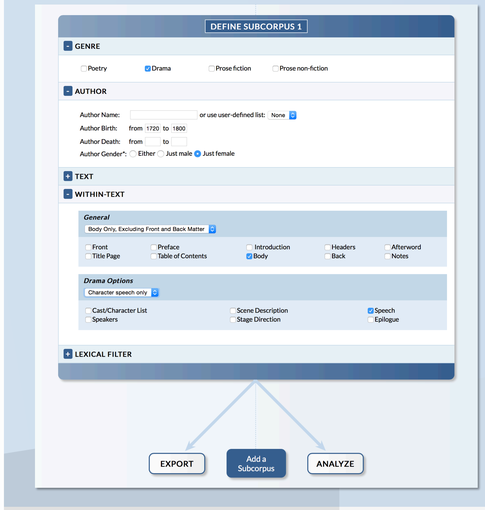
GUTENTAG

An NLP-driven Tool for Digital Humanities Research in the Project Gutenberg Corpus

Julian Brooke*
Adam Hammond©
Graeme Hirst*
UNIVERSITY OF TORONTO ©GUELPH
*UNIVERSITY OF GUELPH

Interface Detail



Definition and motivation

GuntenTag is a tool for medium- and large-scale analysis of texts in the Project Gutenberg corpus. The high-level goal of the project is to create an ongoing two-way flow of resources between computational linguists and digital humanists, allowing computational linguists to identify pressing problems in the large-scale analysis of literary texts, while giving digital humanists access to a wider variety of NLP tools for exploring literary phenomena. GutenTag is intended to be a standalone software tool for non-programmers, but the source code is also available and we welcome others in the computational linguistics community to contribute to its development or adapt it as needed.

Project Gutenberg

Project Gutenberg[1] is a web-based collection of texts which have fallen out of copyright in the United States. The work here is based on the most recently released (2010) DVD image, which has 29,557 documents. Nearly all major canonical works of English literature published before 1923 are included in the collection. The English portion of the corpus consists of approximately 1.7 billion tokens.

INITIAL VERSION

INTERFACE

We use an HTML form as the GUI for GutenTag (*see detail above*), creating a configuration file which can be saved, loaded, and modified in a text editor. Users define the particular sub-corpus of Project Gutenberg they wish to investigate. At a lower level, users can define sets of attributes which are accessible as a single tag in the interface.

SUB-CORPUS FILTERING

Users can define sub-corpora using any of the tags defined in the metadata for the Project Gutenberg corpus (title, author, author birth, author death, and, for some texts, Library of Congress classification and subjects). Some automatically-generated tags (the genre and structure tags) can also be used to narrow the search.

METATEXT CLEANING

Project Gutenberg texts contain header and footer sections with information about the copyright and transcription process. We use fairly sophisticated heuristics to remove this information, including certain kinds of meta-text elements which are inserted within the text boundaries.

GENRE CLASSIFICATION

Since Project Gutenberg has inconsistent metadata with respect to genre, we trained a decision tree classifier that uses hand-identified features reflecting structural aspects of the texts (not the linguistic content) to distinguish four genres: fiction, nonfiction, poetry, and drama. Cross-validation using the training texts (texts in the Project Gutenberg corpus that are marked for genre) indicates 91% accuracy.

STRUCTURAL TAGGING

Project Gutenberg has implicit structuring of texts using spacing and indentation, but this is very inconsistent. GutenTag uses complex heuristics to identify the structure of the text, including elements of the front and back matter as well as text sections (e.g. chapters, acts,) and other genre-specific elements (stage direction, dialogue).

LEXICAL TAGGING

GutenTag uses NLTK tokenization, lemmatization, and POS tagging.[4] Other lexical tags available are manually-built lexicons (MRC psycho-linguistic database[5] and the General Inquirer Dictionary[6]) and a lexicon of style built from the Project Gutenberg corpus.[7] Users can define their own lexicons. GutenTag includes a simple name tagger and connects names and likely spans of dialogue.

TEI OUTPUT

GutenTag uses the popular XML-based Text Encoding Initiative (TEI)[8] format as the default output format when structure (rather than just tokens) is requested, which makes it compatible with other work in the Digital Humanities.

