

Metadata Cards for Describing Project Gutenberg Texts

Ronald P. Reck

RRecktek LLC
Chantilly, VA, USA
reck@rrecktek.com

Abstract

The quantity of data available for linguistic analysis is ever increasing as the Internet expands. However, this is of questionable utility to automated processing when the format of the data is unpredictable. Significant variations can occur, even within a single source. Both data producers and consumers should be able to construct, interpret, and expect a consistently delineated set of metadata for depicting a text-based lexical resource. Standards exist for describing resources but they should be extended in order to support the type and range of information needed for accurate automated processing. Metacards describing the resources would be most beneficial if they extended the existing metadata standards to cover the variation a researcher is likely to encounter. Data producers should be expected to supply enough descriptive information so that a researcher can create the quality of work that others can build upon. This paper describes an effort for creating metacards of Project Gutenberg texts, examples of the variations that occur, and a sample metacard in RDF format.

1. Unpredictable Data is Less Useful

The automated or programmatic processing of corpus data are limited when there are significant or unpredictable variations in the source data. The more data there is, the greater the likelihood of variation, as well as the increased likelihood that the range of variation will cause problems. On the surface it appears that the best solution to the problem would be to have data producers create a consistent and documented data presentation. While consumers would benefit from consistency and documented formats they are not in a position to compel either consistency or documentation. Therefore, until expectations change, researchers need to be prepared to create their own metadata or operate without it.

1.1. Project Gutenberg

A good example of unexpected variation can be seen in the large text archive of freely available text called Project Gutenberg (PG, 2006). Project Gutenberg claims to be “the oldest producer of free ebooks on the Internet” (PG, 2006). PG’s ebook collection is an effort produced by hundreds of volunteers. As of January 2006 Project Gutenberg purports to have more than 17,000 books in electronic format most of which are unencumbered by copyright (PG, 2006).

Two distinctly different kinds of metadata are relevant here. *First order metadata* describes information about the archive file itself. First order metadata is structural and might express the number of files in the archive, the archive format, the archive compression ratio, the archive checksum, and similar information describing qualities of the archive file. The second type of metadata, referred to as *second order metadata*, describes specific characteristics of the content such as the author, title, copyright, or editor.

For the purpose of this discussion, both first order metadata and second order metadata are collapsed into a

single metadata presentation called a metacard. A more accurate presentation would have created a metacard for the archive that would contain first order metadata and a subsequent metacards would contain second order metadata describing the files contained in the archive. Relating the two metacards could be done with a *contains* relationship in the metacard describing the archive. This level of complexity would increase the precision of the following presentation, but would not add true value to the point of the discussion.

1.1.1. First Order Metadata Problems

Tens of thousands of PG ebooks sound like a treasure trove to a computational linguist until one tries to process them. The first problem a researcher can encounter involves the lack of a mechanism to verify the data integrity of an archive. Without a checksum it is unclear whether the downloaded file is complete. A checksum is a value that is calculated to check data integrity. In situations where the zip file itself is invalid or corrupted it is not clear whether the problem is local to the researcher or if the problem lies in the repository or mirror where the file was retrieved from. If the problem was determined to be a local problem the researcher could merely retrieve the file again. In the latter case the researcher might wonder if the mirror itself is the problem and that other file repositories might contain a valid version of the file. All these questions would be moot if the archive checksum was available in a manner consistent with the tens of thousands of software projects that regularly and adequately deal with this type of problem.

The next challenge a researcher is likely to encounter is the inconsistent directory structure. Some PG archives contain directories whereas others do not. Researchers programmatically dealing with PG archives need to compensate for any possible situation. Perhaps there is a directory in the archive, maybe there is not, maybe there

are multiple directories. Perhaps the directory is named in a manner consistent with the naming of the archive; maybe the directory name is arbitrary. To complicate matters, some archives contain multiple files while other archives contain only a single file. In the end, the usefulness of the archives as a lexical resource would be increased if metadata describing each archive was made available by the producer of the data. As it currently stands, each consumer of the data needs to be apprised of the variations and needs a heuristic to recognize and deal with them when they occur.

