Multilingual Propbank Annotation Tools: Cornerstone and Jubilee

Jinho D. Choi, Claire Bonial, Martha Palmer

University of Colorado at Boulder, USA
{choijd;bonial;mpalmer}@colorado.edu

Abstract

This paper demonstrates two annotation tools related to Propbank: Cornerstone and Jubilee. Propbank is a corpus in which the arguments of each verb predicate are annotated with their semantic roles. Propbank annotation also requires the choice of a sense ID for each predicate, defined in the corresponding frameset file. Jubilee expedites the annotation process by displaying several resources of syntactic and semantic information simultaneously; easy access to each of these resources allows the annotator to quickly absorb and apply the necessary syntactic and semantic information pertinent to each predicate for consistent and efficient annotation. Cornerstone is a user-friendly XML editor, customized to allow frame authors to create and edit frameset files. Both tools have been successfully adapted to many Propbank projects; they run platform independently, are light enough to run as X11 applications and support multiple languages such as Arabic, Chinese, English, Hindi and Korean.

1 Introduction

Propbank is a corpus in which the arguments of each verb predicate are annotated with their semantic roles (Palmer et al., 2005). Propbank annotation also requires the choice of a sense ID for each predicate. Thus, for each predicate in the Propbank, there exists a corresponding frameset file encompassing one or more senses of the predicate. All frameset files are written in XML, which is somewhat difficult to read and edit. Although there already exist many XML editors, most of them require some degree of knowledge of XML, and none of them are specifically customized for frameset files. This motivated the development of our own frameset editor, Cornerstone.

Jubilee is a Propbank instance editor. For each verb predicate, we create a Propbank instance that consists of the predicate's sense ID and its arguments labeled with semantic roles. Previously the allocation of tasks, the annotation of argument labels and the frameset tagging were all done as separate tasks. With Jubilee, the entire annotation procedure can be done using one tool that simultaneously provides rich syntactic information as well as comprehensive semantic information.

Both Cornerstone and Jubilee are developed in Java (JDK 6.0), so they run on any platform where the Java virtual machine is installed. They are light enough to run as X11 applications. This aspect is important because Propbank data are usually stored in a server, so annotators need to update them remotely (via SSH). One of the biggest advantages of using these tools is that they accommodate several languages; in fact, the tools have been used for Propbank projects in Arabic (M.Diab et al., 2008), Chinese (Xue and Palmer, 2009), English (Palmer et al., 2005) and Hindi, and have been tested in Korean (Han et al., 2002).

This demo paper details how to create Propbank framesets in Cornerstone, and how to annotate Propbank instances using Jubilee. There are two modes in which to run Cornerstone: multi-lemma and uni-lemma mode. In multi-lemma mode, a predicate can have multiple lemmas, whereas a predicate can have only one lemma in uni-lemma mode. Jubilee also has two modes: normal and gold mode. In normal mode, annotators are allowed to view and edit only tasks that have been claimed by themselves or by one other annotator. In gold mode, adjudicators are allowed to view and edit all tasks that have undergone at least single-annotation.

2 How to obtain the tools

Cornerstone and Jubilee are available as an open source project on Google code.[1] The webpage gives detailed instructions of how to download, install and launch the tools (Choi et al., 2009a; Choi et al., 2009b).

3 Description of Cornerstone

3.1 Multi-lemma mode

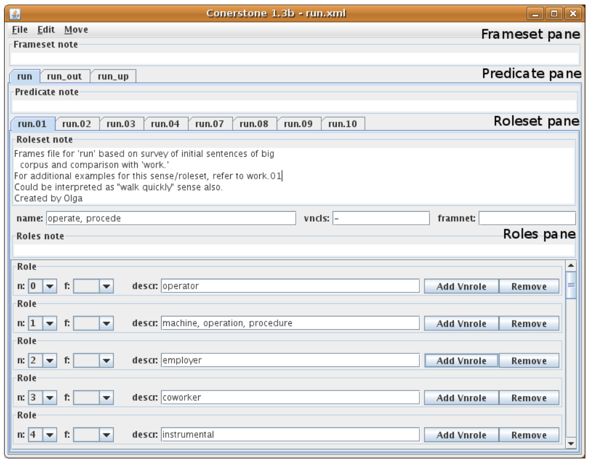
Languages such as English and Hindi are expected to run in *multi-lemma* mode, due to the nature of their verb predicates. In multi-lemma mode, a predicate can have multiple lemmas (e.g., 'run', 'run out', 'run up'). The XML structure of the frameset files for such langauges is defined in a DTD file, `frameset.dtd`.

Figure 1 shows what appears when you open a frameset file, `run.xml`, in multi-lemma mode. The window consists of four panes: the frameset pane, predicate pane, roleset pane and roles pane. The frameset pane contains a frameset note reserved for information that pertains to all predicate lemmas and rolesets within the frameset file. The predicate pane contains one or more tabs titled by predicate lemmas that may include verb particle constructions. The roleset pane contains tabs titled by roleset IDs (e.g., *run*.01, *run*.02, corresponding to different senses of the predicate) for the currently selected predicate lemma (e.g., 'run'). The roles pane includes one or more roles, which represent arguments that the predicate requires or commonly takes in usage.

