Text Morphological Annotation Tool

The Text Morphological Annotation Tool enables a researcher to interact with a tokenizer and a parser to produce a fully tokenized, fully-glossed and -parsed version of a text. When the user opens a plain text file, it is automatically tokenized. The user reviews the tokenization and, if it is correct, the parser is run over the entire document. A color-coded display indicates whether each token is non-parsing (it should not be submitted to the parser), unparsed (the parser did not return any potential parses), ambiguously parsing (the parser returned more than one potential parse), unambiguously parsing (the parser returned exactly one potential parse), or selected (the user has identified one potential parse as the correct one for this token). Finally, the user steps through each token with one or more potential parses and identifies the correct one.

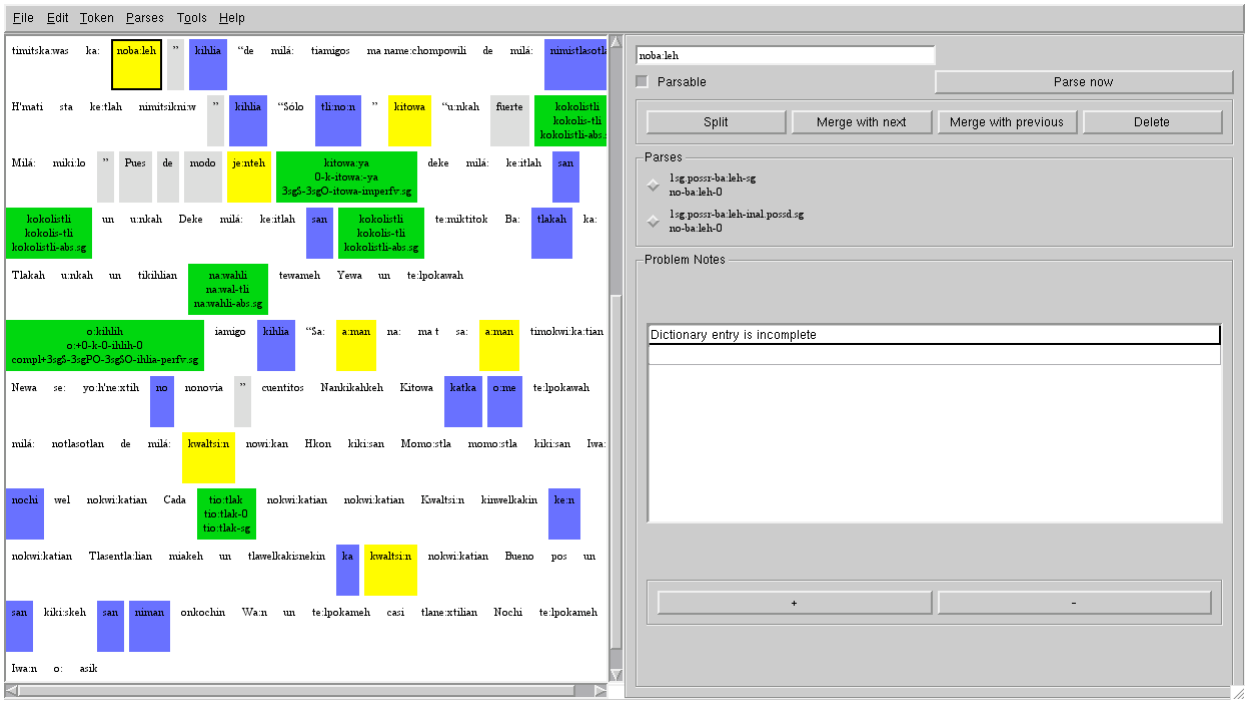
At each stage, errors can be manually corrected. The user can directly edit the text of a token, or can split or merge adjacent tokens. Once corrections have been made, the parser can be re-run over individual tokens or over the entire document. Error sources like incorrect tokenizations, parser failures, and missing dictionary entries can be noted in problem notes for each token to provide feedback to the developers of these resources.

The Text Morphological Annotation Tool is both configurable and extensible. A configuration file controlling encoding, display, and plugin settings can be exported to XML and then imported on another machine, so that a user can share settings with others. INTERLINEAR DISPLAY OF PARSE AND GLOSS INFORMATION CAN BE TOGGLED ON AND OFF. At the same time, new interfaces to tokenizers and parsers can be rapidly deployed through a plugin system. Currently, plugins exist for tools that run on a (local) command-line and tools that run on a remote server and are accessed using XML-RPC server. New plugins are simple to program and are automatically detected by the annotation tool, making it possible to quickly integrate new tokenizers and parsers into the tool.

