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

Julian Brooke
Dept of Computer Science
University of Toronto
jbrooke@cs.toronto.edu

Adam Hammond
School of English and Theatre
University of Guelph
adam.hammond@uoguelph.ca

Graeme Hirst
Dept of Computer Science
University of Toronto
gh@cs.toronto.edu

Abstract

This paper introduces a software tool, GutenTag, which is aimed at giving literary researchers direct access to NLP techniques for the analysis of texts in the Project Gutenberg corpus. We discuss several facets of the tool, including the handling of formatting and structure, the use and expansion of metadata which is used to identify relevant subcorpora of interest, and a general tagging framework which is intended to cover a wide variety of future NLP modules. Our hope that the shared ground created by this tool will help create new kinds of interaction between the computational linguistics and digital humanities communities, to the benefit of both.

1 Introduction

The emerging field of digital literary studies has embraced not only statistical analysis of literary texts in the corpus linguistics tradition, but even more complex methods such as principal components analysis (Burrows, 1987), clustering (Rybicki, 2006), and topic modeling (Goldstone and Underwood, 2012; Jockers, 2013). At the same time, there is sustained interest in computational linguistics in tackling problems that are specific to literature, as evidenced by an annual dedicated workshop as well as various papers at major conferences (Elson et al., 2010; Wallace, 2012; He et al., 2013; Bamman et al., 2014). Though some work in the shared ground between these two fields is explicitly cross-disciplinary, this is still fairly atypical, reflecting a deep cultural barrier (Hammond et al., 2013): in most cases, digital humanists are using off-the-shelf statistical tools with little or no interaction with computer scientists, and computational linguists are developing literature-specific techniques which are unavailable or unknown to the digital humanist community. The high-level goal of the project proposed here is to create an on-going two-way flow of resources between these groups, allowing computational linguists to identify pressing problems in the large-scale analysis of literary texts, and to give digital humanists access to a wider variety of NLP tools for exploring literary phenomena. The context for this exchange of ideas and resources is a tool, GutenTag[1], aimed at facilitating literary analysis of the Project Gutenberg (PG) corpus, a large collection of plain-text, publicly-available literature.

At its simplest level, GutenTag is a corpus reader; given the various eccentricities of the texts in Project Gutenberg (which reflects the diversity of the source texts and the rather haphazard nature of their collection), this application alone serves to justify its existence. A second facet of the tool is a corpus filter: it uses the information contained explicitly within the PG database and/or derived automatically from other sources to allow researchers to build subcorpora of interest reflecting their exact analytic needs. Another feature gives GutenTag its name: the tool has access to tagging models which represent the intersection of literary analysis needs and existing NLP methods. The output of GutenTag is either an XML corpus with tags (at both text and metatextual levels) based on the TEI-encoding standard; or, if desired, direct statistical analysis of the distribution of tags across different subcorpora. None of the features of GutenTag mentioned above are intended to be static: GutenTag is a tool that will grow and improve with feedback from the digital human-

[1] GutenTag is available at http://www.cs.toronto.edu/~jbrooke/gutentag/

Proceedings of NAACL-HLT Fourth Workshop on Computational Linguistics for Literature, pages 42–47,
Denver, Colorado, June 4, 2015.

ities community and new methods from the computational linguistics community.

2 Project Gutenberg

Project Gutenberg is a web-based collection of texts (mostly literary fiction such as novels, plays, and collections of poetry and short stories, but also nonfiction titles such as biographies, histories, cookbooks, reference works, and periodicals) which have fallen out of copyright in the United States. There are versions of Project Gutenberg in various countries around the world, but the development of GutenTag has been based on the US version.[2] The entire contents of the current archive is almost fifty thousand documents, though 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 (and many from other languages) published before 1923 (the limit of US copyright) are included in the collection. The English portion of the corpus consists of approximately 1.7 billion tokens. Although it is orders of magnitude smaller than other public domain collections such as HathiTrust, the Internet Archive, and Google Books, PG has some obvious advantages over those collections: all major modern digitization efforts use OCR technology, but the texts in Project Gutenberg have also been at least proofread by a human (some are hand-typed), and the entire corpus remains sufficiently small that it can be conveniently downloaded as a single package;[3] this last is an important property relative to our interests here, since the tool assumes a complete copy of the PG corpus is present.

3 Reader

The standard format for texts in the PG corpus is plain text, most commonly the Latin-1 character set though some are UTF-8 Unicode. Generally speaking, the actual content is bookended by information about creation of the corpus and the copyright. The first challenge is removing this information–not a trivial task, given that the exact formatting is extremely inconsistent across texts in the corpus. GutenTag employs a fairly complex heuristic involving regular expressions; this handles some of the more troublesome cases by making sure that large sections of the text are not being tossed out. Other common extra-textual elements that we remove during this stage include references to illustrations and notes that are clearly not part of the text (e.g. transcriber's notes).

