The New FrameNet Desktop: A Usage Scenario for Slovenian

Birte Lönneker-Rodman, Collin Baker, Jisup Hong
International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704, USA
{loenneke, collinb, jhong}@icsi.berkeley.edu

Abstract

The FrameNet Desktop application supports frame-semantic lexicon creation and corpus annotation. It is primarily intended for use with the English language at the FrameNet project, but a recent revised version facilitates easier distribution, installation and adaptation to new projects. This paper reports on a case study where the new FrameNet Desktop is used with Slovenian data, in a scenario involving lexicon import from and annotation export to an electronic dictionary.

1 Introduction

FrameNet is a rich lexical semantic resource for English supported by manual annotation of electronic corpora (Ruppenhofer et al., 2006). The FrameNet project has developed the FrameNet Desktop, an application initially intended for frame-semantic lexicon creation and annotation of English texts at the project site. But with a growing number of projects for other languages referring to the FrameNet model, the FrameNet Desktop has recently undergone various changes in architecture and functionality. The aim of these efforts is to make the tool easier to distribute and reuse for purposes including manual annotation in other languages, as well as viewing, summarizing and correcting the output of automatic frame-semantic annotation.

This paper presents a case study aimed at adapting FrameNet Desktop to a new language (Slovenian), and at the same time exchanging data between the FrameNet tool and an electronic dictionary. While the data extracted from the dictionary for import into the Desktop application consists in purely lexical and morphologic data (words, multiwords, and word forms), FrameNet Desktop output contains rich semantic information by associating lexical units with semantic frames, and annotated sentences illustrating their usage in context. These sentences can be fed back into the dictionary, where they will serve as usage examples.

The remainder of the paper is subdivided into two main sections. First, we will provide an overview of the FrameNet Desktop application and recent developments (Section 2). Second, a case study on utilizing the revised FrameNet Desktop for Slovenian and interchanging data between it and an electronic dictionary will be presented (Section 3). A brief summary and outlook in Section 4 concludes the paper.

2 FrameNet Desktop

After explaining FrameNet Desktop core functionality by way of an example (2.1), projects for other languages making use of earlier versions of the Desktop software (2.2) as well as its general architecture (2.3) will be described. Finally, recent developments of the tool are presented in Subsection 2.4.

2.1 Core functionality

The FrameNet Desktop is a suite of tools for (a) entering and editing frame descriptions and lists of lexical units, (b) extracting and annotating representative sentences from a corpus for lexicographic annotation, or annotating every target word in a given corpus during full-text annotation, and (c) organiz-
ing and displaying the results. Annotating target words in the FrameNet Desktop means identifying them as targets, selecting the lexical unit and frame they instantiate, and attaching frame-relevant labels to other constituents of the same sentence.

On the example of the ATTACHING frame, (Fillmore et al., 2003) gives a detailed overview of the annotation process within FrameNet Desktop, which illustrates many functions of the tool. In what follows, we will narrow down their examples to the discussion of a few core functions.

**Frame and lexical unit creation.** Creating a semantic frame involves describing (in English prose) the type of situation or happening it refers to, and drawing a list of words with senses that can be explained with reference to this frame. For instance, in the situation of the ATTACHING frame, somebody attaches (or affixes or joins) one thing to another thing, using some kind of connector. This is reflected in the frame description. A frame also includes a list of participant roles or aspects of the situation, called frame elements (FEs); for ATTACHING, the most prominent ones (“Core FEs”) are: Agent (the person who brings about the attaching), Item (an object affixed to a larger, more stable Goal), the Goal, and a Connector (the bond). Word senses explained in terms of a semantic frame are lexical units (LUs) of that frame. The list of English verb LUs belonging to ATTACHING includes append, attach, connect, and tie, among others. The FrameNet Desktop thus provides a means to create frames along with their informal description and frame elements, as well as lexical units, which are defined as the combination of a frame and a lemma (a lemma in FrameNet is a complex lexical entity combining information on lexeme, part of speech, and word forms; see e.g. (Baker et al., 2003)).

