Error Analysis in the TREACLE project

Mick O'Donnell, Susana Murcia, Rebeca Garcia
Universidad Autónoma de Madrid
1. The TREACLE Project

- A cooperation between Universidad Autonoma de Madrid and University Politecnica de Valencia (Penny McDonald, Keith Stuart, Maria Boquera)
- Funded by Ministerio de Ciencia e Innovación 2010-2012 (FFI2009-14436/FILO)
- Goals:
  - To map at what proficiency level each grammatical structure is best taught in a Spanish context.
  - Adjust the grammar teaching syllabus at our universities in line with these results.
  - Use the error corpus as a resource for teaching examples and online exercises.
  - Automatic proficiency assessment based on structures and errors in student texts.
1.1 The Corpus

- The project involves two corpora:
  
  The **WriCLE** corpus (UAM) - Written Corpus of Learner English. 700,000 words, written by Spanish learners of English at University level. Rollinson and Mendikoetxea (2009).

  The **UPV Learner Corpus** (UPV) containing 150,000 words of shorter texts by ESP students.

- ONLY A 18,000 word subcorpus error annotated so far (28 texts)
2. Role of Error Analysis in the Project

- Error analysis is one way to explore the grammatical competence of students at each level (e.g. Dagneaux et al 1998).
- However, some students make few errors, because they avoid structures they are not sure about.
- More adventurous students take risks and thus make more errors.
- We thus take a two-pronged approach:
  - Automatic syntactic tagging of corpus to see what structures students are attempting;
  - Manual error analysis to see what they do wrong.
- Only both together give the full picture.
3. Software for Error Analysis

- We use (and develop) UAM CorpusTool, software for text annotation
- Multi-layer annotation of a corpus (e.g., we use 3 layers: Document, Grammar, and Error)
- User provides annotation schemes (tags organised into a tag heirarchy) using graphical editor.
- Some schemes built in and optionally available (Error, English Syntax, Appraisal Analysis)
3. Software for Error Analysis

• Tool provides cross-level search facilities

• Tool provides statistical reports (compare two subsets etc.)

• Available for free (Mac, PC) from: http://www.wagsoft.com/CorpusTool/
### Project: WrlcErrorV3-24-08-09

#### Layers in this project:

| Name    | Type     | Segtype | Scheme | | Name    | Type     | Segtype | Scheme | | Name    | Type     | Segtype | Scheme | | Name    | Type     | Segtype | Scheme |
|---------|----------|---------|--------| |---------|----------|---------|--------| |---------|----------|---------|--------| |---------|----------|---------|--------| |---------|----------|---------|--------|

#### Files in this project:

<table>
<thead>
<tr>
<th>Action</th>
<th>Document</th>
<th>Sentence</th>
<th>Error</th>
<th>Grammar</th>
<th>STNFD Parse</th>
<th>Files/A101-1.txt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A101-2.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A101-3.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A101-4.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A102-1.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A102-2.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A102-3.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A103-1.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A103-2.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A105-1.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A106-1.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A106-2.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A109-1.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A109-2.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A110-1.txt</td>
</tr>
<tr>
<td>Action</td>
<td>Document</td>
<td>Sentence</td>
<td>Error</td>
<td>Grammar</td>
<td>STNFD Parse</td>
<td>Files/A131-2.txt</td>
</tr>
</tbody>
</table>
3. Software for Error Analysis

- allows overlapping and embedded error segments
- allows error layer AND layers of other kinds (e.g., we have a layer for DOCUMENT and a layer for syntax)
3. Software for Error Analysis

- Slide of error scheme. Make point that coding criteria recorded here as well.

- spelling-error
  The writer has used an appropriate word, but has spelt it wrongly.

- lexical-transfer-error
  The writer has provided a word spelt similarly or identically to an L1 word, but the word does not exist in L2, or not with the intended meaning.

- lexical-error
  Errors relating to a single word, and not affecting other parts of the phrase or clause. This includes spelling errors and false friends, etc., but does not include cases where wrong inflections are used.

- other-wordchoice-error
  Cases where:
  - the word exists in L2.
  - it is open-class (not a function word)
  - it is not a false-friend/transferred word.
  - it is syntactically appropriate for the slot.
  (e.g., single noun in a single noun slot)
Since the first of January 2006, smoking in public places, such as pubs, restaurants and offices, is forbidden; this is what the new antitobacco law establishes. Smoking was the introductory of tobacco in Europe, regarding the antitobacco law, has become one of the most restrictive countries together with Ireland, Norway and Italy. This law, exaggerated for some people and fair for others, has created a very controversial debate that confronts smokers with non-smokers. In this essay, I intend to present different points of view about the new antitobacco law.

This law establishes smoking zones in pubs, restaurants etc. It limits publicity referring to tobacco and hardens the normative of smoking in public places. In addition, it attempts to improve Spanish citizens health, as it is a fact that the first cause of death in our country is tobacco. A recent study indicates that 38.5% of the population agree with this law whereas 3.11% are against it. According to this results, people should consider that 25.8% of people smoke, 26.7% have given up smoking and 47.5% do not smoke.