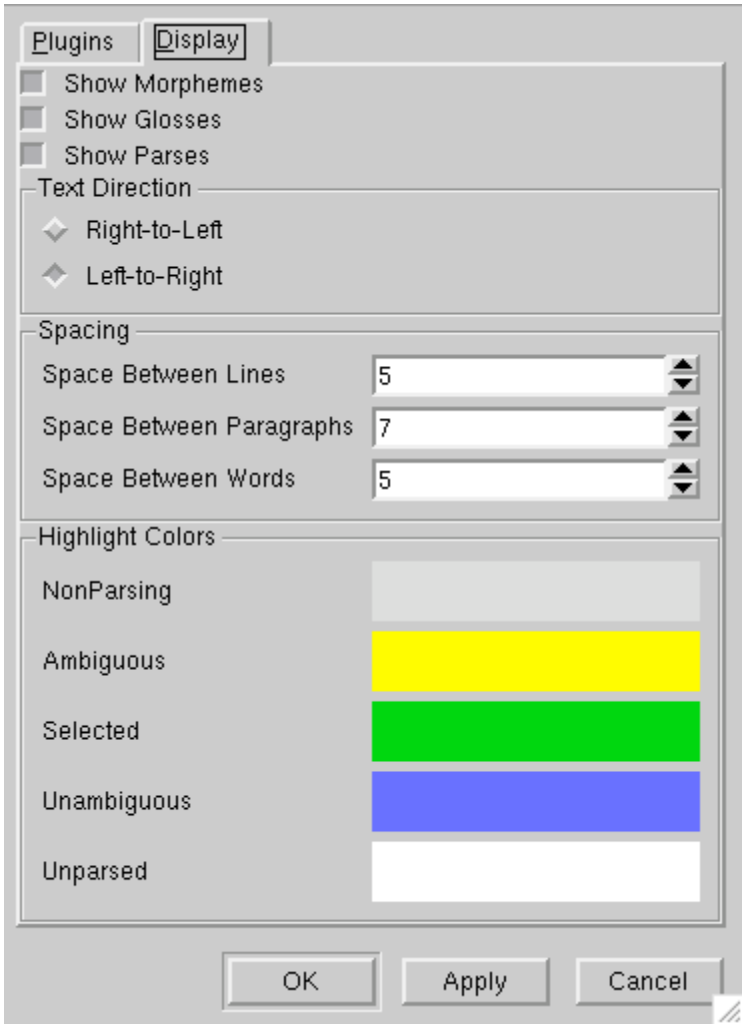
The result of processing a text with this tool is an annotation file (in XML, using annotation graph TOOLKIT, a framework for representing linguistic annotations of time series data) that can easily be displayed or printed in a three-line interlinear form: Surface representation—Parse—Gloss.

==========

Screenshot of main work area of Text Morphological Annotation Tool. In the document display on the left, each token is color-coded based on its parsing status. In the detail display on the right, a user can indicate whether a token should be submitted to the parser with the "Parsable" checkbox. The "Parse now" button re-submits the current token to the parser. The "Split", "Merge with next", "Merge with previous" and "Delete" buttons can be used to manually correct the tokenization of the document. The "Parses" list displays all potential parses returned by the parser so that the user can select the correct parse. The "Problem Notes" list allows the user to add notes to be forwarded to the developers of the dictionary, tokenizer and parser to improve
functionality.



Screenshot of the "Display" preferences dialog box. Through the first two checkboxes, interlinear display of morpheme and gloss information can be turned on and off. The "Show Parses" checkbox indicates whether tokens in the document display should be color-coded based on their parse status. The "Text Direction" section provides support for both left-to-right and right-to-left source languages. The "Spacing" section allows the user to customize the space between words, lines and paragraphs in the document display. The "Highlight Colors" section allows the user to customize the colors used to indicate parsing status in the document display.



Screenshot of the Preferences dialog for a Tokenizer plugin. Each plugin can define a list of settings that it accepts. In this case, the XMLRPC-based tokenizer plugin needs to know the location of the server, the command to run on the server, and what flags to pass to the server.