**Annotation of LU occurrences.** For lexicographic and Natural Language Processing purposes, it is of interest to know how frame elements are linguistically expressed in sentences containing frame LUs. Therefore, FrameNet Desktop facilitates the annotation of sentences relative to one lexical unit, the target. Whole constituents, dependents of the target word within the selected sentence, are annotated on separate layers for frame element, grammatical function, and phrase type, among others. To the present discussion, the FE layer is most central. It has been illustrated by (Fillmore et al., 2003) with the annotation of an occurrence of the lexical unit tie.v in sentence (1).

1 Apparently the healer would tie a black thread round the horse’s ankle, and it usually worked.

On the FE annotation layer for the target tie in this sentence, the FE Agent is assigned to the healer, Item to a black thread, and Goal to round the horse’s ankle. FrameNet Desktop provides a graphical interface for this kind of work, where words and sequences of words are selected, and frame elements from a frame-specific list assigned to them, with a few mouse clicks.

### 2.2 FrameNet Desktop for other languages

Initially, the FrameNet Desktop was designed for work on English. Since then, it has been adapted by various sites setting up FrameNets for different languages. To the knowledge of the authors, variants of earlier FrameNet Desktop versions are in use at Spanish FrameNet (Subirats and Sato, 2004; Subirats, Under review), Japanese FrameNet (Ohara et al., 2004), and German FrameNet1. More or less important changes to the FrameNet Desktop software were carried out at the respective project sites. 

Extent and nature of the changes depend on the resources and aims of the different projects, as well as on language-specific properties, such as differences in character encoding or morpho-syntactic structure.

### 2.3 FrameNet Desktop system architecture

The FrameNet Desktop is a J2EE-based client/server application system using MySQL for data storage. The server side requires a networked server with a Unix-like operating system, such as Linux or Mac OS X. To start up a new FrameNet Desktop project, a MySQL database must be created and seeded with the necessary table structures (Baker et al., 2003) as well as with all data the project chooses to take over from the original FrameNet database (this might include frame and frame element definitions or information about available annotation labels). This MySQL database serves as the data

---

source for a JBoss application server, which mediates all interaction with FrameNet Desktop clients through FrameNet-developed Enterprise JavaBeans (EJBs). The JBoss server also ensures transactional integrity in a multiuser environment. Any number of Desktop clients, within which the manual lexicographic work and annotation finally takes place, can be run from any machine with network access to the server.

2.4 Recent developments

At the FrameNet project, a new version of the FrameNet Desktop software has been deployed that compiles and runs on Java 5 (aka. Java 1.5) and the JBoss 4.0.5.GA J2EE application server. Besides improvement in performance thanks to the newer versions of Java and JBoss, extensive refactoring of the code-base promises to facilitate ongoing internal development as well as make source code distribution to outside groups feasible. Changes range from linearizing package-to-package dependencies and establishing clearer boundaries between the server and client tiers, to rearranging the source directory structure to accommodate popular IDEs like Eclipse and NetBeans, to integrating JBoss and MySQL configuration into the build infrastructure. In the new system, for example, Ant commands can be used to create a new database and seed it with the necessary table structure and to start and stop MySQL and JBoss. These changes will allow FrameNet to distribute Desktop source code along with a fully-functional development and testing configuration that can be easily adapted by an outside project for different languages.

Currently, a revised XML format is being designed and implemented that can be used for both input and output for the FrameNet Desktop software. This is expected to improve data interchange with other applications. Usage scenarios include importing the output of automatic frame-semantic annotation into the Desktop application, manually correcting it therein, and exporting the corrected version for retraining the statistical data model used by the automatic labeler. Another example of how annotated texts exported from FrameNet Desktop can be used within other applications is provided in the case study below (Section 3).

