated by a space should have been concatenated to yield only one token. Similarly, if the target hypothesis accounts for more than one word, the token should have been split up into two.

3.2 Form Errors

Form errors occur in otherwise completely grammatical sentences, which, however do not convey what was obviously intended by the author taking context into account. Lexical errors are also form errors, if the given word is grammatically valid within the sentence.

3.3 Grammatical Errors

3.4 Devising Target Hypotheses

Retain existing input by the author of the text to the greatest extent possible. The text should be only minimally altered in the target hypothesis.

4 Results, Assessment and Future Work