PubAnnotation - a persistent and sharable corpus and annotation repository

Jin-Dong Kim and **Yue Wang**
Database Center for Life Science (DBCLS),
Research Organization of Information and Systems (ROIS),
2-11-16, Yayoi, Bunkyo-ku, Tokyo, 113-0032, Japan
{jdkim|wang}@dbcls.rois.ac.jp

Abstract

There has been an active development of corpora and annotations in the BioNLP community. As those resources accumulate, a new issue arises about the reusability. As a solution to improve the reusability of corpora and annotations, we present *PubAnnotation*, a persistent and sharable repository, where various corpora and annotations can be stored together in a stable and comparable way. As a position paper, it explains the motivation and the core concepts of the repository and presents a prototype repository as a proof-of-concept.

1 Introduction

Corpora with high-quality annotation is regarded indispensable for the development of *language processing technology (LT)*, e.g. *natural language processing (NLP)* or *textmining*. Biology is one of the fields which have strong needs for LT, due to the high productivity of new information, most of which is published in literature. There have been thus an active development of corpora and annotations for the *NLP for biology (BioNLP)*. Those resources are certainly an invaluable asset of the community.

As those resources accumulate, however, a new issue arises about the reusability: the corpora and annotations need to be sharable and comparable. For example, there are a number of corpora that claim to have annotations for protein or gene names, e.g, Genia (Kim et al., 2003), Aimed (Bunescu et al., 2004), and Yapex (Franzén et al., 2002). To reuse them, a user needs to be able to compare them so that they can devise a strategy on how to use them. It is however known that often the annotations in different corpora are incompatible to each other (Wang et al., 2010): while one is considered as a protein name in a corpus, it may not be the case in another.

A comparison of annotations in different corpora could be made directly or indirectly. If there is an overlap between two corpora, a direct comparison of them would be possible. For example, there are one[1], two[2] and three[3] PubMed abstracts overlapped between Genia - Yapex, Genia - Aimed, and Yapex - Aimed corpora, respectively. When there is no or insufficient overlap, an indirect comparison could be tried (Wang et al., 2010). In any case, there are a number of problems that make it costly and troublesome, though not impossible, e.g. different formats, different ways of character encoding, and so on.

While there have been a few discussions about the reusability of corpora and annotations (Cohen et al., 2005; Johnson et al., 2007; Wang et al., 2010; Campos et al., 2012), as a new approach, we present *PubAnnotation*, a persistent and sharable storage or repository, where various corpora and annotations can be stored together in a stable and comparable way. In this position paper, after the motivation and background are explained in section 1, the initial design and a prototype implementation of the storage are presented in section 2 and 3, respectively and future works are discussed in section 4.

2 Design

Figure 1 illustrates the current situation of corpus annotation in the BioNLP community, which we consider problematic. In the community, there

[1] PMID-10357818
[2] PMID-8493578, PMID-8910398
[3] PMID-9144171, PMID-10318834, PMID-10713102

Proceedings of the 2012 Workshop on Biomedical Natural Language Processing (BioNLP 2012), pages 202–205,
Montréal, Canada, June 8, 2012.

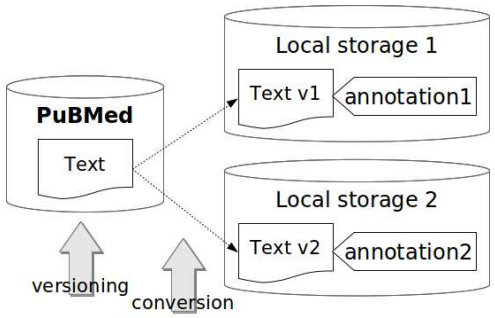

Figure 1: Usual setup of PubMed text annotation

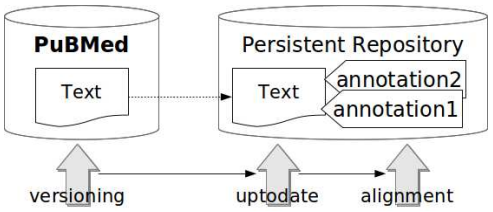

Figure 2: Persistent text/annotation repository

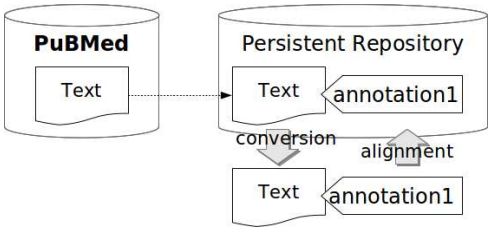

Figure 3: Text/annotation alignment for integration

are several central sources of texts, e.g. PubMed, PubMed Central (PMC), and so on. In this work, we consider only PubMed as the source of texts for brevity, but the same concept should be applicable to other sources. Texts from PubMed are mostly the title and abstract of literature indexed in PubMed. For an annotation project, text pieces from a source database (DB) are often copied in a local storage and annotations are attached to them.