Finally, functionality and user-friendliness of the new FrameNet Desktop have been improved. This includes full Unicode support as well as a character-by-character selection mode. The latter can now be toggled for each session and allows the annotation of units below the word level, which can be useful when working with languages rich in inflection and/or compounding.

3 Case study: A new usage scenario for FrameNet Desktop

This section describes how the FrameNet Desktop is set up and used to work with a new language, Slovenian. To the authors’ knowledge, no FrameNet project has been undertaken for this small Slavic language before. A usage scenario will be presented, involving data interchange between the Desktop application and an electronic dictionary covering Slovenian. The intended goal is to collect frame-semantically annotated sentences illustrating the usage of lexemes covered by the dictionary. The sentences can then be displayed along with the main dictionary entries, to provide the user with examples of the word used in context. An overview of the entire process is given in Figure 1.

The dictionary – provider of lexical information and recipient of annotated example sentences – is briefly described in 3.1, before the steps relevant to FrameNet Desktop will be covered: importing language-independent information (3.2), lexicon data (3.3), and corpus files (3.4), annotating text (3.5), and exporting the annotated corpus for use in the target application (3.6).

3.1 The electronic dictionary

The electronic dictionary used in this case study is Online SLO-DE-SLO, a Slovenian-German/German-Slovenian online dictionary² (Lönneker and Jakopin, 2003; Jakopin and Lönneker, 2004). The motivation behind Online SLO-DE-SLO is to create a lexical resource useful to both human users and Natural Language Processing. The dictionary contains over 8,800 word and multi-word correspondences, as well as 2,150 bilingual usage examples. It records, on average, more than 80,000 requests per month. Data is stored

²http://webapp.rrz.uni-hamburg.de/~slowenisch/ [September 28, 2007].
in a MySQL database, and HTML forms and Perl scripts are used to display dictionary entries in the Web interface.

Whereas the core vocabulary entered into the dictionary was manually drawn from textbooks of both languages, the dictionary has recently been extended by entries derived from electronically available Slovenian texts covering various domains (short stories, cooking, transportation, and transcripts or written drafts of public speeches). These text corpora will also be used for frame-semantic annotation with FrameNet Desktop.

3.2 Importing language-independent information into FrameNet Desktop

Besides the empty table structure of the FrameNet database, new FrameNet-related projects might also take over from the English FrameNet database what they consider being reusable, language- or project-independent, information. Following the example of previous non-English FrameNets (see 2.2 above), we decided to import all frame and frame element definitions from the English FrameNet database by producing an SQL dump file (FrameInformationDump.sql in Figure 1) and loading it into the Slovenian FrameNet database.

This makes available the full list of English frames in FrameNet Desktop, where Slovenian lexical units can be attached to them, and their frame elements used during annotation. Frames can also be modified within FrameNet Desktop; thereby, language-specific differences in frames or frame elements can be accommodated.

3.3 Importing lexicon data

When adding a lexical unit to a frame, the FrameNet Desktop allows the user to create a new lemma, which will be added to the lexicon part of the FrameNet database, together with its corresponding lexeme(s) and word forms, also to be entered manually. Alternatively, the Desktop application allows the user to choose an existing lemma from a list, and to add it to a frame as a new LU.

When starting a FrameNet project for a new language, it might not be desired to start with a completely empty lexicon, and being forced to manually enter all lexemes and word forms; this is especially true for morphologically rich languages, such as Slovenian. A more practical approach is to populate the lexicon part of the FrameNet database with lemmas, lexemes and word forms from an external source, before starting LU creation in the FrameNet Desktop, so that existing lemmas can always be selected from a list.

In our experiment, the relevant lexicon data for Slovenian has been extracted from the Slovenian-
German dictionary. A series of SQL commands is used to prepare tables in the right format and conflate redundant information. For example, the dictionary contains ordered lists of all word forms of a lexeme, which are further described by morphological information. FrameNet Desktop does not store information on individual word forms; therefore, a simple set of distinct forms pertaining to a lexeme is extracted for import into the annotation tool. The file produced by the SQL commands is an SQL dump file (FNLexiconDump.sql in Figure 1), which can easily be imported into an empty FrameNet database. A further advantage of this approach is that dictionary entries and lexemes in the FrameNet database are implicitly linked via identical IDs, thereby increasing interoperability.

