Enhancing Project Gutenberg Metadata  
with Readability Scores

|  |  |
| --- | --- |
| Sofie Grant | Alexandros Plakidas Ntasios |
|  |  |
| School of Computing and Information Systems | School of Computing and Information Systems |
| Grand Valley State University | Grand Valley State University |
|  |  |
| grantsof@mail.gvsu.edu | plakidaa@mail.gvsu.edu |

**Abstract**

Project Gutenberg is a Digital Library that allows a broad audience to browse through and read the contents of tens of thousands of books. Its information retrieval capabilities, however, are limited by the inability to search on the basis of readability. In this paper, we examine the feasibility of enhancing the existing set of e-book metadata by adding readability score and education grade metadata. We also describe and analyze the methods we used to achieve this, as well as evaluate the extent to which we were successful in achieving our initial scope.

# 1. INTRODUCTION

The need to use a certain standard for categorizing books according to their ability to be comprehended by groups of people based on their education or reading abilities has concerned a number of research groups and organizations [9]. This paper delineates the methodology and process of enhancing the Project Gutenberg Digital Library by incorporating reading score metadata. Our goal was to create a triple store database that would allow the indexing and searching of all the English Project Gutenberg e-book records, and output the readability scores along with their evaluated education level as part of the resulting query. We were successful, albeit not to the full extent of our initial scope.

# 2. RELATED WORK

Project Gutenberg has been used as a source for a number of studies regarding the accessibility of e-books by young and adult readers alike. Trieschingg and Hauff [19] worked on an algorithm for automatically detecting unusual or unfamiliar words that increased the reading levels of children’s books using data from Project Gutenberg. Saxena [17] demonstrated a correlation between results of Zipf’s “principle of least effort” [21] with books that have Flesch readability scores between 60 and 80.

Important as Project Gutenberg is as a tool used for researchers, it is also unequivocally important as a resource for students, instructors, or anyone in general who has an intellectual interest in literature. The works of Saxena [17] and Reck [14, 15] are further evidence that the published works available in digital libraries like Project Gutenberg are seen as a valuable resource for a vast reading audience. Reck, in particular, has devoted a great deal of time to creating as many as 18 different metadata properties on a substantial amount of Project Gutenberg books, in order to create assertions for making intersection queries over three major Digital Libraries [15].

The Project Gutenberg digital library features some essential metadata such as author name, book title, genre, and year of the author’s birth and death. It does not, however, feature readability metadata for its vast e-book catalog. This limitation prevents users from finding books that match their education level or reading comprehension skills. The website Reading Square [14] has made a worthwhile attempt at indexing Gutenberg Project texts including their Reading Ease and Fog Index metadata. There are two caveats, however: the website limits its collection to about 1,000 popular books, and there is little information available on the website that helps the user to interpret the numeric scores displayed. Users are obligated to have prior knowledge of these indexes or to locate interpretive information elsewhere and apply it to the raw scores available on the website. The need to locate this information could discourage some users from fully exploiting the website.

# 3. READABILITY OF TEXTS

The importance of measuring the reading difficulty of text is reflected by the variety of formulas and standards available today, as well as the extensive history behind the research effort in this area of interest, which began in the early 20th century [18]. The need for the use of readability measures has also been seen as significant in adolescents with reading disabilities [12]. The most significant works concerning readability are perhaps the Flesch publications in the 1940’s [6], and Kincaid’s readability study for the Navy in the 1970s [10].

## Readability Formulas

In most cases, readability of a document is derived by the semantic and syntactical nature of the text. Formulas count the number of syllables in a word, words in a sentence, or employ a word frequency list to measure the complexity of a document.

Some of the readability standards include the Dale-Chall formula [4], the Fog-Index [8], and the Flesch Reading Ease formula [6]. In our project, we focused on the Flesch Reading Ease and Flesch-Kincaid (F-K) Grade Level Formulas, due to their overall accuracy, popularity and simplicity.

# 4. METHODOLOGY

The Java Virtual Machine environment was used extensively, in order to create and set up tools made in that language. Windows and Linux file processing commands were also used along the way.

## Acquiring the Source Texts and RDF Metadata

We downloaded all the books using a Unix “wget” command with operators to allow only “english” books to be fetched. We then downloaded the RDF/XML catalog of book metadata directly from the Project Gutenberg website [13]. The metadata was compressed and consisted of thousands of folders: each book's metadata record was stored in its own folder. Similarly, the book text files were stored in a nested folder structure that made navigation through them inefficient. We used a Java tool and the Windows copy command to flatten the folder structure of both RDF files and text files.

## Preparation of Source Data for Readability Analysis

A number of steps had to be taken in order to prepare text files for readability analysis. An overview of the steps can be seen in Figure 1.

