Finding Abbreviations in Biomedical Literature: Three BioC-Compatible Modules and Three BioC-formatted Corpora

Rezarta Islamaj Doğan, Donald C. Comeau, Lana Yeganova and W. John Wilbur

National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, 8600 Rockville Pike, Bethesda, MD 20894, MD, USA

*Corresponding author: Tel: 301 435 5926, E-mail: wilbur@ncbi.nlm.nih.gov

Abstract
This paper reports the use of BioC to address a common challenge in processing biomedical text information—that of frequent biomedical entity name abbreviation. We selected three different abbreviation definition identification modules, and used the publically available BioC code to convert these independent modules into BioC-compatible components that interact seamlessly with BioC-formatted data, and other BioC-compatible modules. In addition, we consider three manually-annotated corpora of abbreviations in biomedical text: the Ab3P corpus of 1250 PubMed abstracts, the BIOADI corpus of 1201 PubMed abstracts and the old MEDSTRACT corpus of 199 PubMed citations. Annotations in these corpora have been re-evaluated by four annotators and their consistency and quality levels have been improved. We convert them to BioC-format and describe the representation of their annotations. These corpora are used to test the three abbreviation finding algorithms. The BioC-compatible modules, when compared to their original form, have no difference in their efficiency, running time, or any other comparable aspects, so they can be conveniently used as a common pre-processing step for larger multi-layered text-mining endeavors. Code and data are available for download at the BioC site: http://bioc.sourceforge.net.

Introduction
The BioCreative¹ challenge evaluations since their inception in 2003 have been a community-wide effort for evaluating text mining information extraction systems applied to the biomedical domain. Given the emphasis on promoting scientific progress, BioCreative meetings have consistently sought to make available both suitable information extraction systems that handle life science literature and suitable “gold standard” data for training and testing these systems (1-4). The BioCreative IV Interoperability track² follows the guidelines established in previous BioCreative meetings, and specifically addresses the goal of interoperability—as a major barrier for wide-scale adoption of the developed text mining tools. As a solution, BioC (5) is proposed

¹ http://www.biocreative.org/
² http://www.biocreative.org/tasks/biocreative-iv/track-1-interoperability/
as an interchange format for tools for biomedical natural language processing. BioC is a simple XML format, specified by DTD, to share text documents and annotations. The BioC annotation approach allows many different annotations to be represented, including sentences, tokens, parts of speech, and named entities such as genes or diseases.

In this paper we present the contributions of our team to the BioC repository in the form of BioC-compliant modules that address the abbreviation definition detection task in biomedical text. These modules can be seamlessly coupled with other BioC code and used with any BioC-formatted corpora. We also present BioC-formatted corpora to test the abbreviation definition detection task, which can further be used with any other BioC-compliant tool for many biomedical natural language processing tasks.

**Abbreviation detection in biomedical domain**

The past twenty years have only seen an increase in the interest for automatic extraction of biological information from scientific text, and particularly from MEDLINE abstracts. One characteristic of these documents is the frequent use of abbreviated terms. Abbreviated terms appear not only in the scientific text, but they are also frequent in user queries requesting the retrieval of those documents. Two related specific issues are: 1. The high rate at which new abbreviations are introduced in biomedical texts, and 2. The ambiguity of those abbreviations. Existing databases, ontologies, and dictionaries must be continually updated with new abbreviations and their definitions. In order to help resolve this problem, several techniques have been introduced to automatically extract abbreviations and their definitions from MEDLINE abstracts (6-9).

Abbreviation identification is the task of processing text to extract explicit occurrences of abbreviation-definition pairs. The task requires both the identification of sentences that contain <short-form, long-form> candidate pairs from text, and identification of exact long-form and short-form boundaries. An important clue that is shared by abbreviation detection methods is the presence of parenthetical text and the assumption that parenthetical text signals the presence of an abbreviation definition. Two cases are distinguished:

a) long-form ‘(‘short-form’), and
b) short-form ‘(‘long-form’), with the first alternative being observed much more frequently in practice.

An abbreviation—a short-form—is a shorter term that represents a longer word or phrase, which often refers to an important biomedical entity. The definition—the long-form—is searched for in the same sentence as the short-form, often between the beginning of the sentence and the open parenthesis, for case a).
Abbreviation definition finding algorithms

1. The Schwartz and Hearst algorithm

The Schwartz and Hearst algorithm (7) decides about $<\text{short-form}, \text{long-form}>$ candidates using this rule: If the expression within parentheses contains more than two words then case b) is assumed, otherwise case a) is assumed. A short-form is verified to contain the right number of characters, contains at least one letter character and starts with an alphanumeric character. A long-form candidate is extracted from the string so that it contains at the most $\min(|SF|+5, |SF|*2)$ words, where $|SF|$ is the number of characters in the short-form. The long-form is always longer in size than the short-form. Next, starting from the right end of both strings, the algorithm traverses both strings right to left, matching the characters in the short-form to find the shortest long-form string. The matching of characters has to be sequential in order for both strings. With the exception of the first short-form character that has to match a character at the beginning of a word in the long-form candidate, the rest of the characters can match anywhere in the long-form string, as long as they are in order. The algorithm is very simple, very fast, and the results are very good.

