Flexible annotation of digital literary text corpora with RDF

Andrew U. Frank, Andreas Dittrich

TU Wien, Department of Geodesy and Geoinformation
Gusshausstrasse 27-29/E120.2
A-1040 Vienna, Austria
frank@geoinfo.tuwien.ac.at

1 Introduction

The standardized Resource Descriptor Framework (RDF) can be used for the annotation of text corpora in the humanities. Annotations in text corpora are currently mostly done with standardized tags (e.g. Penn Treebank[11], Stuttgart Tuebingen POS codes[15]), in either sequential files with simple formats per line (e.g. the Portuguese fairy tale corpus 9), or via XML encoding. Other important collections of texts have a nearly uniform format (e.g. the texts in the Gutenberg project) and can therefore be included in systematic studies with little manual effort required.

We started experiments for building corpora for computational comparative literature research with annotations encoded in RDF, in order to understand the practical requirements and limitations of possible technical solutions. RDF overcomes one of the most vexing obstacles of annotating complex texts with network structures. The annotation of a textual structure (sections, paragraphs, sentences...) and layout (pages) requires the combination of two hierarchies, resulting in a network. This is required in many projects where the layout of the text on the pages is relevant.

The focus of our experiments is on comparative literary studies, i.e. comparing texts for literary analysis. The requirements are somewhat different for projects where critical editions of a text are used primarily; TEI serves well for such projects, and discussions of future developments are ongoing in the TEI community[1] . We suggest RDF as a widely recognized standard which can be used to annotate text corpora in a flexible and connectable way. It has the advantage of utilizing the widely used XML format (and can be encoded in XML, if this is

[1] see contributions for TEI conference 2015 by
Pierazzo, Elena: "TEI: XML and Beyond" and
Ciotti / Tomasi / Vitali: "An ontology for the TEI (Simple): one step beyond" .

desired), as it is was designed to connect the extremely large number of documents on the web in a searchable network.

In the next section, the shortcomings of XML for literary annotations and the characteristics of RDF are introduced. The third section describes briefly three experimental projects, and the concluding section argues for RDF as the most promising method for flexible annotations of literary texts which can be linked to other data.

2 Annotation languages:

Annotations add additional data to the text – similar to handwritten glosses in the margin of a printed volume. To make annotations useful, their contents must be structured. Two standardized methods are commonly available:

XML Markup languages go back to the early years of computer technology (e.g. RUNOFF and TEX), and were standardized and extended with the advent of the web. The eXtended Markup Language (XML) is a language that can structure text in a hierarchical tree[2]. Data in XML encoding is flexible, easy to parse, and in principle human-readable. There are many efficient tools for processing it, and tag sets can be adapted to special requirements of application domains, including, but by far not limited to, NLP and documenting different version of a text with the TEI tag set(**author?**) [13].

The major strength of XML, namely the hierarchcial structure which allows very efficient processing, is also its major limitation. Representing non-hierarchical situations is difficult, e.g. the structure of pages in a newspaper, each containing multiple articles or articles spanning multiple pages[3].

RDF The extension of the web to a semantically linked collection of documents[2] has resulted in a standardized method of representing arbitrary networks of linked data [10]. An RDF-encoded collection can be thought of as a labeled graph (i.e. a network), where the nodes are the objects and the labels describe the relations between the objects. The fundamental representation is the triple: object – property – value, describing a named link between two nodes. The nodes stand for uniquely identified concepts which are further described with values, which are either simple values (e.g. names encoded as character strings) or links to other objects (encoded as the identifiers of these objects)[3]. Representations which are easier for human comprehension do exist (e.g., the Turtle format[4]). SPARQL is the query language for collections of RDF triples, searching the graph. It is standardized[5], very general, and syntactically related to SQL. A number of very efficient implementations

[2] http://www.w3.org/TR/REC-xml/

[3] http://www.tei-c.org/release/doc/tei-p5-doc/de/html/NH.html

[4] http://www.w3.org/TR/turtle/

[5] http://www.w3.org/TR/sparql11-query/

exist, which permit searches for objects fulfilling complex conditions, connecting values, and values in other linked objects.

Benchmarks are reported for trillions of triples with modest hardware requirements[6].

Many resources available on the web utilize the RDF format, e.g. a large part of wikipedia (with more than 3 million concepts), a collection describing 7 million geographic features, and many others. These data can be connected – automatically or manually – with units in the literary text. The advantages of RDF for corpus annotations are:

1. RDF builds graphs and captures a network of links of non-hierarchical situations without difficulties (e.g. textual and layout relations can be represented on equal footing).

2. RDF can represent both NLP granularity (a word in a text as a linked element) and "humanities granularity" (a document, a chapter, or a text block like a paragraph).

