PBIE: A DATA PREPARATION TOOLKIT toward DEVELOPING a PARSING-BASED INFORMATION EXTRACTION SYSTEM

Junko Hosaka, Igor V. Kurochkin, Akihiko Konagaya

RIKEN Genomic Sciences Center Suehiro-cho 1-7-22 Tsurumi-ku Yokohama Kanagawa 230-0045 {jhosaka, igork, konagaya}@gsc.riken.jp

Abstract

We have developed a toolkit in which an annotation tool, a syntactic tree editor, and an extraction rule editor interact dynamically. Its output can be stored in a database for further use. In the field of biomedicine, there is a critical need for automatic text processing. However, current language processing approaches suffer from insufficient basic data incorporating both human domain expertise and domain-specific language processing capabilities. With the annotation tool presented here, a set of "gold standards" can be collected, representing what should be extracted. At the same time, any change in annotation can be viewed on an associated syntactic tree. These facilities provide a clear picture of the relationship between the extraction target and the syntactic tree. Underlying sentences can be analyzed with a parser which can be plugged in, or a set of parsed sentences can be used to generate the tree. Extraction rules written with the integrated editor can be applied at once, and their validity can immediately be verified both on the syntactic tree and on the sentence string by coloring the corresponding segments. Thus our toolkit enables the user to efficiently construct parse-based extraction rules.

1 INTRODUCTION

In the field of biomedicine, there is a strong demand for automatic text processing: scientists must keep up with information from a rapidly growing number of scientific journal papers. The extraction of information concerning protein-protein interaction events from written texts provides an example. Approaches in this area have used cooccurrence (Jenssen, 2001), templates and simple rules (Blaschke, 2001), full parsing (Yakushiji, 2001), or grammars developed for a neighboring domain, with modifications regarding e.g. semantic categories (Friedman, 2001). Overall, these approaches are limited to specific interaction sets, or suffer from sparse data, given the difficulty of collecting large amounts of appropriate data.

Data collection for common use is ongoing. However, the work is labor intensive and time consuming. Furthermore, the required corpora are domain-dependent. For example, the GENIA project¹ plans to provide an annotated corpus to the public. It is to be tagged with information concerning proteins, DNA, and RNA. It will also contain morphological information, mainly following definitions given in the Penn Tree Bank². The project started in 2001, and released 2,000 Medline³ abstracts annotated with both domain-specific and morphological information in March, 2003. Similarly, the BioCreative⁴ project prepared a training corpus and a gold standard for their contest, and annotated a number of journal papers on the human genome with functional information defined in the Gene Ontology⁵ with the help of domain specialists.

By contrast, in view of the difficulty of collecting the required data set in a limited time, and considering the lack of generality in template use, we instead apply general-purpose syntactic parsing. This approach is not fully dependent on the reference data.

From our previous work we learned that

- data should be annotated by domain specialists to specify extraction targets
- syntactic rules for general-purpose parsers require improvement

Accordingly, we have developed a **P**arsing-**B**ased Information Extraction (PBIE) toolkit. PBIE allows biologists to annotate from a biomedical point of view. It also helps linguists to improve syntactic rules, and to construct extraction rules that reflect biomedical expertise. Data prepared with the PBIE toolkit will serve as basic data for parsing-based information extraction.

The PBIE toolkit consists of two main tools: the evaluation tool and the data management tool. We have reported elsewhere on the PBIE evaluation tool, which provides annotation and tree editing facilities (Hosaka, 2003). To speed up the data preparation process, we have integrated parsing and extraction modules into the tool, and have renamed the resulting evaluation tool PBIE2.

In this paper, we give an overview of PIBE2, with a short description of its data management tool, and then focus on the two new modules: the parsing and extraction modules. Parsing results presented in this paper were obtained by using the Apple Pie Parser $(APP)^6$ as a plug-in.

PBIE2 works under Windows 2000/XP and requires Microsoft Internet Explorer 6.0 or higher. The data can be stored in Microsoft Access.

