ProLiV - a Tool for Teaching by Viewing Computational Linguistics

Monica Gavrila
Hamburg University, NATS
Vogt-Kölln Str 30, 20251, Germany
gavrila@informatik.uni-hamburg.de

Cristina Vertan
Hamburg University, NATS
Vogt-Kölln Str 30, 20251, Germany
vertan@informatik.uni-hamburg.de

Abstract

ProLiV - Animated Process-modeler of Complex (Computational) Linguistic Methods and Theories - is a fully modular, flexible, XML-based stand-alone Java application, used for computer-assisted learning in Natural Language Processing (NLP) or Computational Linguistics (CL). Having a flexible and extendible architecture, the system presents the students, by means of text, of visual elements (such as pictures and animations) and of interactive parameter set-up, the following topics: Latent Semantics Analysis (LSA), (computational) lexicons, question modeling, Hidden-Markov-Models (HMM), and Topic-Focus. These topics are addressed to first-year students in computer science and/or linguistics.

1 Introduction

The role of multimedia in teaching Natural Language Processing (NLP) is demonstrated by constant development of software packages such as GATE (http://gate.ac.uk) and NLTK (http://nltk.sourceforge.net/index.html). Detailed information about visual tools for NLP, in particular about GATE, is to be found in (Gaizauskas et al, 2001).

ProLiV is a Java application framework, developed in a three-year project (2005-2008) at the University of Hamburg. It helps first-year students to understand and learn, in an easier manner, either complex linguistic theories used in NLP (e.g. question modeling) or statistical approaches for computational linguistics (e.g. LSA, HMM).

The learning process is supported by modules integrating text, visual and interactive elements. In its first released version, ProLiV contains the following modules:

- the Latent Semantic Analysis (LSA) module and the computational lexicons module - for linguists,
- the question modeling module - for computer scientists,
- the Hidden-Markov-Models (HMM) module and Topic-Focus module - for both computer scientists and linguists.

2 The Learning Path

For each module, the learning path is guided by lessons, a terminology dictionary and interactive activities. Exercises and small tests can also be integrated.

The lessons include text, pictures and animations. Hyperlinks between lessons ensure a concept-oriented navigation through the learning content. Additionally key terms within the content are linked with dictionary entries.

Three central issues guided the development of the ProLiV software:

1. choosing the most adequate means (text / picture / animation) to represent lessons content,
2. designing the layout (quantity and size of text, colors) in order to increase the learning success,
3. in case of the animations, defining its components and parameters (speed, animation steps, and graphical elements) to maximize their impact on users.

Regarding the second issue above-mentioned, the layout of the modules follows part of the guidelines found in (Orr et al., 1994) and (Thibodeau, 1997).

Considering the current multimedia development, the trend is using animations to improve the learning process. Animations are assumed to be

Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, pages 13–16,
Suntec, Singapore, 3 August 2009. ©2009 ACL and AFNLP

a promising educational tool, although their efficiency is not fully proved. Researchers, such as (Morrison, 2000), showed that animations can convey more information and be helpful when showing details in intermediate steps of a process, but when building an animation it is very important to consider the background of the student (e.g. linguistics, natural sciences) and his/her psychological functioning. The educational effectiveness of the animations depends on how they interact with the learner. Depending on the student's background, in order to have a helpful material, one has to carefully decide what information the animation contains. As our experiment showed (see Section 2.1), depending on the student and his/her background, an animation can improve the learning process, or bring nothing to it. We found no cases when the animation slowed down the learning process.

The system was experimentally used in seminars at the University of Hamburg. Part of the lessons content was adapted following the user's feedback.

2.1 Animations in ProLiV

Animations are not integrated in all modules of the ProLiV system, but only in the LSA, computational lexicons and question modeling modules.

In order to decide how to organize the information in an animation, we evaluated the animations for the matrix multiplication in the LSA module by asking 11 high-school pupils (between 16 and 19 years old) to choose between the several representations.

We showed the pupils three animations that describe the multiplication of matrices, a static picture and the text representation of the definition. The animations differ in the way the process is presented (abstract vs. concrete) and in user interaction authorization.

The pupils were asked to evaluate all the representations. The question they had to answer was: "*Which of the following representations helps more, when learning about matrix multiplication?*". The scale given was from 1 = very helpful to 5 = not helpful at all.

Analyzing the results, we could not conclude that one representation is a "*real winner*'. The best representation was considered the most flexible animation, that allows the student go backwards and forwards whenever the user needs it, the learning process being adapted to the user's rhythm. All the evaluation results can be seen in Table 1. In order to better see the influence of these representations in the learning process, statistical tests should be run.