Online SLO-DE-SLO dictionary entries exhaustively cover the vocabulary of all corpus texts currently considered for import into the Slovenian FrameNet Desktop (see 3.1 above). Therefore, it is not necessary to add FrameNet lexicon entries in the case study. In the future, it would be desirable to update the Slovenian FrameNet lexicon tables by importing additional data from the dictionary, as the latter offers automatic word form generation. Taking over new lexical data from the dictionary instead of creating it in FrameNet Desktop would thus increase data consistency and minimize the risk of misspelled or overlooked word forms.

3.4 Importing corpus files
FrameNet XML to be loaded into FrameNet Desktop for full-text annotation is produced by the import pipeline (ImportFullText, a Desktop utility implemented in Java) from a text file encoded in UTF-8 character set and provided with paragraph boundary tags. For English, text import comprises the following steps:

1. sentence boundary detection
2. Named Entity Recognition (NER)
3. part-of-speech tagging and lemmatization
4. recognition of tokens that should not receive frame semantic annotation (e.g., punctuation, proper names)
5. XML formatting.

Import of Slovenian texts starts at step 3 of the pipeline, thus presupposing a text file provided with both paragraph and sentence boundary tags. The NER step is skipped, because the NER module used for English texts (BBN IdentiFinder) is language specific and, if run on Slovenian, would output numerous false positives (spurious NE annotations). In the future, a language independent NE recognizer or a specific Slovenian module should be used; however, to our knowledge, no off-the-shelf Slovenian NE recognizer is readily available. Therefore, Slovenian texts imported into FrameNet Desktop currently do not include pre-tagged Named Entities.

The FrameNet Desktop import pipeline presupposes an existing installation of the TreeTagger (Schmid, 1994) for part-of-speech tagging and lemmatization. When importing a corpus, the path to a language-specific TreeTagger parameter file can be provided. Such parameter files are available for numerous languages from the TreeTagger website; however, Slovenian is not among them. We therefore manually annotated a set of Slovenian texts, totaling approximately 14,000 running tokens, with a subset of the MULTEXT-East tagset (Erjavec, 2004) and trained a Slovenian model (parameter file). The “dictionary” file also necessary for training contains over 59,700 word forms.

Once the corpus text has been fully processed by the import pipeline, the resulting XML is loaded into FrameNet Desktop by using the FarinaImport utility. The corpus files and their paragraphs show up as a tree structure in FrameNet Desktop when “Corpus mode” is selected as Tree Mode. Sentences of a paragraph can then be accessed individually for annotation, as illustrated in Figure 2 for the third sentence of the third paragraph from a speech by the Slovenian Prime Minister.

3.5 Annotation within FrameNet Desktop
Once a lexical unit has been attached to a frame, occurrences of its word forms in a corpus text are ready for annotation within FrameNet Desktop. For example, Figure 2 shows the annotation

3\text{http://www.ims.uni-stuttgart.de/projekte/corplex/Tre...\text{27 September, 2007].}

4\text{These resources are suitable for test purposes, but accuracy is not yet adequate for distribution.}
of the target vežejo - ‘(they) tie’, which is automatically recognized by FrameNet Desktop as a word form of the lexeme vezati - ‘(to) tie’, and as an instance of the lexical unit vezati.v in the ATTACHING frame. Clicking on the corresponding pop-up text (Attaching.vezati.v) brings up several annotation layers relevant to this LU. On the frame element (FE) layer, words or word sequences within a sentence can be selected for annotation with one of the frame elements pertaining to the ATTACHING frame, displayed in a list at the bottom of the window. Other layers allow for the annotation of information on relative pronouns and antecedents, or for recording a metaphorical use of the target word, as in the example being annotated in Figure 2, as well as for the representation of other types of linguistic information.