In essence, first order metadata problems can be compared to errors of omission in that the difficulty is based on information not being provided.

1.1.2. Second Order Metadata Problems

Once the first order PG problems are successfully navigated, an entirely new type of challenges makes themselves painfully evident. From an analysis standpoint it is often desirable to know characteristics of a file like the author, title or most importantly, the licensing or copyright restrictions of the material. While the more recent PG texts present at least some of this information, thousands of files neglect to include this information or do so in such a variety of ways that the researcher needs to employ another complex heuristic in order to glean even the most basic of information.

For example, among the 434 files released in 1999 by PG, numerous variations (Table 1) were observed for depicting the title of an ebook (PG, 2006).

Title Variations in PG eBooks
Title: A Midsummer Night's Dream
Project Gutenberg's The Three Musketeers
Project Gutenberg's Etext of Tom Swift And His Wizard Camera
Project Gutenberg's Etext of Tom Swift And His Giant Cannon
<p>Project Gutenberg's Etext of Tom Swift And His Aerial
<pre>Project Gutenberg's Etext of Tom Swift Among The Fire Fighters
The Project Gutenberg Etext of History of England.

Table 1: Title variations in PG ebooks

As these examples illustrate neither the presence of the asterisk at the start of the line, nor the possessive marker, nor the word *title* would be sufficient to consistently determine the text's title. In other words, the inconsistencies create unnecessary challenges for the programmer.

Examples of other variations can include the free variation of words like *Ebook* versus *Etext*, or *preparers* versus *transcribers* versus *producers*. Although, it is possible that these variations actually indicate a

distinction, it is also likely that they do not. While it is understandable that variations would occur, consistent metacards created by data producers would help consumers considerably in terms of automatic processing.

2. Metadata Increases Usefulness

If producers made metadata available for lexical resource researchers, then researchers could select to work with data samples that meet their requirements. Thus, metadata would enable the researcher to retrieve data without the added effort to filter it locally with heuristics. Secondly, if unexpected variations occur researchers could verify whether they were the product of an error or intentionally introduced. These are just a couple of the benefits that would be gained by consumers.

2.1. Generating Metadata Programmatically

While consumers would undoubtedly benefit from as much metadata as possible, the ability to determine second order metadata programmatically is quite limited and is often impossible. Second order metadata, as defined here, often requires specific knowledge about the origin or history of a work. Much of the metadata generated in this effort describes only the archive file, or physical qualities about the ebook. Metadata that describes the contents of the file generally came from inside the file itself or is the product of programmatic analysis. Structural similarities between files made it possible to determine many elements of second order metadata once the variability was decomposed. The use of regular expressions made compensating for variations much more manageable than it would otherwise have been.

2.2. Metadata Presentation in Metacards

An ideal solution for presenting metadata is in the form of a metacard. A metacard could be created by data producers and consumers alike. In this analysis a single metacard was created for each of the text based lexical resources using Resource Description Framework (RDF, 2004). Clearly, it would be most beneficial if the data producers created metacards rather than each consumer having to create them on their own.

In the worst case scenario, consumers could download data and then create metacards themselves to facilitate their lexical analysis. Once the metacards were created consumers could then share them with other interested parties throughout the community.

2.3. Existing Metadata Standards

Many metadata standards exist today, but the information they express is not equally important to all consumers. The most important kind of metadata is the type that facilitates programmatic analysis. Once a researcher can determine if the file is intact it is less important to provide metadata that they could create themselves. The second most important type of metadata is one that depicts license restrictions on the content. Of tertiary importance is metadata expressing attributes of

the archive that would otherwise require domain specific knowledge such as the title, author, or genre. This is not as significant of a problem when working with a file at a time since research can often supplement deficient or inaccurate information. Automating a task to analyze a repository such as Project Gutenberg had better employ a robust strategy or it is basically futile.

