May 12, 2017

Final Take-Home Exam

Arabic as a Foreign Language (AFL) Tool

EL AMRI Ali

72241
ali.elamri@aui.com
shoodey.github.io

AFL TOOL

I. The objective

The main objective is to provide a design for an interface that caters to the Arabic text's readability by considering features that characterize the text often considered as shallow features in addition to making use of the best features present in both evaluated websites while trying to improve some of them using concretely suggested ideas.

II. The project

The project can be viewed and interacted with online, it includes the project itself as well as a readme (.md file) file with instruction to either view it online, or download it and use it locally.

III. Write-Up

1. General Overview

The tool is designed to leverage the best features encountered in the websites I have evaluated, it also tried to overcome some of their weaknesses and improve on them.

I have decided to go with a simplistic interface, composed of a top bar that for the moment contain links to this document (both word and pdf), in term, it should contain links to an interactive guide (much like the one I mentioned in the genetics tutor evaluation) and some information about the project itself.

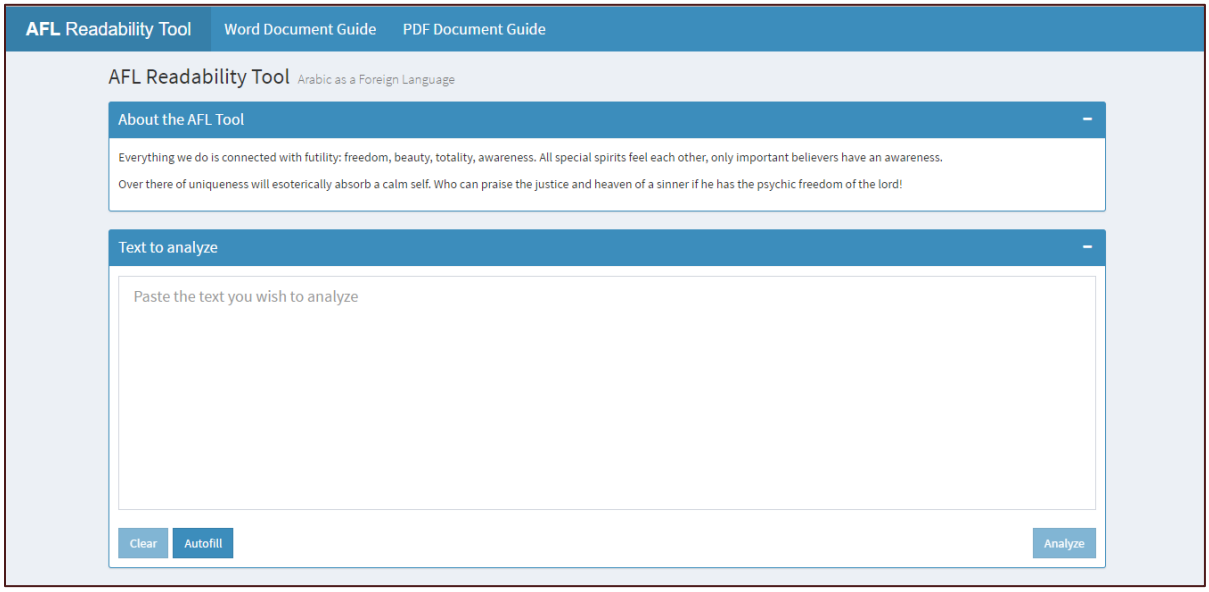

Figure 1 - Main page

2. Using the tool

2.1 Text to Analyze

The landing page contains a simple text area where to user can type or paste the text he wishes to analyze.

For testing purposes, you can press the autofill button that will fill the box with the second excerpt from The little prince translated to Arabic using google translate (Imperfect but it did the job).