3.2 Uni-lemma mode

Languages such as Arabic and Chinese are expected to run in *uni-lemma* mode. Unlike multi-

[1] `http://code.google.com/p/propbank/`



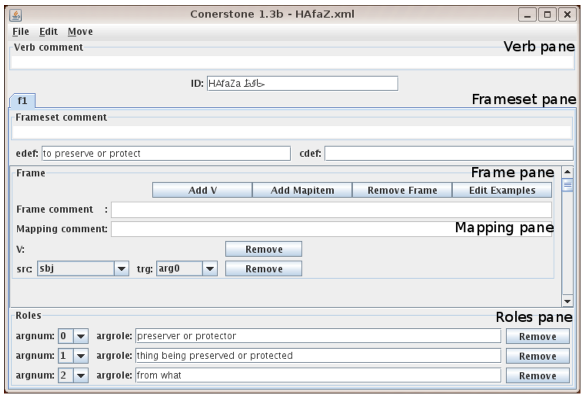
Frames file for 'run' based on survey of initial sentences of big
  corpus and comparison with 'work.'
For additional examples for this sense/roleset, refer to work.01
Could be interpreted as "walk quickly" sense also.
Created by Olga

name: operate, procede vncls: – framnet:

n	f	descr
0		operator
1		machine, operation, procedure
2		employer
3		coworker
4		instrumental

Figure 1: Open `run.xml` in multi-lemma mode

lemma mode, which allows a predicate to have multiple lemmas, uni-lemma mode allows only one lemma for a predicate. The XML structure of the frameset files for such langauges is defined in a DTD file, `verb.dtd`.



ID: HAfaZa حافظ

edef: to preserve or protect cdef:

src: sbj trg: arg0

argnum	argrole
0	preserver or protector
1	thing being preserved or protected
2	from what

Figure 2: Open `HAfaZ.xml` in uni-lemma mode

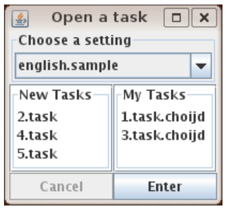
Figure 2 shows what appears when you open a frameset file, `HAfaZ.xml`, in uni-lemma mode. The window consists of four panes: the verb pane, frameset pane, frame pane and roles pane. The verb pane contains a verb comment field for information helpful to annotators about the verb, as well as the attribute field, `ID`, which indicates the predicate lemma of the verb, represented either in the Roman alphabet or characters in other languages. The frameset pane contains several tabs titled by frameset IDs (corresponding to verb senses) for the predicate. The frame pane contains a frame comment for optional information about the frame and the mapping pane, which includes mappings between syntactic constituents and semantic arguments. The roles pane consists of a set of arguments that the predicate requires or commonly takes.

4 Description of Jubilee

4.1 Normal mode

Annotators are expected to run Jubilee in normal mode. In normal mode, annotators are allowed to view and edit only tasks claimed by themselves or one other annotator when the max-number of annotators allowed is two. Jubilee gives the option of assigning a different max-number of annotators as well.

When you run Jubilee in normal mode, you will see an open-dialog (Figure 3). There are three components in the open-dialog. The combo-box at the top shows a list of all Propbank projects. Once you select a project (e.g., `english.sample`), both `[New Tasks]` and `[My Tasks]` will be updated. `[New Task]` shows a list of tasks that have either not been claimed, or claimed by only one other annotator. `[My Tasks]` shows a list of tasks that have been claimed by the current annotator.

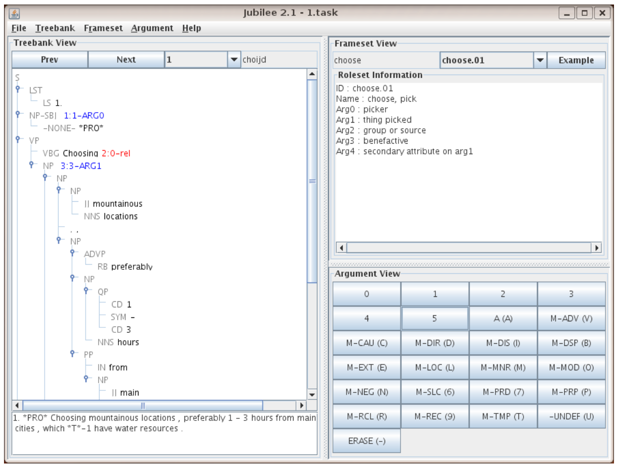


Choose a setting: english.sample

New Tasks	My Tasks
2.task	1.task.choijd
4.task	3.task.choijd
5.task	

Figure 3: Open-dialog

Once you choose a task and click the `[Enter]` button, Jubilee's main window will be prompted (Figure 4). There are three views available in the main window: the treebank view, frameset view and argument view. By default, the treebank view shows the first tree (in the Penn Treebank format (Marcus et al., 1993)) in the selected task. The frameset view displays rolesets and allows the annotator to choose the sense of the predicate with respect to the current tree. The argument view contains buttons representing each of the Propbank argument labels.



```
S
  LST
    LS 1.
  NP-SBJ  1:1-ARG0
    -NONE- *PRO*
  VP
    VBG Choosing 2:0-rel
    NP  3:3-ARG1
      NP
        NP
          JJ mountainous
          NNS locations
        , ,
        NP
          ADVP
            RB preferably
          NP
            QP
              CD 1
              SYM -
              CD 3
            NNS hours
          PP
            IN from
            NP
              JJ main
```