2. The Ab3P Algorithm

The Ab3P algorithm, developed by Sohn et al. (6) is another pattern-matching approach to abbreviation definition detection. This algorithm defines 17 pattern-matching rules which the authors called strategies. Depending on the matching strategy and the length of the short form, they estimate an accuracy measure called pseudo-precision. Pseudo-precision provides a computed reliability estimate for an identified $<\text{short-form}, \text{long-form}>$ pair, without any human judgment. This algorithm is also very fast and provides high precision results.

3. The NatLAb algorithm

Rule-based methods, such as the Schwartz and Hearst algorithm and Ab3P, are successful at identifying abbreviation definition pairs with high precision. However, such approaches are unable to identify non-typical pairs, such as three dimensional (3-D), or out-of-order matches, such as melting temperature (T(m)). Machine learning methods have the potential of recognizing such non-trivial or irregular pairs and improving the recall, if enough training data is provided. NatLAb (Natural Learning for Abbreviations), developed by Yeganova et al., (8) is an example of such algorithm. NatLAb is a supervised learning approach whose features, inspired by the basic rules defined in Sohn et al., describe a mapping between a character in a Potential Short-form and a character in a Potential Long-form. However, in contrast to Ab3P, Yeganova et al., do not combine these features into hand-crafted strategies. They provide the learner with

---

3 http://biotext.berkeley.edu/software.html
all these features and feature pairs and let the training process weight them. Feature weights are then used to identify abbreviation definitions.

Converting into BioC

1. BioC-compliant modules

We found it straightforward to convert the original software tools for Abbreviation Definition Recognition into BioC-compliant tools. We would also like to point out that the original Schwartz and Hearst software is written in Java, the original Ab3P software is written in C++ and the original NatLab software is written in Perl. As a result, each implementation used a different BioC library; the necessary links were established so that Schwartz and Hearst algorithm could flow seamlessly with the rest of the BioC-Java code, the Ab3P algorithm with the BioC-C++ code, and NatLab with a SWIG-Perl- BioC implementation (10). The BioC-compliant algorithms differ from the originals in these main points:

- The BioC format includes the precise location of annotations in the original text. The original algorithms did not track the location of their recognized abbreviations. Retrofitting this tracking required considerable effort.
- Any text element in a BioC collection is considered a valid input string to inquire for abbreviation definitions. The module accepts BioC-formatted data, and searches for abbreviations regardless of whether text is organized in sentences, passages, or passages of multiple paragraphs. The precise text offsets are produced accordingly.
- The results are produced in BioC format and recognized abbreviations and their definitions can be compared with the output of any other BioC-compliant abbreviation definition recognition tool which uses the same keyfile. This is demonstrated below where we describe the BioC format of three independent abbreviation corpora. All three corpora are provided as input to the three BioC-modules, and results are compared in the Results section.

2. BioC-formatted corpora

The first step in preparing a given corpus into BioC format is deciding how to represent the information present in the corpus. Figure 1 illustrates the BioC format for abbreviation annotations that is used in the three abbreviation corpora used for this study. The corpora we considered for this task consist of annotations in the form of \(<\text{short-form, long-form}>\) pairs. To capture this, and make the corpora versatile for other possible biomedical information retrieval studies, we use this markup:

- For annotation elements, the \textit{infon type =ABBR} semantically identifies the annotations as abbreviations. This allows the possibility of having multiple layers of annotations on the same textual data, even including annotations on other entity types that may also overlap. All such annotations can be added without confusion.
- An additional infon element identifies two parts of an abbreviation: ShortForm and LongForm thus preserving the corpora original representation of an abbreviation definition as a \(<short-form, long-form>\) pair.
- Finally, a relation element reflects the pairing between a short-form and a corresponding long-form. The same infon type “ABBR” is repeated for the relation to make it easier at a semantic level to distinguish what is being annotated and how they relate together.