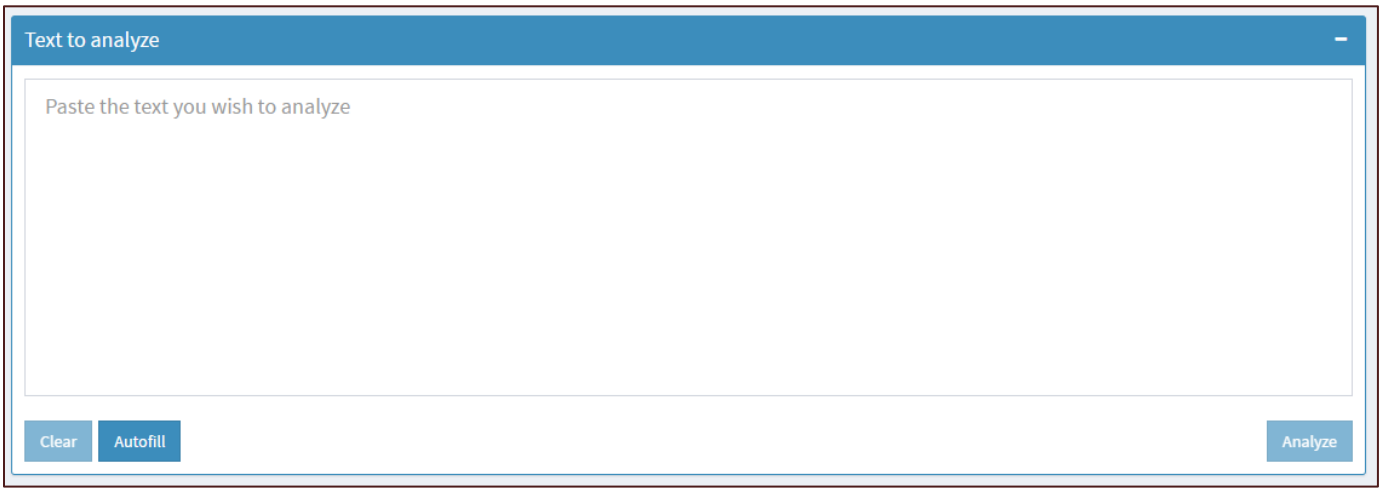

Figure 2 - Text Area

Before filling the box, both the analyze and clear button are disabled to allow for **error prevention**.

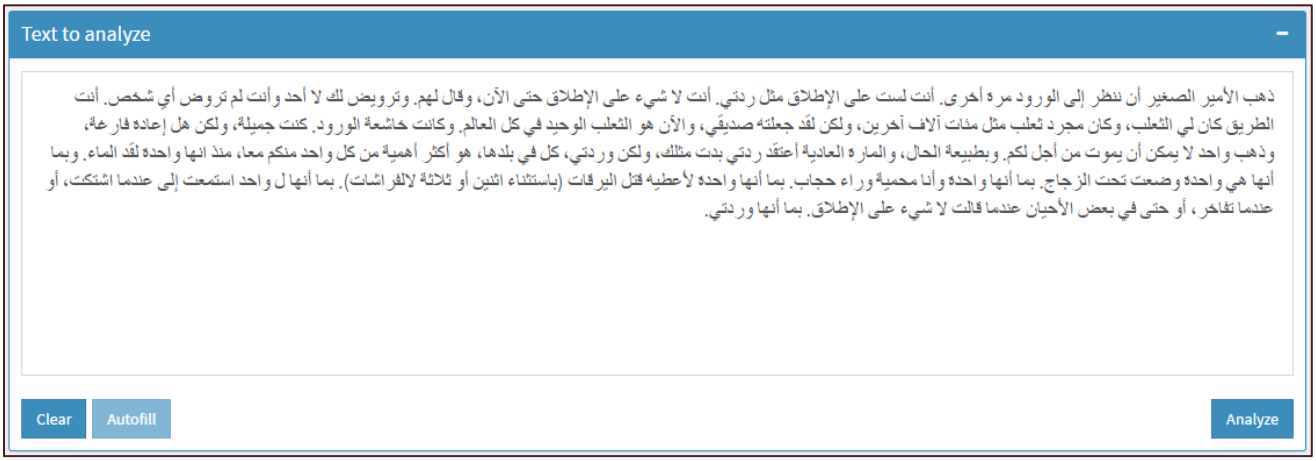

Figure 3 - Autofilled text

Autofilling has given access to the analyze button, by clicking it, the user is prompted a message giving **visibility of system status.**