Most texts are structured to some degree, and this structure is reflected inconsistently in the raw Gutenberg texts by implicit indicators such as extra spacing, capitalized headings, and indentations. The structure depends on the type of literature, which may or may not be indicated in the datafile (see Section 4). Most books contain at least a title and chapter/section/part headings (which may be represented by a number, a phrase, or both); other common elements include tables of contents, introductions, prefaces, dedications, or initial quotations. Plays have an almost entirely different set of elements, including character lists, act/scene breaks, stage directions, and speaker tags. GutenTag attempts to identify common elements when they appear; these can be removed from the text under analysis if desired and/or used to provide structure to the text in the final output (as special tags, see Section 5). Note that this step generally has to occur before tokenization, since many markers of structure are destroyed in the tokenization process.

GutenTag is written in Python and built on top of the Natural Language Toolkit (Bird et al., 2009): for sentence and word tokenization, we use the NLTK regex tokenizer, with several pre- and post-processing tweaks to deal with specific properties of the corpus and to prevent sentence breaks after common English abbreviations. We are careful to preserve within-word hyphenation, contractions, and the direction of quotation marks.

4 Subcorpus Filter

Taken as a whole, the Gutenberg corpus is generally too diverse to be of use to researchers in particular fields. Relevant digital humanities projects are far more likely to target particular subsections of the corpus, e.g. English female novelists of the late 19th century. Fortunately, in addition to the raw texts, each document in the PG corpus has a correspond-

[2] http://www.gutenberg.org

[3] http://www.gutenberg.org/wiki/Gutenberg:The_CD_and_DVD_Project

ing XML data file which provides a bibliographic record, including the title, the name of the author, the years of the author's birth and death, the language in which the text is written, the Library of Congress classification (sometimes multiple), and the subject (often multiple). GutenTag provides a complete list of each of the non-numerical tags for reference and allows the user to perform an exact or partial string match to narrow down subcorpora of interest or to combine lists of independently defined subcorpora into a single subcorpus.

Although they are extremely useful, there are numerous problems with the built-in PG annotations. While Library of Congress classification is generally reliable for distinguishing literature from other books, for instance, it does not reliably distinguish between genres of literature. Therefore, GutenTag distinguishes prose fiction from drama and poetry by (at present) simple classification based on the typical properties of these genres. For drama, it looks to see if there are significant numbers of speaker tags (which unfortunately appear in numerous distinct forms in the corpus); to distinguish poetry from prose fiction, it uses line numbers and/or the location of punctuation (in poetry, punctuation often appears at the end of lines of verse); collections of short stories can often be distinguished from novels by their titles (e.g. *and other stories*). We make these automatic annotations available as a "genre" tag to help users create a more-exact subcorpus definition.

Other useful information missing from the PG 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. When possible, we collect additional information about texts and their authors from other structured resources such as Open Library, which has most of the same texts but with additional publication information and metadata, and Wikipedia, which only references a small subset of titles/author, but usually in more detail. A more speculative idea for future work is to derive information about less-popular texts and authors from unstructured text.

We did not carry out a full independent evaluation of the (non-trival) subcorpus filtering and reader features of GutenTag, but we nevertheless took steps to ensure basic functionality: after developing some initial heuristics, we sampled 30 prose texts, 10 poetry texts, and 10 plays randomly from the PG corpus based on our automatic classification, resampling and improving our classification heuristics until we reached perfect performance. Then, using those 50 correctly-classified texts, we improved our heuristics for removing non-textual elements and identifying basic text structure until we had perfect performance in all 50 texts (as judged by one of the authors). Needless to say, we avoided including heuristics that had no possibility of generalization across multiple texts (for instance, hand-coding the titles of books). We also used these texts to confirm that sub-corpus filtering was working as expected. GutenTag comes with a list of the texts that were focused on during development, with the idea they could be pulled out using sub-corpus filtering and used as training or testing examples for more-sophisticated statistical techniques.

5 Tagging

Once a text has been tokenized, a tag can be defined as a string identifier, possibly with nominal or numerical attributes, which is associated with a span of tokens. Tags of the same type can be counted together, and their attributes can be counted (for nominal attributes) or summed or averaged (for numerical attributes) across a text, or across a subcorpus of texts. The particular tags desired in a run of GutenTag are specified by the user in advance. The simplest tag for each token is the token itself, or a lemmatized version of the token. Another tag of length one is the part-of-speech tag, which, in GutenTag, is currently provided by the NLTK part-of-speech tagger. GutenTag also supports simple named entity recognition to identify the major characters in a literary work, by looking at repeated proper names which appear in contexts which indicate personhood. Any collection of words or phrases can be grouped under a single tag using user-defined lexicons, which can be nominal or numerical; as examples of this, the GutenTag includes word properties from the MRC psycholinguistic database (Coltheart, 1980), the General Inquirer Dictionary (Stone et al., 1966) and high-coverage stylistic and polarity lexicons (Brooke and Hirst, 2013; Brooke and Hirst, 2014) which were built automatically using the vari-

ation within the PG corpus itself.