3. RDF is optimized for graph search in networks of billions of elements (many NLP operations are graph searches, e.g. finding Hearst (1992) patterns).

Several authors have advocated the use of RDF for NLP encoding (e.g **(author?)** 7). Others have shown how the combination of NLP and RDF can be used to build resources for other disciplines (e.g. **(author?)** 14, 16). A text with treebank tags, dependencies, and coreferences produces approximately 10 triples for each word, which means that current large corpora (BNC, ANC) yield less than a trillion triples when fully annotated and triplified.

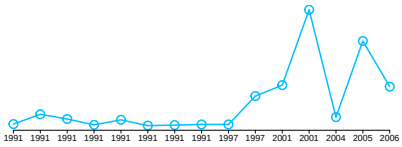
3 Experiments

A text corpus in general is - in the language of software engineering - an abstract data type with methods to

- add and remove texts to the corpus,
- obtain information about the texts included, and
- query the corpus with a flexible query language.

When building an annotated corpus, one must decide (i) on the units to annotate and (ii) the information that should be annotated. The experiments we report are representative for three typical forms of inquiry of comparative literature; each

[6][http://www.w3.org/wiki/RdfStoreBenchmarking
https://www.w3.org/wiki/LargeTripleStores#Oracle_Spatial_and_Graph_with_Oracle_Database_-_1.08_Trillion_triples_.28edges.29]
http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/results/V7/



(a) A chronological ordered frequency of occurrences of real place name references

(b) Named places in Vienna mentioned in the text

motivates a specific requirement, but we find that all requirements are present in most literature studies to varying degree:

study of the *œuvre of a single author*: annotation on a high semantic level to study the use of space, time, and persons in the author's œuvre;

study of a *literary genre*: annotation of the linguistic structure of text (parts of speech, dependencies, correferences etc.) to analyze style;

large *collections of texts* with bibliographic data and annotations of different levels of granularity, to compare stylistic and vocabulary characteristics.

3.1 A corpus-based examination of the œuvre of a single writer

Goal Subject of the examination is the œuvre of the Austrian author Ilse Aichinger, which is the object of the project `:aichinger:`. The goal is to briefly show (a) its conceptual framework and basis in the literary studies, and (b) its technical methods[7].

The focus of the study is on the spatiality of Aichinger's texts. In her work, "places carry the plot" [1], and places are important to her in all her writings, spanning more than half a century: they vocalize memories of her experiences during the Second World War in Vienna. "The places, which we looked at, look at us," as she writes in her prose-poetic short text "City Center" („Stadtmitte"[\autocite[11]{AichingerK}]). It is assumed that the places trigger a process of remembrance and evoke collocated events from different periods. Nevertheless, it is only in Aichinger's later texts that references to recognizable place names become frequent, as was shown during exploratory analysis. The frequency of the occurrence of real place names (identified as words which end with "*gasse", "*straße" or "*platz") varies enormously throughout her work., as is shown in figure1a.

Approach To construct the corpus systematically, we had to (a) include the text from an authoritative edition, (b) anotate each paragraph, and (c) identify places, persons and times. It seemed to us that a paragraph (or a similarly-sized unit of text) is the optimal granularity for the intended literary analysis. The text therefore

[7] more detail can be found at http://gams.uni-graz.at/o:dhd2015.v.032

had to be broken up into paragraph units, and annotations linked to paragraphs. This is in contrast to many corpora that are built to produce annotated editions of literary text and are annotated with text variants etc. linked to words or short sequences of words and tagged with TEI tags.

The steps are:

1. Obtain a text with simple character encoding in UTF8 from scanning and optical character recognition (OCR) processing.
2. Manually mark the different parts of the text as titles, subtitles, tables of content, and break text up into paragraphs.
3. Manually annotate the paragraphs for places, persons, and times mentioned.
4. Convert the text to RDF format and insert it into a SPARQL endpoint.

We found it advantagous to just annotate the persons, places, and times within the paragraphs with triples (paragraph-id, person-relation, person-id), where person-id was produced on the spot (e.g. theFishmonger). A similar procedure was used for the places and times. Adding additional details and links to other data, e.g. dbpedia, or addresses to link with Google Maps is better left for later.

Analysis is possible using SPARQL; for example, one can retrieve all paragraphs where a location is mentioned with the following query, which finds all google names gname and the title of the text tit for all locations place mentioned in a paragraph para in the text :

```
SELECT ?tit   ?gname WHERE {
 ?txt lit:titel ?tit .
 ?para lit:in ?txt .
 ?para lit:ort ?place .
 ?place lit:google ?gname .
}
```

From this data, a map can be produced using Google Maps and Fusion Tables[8]. The output for part of a volume of the work and places mentioned in Vienna is shown in Figure 1b.