2 OVERVIEW of PBIE2

In PBIE2 we have combined three modules: an annotation module, a parsing module, and an extraction module. The annotation and parsing modules can each have a maximum

³ http://www.ncbi.nlm.nih.gov/PubMed/

⁴ http://www.pdg.cnb.uam.es/BioLink/BioCreative.eval.html

⁵ http://www.geneontology.org/

⁶ http://www.cs.nyu.edu/cs/projects/proteus/app/

http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/

² http://www.cis.upenn.edu/~treebank/home.html

of two windows. These windows can be used for evaluation by comparing two different extractions, e.g. comparing segments extracted manually by a domain specialist with segments automatically extracted by linguists, or comparing two different parsing results. PBIE2 also accommodates the plug-in of external parsers and an editor for extraction rule construction. Users can load a sentence file, run the parser, write extraction rules, and test the appropriateness of the rules. The program represents annotations using colored text and represents parsed sentences using syntactic trees. Any change in annotation can be simultaneously viewed on the syntactic tree. With the annotation tool, specialists in biomedicine typically mark phrases that should be extracted. Predefined categories should be used. Linguists can edit the syntactic trees by moving sub-trees, inserting new syntactic nodes, or deleting intermediate nodes. They can select a node name from the list of predefined names, or create new ones. The program has comments windows for both annotation and syntactic tree. The user can choose comments from the comment menu, or write texts freely.

The PBIE toolkit uses an XML format. Its data management tool prepares input data by selecting parsing and extraction pairs for comparison. The tool also converts data to and from the database. Comparing two extraction sets and taking one of them as a gold standard, the program calculates the performance (recall, precision and Fvalue). It also marks the unmatched phrases, which is helpful during human examination.

2.1 STRUCTURE of PBIE2

PBIE2 has an execution file, four files in XML format (.xml, .xpt), and a file for plugging in an external parser. The latter five files can be customized by the user, and the xpt file can be renamed. The file configuration is shown in Table 1:

Files	Contents	
PBIE2.exe	Execution file	
category.xml	List of extraction categories	
	Definition of start-up parameters	
comment.xml	List of comments	
nodenames.xml	List of POS/syntactic nodes	
pattern.xpt	List of extraction rules	
parser.dll	File for a parser plug-in	

Table 1: File configuration of PBIE2

The file category.xml contains an ontology specific to the current domain. It can also contain explanations of each ontological category, along with examples. These category explanations are used to label extraction targets. The current PBIE2 has eight categories, such as "Agent Compound", "Agent Process" and "Agent Protein". We prepared this set based on the preceding study of 400 sentences in cooperation with a biologist. The file nodenames.xml contains 70 node names used in the APP. They are basically the same as the node names defined for the Penn Tree Bank.

2.2 FILE FORMAT for PBIE2

PBIE2 takes text files (.txt) and XML files (.xst). A txt file should contain a list of sentences, with each sentence on

its own line. An xst file is a PBIE2 working file. The user can receive two kinds of syntactic parsing results, extracted phrases and comments for a single sentence. PBIE2 can convert text files to xst files, which can be used for parsing. The xst data structure is as follows. The input sentence is marked up with the tag "original":

```
<sentence id = s_id sentenceid = sent_id >
<original> original sentence </original>
       <parsed> parsed sentence </parsed>
        <extracted>
                <phrase type = ph_type start = s_no</pre>
               end = e no >
               extracted_phrase </phrase>
       </extracted>
       <parsed1 sign = pconf_on/off >
       parsed_sentence1 <parsed1>
        <extracted1 sign = econf_on/off >
               <phrase type = ph1 type start= s1 no
               end = e1 no >
               extracted phrase1 </phrase>
       </extracted1>
       <commentp> com_parsing </commentp>
       <commente> com_extraction </commente>
       <commentp1> com_parsing1 </commentp1>
       <commentel> com_extraction1 </commentel>
```

</sentence>

The tags "parsed", "extracted", "commentp" and "commente" belong to the first result set, and "parsed1", "extracted1", "commentp1" and "commente1" belong to the second set. The "start" and "end" tags indicate the position from the beginning of the sentence, and "sign" denotes whether the results have been confirmed.

2.3 STARTING PBIE2

PBIE2 provides seven tool combinations, as shown in Table 2. The user can select a combination using the startup menu, and can also change the combination while working in order to accommodate the current task. PBIE2 combines annotation and parsing results, each with a maximum of two windows supporting evaluation or comparison. We call these windows comparison tools. If there are two windows, one of them is read-only, and can be used only for viewing. In the tree editor, the user can run a parser which has been plugged in. PBIE2 also provides facilities for editing extraction rules:

No.	Tools
1	Extraction marking
2	Extraction comparison
3	Sentence tree editor
4	Sentence tree comparison
5	Extraction marking and sentence tree editor
6	Extraction and sentence tree comparison
7	Extraction rule editor, extraction and sentence
	tree comparison

Table 2: PBIE2 start-up menu

As shown in Table 2, there is only one tool combination which includes the extraction rule editor (combination number 7). As our information extraction method is based upon syntactic parsing results, the user must refer to the syntactic structure in order to construct extraction rules. To evaluate the validity of the rules, it is crucial to compare their results with the gold standard, or with results obtained using different rules.

