Friday, May 6, 2016

Getting Started

EasyRead

- External Dependencies
  - Java 7+
  - Stanford Core NLP 3.4.1
  - Java WordNet Library 1.9
- Other Projects Already under Source Control
  - Simple NLG 4.4.2
    - SimpleNLG is responsible for suggested words being converted to the proper tense or number for a certain part of speech.
  - TexHyphJ
    - If EasyReady can't find the number of syllables from Mashape, this hyphenation algorithm is used as a fallback.
  - *While these projects are already under source control, if you make changes to the project that cause them to stop functioning there will be errors*

Getting Up and Running

- Install Play and MySQL
- Create a Database called *ezread*
  - update *application.conf* with necessary path changes
    - Make Sure
      - *play.evolutions.enabled = true*
      - The file *WordsWithSyllables.sql* is there to dump in the list of words I originally worked with for this project.

Concepts

- EasyRead does passage simplification by using Stanford CoreNLP to parse text that is added to the system and checking the words in that passage against words EasyRead is already familiar with.
  - Parsing the text means adding a List of Sentences to the SimplePassage object and a list of POS objects to the words that aren't already there.
  - If the word is in the System
    - Its Part of Speech is added to the database if it's not already there and you're done with the analysis step
  - For New Words
    - NEW WORDS ARE NOT ADDED TO THE DATABASE
      - While unknown words cannot be marked as too difficult themselves, EasyRead's sentence difficulty algorithms are based on Numbers of words, syllables, characters and other measurable factors. So the addition of unknown words can still be monitored through sentence difficulty.
    - We check if the word is a more complicated form of a word that we already know using the lemmatizer in Stanford CoreNLP
      - If It is, you're done!
      - If it is not
        - We fetch the number of syllables from **Mashape Words** or the Hyphenator and use that to increment the number of syllables for the passage
        - We increment the number of words and number of characters for the passage
  - For all those words, EasyRead will try and get suggestions for it if we can say for sure that it is too difficult (i.e. it is in the database)
    - Suggestions for Words are words for which WordNet has a set of synonyms for the word as it appears in the passage OR its base form.
      - If StanfordCoreNLP indicates that the lemma is different than the text of a token, an instance of the *InflectedWordForm* (EasyRead's model — not the one included in SimpleNLG) that relates the stem (lemma) to the text as it appears.

- In the front end Suggestions are ordered by frequency
  - Frequency is determined by any class that can implement EasyRead's *PhraseValidator* Interface
    - At present, Google Books NGrams is being used
      - EasyRead requests Google's page that presents the front end for this and scrapes that page for a javascript array containing the frequencies
      - This class adds frequencies to all of the words and sets any Suggestion of lower frequency that the original as 0. (Frequency of 0 means it won't be displayed in the front end)
  - Suggestions are also algorithmically checked for difficulty through the same methods as entire Passages
    - No Suggestion that makes the entire sentence more difficult is displayed in the front end.

- EasyRead's Grade Level algorithm makes use of
  - Age of Acquisition stored in the database
    - Stored per word
  - Flesch Kincaid Readability Test
  - Automated Readability Index
  - Coleman Liau Index
  - Any of these algorithms that produce a non-negative, non-infinite result for a passage or sentence are included.
    - Coleman Liau Index is only for passages with 100 words or more
  - These algorithms are applied to an entire passage during analysis and to every sentence whenever changes are made to it
  - These Grade level determinations are used in two different ways
    - Every passage has a specific grade level that its original text is determined to be.
    - Upon analysis a version of the markup is generated for each grade level 0 — 13 (College) and the instructor can view and edit a different version at every grade.

- So you can view a 3rd Grade Passage at a Kindergarten level and see the Suggestions for that.

Known Issues

- Detecting the end of sentences properly when words or prefixes with a period in them are used
  - Ex. Mr. or Ms.
- New words that are too complex and updated instantly in the GUI, but you can't be sure that all the suggestions are there until you analyze
- Only words that we have in the database can be underlined.
  - So you can use a really difficult world and, if we don't have it in the database, it won't be underlined.
- The sentence difficulty algorithm works based on Flesch Kincaid Readability Test and the Automated Readability index, so the increase in difficulty will still be picked up then.
- When an instructor is viewing a student's answers, the entire history isn't visible, just the most recent attempt.