Among others, the problem we focus on in this situation is the variations that are made to the texts. Suppose that there are two groups who happen to produce annotations to a same PubMed abstract. The abstract will be copied to the local storages of the two groups (illustrated as the local storage 1 and 2 in the figure). There are however at least two reasons that may cause the local copies to be different from the abstract in PubMed, and also to be different from each other even though they are copies of the same PubMed abstract:

Versioning This variation is made by PubMed. The text in PubMed is changed from time to time for correction, change of policy, and so on. For example, Greek letters, e.g., α, are spelled out, e.g., `alpha`, in old entries, but in recent entries they are encoded as they are in Unicode. For the reason, there is a chance that copies of the same entry made at different times (*snapshots*, hereafter) may be different from each other.

Conversion This variation is made by individual groups. The texts in a local storage are sometimes changed for local processing. For example, most of the currently available NLP tools (for English), e.g., POS taggers and parsers that are developed based on Penn Treebank, cannot treat Unicode characters appropriately. For such NLP tools to be used, all the Unicode characters need to be converted to ASCII character sequences in local copies. Sometimes, the result of some pre-processing, e.g. tokenization, also remains in local copies.

The problem of text variation may not be such a problem that makes the reuse of corpora and annotations extremely difficult, but a problem that makes it troublesome, raising the cost of the entire community substantially.

To remedy the problem, we present, a persistent and sharable storage of corpora and annotations, which we call *PubAnnotation*. Figure 2 illustrates an improved situation we aim at with PubAnnotation. The key idea is to maintain all the texts in PubAnnotation in their canonical form, to which all the corresponding annotations are to be aligned. For texts from PubMed, the canonical form is defined to be exactly the same as in PubMed. With the definition, a text entry in PubAnnotation needs to be updated (*uptodate* in the figure) as the corresponding text in PubMed changes (*versioning*). Accordingly, the annotations belonging to the entry also need to be re-aligned (*alignment*).

There also would be a situation where a variation of a text entry is required for some reason, e.g. for

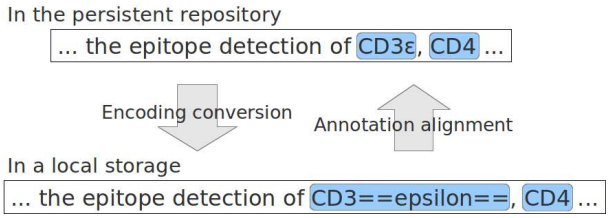

Figure 4: Text/annotation alignment example

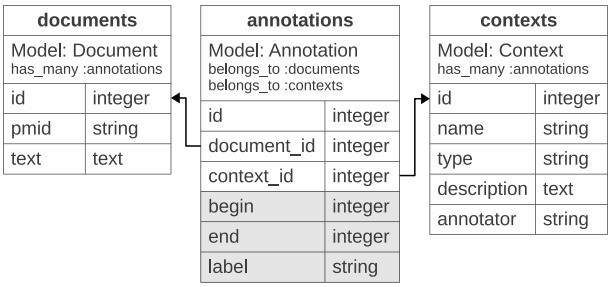

Figure 5: DB schema of persistent annotation repository

application of an NLP tool that cannot handle Unicode characters. Figure 3 illustrates a required process to cope with such a situation: first, the text is exported in a desired form (*conversion* in the figure); second, annotations are made to the text; and third, the annotations are aligned back to the text in its canonical form in the repository.

Figure 4 shows an example of text conversion and annotation alignment that are required when the Enju parser (Miyao and Tsujii, 2008) needs to be used for the annotation of protein names. The example text includes a Greek letter, ε, which Enju cannot properly handle. As Enju expects Greek letters to be spelled out with double equal signs on both sides, the example text is converted as so when it is exported into a local storage. Based on the pre-processing by Enju, the two text spans, `CD==epsilon==` and `CD4`, are annotated as protein names. When they are imported back to PubAnnotation, the annotations are re-aligned to the canonical text in the repository. In this way, the texts and annotations can be maintained in their canonical form and in alignment respectively in PubAnnotation. In the same way, existing annotations, e.g. Genia, Aimed, Yapex, may be imported in the repository, as far as their base texts are sufficiently similar to the canonical entries so that they can be aligned reliably. In this way, various existing annotations may be integrated in the repository,

To enable all the processes described so far, any two versions of the same text need to be aligned, so that the places of change can be detected. Text alignment is therefore a key technology of PubAnnotation. In our implementation of the prototype repository, the Hunt-McIlroy's longest common subsequence (LCS) algorithm (Hunt and McIlroy, 1976), as implemented in the `diff-lcs` ruby gem package, is used for the alignment.