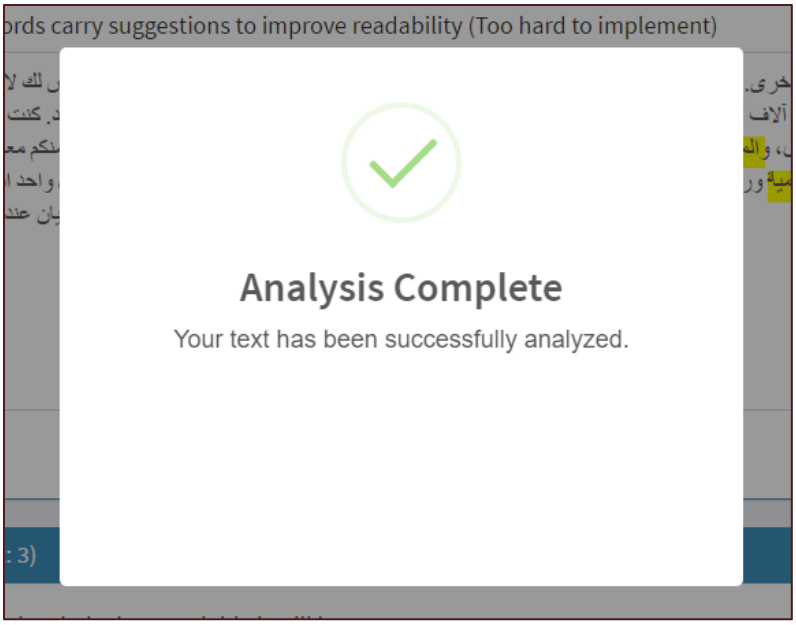

Figure 4 - Visibility of system status

2.2 Highlights and Suggestions

Once the analysis is complete, the text area now contains the following:

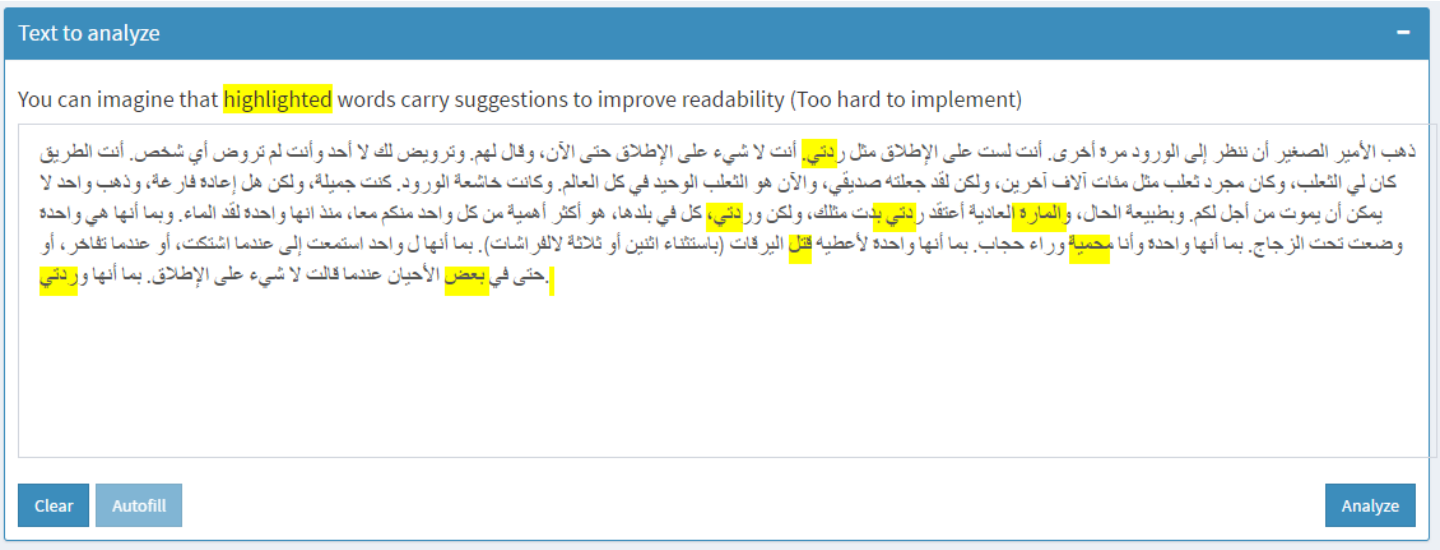

Figure 5 - Highlighted text area

Some random text is now highlighted, but we can imagine that an algorithm detects weak or too complex words that carry some suggestions to change them according to a desired level of complexity that the user can adjust.