Non-smokers, who are in favour of the antitobacco law, support that the law is going to improve society's health and is not against nobody's rights, in fact, it protects the right to health, which is reflected in the
The new points system for driving offences will be established in Spain before the summer of 2016.

With this new system, the driving licence will consist of a number of points that can be lost for various driving offences.

I personally agree with the establishment of this new law, as I feel that this system will help reduce traffic accidents and improve road safety.
3.5 Data Representation

- XML - machine readable by other projects
- STANDOFF ANNOTATION: allows multiple annotation layers for each text file.

```xml
<document>
  <header><textfile>Files/A101-3.txt</textfile></header>
  <segments>
    <segment id="44" start="11" end="16"
             features="error;lexical-error;spelling-error"
             state="active" correction="Mayor" />
    <segment id="45" start="77" end="86"
             features="error;lexical-error;spelling-error"
             state="active" correction="vehicles" />
  </segments>
</document>

- Beginning and end of all errors is recorded (Leuven aproach just records start in general)
4. Basic Principles of Error Coding in Treacle

Basic PHILOSOPHY

• The primary criteria behind the error scheme is to allow errors to be related to the English grammar teaching syllabus (Quirk and Greenbaum model assumed)

• We are thus not interested in lexically organized "dictionaries of errors"
  • Rather, we focus on the grammatical topic in which the error would be taught.

• We also avoid connecting errors to word classes (e.g. adverb error)
  • Rather, we associate them to the grammatical unit which provides the context for the error (phrase or clause)
  • E.g., “He runs quick” is not an adverb error, but rather an error at clause level (inappropriate Adjunct filler).
• **Code the text vs. code the correction**: In coding errors, we can code in respect to:
  • what the learner actually writes, or
  • what the corrected text should be.

• For instance, if a learner writes
  
  a woman beautiful

  ...is this a noun premodifier problem? (what should have been a premodifier was placed after the noun)

  Or a postmodifier problem? (incorrect type for a postmodifier)

• In general, we follow the principle: *if there is a conflict, we code in relation to what the learner has written, not to what they should have written.*

• Rationale: there are various possible corrections to some errors, and if we code to the corrected text, the coder’s choice of correction determines the error category.
4. Basic Principles of Error Coding in Treacle

- When segmenting errors, we use **minimal segmentation** – only select as much as you need to make the correction, with the exception that you should never select parts of words.
- We don’t need to identify whole syntactic units, because the automatic syntactic analysis identifies clause and phrase boundaries.
- Examples:
  - *in the other hand*
  - These *person*
  - They *advocate immigration fully*
4. Basic Principles of Error Coding in Treacle

The Coding Criteria Document

• We maintain a coding criteria document recording all decisions we reach in coding, organised to follow the structure of the coding scheme (20 pages at present)

• Coding criteria are also recorded in the coding scheme so that criteria are visible as one tags errors.
Root

- **lexical-error...**
  Errors relating to a single word, and not affecting other parts of the phrase or clause. This includes spelling errors and false friends, etc., but does not include cases where wrong inflections are used.

- **grammar-error...**
  Errors where some grammatical rule is broken (wrong class for slot, word order, agreement problem, missing but necessary element, present but unnecessary element, etc.)

- **punctuation-error**
  Errors in the use of punctuation

- **pragmatic-error...**
  Text which is grammatically correct, but the text is in some way incoherent with the surrounding text or context of the text. For instance, a reference to a woman as "he", or a reference to a future event using past tense (Tomorrow I went to the shop.)

- **phrasing-error...**
  Where the text is grammatically correct, but not what a native would say.
  E.g., I have ten years. (I am 10 years old)
  E.g., People with a bad behaviour (people who behave badly)

- **uncodable-error**
  Use this category if you cannot decide what the writer actually intended to say.
Grammar-error

- np-error...
- adjectival-phrase-error...
- adverb-phrase-error...
- prep-phrase-error...
- vp-error...
- clause-error...
- clause-complex-error...
- special-structure-error...
- other-grammatical-error
Grammar-error

np-error \[
\text{NOM-GROUP-ERRORS-TYPE}
\]

- determiner-error...
- premodifier-error...
- head-error...
- postmodifier-error...
- np-complex-error...
- proper-name-error...
- pronoun-error...
Grammar error

determiner-order