3.2 Ontology in literary text

Goal We wanted to test the hypothesis that different literary texts use a different "conceptualization of part of reality" [5]. We constructed a corpus of literary texts, for which we assumed that the ontologies differ. The texts are mostly from the Gutenberg project and include fairy tales (from Grimm, Hauff, the Arabian Nights, and similar) but also other texts with non-standard ontology (e.g. science fiction). Some fairy tales differ in their ontology clearly from the real world as we experience it daily. As a first - easy - target, we want to identify the stories in which animals act as rational agents and communicate verbally. Fairy tales should prove

[8] https://support.google.com/fusiontables/answer/2571232

a relatively easy genre for such an analysis, and we expect to identify the class of fairy tales where animals act as rational agents automatically.

Approach Texts are manually broken into paragraphs marked up with a schema similar to the previous project. A version of the text is marked in a way which passes transparently through the CoreNLP suite and can be identified in the output. The text paragraphs are thus linked with the sentence granules of CoreNLP. The translation of the XML-tagged output to RDF and loading it in a SPARQL endpoint is straightforward. Compared to the processing time required for NLP, the conversion and loading into the RDF storage is negligible (a few minutes). Most time-consuming is the linguistic parser and tagger, which for a text of 100,000 words on an (slow) PC takes approximatively one hour; the other steps require less than a minute and load 1.5 billion triples.

The ontology is hidden in the text [4], and we extract from the treebank annotations for example the capabilities of animals for verbal speech, as indication of ontological commitments in a fairy tale. The plan is to identify (human) persons and the verbs of rational actions typically associated with them. The RDF-encoded POS can be linked with the RDF-encoded wordnet[9] and RDF Framenet [8]. The tales in which animal agents act rationally, i.e. are subject to active forms of verbs of rational action, are the desired class.

To identify the "rational animals as agents" class of fairy tales automatically and compare it with human judgment is a first step in an effort to understand the ontological differences between texts. To achieve this goal, the analysis must combine observations at every structural level of a text - from the word to groups of texts - and multiple methods of analysis (lexical, grammatical, or narrative structure).

3.3 "All methods for all texts"

Goal This experiment shows how to use Big Data methods for computational comparative literature. Comparative literature shuld be corpus-based in order to document the texts included in a study, and it must include a fixed set of analytical methods which are applicable to the texts in the corpus. The application of a fixed collection of methods to a fixed corpus of text results in comparable and repeatable results.

A corpus from as many literary texts as we could get hold of is assembled, and a collection of digital analysis methods are applied to each text. The methods vary from simple counts of words, to methods of comparing vocabulary or use of syntactic constructions, etc. Any method published with sufficient detail to be implementable, e.g. from vocabulary or syntactic similarity, to use of space and time in the text or narratology [12].

[9] http://wordnet-rdf.princeton.edu/

Each method produces a characteristic of the text analyses, and for each text, a vector of characterizations is obtained. Cluster analysis of these vectors results in various groupings which are then used for further studies by comparative literature researchers.

Technical solution For this project, the possibility of RDF to include bibliographic details of the texts included in the corpus is crucial for the handling of a very large number of texts, each with its own vector of characterizations. The results of NLP processing at the word and sentence level are used for style analysis; vocabulary analysis, syntactic structure statistics and similar apply to larger units.

The methods will interface the corpus with a SPARQL query; tools to access a SPARQL endpoint and to process the data are available in most current computer languages, giving maximum freedom in the implementation of observation methods. The key is the fully automated application of the methods to texts in the corpus, which can be repeated if new methods are added or existing methods are changed. This gives reliable and repeatable evaluations of the texts.

4 Conclusion

A common set of requirements emerges from the three different experiments:

- multiple granularity of the analysis and appropriate subdivisions of the text,
- network links between text units,
- automated processing with minimal human intervention only when a text is integrated in the corpus.

A corpus-based analysis has the advantage of generating reproducible results. A corpus useful for the humanities, e.g. for literature analysis, must merge the Natural Language Processing results, which are mostly expressed at the word and sentence granularity, with the structuring tools for literary texts in paragraphs, chapters etc., but also for the layout on pages. Each text must have the relevant bibliography data, and many texts must be available for concurrent analysis. A corpus for literary analysis, especially for comparative literature, must include many texts, possibly from multiple languages; a corpus for literary analysis is different from a corpus constructed for editorial work and publishing critical editions, where a single œuvre is included.

In the language of software engineering, a corpus can be seen as a abstract data type with methods to

- insert and remove texts,
- obtain bibliographic information about the texts included , and
- query the texts with a flexible query language.