2.3 Desired Complexity Level

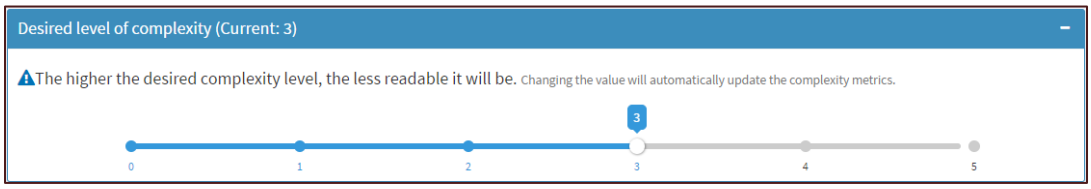

Figure 6 - Desired Complexity Level

This box shows the current complexity level (3) and a slider to change to desired value. This is an addition that I think was missing in the websites I have evaluated which enforces **user control and freedom.**

2.4 Complexity Metrics

Underneath the complexity level slider, are displayed complexity metrics which values are changed automatically when the slider's value is changed.

I have used a double value graph for each metric to emphasize on the different between the current and desired levels.

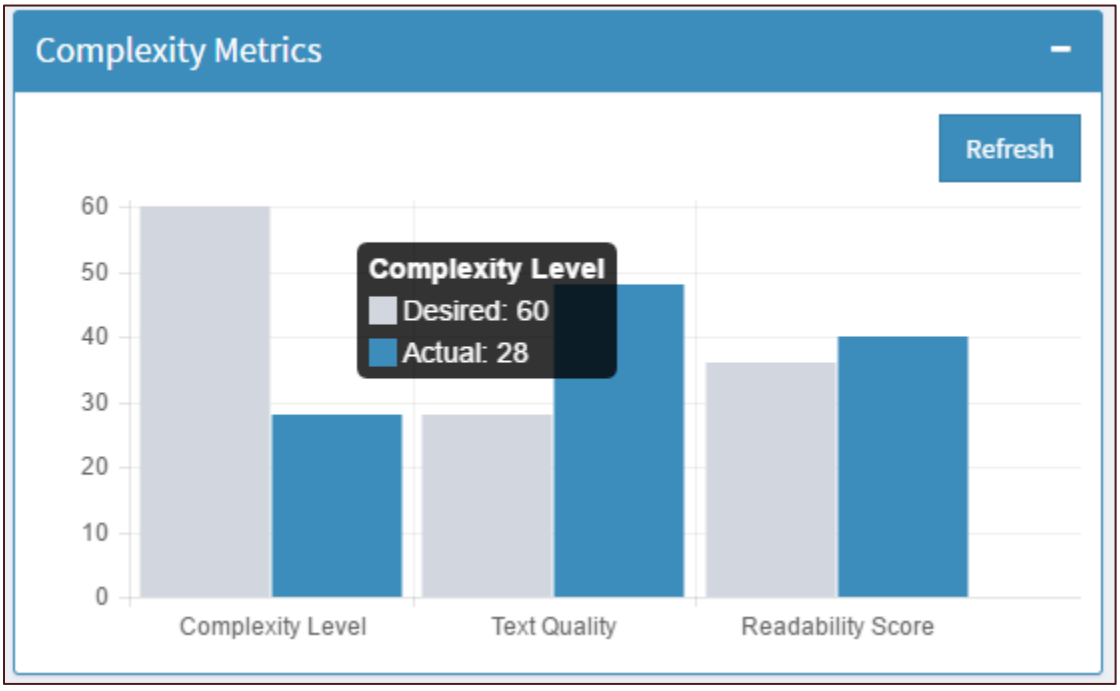

Figure 7 - Complexity Metrics

2.5 General Metrics

The general metrics section contains counts of elements contained in the text which are mentioned in the frequent readability features we should consider:

- Frequent Readability Features You Should Consider

Here is a list of "shallow features" that seem to be interesting for Arabic as a Foreign Language (AFL). Some of the terminology is explained as it occurs.

- Total **tokens** per document
  Tokens are the 'words' delimited by punctuation or whitespace. The set of 'tokens' in a document may contain many duplicates, e.g. for commonly occurring words such as 'the'.

- Total **types** per document
  Types are the unique set of words that occurs in a document. E.g., there may be many occurrences of 'the' but only one 'type'. For words that are inflected or have clitics (see next bullet item), the type represents the dictionary entry form (lemma). E.g. 'cats' and 'cat' both have the type 'cat'; 'eat', 'ate', and 'eaten' all have the type 'eat'. For Arabic, مدرستنا - مدارس - المدرسة are all considered occurrences of a single type, the lemma مدرسة.