1. *PRO* Choosing mountainous locations , preferably 1 - 3 hours from main cities , which *T*-1 have water resources .

choose — choose.01

ID : choose.01
Name : choose, pick
Arg0 : picker
Arg1 : thing picked
Arg2 : group or source
Arg3 : benefactive
Arg4 : secondary attribute on arg1

Figure 4: Jubilee's main window

4.2 Gold mode

Adjudicators are expected to run Jubilee in gold mode. In gold mode, adjudicators are allowed to view and edit all tasks that have undergone at least single-annotation. When you run Jubilee in gold mode, you will see the same open-dialog as you saw in Figure. 3. The `[New Tasks]` shows a list of tasks that have not been adjudicated, and the `[My Tasks]` shows a list of tasks that have been adjudicated. Gold mode does not allow adjudicators to open tasks that have not been at least single-annotated.

5 Demonstrations

5.1 Cornerstone

We will begin by demonstrating how to view frameset files in both multi-lemma and uni-lemma mode. In each mode, we will open an existing frameset file, compare its interface with the actual XML file, and show how intuitive it is to interact with the tool. Next, we will demonstrate how to create and edit a new frameset file either from scratch or using an existing frameset file. This demonstration will reflect several advantages of using the tool. First, the XML structure is completely transparent to the frame authors, so that no knowledge of XML is required to manage the frameset files. Second, the tool automates some of the routine work for the frame authors (e.g., assigning a new roleset/frameset ID) and gives lists of options to be chosen (e.g., a list of function tags) so that frameset creation, and the entire annotation procedure in turn, become much faster. Third, the tool checks for the completion of required fields and formatting errors so that frame authors do not have to check them manually. Finally, the tool automatically saves the changes so the work is never lost.

5.2 Jubilee

For the treebank view, we will compare Jubilee's graphical representation of the trees with the parenthetical representation of former tools: the clear visual representation of the phrase structure helps the annotator to better understand the syntax of the instance and to annotate the appropriate node within the correct span. For the frameset view, we will detail what kind of semantic information it provides as you choose different rolesets. This will highlight how Jubilee's support of roleset ID annotation not only speeds up the annotation process, but also ensures consistent annotation because the roleset information provides a guideline for the correct annotation of a particular verb sense. For the argument view, we will illustrate how to annotate Propbank arguments and use the operators for concatenations and links; thereby also demonstrating that having each of these labels clearly visible helps the annotator to remember and evaluate the appropriateness of each possible argument label. Finally, we will show how intuitive it is to adjudicate the annotations in gold mode.

6 Future work

Both Cornerstone and Jubilee have been successfully adapted to Propbank projects in several universities such as Brandeis University, the University of Colorado at Boulder, and the University of Illinois at Urbana-Champaign. We will continuously develop the tools by improving their functionalities through user-testing and feedback, and also by applying them to more languages.

Acknowledgments

Special thanks are due to Prof. Nianwen Xue of Brandeis University for his very helpful insights as well as Scott Cotton, the developer of RATS and Tom Morton, the developer of WordFreak, both previously used for PropBank annotation. We gratefully acknowledge the support of the National Science Foundation Grants CISE-CRI-0551615, Towards a Comprehensive Linguistic Annotation and CISE-CRI 0709167, Collaborative: A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu, and a grant from the Defense Advanced Research Projects Agency (DARPA/IPTO) under the GALE program, DARPA/CMO Contract No. HR0011-06-C-0022, subcontract from BBN, Inc. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Jinho D. Choi, Claire Bonial, and Martha Palmer. 2009a. Cornerstone: Propbank frameset editor guideline (version 1.3). Technical report, Institute of Cognitive Science, the University of Colorado at Boulder.

Jinho D. Choi, Claire Bonial, and Martha Palmer. 2009b. Jubilee: Propbank instance editor guideline (version 2.1). Technical report, Institute of Cognitive Science, the University of Colorado at Boulder.

C. Han, N. Han, E. Ko, and M. Palmer. 2002. Korean treebank: Development and evaluation. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation.*

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

M.Diab, A.Mansouri, M.Palmer, O.Babko-Malaya, W Zaghouani, A.Bies, and M.Maamouri. 2008. A pilot arabic propbank. In *Proceedings of the 7th International Conference on Language Resources and Evaluation.*

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Nianwen Xue and Martha Palmer. 2009. Adding semantic roles to the chinese treebank. *Natural Language Engineering*, 15(1):143–172.