A translation of the Slovenian sentence under consideration in Figure 2 is provided in Example (2).

(2) Lepo je med prijatelji, s katerimi nas vežejo skupne korenine, deliti us-ACC tie common roots, to-share tudi radost in ponos, ki ju vzbuja slovenska himna.

‘It is nice to share with friends, to whom we are tied by common roots, also the joy and pride evoked by the Slovenian anthem.’

3.6 Export from FrameNet Desktop

Corpora or documents annotated within FrameNet Desktop are exported with the StaticReportGenerator utility. This facilitates the creation of a wide variety of “reports”. For the scenario under consideration, where we are interested in displaying annotated sentences as usage examples in an electronic dictionary, the most interesting report types are XML and HTML renderings of an annotated corpus. These represent sentence as sentence as well as target word by target word, the frame element information attached to sentence constituents.

To illustrate, the XML representation of the annotation pertaining to the target lexical unit vezati.v in the ATTACHING frame, as shown in Figure 2, is given in Figure 3. The annotation with respect to vezati.v is subsumed under the annotationSet element, which provides as values of its attributes the names of the frame and lexical units, as well as their IDs in the MySQL database. Target, frame element (FE), and Other information attached to particular strings within the sentence is given as positions of their first and last character within the sentence string, as represented as values of label at-
Figure 3: FrameNet XML representation of lexical unit annotation.

tributes. The XML report does not include information on the ATTACHING frame, such as its description or the default colors associated with each of its frame elements. Also, it does not explicitly state information on the lemma or lexeme underlying the annotated LU. These pieces of information, necessary to associate the sentence and its annotation with the relevant entry in the electronic dictionary, have to be retrieved from other types of FrameNet Desktop reports or from the MySQL database itself.

The HTML representation of annotated corpora consists of several files per target. One of them contains color markup of those annotated sentence constituents that are either target word or frame elements (see Figure 4). The information that the target has been annotated with respect to the ATTACHING frame is contained in a different HTML file, but none of the HTML files contains lexical unit information. As a result, reusing color information is easier from the HTML than from the XML representation of annotated sentences, but retrieving the correct lexeme for linking to a dictionary entry is even more difficult.

We are currently studying the most elegant way to combine annotation set information with frame and lexeme information and to link the example to the correct dictionary entry. Once a method has been found, annotated example sentences could be shown within the dictionary as in Figure 5. The figure shows a bilingual setting, with an aligned sentence in German annotated for the corresponding German target word as well.

The user interface of the dictionary should display information useful to and accessible by humans, not necessarily experts in linguistics. Therefore, the frame-semantic markup simply represents target words and frame elements in different colors (see Figure 5). Definitions of the semantic frame and its frame elements (displayed below the examples) are kept to a minimum; a hyperlink leads to full frame-semantic information in the most current version of FrameNet.

4 Conclusion

We have presented FrameNet Desktop, an application for frame-semantic lexicon creation and corpus annotation. The recent remodeling of the FrameNet Desktop source code is intended to facilitate easy set-up of the tool, making it more accessible to new FrameNet-related projects. We used the new FrameNet Desktop version in a case study on a Slovenian FrameNet, aimed at data interchange with an electronic dictionary for this language. The actual set-up of a functional Slovenian FrameNet Desktop, as documented in this paper, including import of Slovenian lexicon data and corpus texts, but excluding
training the Slovenian POS-tagger, took about two weeks. This is encouraging for any projects considering using the new FrameNet Desktop for their work.

The Slovenian Desktop does not yet offer the full functionality of its English counterpart, due to missing Named Entity Recognition and phrase type definitions. However, as the intended usage scenario focuses on collecting frame-semantic annotation in the form of frame element information, the current Slovenian FrameNet Desktop is already useful for preparing annotated example sentences that can be imported into an electronic dictionary.

5 Acknowledgments

This work was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD), granted to the first author.

References