  "money enough"

determiner-present-not-required
  "THE good intentions are not always sufficient"
  "if THE smoking is legalised"

determiner-absent-required
  "in () last 15 years"
  Worse problem is ...

determiner-choice-error
  "add FEW water"

determiner-agreement
  "THIS people"

inappropriate-pluralisation-of-determiner
  "others humans"

partitive-expression-error
  "most OF young people"

genitive-formation-error...
  Errors in making a genitive determiner
  which includes pronouns ('my book'),
  proper nouns ('John's book') and NPs
  ('the boy's book').
6. Ensuring Inter-Coder Reliability

• We have tried as far as possible to make the coding criteria clear and unambiguous.

• These criteria are available in a 20 page coding criteria document, and also within the coding program itself.

• However, to test how reliably different coders replicate the same results, 7 of us coded 6 new texts (2500 words) for errors, with no discussion between us.

• Around 500 errors in this corpus.
6. Ensuring Inter-Coder Reliability

• Software was written to compare a set of error-coded texts, and produce a “consensus coding”.
• Only includes segments identified by at least 50% of the participants (based on segment bounds only)
• Takes the most common features assigned to each segment.
6. Ensuring Inter-Coder Reliability

- The Consensus included 453 errors

<table>
<thead>
<tr>
<th></th>
<th>Mick</th>
<th>Penny</th>
<th>Susana</th>
<th>Keith</th>
<th>Ainhoa</th>
</tr>
</thead>
<tbody>
<tr>
<td>Errors recognised</td>
<td>549</td>
<td>540</td>
<td>664</td>
<td>431</td>
<td>604</td>
</tr>
<tr>
<td>Segments present in consensus regardless of coding</td>
<td>371</td>
<td>407</td>
<td>422</td>
<td>289</td>
<td>273</td>
</tr>
<tr>
<td>% of consensus segments</td>
<td>82%</td>
<td>84%</td>
<td>90%</td>
<td>63%</td>
<td>59%</td>
</tr>
<tr>
<td>Segments coded identically to consensus</td>
<td>275</td>
<td>280</td>
<td>328</td>
<td>220</td>
<td>235</td>
</tr>
<tr>
<td>% of consensus segments</td>
<td>60%</td>
<td>61%</td>
<td>72%</td>
<td>48%</td>
<td>51%</td>
</tr>
<tr>
<td>Segments not in consensus</td>
<td>179</td>
<td>133</td>
<td>242</td>
<td>142</td>
<td>331</td>
</tr>
<tr>
<td>Segments in consensus, but coded differently</td>
<td>169</td>
<td>188</td>
<td>129</td>
<td>216</td>
<td>164</td>
</tr>
<tr>
<td>Total different from consensus</td>
<td>348</td>
<td>321</td>
<td>371</td>
<td>358</td>
<td>495</td>
</tr>
<tr>
<td>% of their segments</td>
<td>63%</td>
<td>59%</td>
<td>56%</td>
<td>83%</td>
<td>82%</td>
</tr>
</tbody>
</table>
6. Ensuring Inter-Coder Reliability

- The Inter-Coder Reliability software produces a document showing each segment identified and how people coded it.

<table>
<thead>
<tr>
<th>Consensus:</th>
<th>grammar-error: vp-error: perfect-formation-error</th>
<th>gives</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Laura</td>
<td></td>
<td>is</td>
</tr>
<tr>
<td>✓ Maria</td>
<td></td>
<td>has given</td>
</tr>
<tr>
<td>✗ Ainhoa</td>
<td>grammar-error: vp-error: subject-finite-agreement</td>
<td></td>
</tr>
<tr>
<td>✗ Mick</td>
<td>grammar-error: vp-error: passive-formation-error</td>
<td>is given</td>
</tr>
<tr>
<td>✗ Susana</td>
<td>grammar-error: vp-error: modal-tense-aspect-selection-error</td>
<td></td>
</tr>
</tbody>
</table>

**Comments:** Mick: Maybe has been given.
Some Comments:

- The levels of agreement are lower than desired.
- However, this was just the first of a planned cycle of three such studies, with each one intended to reveal differences in coding practices, leading to stronger agreement.
- On the basis of this first study, the coding criteria document was revised to cover cases not covered (e.g., segmentation of punctuation marks)
- Also, on analysis of the “Consensus”, often we all agree on an error, but segmented it differently, so disagreement was not real.
- Often real ambiguity as to what the student meant, so hard to decide:
6. Some Results

- Errors coded: 1842
- 28 essays coded, containing 18,400 words
- Just started… …the following results are early, so take them with a grain of salt.
By examining the types of errors made at each proficiency level, we can determine how much teaching time to spend on each area.
7. Some Results of Our Coding

- Degree of teaching effort should relate to degree of occurrence of particular errors at the level they are at.
- Looking at graphs per proficiency level provides specific information as to what each group needs.
- Use of corpus to derive exercises or examples.
8. Future Work

• In parallel work, we are using a parsed corpus to see which syntactic structures start to be used.
• We need to combine this work with the error analysis work.
• 3 stages:
  • Don’t use the structure
  • Use the structure with errors
  • Use the structure correctly
• We are also interested in automatically assigning proficiency level based on the set of errors they make (automatic classification using a ‘bag of words’ approach where the ‘words’ are errors)