2.3.1. Dublin Core Metadata Element Set

Dublin Core Metadata Element Set (DCMES) – The Dublin Core Metadata Element Set is comprised of 15 optional elements and entails only a subset of the more encompassing Dublin Core Metadata Initiative (DCMI Usage Board, 2003). Currently, there are two formally endorsed versions of the Dublin Core Metadata Element Set 1.1 (DCMI Usage Board, 2003). They are the ISO Standard 15836-2003 and the NISO Standard Z39.85-2001. These 15 elements are so well considered that the entire set is applicable to the metacard creation effort described by this paper. It is clear this is the product of careful forethought as the creators state, “there are no fundamental restrictions to the types of resources to which Dublin Core metadata can be assigned” (DCMI, 2003).

In this study, four Dublin Core elements were used.

- *dc:title* – used to express the title of an ebook
- *dc:language*¹ – used to express the language an ebook was presented in
- *dc:creator* – used to express the author of an ebook
- *dc:available* – used to express the date an ebook became available in Project Gutenberg in the format YYYY-MM

2.3.2. ISLE Metadata Initiative

The International Standard for Language Engineering (ISLE) Metadata Initiative (IMDI) is a metadata standard proposal for describing multi-media and multi-modal language resources (ISLE, 2006). This proposal recognizes a distinction between top level *catalogue* metadata elements for describing *published corpora* and *session* level metadata elements targeted at describing multi-modal multimedia and written language corpora. Broader in scope than the metacards proposed here, IMDI creators intended to provide metadata for automatic resource discovery as well as *human readable* descriptions.

The IMDI initiative involves a set of proprietary tools that support its use such as the IMDI Editor and IMDI BCBrowser (ISLE, 2006). The ISLE metadata is very expressive and can represent much more detail than the metadata cards described in this paper. IMDI covers some of the same areas as Dublin Core, and where there is overlap between the two standards the Dublin Core version was used.

An interesting element described by IMDI is *CoreMediaFile Type* (ISLE, 2006). This element is encoded as a top-level media type from Multipurpose

¹Our use of *dc:language* uses the three letter country codes of ISO639-2 instead of the two letter country codes of ISO639-1.

Internet Mail Extension (MIME) as described in RFC2046 (1996). This element would make a well needed addition to the metacards described here since PG contains several MP3 audio files. Programmatic analysis of audio files was outside of the computationally based scope of the current effort. It was not determined whether this was possible with automated processing but it likely it is.

Special editing tools are intended to support the re-use of existing ISLE metadata transcriptions to create new ones. This time and energy saving feature is clearly intended to support manual creation of metadata. The only elements that are regarded as mandatory are the ones needed for the correct functioning of the tools for working with the metadata descriptions.

This flexibility is likely to encourage adoption and use of ISLE metadata. However, its usefulness is somewhat limited by the lack of clearly articulated requirements.

2.3.3. Open Language Archives Community

The Open Language Archives Community (OLAC) is an international partnership of institutions and individuals who are creating a virtual library of language resources by developing consensus of best current practices while developing a network of interoperable repositories and services for housing and accessing resources (2006). OLAC’s strategy tackles two problems at once. It both prescribes a data format for metadata and a repository for storage of that metadata.

The metadata described by OLAC fits into three distinct categories (2006). The first category of OLAC metadata follows the guidelines for embedding Dublin Core in XML. The second category of OLAC metadata uses the *xsi:type* mechanism to access to the full power of XML Schema. This permits the narrowing and restricting of element content. Lastly, OLAC metadata records may use extensions from other namespaces. The process for creating these extensions is well documented and quite accessible to interested parties wishing to create and express metadata in purely XML format.

The effort described here expressed metadata in RDF format for use in RDF aware data stores such as Siderean Software and Oracle 10g. For this reason the metacards are in RDF instead of XML, hence OLAC’s metadata approach was not adopted.

2.3.4. Friend of a Friend

The Friend of a Friend (FOAF) project is based around creation of machine readable information about people, groups and companies (2006). The FOAF vocabulary is based on RDF/OWL and is quite straight forward and easy to understand even for humans. This contrasts some other uses of RDF. Three elements from the FOAF project were used in the PG metacards.