Tags above the word level include, most prominently, structural elements such as chapters identified in the corpus-reader step. Another tag supported in GutenTag is the TEI "said" tag which is used to identify quoted speech and assign it to a specific character. The current version of "said" identification first detects the quotation convention being used in the text (i.e. single or double quotes), matches right and left quotes to create quote spans, and then looks in the immediate vicinity around the quotes to identify a character (as identified using the character module) to whom to assign the quotation. Though currently functional, this is the first module in line to be upgraded to a fully statistical approach, for instance based on the work of He et al. (2013).

As far as GutenTag is concerned, a tagger is simply a function which takes in a tokenized text and (optionally) other tags which have been identified earlier in the pipeline, and then outputs a new set of tags. Even complex statistical models are often complex only in the process of training, and classification is often matter of simple linear combinations of features; adding new tagging modules should therefore be simple and seamless from both a user's and a developer's perspective. To conclude this section, we will discuss some of the ideas for kinds of tagging that might be useful from a digital humanities perspective as well as interesting for computational linguists. Some have been addressed already, and some have not. The following is intended not as an exhaustive list but rather as a starting point for further discussion.

At the simpler end of the spectrum, we can imagine taggers which identify some of the classic poetic elements such as rhyme scheme, meter, anaphora, alliteration, onomatopoeia, and the use of foreign languages (along with identification of the specific language being used). Metaphor detection is of growing interest in NLP (Tsvetkov et al., 2014), and would undoubtedly be useful for literary analysis (as would simile detection, a somewhat simpler task). Another challenging but important task is the identification of literary allusions: we envision not only the identification of allusions, but also the establishment of direct connections between alluding and alluded works with the PG corpus, which we could then employ to derive metrics of influence and canonicity within the corpus. We are also interested, where appropriate, in identifying features relevant to narratives: when analyzing a novel, for example, it would be interesting to be able to tag entire scenes with a physical location, a time of day, and a list of participants; for an entire narrative, it would be useful to identify particular points in the plot structure such as climax and dénouement, and other kinds of variation such as topic (Kazantseva and Szpakowicz, 2014) and narrator viewpoint (Wiebe, 1994).

6 Interfaces

GutenTag is intended for users with no programming background. The potential options are sufficiently complex that a run of GutenTag is defined within a single configuration file, including any number of defined subcorpora, the desired tag sets (including various built-in tagging options and user-defined lexicons), and options for output. We also offer a web interface for small-scale, limited analysis for those who do not want to download the entire corpus.

Given our interest in serving the digital humanities community, it is important that the output options reflect their needs. For those looking only for a tagged corpus, the Text Encoding Initiative (TEI) XML standard[4] is the obvious choice for corpus output format. The only potential incompatibility is with overlapping but non-nested tags (which are supported by our tag schema but not by XML), which are handled by splitting up the tags over the smaller span and linking them using an identifier and "next" and "prev" attributes. Numerical attributes for lexicons are handled using a "value" attribute. Again, users can choose whether they want to include structural elements that are not part of the main text, and whether they want to include these as part of the text, or as XML tags, or both.

For those who want numerical output, the default option is a count of all the desired tags for all the defined subcorpora. The counts can be normalized by token counts and/or divided up into scores for individual texts. We also allow counts of tags occurring only inside other tags, so that, for instance, different sub-genres within the same texts can be compared. GutenTag is not intended to provide more-

[4] http://www.tei-c.org/Guidelines/

sophisticated statistical analysis, but we can include it in the form of interfaces to the Numpy/Scipy Python modules if there is interest. We will include support for direct passing of subcorpora and the portions of subcorpora with a particular tag to MALLET (McCallum, 2002) for the purposes of building topic models, given the growing interest in their use among digital humanists.

7 Comparison with other tools

GutenTag differs from existing digital humanities text-analysis offerings in its focus on large-scale analysis using NLP techniques. Popular text-analysis suites such as Voyant[5] and TAPoR[6] present numerous useful and user-friendly options for literary scholars, but their focus on individual texts or small groups of texts as well as output which consists mostly of simple statistical measures or visualizations of surface phenomena means that they are unable to take advantage of the new insights that larger corpora and modern NLP methods can (potentially) provide. As digital humanists become increasingly interested in statistical approaches, the limiting factor is not so much the availability of accessible statistical software packages for doing analysis but rather the ability to identify interesting subsets of the data (including text spans *within* texts) on which to run these tools; GutenTag supplements these tools with the goal of producing more diverse, meaningful, and generalizable results.