FUTURE VERSIONS

INTERFACE

One long-term goal of the project is to be able to access the full functionality of GutenTag via the web. Given our diverse user base, we may need to upgrade the interface to improve usability.

SUB-CORPUS FILTERING

Useful information missing from the Project Gutenberg database includes the text's publication date and place and information about the author such as their gender, nationality, place of birth, education, marital status, and membership in particular literary schools. We intend to collect this information from structured resources such as Open Library, and Wikipedia, and perhaps even unstructured text.

METATEXT CLEANING

Individual Project Gutenberg transcribers used different formats for inserting their notes, and so there are probably some cases we have not yet come across. We don't yet properly support languages other than English.

GENRE CLASSIFICATION

We could subdivide our four main genres into any number of sub-genres, though it might be difficult to do this without integrating content features (which might invalidate some uses of the tag). A more sophisticated classifier might be preferred, and we should integrate more features.

STRUCTURAL TAGGING

We built our structural tagging module by focusing on the structure of 50 texts from diverse genres (20, fiction, 10 nonfiction, 10 drama 10 poetry); this is an insufficient sample. Some structural tagging would likely benefit from statistical machine learning approaches. Other kinds of structure that would require sophisticated NLP modules include those reflecting time, location, viewpoint,[2] topic,[3] and narrative structure.

LEXICAL TAGGING

We intend to add multi-word lexical tagging and to upgrade the name tagger to distinguish various types (eg. characters vs. locations). Future modules would include tagging of elements such as meter, anaphora, alliteration, onomatopoeia, foreign languages, allusions, simile, and metaphor.

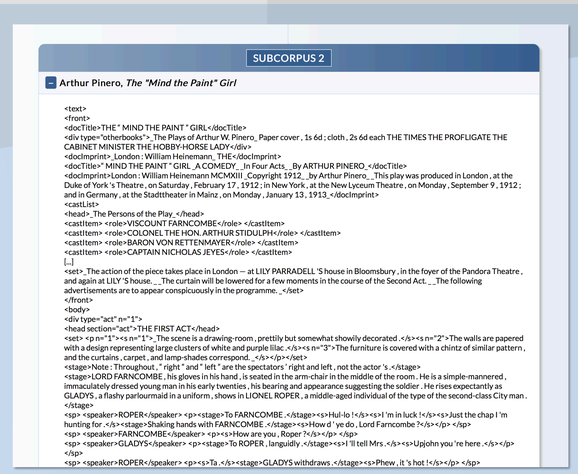
TEI OUTPUT

Though we have tried to stay as close as possible to the TEI standard, we have omitted certain tags because we felt that they were too detailed or too challenging to deal with automatically. We would be interested in hearing feedback on other tags we should include, and on existing tags that we are handling poorly.

FUTURE VERSIONS

Other tools

Tools similar to GutenTag include software for automatic analysis of texts for literary purposes such Voyant,[9] literary corpus tools like PhiloLogic,[10] general purpose NLP tool packages such as NLTK (which GutenTag is built on), and a (very simple) existing Project Gutenberg reader, Gutenberg.[11] The overlap between these other tools and GutenTag is, however, fairly small: no existing tool offers sophisticated language analysis with literature-specific tagging appropriate for large-scale analysis. Our intent is that GutenTag will become a growing repository for NLP solutions to tasks relevant to literary analysis, and it is this wide-ranging, inherently cross-disciplinary focus that is the clearest difference between GutenTag and other tools.



TEI Output Detail

Acknowledgments This work was supported by the Natural Sciences and Engineering Research Council of Canada, the MITACS Elevate program, and the University of Guelph.

References

1. http://www.gutenberg.org/
2. Janyce M. Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287. June.
3. Anna Kazantseva and Stan Szpakowicz. 2014. Hierarchical topical segmentation with affinity propagation. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*.
4. Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.
5. Max Coltheart. 1980. *MRC Psycholinguistic Database User Manual: Version 1*. Birkbeck College.
6. Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.
7. Julian Brooke and Graeme Hirst. 2013. Hybrid models for lexical acquisition of correlated styles. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP '13)*.
8. http://www.tei-c.org/Guidelines/
9. http://voyant-tools.org/
10. https://sites.google.com/site/philologic3/home
11. https://pypi.python.org/pypi/Gutenberg/0.4.0