Documents may contain multiple abbreviation definitions. The location element links the definition to the exact textual coordinates it is mentioned, and also allows for defining a mention composed of multiple con-continuous substrings. Location information was not present in original annotations, so this is an enrichment over the original versions.

```xml
<annotation id="SF1014">
  <infon key="type">ABBR</infon>
  <infon key="ABBR">ShortForm</infon>
  <location offset="79" length="2"/>
  <text>FA</text>
</annotation>

<annotation id="LF1014">
  <infon key="type">ABBR</infon>
  <infon key="ABBR">LongForm</infon>
  <location offset="63" length="14"/>
  <text>Fanconi anemia</text>
</annotation>

<relation id="R1014">
  <infon key="type">ABBR</infon>
  <node refid="SF1014" role="ShortForm"/>
  <node refid="LF1014" role="LongForm"/>
</relation>
```

**Figure 2.** Illustration of abbreviation annotation in BioC format.

In order to test the abbreviation definition recognition modules, we converted three abbreviation definition recognition corpora to BioC format. These corpora are: the Ab3P corpus (6), the BIOADI corpus (9) and an earlier version of the Medstract corpus (11). The corpora consist of 1250, 1201 and 199 PubMed citations respectively, and the number of total (and unique) abbreviations annotated in each corpus is: 1223 (1113), 1720 (1491), and 159 (152), respectively, as shown in Table 1. To further highlight the inherent ambiguity in this task, there are 998, 1330, and 146 unique short-forms in the Ab3P, BIOADI and Medstract corpora respectively.

The original versions of these corpora consisted of text files where documents were separated by blank lines. For each document we were given a passage of text (in the Ab3P corpus this was divided into PubMed title and PubMed abstract lines, in the BIOADI corpus title and abstract lines were concatenated together), followed by a list of \(<short-form, long-form>\) pairs of abbreviations mentioned in the text. PubMed document IDs were given for the Ab3P and
BIOADI corpora, while Medstract documents had Journal citation information and their author list. In converting these corpora to BioC format first we identified PMIDs for all articles, and kept all relevant text for abbreviation definition detection (title and abstract). Next, we identified the correct offsets for each defined abbreviation in the corresponding text. This step included multiple occurrences of each definition within the same passage at times, as well as correct identification of multiple offsets for multiple substrings of some long-form definitions. Naturally, all three corpora went through another step of manual annotation, as many definitions were evaluated and discussed among the four authors. As a result, the final numbers, shown in Table 1, reflect a difference in the number of total annotations per corpus, when compared to the original publications, but we believe that our thorough review has produced better consistency and higher quality corpora.

Results

We tested the three abbreviation identifying modules on the Ab3P, BIOADI and Medstract corpora as shown in Table 1. Results are based on the new gold standard annotations in the three abbreviation corpora. When compared to the outputs of the algorithms original versions, the BioC-compliant modules produced the same results. The BioC versions, however, have the advantage of being easily combined with other BioC-compliant tools to produce a more complex biomedical text processing system.

Table 1. Results of BioC-compliant abbreviation detection modules when tested on BioC-formatted abbreviation corpora.

<table>
<thead>
<tr>
<th>Corpora</th>
<th>Ab3P</th>
<th>BIOADI</th>
<th>MEDSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of abstracts</td>
<td>1250</td>
<td>1201</td>
<td>199</td>
</tr>
<tr>
<td>Number of defined abbreviations</td>
<td>1223</td>
<td>1720</td>
<td>159</td>
</tr>
<tr>
<td>Unique number of abbreviations (across the whole corpus)</td>
<td>1113</td>
<td>1421</td>
<td>152</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ab3P Results</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.971</td>
<td>0.952</td>
<td>0.993</td>
</tr>
<tr>
<td>Recall</td>
<td>0.836</td>
<td>0.770</td>
<td>0.906</td>
</tr>
<tr>
<td>F-score</td>
<td>0.898</td>
<td>0.851</td>
<td>0.947</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shwartz&amp;Hearst Results</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.950</td>
<td>0.943</td>
<td>0.986</td>
</tr>
<tr>
<td>Recall</td>
<td>0.788</td>
<td>0.765</td>
<td>0.893</td>
</tr>
<tr>
<td>F-score</td>
<td>0.861</td>
<td>0.844</td>
<td>0.937</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NetLab Results</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.927</td>
<td>0.853</td>
<td>0.924</td>
</tr>
<tr>
<td>Recall</td>
<td>0.879</td>
<td>0.830</td>
<td>0.918</td>
</tr>
<tr>
<td>F-score</td>
<td>0.903</td>
<td>0.841</td>
<td>0.921</td>
</tr>
</tbody>
</table>
Conclusions
We present three easy-to-use, portable, BioC-compatible, interoperable abbreviation definition recognizing modules in biomedical text. The original tools corresponding to Ab3P, Schwartz and Hearst and NatLAb algorithms, have only been altered to read and produce the enriched BioC format. The new BioC-compatible modules faithfully preserve their original efficiency, running time power, or other complexity-related aspects, so they can be confidently used as a common pre-processing step for larger multi-layered text-mining endeavors.

We also present three BioC-formatted abbreviation definition recognition corpora that can be used to test the above modules, as well as to study new natural language processing tools. The new versions of the modules, as well as the accompanying corpora, are freely available to the community, through the BioC website: http://bioc.sourceforge.net.

Acknowledgements
This research was supported by the Intramural Research Program of the NIH, National Library of Medicine.

References