- *foaf:name* – used to express the name of the PG text editor/translator/producer
- *foaf:mbox* – used if a producer’s email address was provided and if determinable

- *foaf:sha1*² – adopted to express the checksum of the archive file

2.4. Extending Metadata Elements

While clearly, there are a range of useful elements in existing standards, the analysis here has created eleven metadata elements that are not part of any of the cited existing standards applicable to PG lexical resources. They include:

- *charactercount* – A count of the characters in the uncompressed archive. This value is determined through the use of the *wc* command.
- *characterSet* – This is a product of the *file* command.
- *cratio* – This element expresses the ratio of the compressed archive to the uncompressed archive and therefore is derivable from two other elements. It comes from the default output of the *zipinfo* command (version 2.40).
- *csize* – The element expresses the number of bytes the compressed archive takes up on disk. It comes from the default output of the *zipinfo* command (version 2.40).
- *etext* – This is the PG number for the text. Each PG text has a unique number.
- *fcoun*t – This element is the number of files that are contained in the archive as determined by the *zipinfo* command (version 2.40).
- *ftype* – The file type specified by this metadata element comes from the output of the *file* command (version 3.39). This program is believed to exceed the System V Interface Definition of FILE(CMD)2.
- *linecount* – A count of the lines in the uncompressed archive. This value is determined through the use of the *wc* command.
- *Producer* – This is the PG producer (a.k.a. transcriber, translator, or editor) for the text.
- *ucsize* – The element expresses the number of bytes the uncompressed archive takes up on disk. It comes from the default output of the *zipinfo* command (version 2.40).
- *wordcount* – A count of the words in the uncompressed archive. This value is determined through the use of the *wc* command.

3. Metacard

The metacard format and elements described in this paper were created for 15,511 books of PG. Of those texts there were significant problems with approximately 3 percent of the texts. The problems were caused by incomplete, inconsistent, or incorrect internal metadata, characters outside the range of the current operating system supported character sets corrupted files or archives, non- textual formats such as pictures or audio.

²The checksum created in this situation was done using Secure Hash Algorithm 1 (SHA1). The *sha1sum* 160-bit checksum (as described in FIPS-180-1) was calculated using the *sha1sum* command that is included with *coreutils* 4.5.3.

The program for creating the metacards was written using the Perl programming language running under RedHat AS 3.0.

The following table (Table 2) details the size of the PG data set that was analyzed for this project.

Data	Count	Data in MB Compressed	RDF Assertions	Words in Billions
ebooks	15,511	16155	N/A	8.3
metacards	15,022	63	912,806	N/A

Table 2: Data set

To help put the data analyzed into proper perspective, the full metacards required around two hundred megabytes of disk space. The RDF assertions made by the metacards numbered 912 thousand.

3.1. Sample Metacard

In this section, the creation of the sample metacard found in Figure 1 is dissected element by element. In this metacard example as well as the rest of the metacards created for this effort, the first seven lines are referred to as the prologue and establish the namespaces for the tags that follow. Following the prologue section, the *book:Book* tag is a container element that holds the rest of the metacard values. The *book:Book* element was chosen to facilitate ease in integrating the model in Siderean’s Seamark server. In the metacard example provided, the container element name does not have a large significance.

While gathering the information required for the creation of the sample metacard, most of the second order metadata was easily discovered in the first thirty-eight lines of the ebook file. Such elements as *dc:title*, *dc:language*, *dc:creator*, *pg:characterSet*, and *dcterms:available* were found in the ebook with lines starting with the words Title, Language, Author, Character set encoding, and Release Date respectively. In addition, the *etext* number was discovered on the Release Date line in the PG ebook.