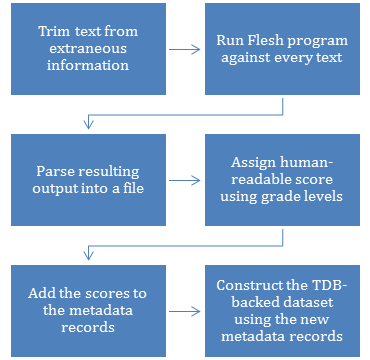


Figure 1 - Steps followed for text analysis and metadata enhancement

The text files contained disclaimers and other content at the head and the base of each file that was extraneous to the original book. This extra information would skew our results if analyzed, therefore it had to be stripped away.

We used a batch script developed by Krugel [11] to achieve that. We could not verify that every single file had all header and footer information removed, but the ones that we randomly tested proved free from that information. The text files that were downloaded consisted of both UTF-8 and ASCII encoded versions, as well as some with an additional Unicode encoding. We decided to keep the ASCII versions and run a script that would delete the rest, with the exception of books that were only stored in UTF-8 format.

Flesh [7], the program that we used to analyze the e-books, would process each text file and output the Flesch-Kincaid Grade and Flesch Reading Ease levels among other measurements. The application would only analyze one file at the time, but the authors of Flesh had also made a command line version available to allow more flexibility. We wrote a script that was able to automate the process of running the command line version of Flesh over many files.

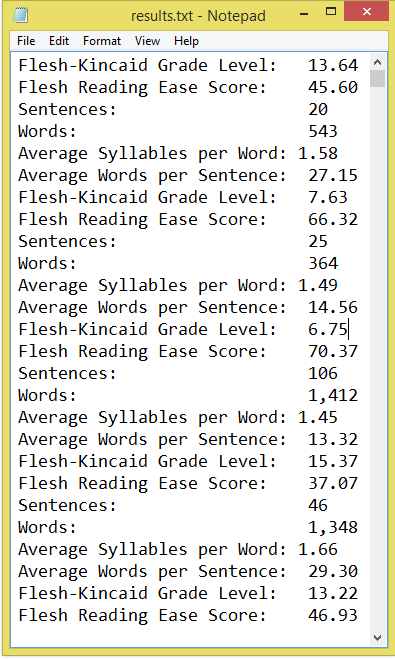


Figure 2 - Batch file result output of Flesh command line tool

Reck found that the RDF files have a high degree of “variability and poor consistency” [16] which added a great deal of extra workload. We found the same to be true for the text file naming format. Thousands of files were named after parts of their title instead of their record number. We identified and renamed these outlying files.

Once we had an uninterrupted sequence of text files, we were then able to apply the Flesh tool to process the Flesch-Kincaid Grade and Flesch Reading Ease levels of each book and output the results into a single result file (Figure 2). We wrote an additional script that would parse the result file and remove extra data provided by Flesh, such as word and sentence count. That file would then be further processed by another script that would assign a more human-readable score by means of education levels based on the Flesh-Kincaid Grade Level score (table 1, figure 3). When that part of the process was done, our data was ready to be inserted as book metadata into the original RDF/XML files, in order to construct the triple store dataset.

|  |  |
| --- | --- |
| **Grade Level Score Range** | **Grade Level** |
| 1.00-1.99 | 1st Grader |
| … | … |
| 12.00-12.99 | 12th Grader |
| 13.00-16.00 | College |
| >16.00 | Advanced College |

Table 1 - Grade Level Score Evaluation

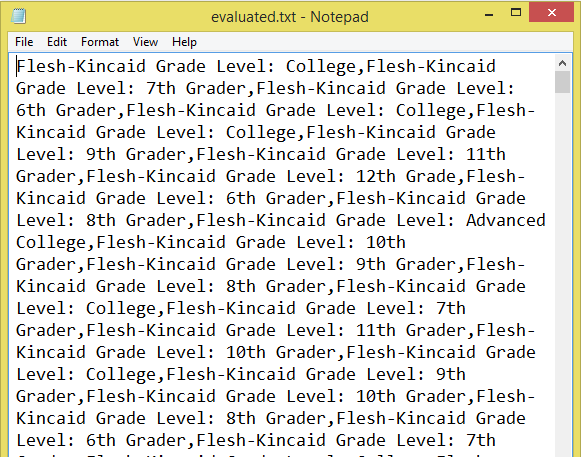


Figure 3 - Comma delimited output results after grade level evaluation

## Setting up a Triple Store Database with Apache Jena

Apache Jena is an open source Java framework that contains APIs which support local and persistent triple store databases [1]. Its high performance in managing large triple stores made it a suitable candidate for our goal. There are two types of triple store databases that Jena allows: TDB and SDB. The TDB API is focused on RDF formatted triple stores and allows more scalability and better performance, whereas SDB offers persistent triple stores using a number of supported relational databases [2], [3]. We opted for the TDB API to construct our triple store database.