GutenTag also has some overlap in functionality with literary corpus tools such as PhiloLogic[7], but such tools are generally based on manual annotations of structure and again offer only surface treatment of linguistic phenomena (e.g. identification of keywords) for text retrieval. We also note that there is a simple Python corpus reader for Gutenberg available[8], but it is intended for individual text retrevial via the web, and the only obvious overlap with GutenTag is the deletion of copyright header and footers; in this regard GutenTag is noticeably more advanced since the existing reader relies only on presence or absence of keyphrases in the offending spans.

[5]http://voyant-tools.org/

[6]http://tapor.ca/

[7]https://sites.google.com/site/philologic3/

[8]https://pypi.python.org/pypi/Gutenberg/0.4.0

There are of course many software toolkits that offer off-the-shelf solutions to a variety of general computational linguistics tasks: GutenTag makes direct use of NLTK (Bird et al., 2009), but there numerous other popular options–our choice of NLTK mostly reflects our preference for using Python, which we believe will allow for quicker and more flexible development in the long run. What is more important is that GutenTag is intended to make only modest use of off-the-shelf techniques, because we strongly believe that using NLP for literary analysis will require building literature-specific modules, even for tasks that are otherwise well-addressed in the field. In numerous ways, literary texts are simply too different from the newswire and web texts that have been the subject of the vast majority of work in the field, and there are many tasks fundamental to literary study that would be only a footnote in other contexts. Our intent is that GutenTag will become a growing repository for NLP solutions to tasks relevant to literary analysis, and as such we hope those working in digital humanities or computational linguistics will bring to our attention new modules for us to include. It is this inherently cross-disciplinary focus that is the clearest difference between GutenTag and other tools.

8 Conclusion

In the context of computational analysis of literature, digital humanists and computational linguists are natural symbionts: while increasing numbers of literary scholars are becoming interested in the insights that large-scale computational analysis can provide, they are often limited by their lack of technical expertise. GutenTag meets the needs of such scholars by providing an accessible tool for building large, highly customizable literary subcorpora from the PG corpus and for performing pertinent advanced NLP tasks in a user-appropriate manner. By thus drawing increasing numbers of literary scholars into the realm of computational linguistics, GutenTag promises to enrich the latter field by supplying it with new problems, new questions, and new applications. As the overlap between the spheres of digital humanities and computational linguistics grows larger, both fields stand to benefit.

Acknowledgements

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

References

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A bayesian mixed effects model of literary character. In *Proceedings of the 52st Annual Meeting of the Association for Computational Linguistics (ACL '14)*.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

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)*.

Julian Brooke and Graeme Hirst. 2014. Supervised ranking of co-occurrence profiles for acquisition of continuous lexical attributes. In *Proceedings of The 25th International Conference on Computational Linguistics (COLING 2014)*.

John F. Burrows. 1987. *Computation into Criticism: A Study of Jane Austen's Novels and an Experiment in Method*. Clarendon Press, Oxford.

Max Coltheart. 1980. *MRC Psycholinguistic Database User Manual: Version 1*. Birkbeck College.

David K. Elson, Nicholas Dames, and Kathleen R. McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*.

Andrew Goldstone and Ted Underwood. 2012. What can topic models of PMLA teach us about the history of literary scholarship? *Journal of Digital Humanities*, 2.

Adam Hammond, Julian Brooke, and Graeme Hirst. 2013. A tale of two cultures: Bringing literary analysis and computational linguistics together. In *Proceedings of the 2nd Workshop on Computational Literature for Literature (CLFL '13)*.

Hua He, Denilson Barbosa, and Grzegorz Kondrak. 2013. Identification of speakers in novels. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)*.

Matthew Jockers. 2013. *Macroanalysis: Digital Methods and Literary History*. University of Illinois Press, Champaign, IL.

Anna Kazantseva and Stan Szpakowicz. 2014. Hierarchical topical segmentation with affinity propagation. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Jan Rybicki. 2006. Burrowing into translation: Character idiolects in Henryk Sienkiewicz's trilogy and its two English translations. *Literary and Linguistic Computing*, 21:91–103.

Philip J. Stone, Dexter C. Dunphy, Marshall S. Smith, and Daniel M. Ogilivie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL '14)*.

Byron C. Wallace. 2012. Multiple narrative disentanglement: Unraveling *Infinite Jest*. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT '12)*.

Janyce M. Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287, June.