Although the content for the elements mentioned above were fairly straightforward, the content for the content of the *dc:producer* element was somewhat inaccurately placed within the layout of the file. The content for this element was detected on a line following the words ‘*Transcribed by*’. It would have been more accurate if ‘*Transcribed by*’ preceded the line stating “*** START OF THE PROJECT GUTENBERG EBOOK, A HORSE’S TALE ***” (PG, 2006) since presumably the author, Mark Twain, did not have the assistance of transcriber, David Price. David Price’s email address followed his name.

The remaining elements in the *pg* namespace were determined using the commands as explained in section 2.4 titled ‘extending metadata elements’.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:book="http://www.siderean.com/ia/ns/bookdemo/"
  xmlns:pg="http://iama.rrecktek.com/daml/ont/pg#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/" >

  <book:Book about="ftp.archive.org/pub/etext/etext97/hrstl10.zip">
    <dc:title>A Horse's Tale</dc:title>
    <dc:language rdf:resource="http://skosaurus.rrecktek.com/ont/language#eng"/>
    <dc:creator rdf:resource="http://skosaurus.rrecktek.com/ont/author#mark_twain"/>
    <pg:charset rdf:resource="http://skosaurus.rrecktek.com/ont/character_set#US-ASCII"/>
    <dcterms:available rdf:datatype="http://www.w3.org/2000/10/XMLSchema#date">1997-10</dcterms:available>
    <pg:etext>1086</pg:etext>
    <pg:producer>David Price</pg:producer>
    <foaf:person rdf:parseType="Resource">
      <foaf:name>David Price</foaf:name>
      <foaf:mbox rdf:resource="mailto:ccx074@coventry.ac.uk"/>
    </foaf:person>
    <pg:linecount rdf:datatype="http://www.w3.org/2000/10/XMLSchema#int">2365</pg:linecount>
    <pg:wordcount rdf:datatype="http://www.w3.org/2000/10/XMLSchema#int">19257</pg:wordcount>
    <pg:charactercount rdf:datatype="http://www.w3.org/2000/10/XMLSchema#int">107174</pg:charactercount>
    <foaf:sha1>386126b01230dd062894742701cb208c525471db</foaf:sha1>
    <pg:ftype>ASCII English text, with CRLF line terminators</pg:ftype>
    <pg:fcount rdf:datatype="http://www.w3.org/2000/10/XMLSchema#int">1</pg:fcount>
    <pg:csize rdf:datatype="http://www.w3.org/2000/10/XMLSchema#int">44497</pg:csize>
    <pg:ucsize rdf:datatype="http://www.w3.org/2000/10/XMLSchema#int">109539</pg:ucsize>
    <pg:cratio rdf:datatype="http://www.w3.org/2000/10/XMLSchema#int">59.4</pg:cratio>
  </book:Book>
</rdf:RDF>

```

Figure 1: Sample Metacard for ebook 'A Horse's Tale'

3.2. The Range of Metacard Values

In this section, the dataset encompassing the majority of the PG ebooks is characterized by the four characteristics of language, authors, character set, and compression ratio.

3.2.1. Language

Language could only be determined conclusively in 75 percent (11,288) of the 15,022 texts in our data sample. The texts contained content in 25 different languages. The languages translate as follows: 91 percent (10,379) of texts were in English, 4 percent (468) in French, 2 percent (324) in German and the remaining languages were represented in less than 20 files each.

3.2.2. Authors

The 15,022 texts analyzed came from 5,225 different authors. Mark Twain is credited with 132 works, and the second most prolific author was Honore de Balzac with 119 publications.

The generic label 'Various authors' was on 7 percent (1,133) of texts, and less than one percent (120) of texts was labeled anonymous. The top 100 authors accounted for 28 percent or 4,184 of the texts.

3.2.3. Character Set

The 15,022 texts had 155 different file type labels. The largest category of character sets was "ASCII English text, with CRLF line terminators" which accounted for 82 percent (12,369) of the texts examined.

3.2.4. Compression Ratio

Out of the 15,022 texts, 93 percent (13,999) of the archive files were compressed between 56 to 66 percent.

4. Conclusion