As mentioned before, a number of scripts had to be created in order to allow the Flesch readability scores to be added to the original RDF graphs. The scripts would go through every book and check first if it contained the term EN (the abbreviation for the English language) in its Dublin core language tag [5]. They would then upload the updated metadata records of the English language books and the unedited metadata records of the non-English language books to the triple store database.

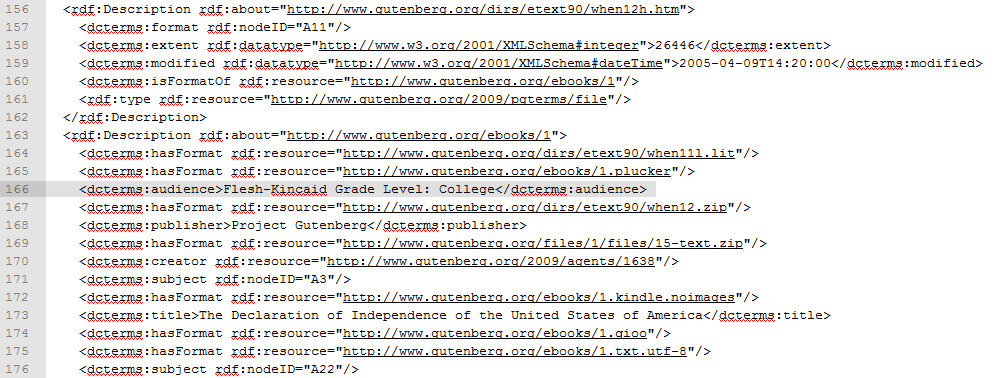


Figure 4 - A modified RDF file after insertion of Flesh-Kincaid Grade Level metadata using the “audience” tag

## Querying the Database

Apache Jena’s TDB API includes SPARQL query engine. It allows querying “required and optional graph patterns” and supports “aggregation, subqueries, negation [...] and constraining queries by source RDF graph” [20]. The result can be an RDF graph or a result set. The last script that needed to be created, worked as follows: it would run one query to verify each book had a metadata record uploaded to TDB and a second query that verified the English language book metadata records now included the Dublin Core language tag.

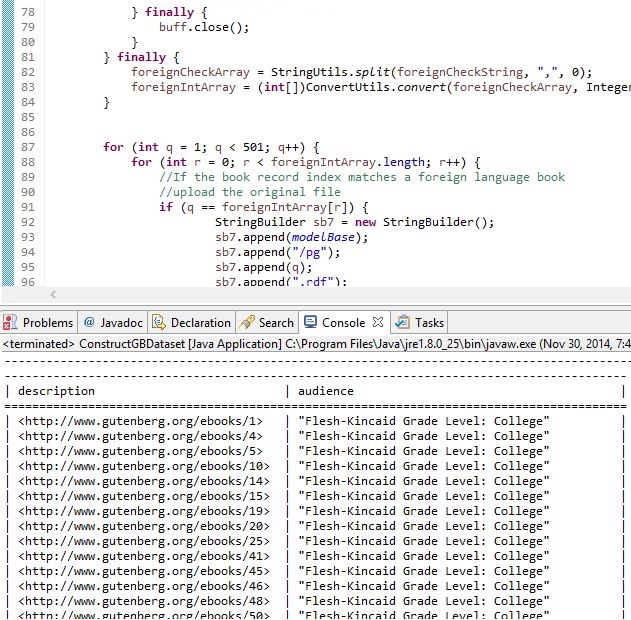


Figure 5 - Query engine output showing some of the books’ identifying numbers and human-interpreted readability levels

# 5. CHALLENGES

Project Gutenberg’s database provided us with a number of challenges right from the start. The first challenge was to download all the books, excluding non-English ones, in order to match our readability analysis scope. Despite using the proper command operators, the books that were downloaded were not all written in English. This prompted us to make a custom script that would skip the non-English documents by detecting the language tag of each RDF file.

Once all the books were downloaded, we found that -as previously mentioned- that Project Gutenberg’s RDF and text file formats had poor consistency, inevitably generating a higher amount of workload than anticipated. Several thousands of texts were missing the simple numerical scheme found in the majority of the body of texts, and instead incorporated an alphanumerical system based on the book title. Because of a lack of a specific pattern, those texts had to be manually cross-referenced and renamed using an index file found at https://www.gutenberg.org/dirs/GUTINDEX.ALL.

The index identified a majority of files that incorporated the alphanumeric scheme. There were some outlying files that were stated on the GUTINDEX.ALL list to have been named according to the numeric scheme that were found to have been since renamed with the alphanumeric scheme. We were able to identify these outlying files and add them to our data set after some added effort.