The start-up parameter can be defined in the file category.xml, as shown in Table 1. Parameters control e.g. which tool should be used as the start-up tool and which tools should be available for the current task.

A sample XML specification including the definition for the extraction category "Agent" is shown below. Tool availability mode seven is specified, in which all seven tools can be used; the start-up menu is set to non-display mode; the starting tool number is seven; and the color editor is available. The values for the background color, (bckcolor) and the text letter color (txtcolor) are given in standard RGB numbers. However, rather than write this number, the user can choose the desired color from a color table. The colors are used for marking phrases in the marking window:

<?xml version="1.0"?> <categories mode="7" startup="0" tool="7" coloreditor="1">

<agent id="1" name="Agent" category="agent" bckcolor="16710867" txtcolor="0">

Agent

Example:

We find that ACK-2 can be activated by cell adhesion in a Cdc42-dependent manner. Agent: cell adhesion

Figure 1 shows the PBIE2 interface immediately after execution. The extraction category "Agent" is selected:

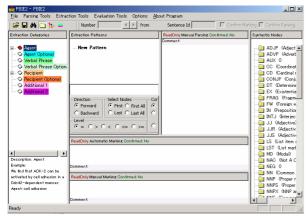


Figure 1: PBIE2 with the selected extraction category "Agent"

Syntactic nodes contain morphological and syntactic categories in abbreviated and full forms. These should be consistent with the node names used in the syntactic tree, except when new node names are created with the tree editor.

3 PARSING

To parse sentences using PBIE2, the user first loads a sentence list. There are two ways: one can open an xst file in PBIE2 format, or can open a txt file in which sentences are listed line by line, and then run the conversion command on it. Once the sentence list has been loaded, the parser which has been plugged in can be run. Figure 2 shows PBIE2 when a sentence list has already been brought in. Tool combination number 5 has been used in this example:

sample txt - PBIE2	_02
Eile Parser Extraction Tools Evaluation Tools Options About program	
Contract Contrect Contrect Contract Contract Contract Contract Contract Contre	Syntactic Nodes
<	CD. (Cardinal number) CONJP (Conjunction phr- DT (Determiner) EX (Existential there) FRAG (Froegen word) FW (Foreign word) FW (Foreign word) FW (Propertion or subor)
The enzyme can be activated by detergents in vitro and by short flight intervals in vitro , w indicates that changes in the membrane environment modulate trehelase activity under physiological conditions .	Same Area A
Parse Sentence	

Figure 2: Loading a sentence list

4 INFORMATION EXTRACTION

For the construction of extraction rules, information concerning direction, level, etc. in the syntactic tree can be used. To pinpoint target segments, the user can also employ xPath⁷ expressions.

4.1 EDITOR for EXTRACTION RULES

The extraction rule edit window consists of three sections. In the top section, rule structures are defined. Each rule consists of a set of patterns, and each pattern has a set of steps. Steps mostly correspond to the extraction categories listed on the left of the edit window. The user can select a category name from the list of predefined names by drag-ging-and-dropping. There are also certain special steps, such as the end marker step. Edit functions such as copy, paste, and delete can be performed by choosing the appropriate command from the menu that appears when the mouse is clicked in the region. Figure 3 shows PBIE2 when the edit menu has popped up:

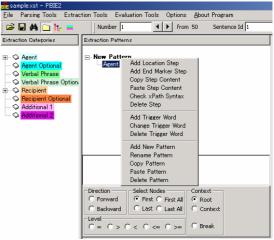


Figure 3: Extraction rule editor

In the middle section of the edit window, the user can define precise rules using xPath expressions. In the bottom section, he or she can assign values to predefined attributes, e.g. indicating the direction and the level in the syntactic tree, using selection buttons.