3 Prototype implementation

As a proof-of-concept, a prototype repository has been implemented. One aspect considered seriously is the *scalability*, as repository is intended to be "persistent". Therefore it is implemented on a relational database (Ruby on Rails with PostgreSQL 9.1.3), instead of relying on a plain file system.

Figure 5 shows the database schema of the repository.[4] Three tables are created for *documents*, *annotations*, and (annotation) *contexts*, respectively. The annotations are stored in a stand-off style, each of which belongs to a *document* and also to an *annotation context* (context, hereafter). A context represents a set of annotations sharing the same set of meta-data, e.g., the type of annotation and the annotator. For brevity, we only considered PubMed as the source DB, and named entity recognition (NER)-type annotations, which may be simply represented by the attributes, `begin`, `end`, and `label`.

The prototype repository provides a RESTful interface. Table 1 shows some example which can be accessed with the standard HTTP GET method. A new entry can be created in the repository using a HTTP POST method with data in JSON format. Figure 6 shows an example of JSON data for the creation of annotations in the repository. Note that, the base text of the annotations needs to be passed together with the annotations, so that the text can be compared to the canonical one in the repository. If a difference is detected, the repository will try to align the annotations to the text in the repository.

[4] Although not shown in the figure, all the records are stored with the date of creation.

`http://server_url/pmid/8493578` to retrieve the document record of a specific PMID
`http://server_url/pmid/8493578.ascii` same as above, but in US-ASCII encoding (Unicode characters are converted to HTML entities).
`http://server_url/pmid/8493578/annotations` to retrieve all the annotations to the specific document.
`http://server_url/pmid/8493578/contexts` to retrieve all the annotation contexts created to the specific document.
`http://server_url/pmid/8493578/annotations?context=genia-protein` to retrieve all the annotations that belong to genia-protein context.
`http://server_url/pmid/8493578/annotations.json?context=genia-protein` the same as above, but in JSON format.

Table 1: Examples of RESTful interface of the prototype repository

```
{
 "document":
    {"pmid":"8493578",
     "text":"Regulation ..."},
 "context":
    {"name":"genia-protein"},
 "annotations":
    [
      {"begin":51,"end":56,
       "label":"Protein",
      {"begin":75,"end":97,
       "label":"Protein",
    ]
}
```

Figure 6: The JSON-encoded data for the creation of two protein annotations to the document of PMID:8493578.

4 Discussions and conclusions

The current state of the design and the prototype implementation are largely incomplete, and there is a much room for improvement. For example, the database schema has to be further developed to store texts from various source DBs, e.g., PMC, and to represent various types of annotations, e.g., relations and events. The issue of governance is yet to be discussed. We, however, hope the core concepts presented in this position paper to facilitate discussions and collaborations of the community and the remaining issues to be addressed in near future.

Acknowledgments

This work was supported by the "Integrated Database Project" funded by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

Razvan Bunescu, Ruifang Ge, Rohit J. Kate, Edward M. Marcotte, Raymond J. Mooney, Arun K. Ramani, and Yuk Wah Wong. 2004. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155.

David Campos, Srgio Matos, Ian Lewin, Jos Lus Oliveira, and Dietrich Rebholz-Schuhmann. 2012. Harmonization of gene/protein annotations: towards a gold standard medline. *Bioinformatics*, 28(9):1253–1261.

K. Bretonnel Cohen, Philip V Ogren, Lynne Fox, and Lawrence Hunter. 2005. Empirical data on corpus design and usage in biomedical natural language processing. In *AMIA annual symposium proceedings*, pages 156–160.

Kristofer Franzén, Gunnar Eriksson, Fredrik Olsson, Lars Asker, Per Lidén, and Joakim Cöster. 2002. Protein names and how to find them. *International Journal of Medical Informatics*, 67(13):49 – 61.

James W. Hunt and M. Douglas McIlroy. 1976. An Algorithm for Differential File Comparison. Technical Report 41, Bell Laboratories Computing Science, July.

Helen Johnson, William Baumgartner, Martin Krallinger, K Bretonnel Cohen, and Lawrence Hunter. 2007. Corpus refactoring: a feasibility study. *Journal of Biomedical Discovery and Collaboration*, 2(1):4.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic hpsg parsing. *Computational Linguistics*, 34(1):35–80, March.

Yue Wang, Jin-Dong Kim, Rune Sætre, Sampo Pyysalo, Tomoko Ohta, and Jun'ichi Tsujii. 2010. Improving the inter-corpora compatibility for protein annotations. *Journal of Bioinformatics and Computational Biology*, 8(5):901–916.