Additionally, among the 500 Project Gutenberg titles were a small number of files that were not available in text format. For example, metadata record 116 refers to a small collection of video recordings of the Apollo 11 lunar landing by the United States. Given the limitations of our project, we elected to simply add these records to our list of non-English language books. This would prevent them from being included in the edited record set.

Many scripts had to be custom-made in order to relay the correct data to the Flesh reading level analyzer, and parse it properly for Apache Jena’s triple store database.

# 6. EVALUATION

Given the challenges above, the time resources in our disposal were not enough to process the number of books we initially set out to analyze. Out of our initial dataset of more than 40,000 books, we were able to successfully construct triple store graphs out of 500 books, and query their metadata. However, we proved that it is possible to utilize Flesh to analyze batches of text files, inject triples into an existing dataset of Project Gutenberg books and expose that dataset to a new triple store database.

# 7. CONCLUSION

Given the limited scope of our end result, a future version of our tools that would allow the quicker and sequence-agnostic nature of the files would potentially allow for the ability to process all 40,000+ English texts. A future endeavor in the same topic, would allow us to explore the ability to host a mirror database that would use clustering to filter books according to the grade levels they are suitable for.

# 8. ACKNOWLEDGMENTS

We would like to acknowledge the support and assistance of other scholars that developed scripts which helped the advancement of our project. Namely, Jack Frink and Johannes Krugel.

# 9. REFERENCES

1. Apache Software Foundation. 2011. Apache Jena. Available: https://jena.apache.org
2. Apache Software Foundation. 2011. Apache Jena TDB. Available: https://jena.apache.org/documentation/tdb
3. Apache Software Foundation. 2011. Apache Jena SDB. Available: https://jena.apache.org/documentation/sdb
4. Dale, E., Chall, J.: A Formula for Predicting Readability. Educational Research. Bulletin 27 (1948) 11–20
5. Dublin Core Dublin Core Metadata Initiative. 2014. DCMI Metadata Terms. Available:

http://dublincore.org/documents/dcmi-terms

1. Flesch R. (1948). A new readability yardstick. Journal of Applied Psychology 32: 221–233. doi:10.1037/h0057532
2. Frink, J. 2007. Flesh. Available: http://flesh.sourceforge.net
3. Gunning, R. (1968) The Technique of Clear Writing. 2nd edn. McGraw-Hill
4. L. Kane, J. Carthy, and J. Dunnion, “Readability Applied to Information Retrieval,” in Advances in Information Retrieval, M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, Eds. Springer Berlin Heidelberg, 2006, pp. 523–526
5. Kincaid, J. P., Fishburne, R. P., Rogers, R. L., & Chissom, B. S. (1975). Derivation of New Readability Formulas (Automated Readability Index, Fog Count, and Flesch Reading Ease formula) for Navy Enlisted Personnel. Research Branch Report 8-75. Chief of Naval Technical Training: Naval Air Station Memphis
6. Krugel, J. PAPI: Practical Approximate Pattern Matching with Index Structures. Available: http://www14.in.tum.de/spp1307
7. Leko, M.M., Mundy, A.C., Kang, H., Datar, D.S. 2013. If the Book Fits: Selecting Appropriate Texts for Adolescents With Learning Disabilities. Intervention in School and Clinic 2013 48: 267 (4 February 2013), 267-270 DOI: 10.1177/1053451212472232
8. PG - Project Gutenberg (2006) Free eBooks. Available: http://www.gutenberg.org
9. Readability Square. 2013 Readability Square. Available: http://readability.mackayst.com
10. Reck, R.P. (2006) Metadata Cards for Describing Project Gutenberg Texts. OntoLex 2006: Interfacing Ontologies and Lexical Resources for Semantic Web Technologies. Genoa, Italy
11. Reck, R.P. (2007) Generating and Rendering Scores for Project Gutenberg Texts. Proceedings of the Corpus Linguistics Conference. CL2007. University of Birmingham, UK. (27-30 July 2007)
12. Saxena, A. (2008) Readability of Open Education Resources. 5th Pan-Commonwealth Forum on Open Learning
13. Thorndike, E. J. (1921) The Teacher’s Word Book. New York: Teachers College Columbia University
14. D. Trieschnigg and C. Hauff, “Classic Children’s Literature - Difficult to Read?,” in Advances in Information Retrieval, P. Clough, C. Foley, C. Gurrin, G. J. F. Jones, W. Kraaij, H. Lee, and V. Mudoch, Eds. Springer Berlin Heidelberg, 2011, pp. 691–694
15. World Wide Web Consortium. (2013) SPARQL 1.1 Query Language. Available: http://www.w3.org/TR/sparql11-query
16. Zipf G.K. (1949) Human Behavior and the Principle of Least Effort. Cambridge, MA: Addison-Wesley Press