⁷ http://www.w3.org/TR/1999/REC-xpath-19991116

4.2 **EXTRACTION RULES and EXTRACTION EXECUTION**

The definition file for extraction rules, pattern.xpt, contains a set of patterns and their steps. The data structure of a step follows. The order of the attributes is free:

<step name=cat_name category=cat_value context = cont_value direction= dir_value level=lel_value position=pos value", break = break value>

<stepxpath> xPath expression </stepxpath> </step>

Following is a rule example showing extraction of a verbal phrase and a recipient segment. The corresponding PBIE2 interface is shown in Figure 4:

<step name="Verbal Phrase" category="verbal" context="root" direction="after" level="any" position="first" break="yes">

<stepxpath>//VP[descendant::VBN="activated"][VP or VBN or VBZ][not (descendant::SBAR//VBN)][not(CC)]

</stepspath>

</step>

<step name="Recipient" category="recipient" context="current" direction="before" level="equal" position="first" break="no">

<stepxpath>//NPL|//NP|//NNPX</stepxpath> </step>

We applied these rules along with a step for extracting a recipient. Figure 4 shows the resulting view:

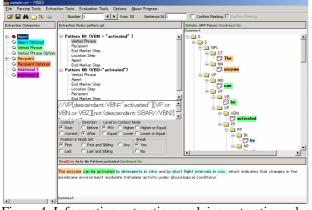


Figure 4: Information extraction applying extraction rules

5 **RELATED WORKS**

Several extraction tools were developed in the MUC (Message Understanding Conference), such as the Alembic Workbench (Aberdeen, 1996) and FASTUS (Appelt, 1995). These tools have mainly been used for extracting named entities from newswire articles. To our knowledge, there is no report of their application to the biomedical domain.

Based upon experience with processing biomedical texts, Yakushiji (2003) developed a debugging tool willex, and applied it to an HPSG-style grammar. The grammar was first constructed for general purposes, and thus would have to be specialized to handle biomedical texts. The main purpose was to increase the coverage of relevant grammars.

Linguistic annotation is needed for better performance in processing biomedical texts. Grover (2002) has been developing a tool with which one can analyze texts, and annotate linguistic information, such as parts of speech and subcategory information.

6 CONCLUSION

We have introduced our PBIE toolkit, designed to facilitate preparation of the basic data for parsing-based information extraction. We have especially focused on the PBIE2 evaluation tool. Using it, the user can annotate extracted segments to enable their use as a "gold standard", and can edit syntactic trees. He or she can also run a parser that is plugged in, write extraction rules, and apply these rules to the parsing result at hand. To further speed up the development cycle, we are planning to implement a more powerful search module.

We have been developing the toolkit to extract information concerning protein-protein interactions in the immunology domain. The toolkit is designed to be flexible, however, and especially to allow the easy customization of predefined extraction categories and comments. Thus its application to other domains should not pose great difficulties. We are next planning to use the PBIE toolkit for extracting information concerning drug-biomolecule interactions (Yoshikawa, 2003).

Acknowledgements

We thank Dr. S. Yoshikawa and Dr. K. Matsumura for their advice on pharmacology and Dr. M. Seligman for reading our draft.

REFEENCES

- Aberdeen, John, et al.: "MITRE: Description of the Alem-bic system as used in MET", Proc. of the TIPSTER 24-Month Workshop, pp.461-462, 1996. Appelt, Douglas E., et al.: "SRI international FASTUS
- system MUC-6 test results and analysis", Proc. of the Sixth Message Understanding Conference, pp.237-248, 1995
- Blaschke, Christian and Valencia, Alfonso: "The potential use of SUISEKI as a protein interaction discovery tool", Genome Informatics, Vol.12, pp.123-134, 2001. Friedman, Carol, et al.: "GENIES: a natural-language
- processing system for the extraction of molecular path-ways from journal articles", Proc. of ISMB-2001,
- man genes for high-throughput analysis of gene expression", Nature Genetics, Vol.28, pp.21-28, 2001. Hosaka, Junko, et al.: "PBIE Toolkit for Data Collection:
- Toward Parsing-based Information Extraction", Proc. of the IJCNLP-04, 2004 (forthcoming).
- the IJCNLP-04, 2004 (forthcoming).
 Yakushiji, Akane, et al.: "Event extraction from biomedical papers using a full parser", Proc. of PSB-2001, Vol.6, pp.408-419, 2001.
 Yakushiji, Akane, et al.: "A Debug Tool for Practical Grammar Development", ACL-03 Companion Volume to the Proc. of the Conference, pp.173-176, 2003.
 Yoshikawa, Sumi and Konagaya, Akihiko: "DIO: Drug Interaction Ontology Application to Inferences in Pos-
- Interaction Ontology Application to Inferences in Pos-sible Drug-drug Interactions", Proc. of The 2003 Inter-Mathematics national Conference on and Engineering Techniques in Medicine and Biological Sciences (METMBS '03), pp. 231-236, 2003.