- Total **morphemes** (coarse **clitics**) per document.
  Morphemes are the smallest unit of meaning in a word. It can be the stem or root of the word (in 'cats' that would be 'cat') or it can be an inflection (e.g., 's' meaning plural in 'cats', 'ed' indicating the past tense in a verb like 'liked'). In Arabic morphemes can be prefixes, suffixes or circumfixes (i.e., they can surround a word); e.g., the ي and the ون in يكتبون.
  Clitics are things that attach to a word (at the beginning or at the end). Some are inflectional, as in a verb prefix or suffix that tells you which person (1st, 2nd, 3rd), gender (masculine, feminine) and number (singular, plural), you are talking about. Others are particles that attach. In Arabic the pronouns of possession at the end of the noun (e.g. the ه in كتابه), conjunctions (e.g. و or ف), prepositions (e.g. ل), the future particle س, the question particle أ, etc.

- Total characters per document (excluding **diacritics**).
  Diacritics are the unwritten signs (the شكل).
- Total **open-class** tokens per document.
  Open-class words are nouns, verbs, adjectives, adverbs. They are considered " open class" because as the language changes, more can be added to reflect new objects, actions, and properties of the world and its functioning.
- Average sentence length in tokens.
- Average sentence length in morphemes (coarse clitics).
- Average sentence length in characters (without diacritics).
- Total ambiguous types per document (essentially homographic lemmas that get two or more distinct alternative morphological analyses).
  Homographic means that they are written the same way; ambiguous means that they can be interpreted differently. An example for Arabic is the word علم which, written without diacritics can be a noun or a verb, and actually have multiple meanings as a noun or as a verb.
- Total frequent types per document (refers to occurrence of words included in a pre-compiled wordlist containing frequently-used words that someone should know at a given level of language proficiency).
- Total frequency of tokens in document matching types mentioned in previous bullet.
- Total closed-class tokens per document.
  Closed-class words, in contrast to open-class words, are function words with a limited membership, including prepositions, conjunctions, pronouns, determiners, etc.

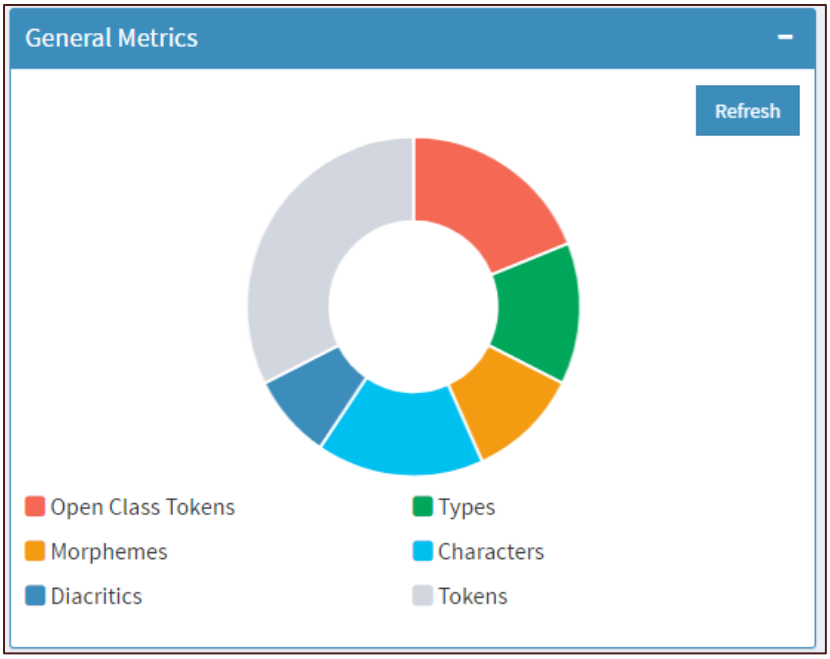

Figure 8 - General Metrics

2.6 Percentage Metrics

Other metrics worth mentioning that together form 100% of the content:

