Overview of This Work

This work is a step in a long-term research program on grammatical correction of texts written by English as a Second Language writers. Earlier steps addressed both methodological and machine-learning/algorithic issues and culminated in successful programs for this domain. See Rozovskaya and Roth (NAACL’10), Rozovskaya and Roth (ACL’11), Rozovskaya and Roth (EMNLP’13), Rozovskaya et al. (CoNLL Shared Task 2013).

Common Verb Errors Made by Non-Native English Writers

• Subject-verb agreement
  “We *discusses/discuss this very time.”
• Tense
  “We *discuss/discussed this last week.”
• Form
  “They left without *discuss/discussing this with me.”

Special Challenges

• Most of the earlier work focused on article and preposition usage errors.
• Verb-related errors received very little attention in the error correction literature (though they are more common than article/preposition errors).
• The set of Verbs is not a closed class.
• Verb errors involve several grammatical phenomena.

Contributions

• We present a holistic, linguistically-motivated framework for correcting grammatical verb mistakes.
• We propose and evaluate:
  • methods of selecting verb candidates
  • an algorithm for determining verb finiteness
  • a finiteness-based verb error correction system
• We show that the specific challenges of verb error correction are better addressed by making use of the notion of verb finiteness in a linguistically-aware framework.
• We develop an annotation for a subset of the FCE dataset that specifies gold verb candidate information and verb finiteness.

Key Results

<table>
<thead>
<tr>
<th>Model</th>
<th>Avg. Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combined</td>
<td>81.39</td>
</tr>
<tr>
<td>Finiteness-based</td>
<td>87.85</td>
</tr>
</tbody>
</table>

Table 1: Verb finiteness contribution to error identification.

The Computational Model for Correcting Verb Errors

1. Candidate selection
2. Verb finiteness prediction
3. Feature generation – special features that depend on finiteness value and error type
4. Error identification
5. Error correction

The combined model is agnostic to the finiteness of the verb; the finiteness-based model uses the verb finiteness prediction made by the verb finiteness classifier.

Identifying Verbs in Noisy Learner Text: Candidate Selection Methods

Candidate selection is difficult for verbs:
• The class of verbs is open
• Learner text is noisy – leads to many POS errors

<table>
<thead>
<tr>
<th>Candidate selection methods</th>
<th>Recall on selecting erroneous verbs as candidates (%)</th>
<th>Avg. precision on error identification (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>All VNs</td>
<td>97.00</td>
<td>79.10</td>
</tr>
<tr>
<td>tokens POS tagged as verbs</td>
<td>99.06</td>
<td>86.48</td>
</tr>
<tr>
<td>tokens that are valid verb lemmas</td>
<td>99.50</td>
<td>87.85</td>
</tr>
<tr>
<td>tokens that are valid, inflected</td>
<td>96.09</td>
<td>86.81</td>
</tr>
</tbody>
</table>

Table 3: Impact of candidate selection methods on error identification performance.

Verb Finiteness

<table>
<thead>
<tr>
<th>Verb Type</th>
<th>Example</th>
<th>Agreement</th>
<th>Verb properties</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite</td>
<td>He discussed this with me.</td>
<td>3rd person, sing.</td>
<td>Past Simple</td>
<td>Infinitive</td>
</tr>
<tr>
<td></td>
<td>He discussed this with me.</td>
<td>3rd person, sing.</td>
<td>Present Simple</td>
<td>To-Infinite</td>
</tr>
<tr>
<td>Non-finite</td>
<td>He left without discussing it.</td>
<td></td>
<td>Gerund</td>
<td></td>
</tr>
<tr>
<td></td>
<td>They let him discuss this.</td>
<td></td>
<td>Infinitive</td>
<td>To-Infinite</td>
</tr>
<tr>
<td></td>
<td>To discuss this now would be ill-advised.</td>
<td></td>
<td>Infinitive</td>
<td>To-Infinite</td>
</tr>
</tbody>
</table>

Table 4: Examples of contexts that license finite and non-finite verbs.

Intuition:
Verb finiteness should benefit verb error correction, because properties associated with each type are mutually exclusive.

Machine-Learning Components: Error Identification and Error Correction

Error identification – the goal is to identify errors and predict error type. We train a 4-class machine-learning classifier that operates in the label space (Correct, Agreement, Tense, Form).

Error correction – three components – one for each type of mistake - applied to the output of the error identification model. Each component is a multiclass classifier and is run on the instances identified as errors of a particular error type.

We train all of the models with the SVM learning algorithm implemented in JLIS (Chang et al., 2010).