As explained in the preceding discussion of this project, metadata is invaluable to researchers. Creation of metadata is worthwhile effort for data producers and consumers alike.

When metadata cannot be determined through a programmatic means, discovery or correction of poor, inaccurate or absent metadata can involve significant labour. Lexical researchers can save considerable time and effort if community expectations change to reflect the need for accurate machine readable information depicting lexical resources.

5. Future Direction

The metadata cards described here depict only an initial set of useful metadata that can be generated programmatically. Metadata for text based lexical resources can be extended further to include other information such as the string frequencies for each of the terms in an ebook. Other useful types of metadata could include measurements that indicate the complexity of a written work. Complexity measurements might include a SMOG Index, a Flesch-Kincaid score, a Gunning-Fog Index, or a Coleman-Liau Index. These measurements characterize the understandability for a piece of writing. Measurements of this type lend themselves to a programmatic analysis which could provide a richer understanding of language.

6. Acknowledgements

I would express my gratitude to Olga Lorincz-Reck, Ruth A. Reck, and Kenneth Sall who were all kind enough to lend their support in the writing of this paper. Mike DiLascio of Siderean Software was instrumental in permitting the use of Seamark Server version 4.0 for faceted navigation of the RDF metacards.

7. References

- Beckett, D., Miller, E., & Brickley, D. (2002). *Expressing Simple Dublin Core in RDF/XML*. Retrieved from: <http://dublincore.org/documents/dcmes-xml/>
- Darwin, Ian F. (1999). *Manpage for file*. Retrieved from: <http://man.he.net/?topic=file§ion=all>
- DCMI Usage Board (2003). *Metadata terms*. Retrieved from: <http://dublincore.org/documents/2003/03/04/dcmi-terms/>
- FIPS 180-1 (1995). *Secure Hash Standard*. Retrieved from: <http://www.itl.nist.gov/fipspubs/fip180-1.htm>
- FOAF (2006). *The friend of a friend project*. Retrieved from: <http://www.foaf-project.org/>
- Free Software foundation (2002). *Manpage for wc*. Retrieved from: <http://man.he.net/?topic=wc§ion=all>
- ISLE (2003). *International Standard for Language Engineering*. Retrieved from: <http://www.mpi.nl/ISLE/index.html>
- ISO 15836 (2003). Available at: <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=37629&ICS1=35&ICS2=240&ICS3=30>
- ISO 639-1 (2002). Available at: <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=22109&ICS1=1&ICS2=140&ICS3=20>
- ISO 639-2 (1998). Available at: <http://www.loc.gov/standards/iso639-2/langhome.html>
- OLAC (2006). *Open Language Archives Community*. Retrieved from: <http://www.language-archives.org/>
- Oracle 10g (2006). *Oracle Database 10g Downloads*. Retrieved from: <http://www.oracle.com/technology/software/products/database/oracle10g/index.html>
- NISO Z39.85 (2001). Available at: <http://www.niso.org/standards/resources/Z39-85.pdf>
- Perl (2006). *Practical Extraction and Report Language*. Retrieved from: <http://www.perl.com>
- PG - Project Gutenberg (2006). *Free eBooks*. Retrieved from: <http://www.gutenberg.org/>
- PG (2006). *A horse's tail*. Retrieved from: <ftp.archive.org/pub/etext/etext97/hrstl10.zip>
- RDF (2004). Resource Description Framework. Retrieved from: <http://www.w3.org/RDF/>
- Redhat (2006). *Red Hat Enterprise Linux AS*. Retrieved from: <https://www.redhat.com/rhel/details/servers/>
- RFC 1766 (1995). *Tags for the Identification of Languages* Retrieved from: <http://www.faqs.org/rfcs/rfc2046.html>
- RFC 2046 (1996). *Multipurpose Internet Mail Extensions Part Two: Media Types* Retrieved from: <http://www.faqs.org/rfcs/rfc2046.html>
- Roelofs, Greg (2002). *Manpage for zipinfo*. Retrieved from: <http://man.he.net/?topic=zipinfo§ion=all>
- Siderean Software (2005). Retrieved from: <http://www.siderean.com/>