Teaksta

Technical comments

Heli Uibo

Teaksta – North Saami version of WERTi

- Based on WERTi core bransch
- WERTi interface (not as a Firefox plugin)
- Running on a development server gtlab.uit.no
- Requires Apache Tomcat and Java Runtime Environment
- The work started in Dec. 2012.

What is going on behind the scenes after the user has pressed the button "Go!"

1. preprocessing:
   - extracting the textual content from the webpage
   - tokenisation
   - sentence boundary detection
   - linguistic annotation
2. postprocessing:
   - Selection of the tokens that are relevant for the topic
   - Enriching the HTML code
3. Load the enhanced page to the browser

Linguistic annotation pipeline

- morphological analysis (FST)
- morphological disambiguation (CG)
- shallow syntactic parsing (CG)

Enhancement

Enriching selected spans in the original HTML code of the webpage with additional attributes.

- Separate enhancement class for each topic:
  - SubjectEnhancer
  - ObjectEnhancer
  - InfiniteVerbEnhancer
  - NounSgEnhancer

  etc.

New in Teaksta: generated distractors and answers

- Distractors are generated using the reverted FST.
- For each topic there is a specific list of forms that will be generated (the distractors should not be totally insensible).
- For the "practice" exercise for some topics even the correct answer is generated (because of parallel forms the user cannot know which of the forms was occuring in the original text).

Example: generated distractors and answers

```
<span id="WERTi-span- Ind Prt-2"
        class="wertiviewtoken  wertiviewVerbConjugation"
        lemma="leat"
        distractors="leat lean leat leahkit leame leamen
          leahkime lea "
        answer="lei leai "> lei
</span>
```

Problems

- Generation of distractors takes a lot of time (ca 2 min for webpages with a lot of text).
  - Probably can be solved by optimising the code.

ToDo

- For each topic provide a list of webpages recommended by the teacher (some grammatical categories are infrequent in e.g. newspaper texts)
  - Every teacher should have a possibility to provide her own list of webpages.
  - Perhaps save the pre-selected webpages together with their enhancement (avoid repeated runtime processing of these pages).
- Do not ignore the words with ambiguous analysis that actually are not ambiguous.
- Last but not least: speed up the system!