Production Publishing Considerations for Annotating the Scientific Literature

Ron Daniel, Elsevier Labs

2011-03-24
Introduction

• Publishers like Elsevier are using text-mining to:
  – Help answer research questions,
  – Direct material to interested readers, and
  – Extract disparate facts from broad swathes of the literature to create knowledgebases.

• Purely automatic techniques do not have needed level of accuracy.
  – Need to have a production workflow integrating automated and manual methods.
  – Cost-benefit tradeoffs will differ for different publications.

• Production workflows impose requirements beyond representing linguistic and ontological annotations.

• Requirements include:
  – Staff and Infrastructure that are XML-capable but not RDF-capable.
  – Making provision for debugging.
  – QA of tags (aka Provenance)
  – Communicating with contractors who send markup, and also suggest new ‘topics’ for the taxonomy.
  – Different workflows for different publications.
  – Simple handling of fragments.
  – The need to deal with unusual and evolving RDF infrastructure.

• This talk explains these requirements and discusses our initial experiences with an annotation format that was designed to deal with them.
XML-capable, not RDF-capable, Staff and Tools

Requirements & Background

- Elsevier, and its contractors, have used XML for years.
- Extensive XML-capable infrastructure, QA tools, repositories, etc.
- Some RDF knowledge, but very limited in proportion.

Decisions

- Use RDF-XML format, not others.
- Adjust the syntax to be as comfortable as possible to XML-knowledgeable staff while retaining a useful RDF model.
- Use XML Schemas to validate the syntax so that documents which validate will produce correct RDF.

Terminology:
Chunks of RDF metadata for content items are called “Satellites”.
Debugging

Requirements & Background

• Data will be transferred from suppliers to Elsevier and back.
• Problems will happen.
• People in the middle tracking down the problem will not have fancy tooling. Text editors will be common.

Decisions

• Include some redundant, human-readable, info in the satellites to help people technicians know what they are looking at.
• Some minimal metadata about the item: title, publication version, content type.
• Readable labels for subject codes:

```xml
<med:disease>
  <skos:Concept rdf:about="...code...">  
    <skos:prefLabel>Diabetes</skos:prefLabel>
  </skos:Concept>
</med:disease>
```

Not the minimal:

```xml
<med:disease rdf:resource="...code..."/>
```
Scoring

Requirements & Background

• Need to be able to provide lists of content items ranked according to their ‘score’ for their subject tags.

• Used in user interfaces, QA procedures, etc.

Decisions

• Provide ID on statement to be scored.
• Annotation Structure introduced here.

```xml
<rdf:Description rdf:about="...item...">
  <med:disease rdf:ID="stmt-1">
    <skos:Concept rdf:about="...code...">
      <skos:prefLabel>Diabetes</skos:prefLabel>
    </skos:Concept>
  </med:disease>

  <tag:relatedAnnotation>
    <tag:TaggingAnnotation rdf:about="#anno-1">
      <tag:annotatesStatement rdf:resource="#stmt-1"/>
      <tag:score>0.8939283</tag:score>
    </tag:TaggingAnnotation>
  </tag:relatedAnnotation>
</rdf:Description>
```
Requirements & Background

• Need more info about a tag than its score:
  – Who created it?
  – When?
  – What version/configuration of a tool?
  – Has it been reviewed?
  – If so, when and by whom?
  – Was it approved or rejected?
  – etc.

Decisions

• Use Annotation Structure
• Use pav:createdOn and createdBy instead of dc:date and creator to reduce confusion.

```xml
<tag:TaggingAnnotation rdf:about="#anno-1">
  <tag:annotatesStatement rdf:resource="#stmt-1"/>
  <tag:score>0.8939283</tag:score>
  <pav:createdOn>2010-07-23T15:45:00Z</pav:createdOn>
</tag:TaggingAnnotation>
```
Communication w/ Contractors

Requirements & Background

• Need to be able to tell suppliers about tagging errors so:
  – They can fix them.
  – They can remember them and not repeat them in the future if re-processing the same content
  – They can improve their testing and training sets.
• Suppliers need to tell publisher about potential new concepts to be put into the taxonomy.

Decisions

• Review history and corrections can be sent back in a TaggingAnnotation like shown before.
• Don’t want full history to accompany article, OK to leave stuff like revision history off basic tagging satellite.
• Still defining vocabulary suggestion procedure in co-op with Elsevier’s taxonomy maintenance team.
Fragment Handling

Requirements & Background

- Need to be able to address portions of resources:
  - to show what is being annotated
  - to show ‘evidence’ that led to a particular categorization
  - to show surrounding context in order to detect breaking edits
- Need a widely implemented method so that not-very-sophisticated partners can use available tools with minimum effort.

Decisions

- Use XPointer for XML documents, Media Fragments for image, video, audio.
  - XMLDocumentRegion, MediaRegion akin to AO Selectors
- xpath1() XPointer scheme so pointer can be dropped into any XPath engine and get back the section being annotated.
- Complex case – arbitrary string in document:
  
  ```xml
  <rgn:XMLDocumentRegion rdf:about="http://.../S0140-6736(95)90494-8#xpath-e(substring(id('sb-3')/p[2],15,6))">
    <dct:isPartOf rdf:resource="http://.../S0140-6736(95)90494-8"/>
    <rgn:startingOffset>15</rgn:startingOffset>
    <rgn:stringLength>6</rgn:stringLength>
    <rgn:matchedString>biopsy</rgn:matchedString>
    <rgn:prefixString>Testing of the </rgn:prefixString>
    <rgn:suffixString>indicated nothi</rgn:suffixString>
  </rgn:XMLDocumentRegion>
  ```
Novel Infrastructure – Linked Data Repository

Requirements & Background

- Idea for LDR was a graph store, not a triple store.
  - Would let people pull graphs of interest and put into a triple store for an application.
  - LDR would not provide much beyond retrieval of graph by URL.
- Plan did not survive contact with the customer.
  - Want to do lots of queries to get back minimum amount of most relevant info.

Decisions

- Have a common header for satellites that will provide minimal common info for working with graphs in the LDR.
- `<satHead>` is about the satellite, not the resource.
- Can provide info on synonymous URLs.
Lots More

Requirements & Background

Decisions

- W3C Media Fragment Identifiers
- XPaths to identify XML regions to be annotated.
- Other satellite types beyond TaggingAnnotation – Basic metadata, various micro-satellites for specific needs.
- Modular structure of namespaces
- XML Schema with fairly open content models for most namespaces.
- For each project’s specific satellites, a more rigorous XML Schema is defined so that suppliers know what to send and how to check it.
Contact Info

Ron Daniel       +1 619 208 3064       r.daniel@elsevier.com