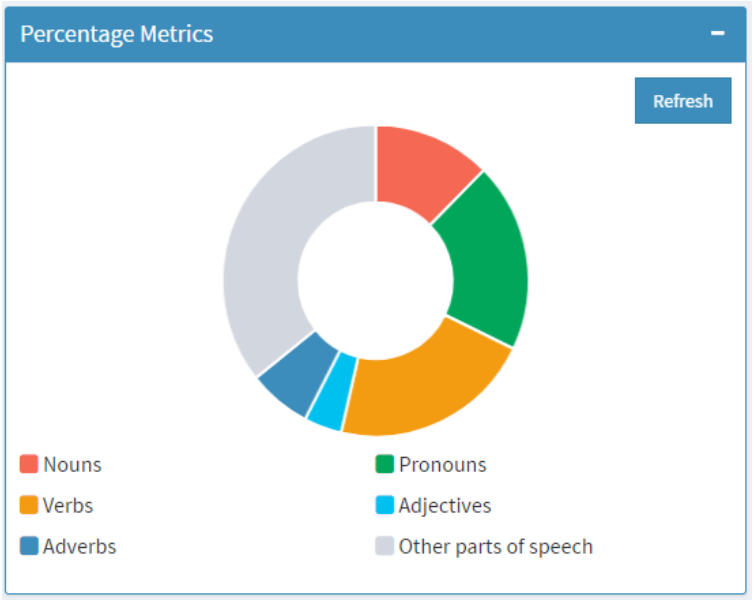

Figure 9 - Percentage Metrics

I have opted for a pie or doughnut char for both general and percentage metrics as they offer more visibility when it comes to data that together can be considered a whole. In addition, colors offer an addition visibility and clear separation between the different elements.

2.7 Other Statistics

Finally, I have summed up some worth mentioning statistics about the analyzed text such as averages and timing as mentioned in the frequent readability features we should consider.

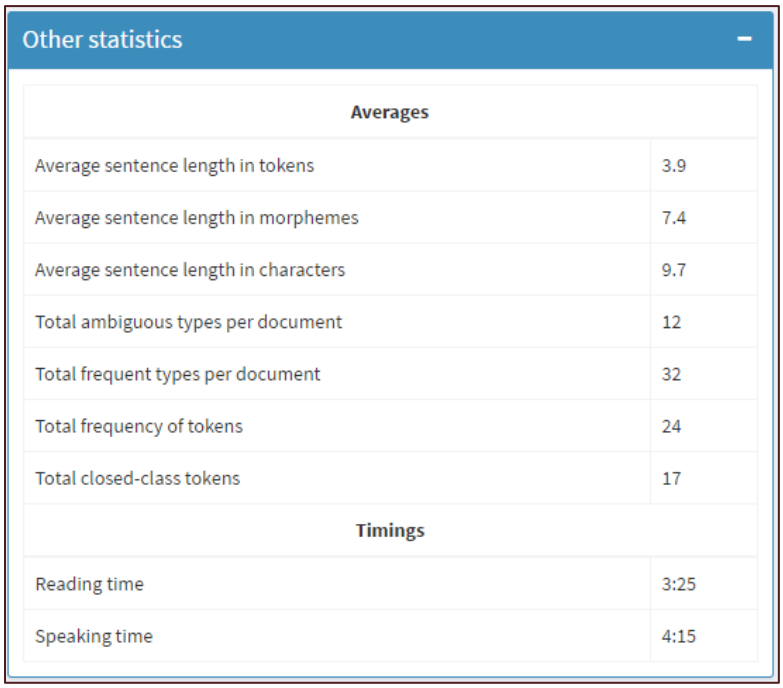

Figure 10 - Other Statistics

3. Evaluation

Applied Heuristics	Score/5	Comments
Visibility of System Status	5	Through the notification prompt
Match between system and the real world	5	By using common words and no complex formulas
User control and freedom	5	Clear button
Consistency and standards	5	Minimal interface and no ambiguity
Error prevention	5	Cannot analyze if text box is empty
Recognition rather than recall	5	Highlighted text easy to use and boxes are collapsible
Flexibility and efficiency of use	5	No reloading of page, unseen by the user
Aesthetic and minimalist design	5	Minimalist flat design
Help and documentation	No grade	Nothing yet ☹
*Required evaluation 1	5	the interface displays the characteristics of the text and makes clear the level of complexity of the text with respect to features.
*Required evaluation 2	5	Desired level of complexity adjustable.
*Required evaluation 3	5	Highlighted text and suggestions.
*Required evaluation 4	5	Could be implemented.
*Required evaluation 5	Different colors in highlighting and even underlining or using different styles	
*Required evaluation 6	We can have different styling for the different features.	
Score	Heuristics: 45/45 Required Evaluation: 20/20 (Req Ev. 5& 6 do not carry scores) Total: 100/100	

A perfect is maybe too much for an interface (somewhat) quickly put together that (for the time being) only uses static data.

The interface can be greatly improved by spending more time on the design and hooking it to some language complexity formulas and algorithms.