The construction of corpus software can begin with readily available open-source software, mostly an RDF triple store (e.g. 4store[10]), extended with smaller, discipline specific programs to convert input documents into an RDF format with the necessary annotations. The analysis programs access the store using SPARQL in the same way administrative programs access databases using SQL and may store the results again as RDF triples in the same store. This layered structure [17] is fundamental in software engineering, and is probably the major contribution to the success of the www; it could be used to build the core software to manage corpora for diffferent application domains in digital humanities. The particulars of different domains and applications are then relatively thin and inexpensive layers which are stacked on top of the foundations. For example, a program to convert the result of the Stanford NLP processor to RDF is a dozen pages long and can be reused with small adaptations for treebanks of other languages.

The generalities of many applications of corpus techniques in digital humanities have certain requirements:

- The analysis of the corpus must be possible at different levels of granularity of the text (e.g. paragraph, page, chapter, volume or multi-volume work, but also pages with their layout), and with different subsets of the corpus.
- It is necessary that treebank tags and other results from Natural Language Processing are available for each subset of a text, such that arbitrary larger subsets from the corpus can be analyzed.

It is likely that many additional widely usable tools can be identified. For example, the annotation of the Aichinger corpus could have been advanced by using a Named Entity Recognizer (e.g. the output form the CoreNLP) to annotate each paragraph with the entities automatically recognized, and to link them with a gazetteer for cartographic output or a list of places of historic times (e.g. for the medival period[11]), reducing further the amount of human-produced annotations. The same process is likely useful for preprocessing data for historic research.

The potential for connection of annotations with other sources is important: in the Aichinger project, a connection to Google Maps (for the production of interactive maps), and to the movie database[12] has proven valuable in assisting the literary analysis.

Some manual preparation of texts to be included in a corpus cannot be avoided. Not only are there non-relevant text parts to delete, but there is also the implicit structure of the text with titles and subtitles, which must be marked up to be preserved in the corpus; the use of directories and subdirectories can achieve the same end, but is far less flexible. With a brief markup, taking a few minutes for a text, this structure can be identified.

[10] http://4store.org/

[11] http://www.leeds.ac.uk/arts/info/125136/international_medieval_bibliography

[12] http://datahub.io/de/dataset/linkedmdb

We found that RDF is flexible and efficient; the openness of RDF and the ease of connecting one RDF collection with other resources is an additional benefit. We recommend standardizing a simple markup language to prepare texts for their conversion into RDF resources.

Literatur

[1] Ilse Aichinger. *Zu keiner Stunde*. Fischer, 1991.

[2] Tim Berners-Lee, James Hendler, Ora Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[3] Peter Pin-Shan Chen. The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, 1976.

[4] Philipp Cimiano, Christina Unger, and John McCrae. *Ontology-based interpretation of natural language*, volume 7. Morgan & Claypool Publishers, 2014.

[5] Thomas R Gruber et al. Toward principles for the design of ontologies used for knowledge sharing. *International journal of human computer studies*, 43(5):907–928, 1995.

[6] Marti A Hearst. Automatic acquisition of hyponyms from large text corpora. pages 539–545, 1992.

[7] Sebastian Hellmann, Jens Lehmann, Auer, Sören, and Martin Brümmer. Integrating nlp using linked data. In *The Semantic Web–ISWC 2013*, pages 98–113. Springer, 2013.

[8] Nancy Ide. Framenet and linked data. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore*, volume 1929, pages 18–21, 2014.

[9] Paula Vaz Lobo and David Martins De Matos. Fairy tale corpus organization using latent semantic mapping and an item-to-item top-n recommendation algorithm. In *LREC*, 2010.

[10] Frank Manola, Eric Miller, Brian McBride, et al. RDF primer. *W3C recommendation*, 10(1-107):6, 2004.

[11] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[12] Vladimir Jakovlevič Propp, EM Meletinskij, and Christel Wendt. *Morphologie des Märchens*. Carl Hanser Verlag, 1972.

[13] Adam Przepiórkowski. TEI P5 as an XML standard for treebank encoding. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT8)*, pages 149–160, 2009.

[14] José Saias and Paulo Quaresma. Using NLP techniques to create legal ontologies in a logic programming based web information retrieval system. In *Workshop on Legal Ontologies and Web based legal information management of the 9th International Conference on Artificial Intelligence and Law, Edinburgh, Scotland*, 2003.

[15] Anne Schiller, Simone Teufel, and Christine Thielen. Guidelines für das tagging deutscher Textcorpora mit STTS. *Manuscript, Universities of Stuttgart and Tübingen*, 66, 1995.

[16] Erich Schweighofer. *Semantic indexing of legal documents*. Springer, 2010.

[17] Hubert Zimmermann. OSI reference model–the ISO model of architecture for open systems interconnection. *Communications, IEEE Transactions on*, 28(4):425–432, 1980.