Representation	Average Result
Definition (formula)	3.5
Picture	2.91
Animation 1	3.64
Animation 2	**2.09**
Animation 3	2.45

Table 1: Evaluation of the animations in the matrix multiplication (*Animations 1 and 3 have no user interaction; Animations 1 and 2 are more abstract*)

3 System Architecture

In Figure 1 we present the ProLiV System architecture, consisting of:

- a file repository (lessons, dictionary, tests, and exercises),
- a tool repository,
- an aggregating module combining elements from file and tool repository (Main Unit),
- the graphical user interface (G.U.I.)

For each topic a stand-alone module is connected with the G.U.I module via the Main Unit. Modules related to new topics can be inserted any time with no particular changes of the system.

The ProLiV architecture follows the guideline considerations found in (Galitz, 1997).

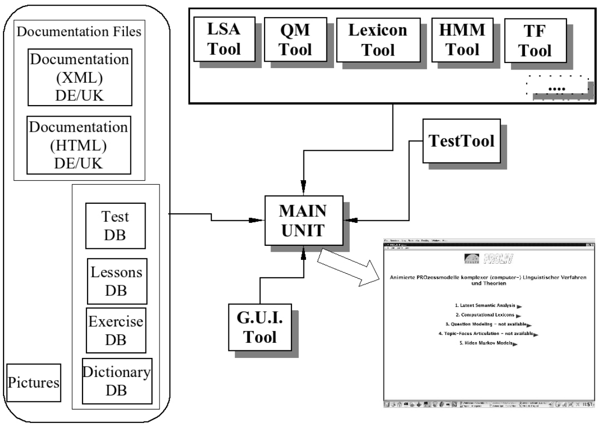
Documentation Files
Documentation (XML) DE/UK
Documentation (HTML) DE/UK
Test DB
Lessons DB
Exercise DB
Dictionary DB
Pictures
LSA Tool
QM Tool
Lexicon Tool
HMM Tool
TF Tool
....
TestTool
MAIN UNIT
G.U.I. Tool
PROLIV
Animierte PROzessmodelle komplexer (computer-) Linguistischer Verfahren und Theorien
1. Latent Semantic Analysis
2. Computational Lexicons
3. Question Modeling - not available
4. Topic-Focus Articulation - not available
5. Hiden Markov Models

Figure 1: The ProLiV Architecture

The flexibility of the system is also given by the fact that the G.U.I.[1] is generated according to an XML[2] description, developed within the project (see DTD Description).

The XML description contains the information in the lessons (definitions, theory, examples, etc.) and the G.U.I. specifications (colors, fonts, links, arrangement in the interface, etc.). Having an XML file as input, the system generates automatically the G.U.I. presented to the student. The information shown to the user can be extended or modified with almost no implementation effort. New lessons or modules can be integrated, by extending or adding XML files. Due to the same fact, also the content adaptation of the system to other languages[3] is very easy.

```
The DTD Description:

<?xml version=’’1.0’’?>
<DOCTYPE LESSONS[
<!ELEMENT LESSONS (LESSON+)>
<!ELEMENT LESSON (TITLE+, (TEXT|FORMULA|
INDEXI|INDEX|BOLD|
ITALIC|TERM|LINK|DEF|
EXM|OBS|T|OTHER)+>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT TEXT (#PCDATA)>
<!ELEMENT FORMULA (#PCDATA)>
<!ELEMENT INDEX (#PCDATA)>
<!ELEMENT INDEXI (#PCDATA)>
<!ELEMENT BOLD (#PCDATA)>
<!ELEMENT ITALIC (#PCDATA)>
<!ELEMENT TERM (#PCDATA)>
........................
<!ELEMENT T (#PCDATA)>
<!ELEMENT OTHER (#PCDATA)>

<!ATTLIST LESSON NO CDATA #REQUIRED>
<!ATTLIST DEF NO CDATA #REQUIRED>
<!ATTLIST EXM NO CDATA #REQUIRED>
<!ATTLIST OBS NO CDATA #REQUIRED>
<!ATTLIST QUIZZ NO CDATA #REQUIRED>
<!ATTLIST EX NO CDATA #REQUIRED>
<!ATTLIST T NO CDATA #REQUIRED>
<!ATTLIST OTHER STYLE CDATA #REQUIRED>
```

The G.U.I. follows the same design rules in all modules and the layout and format decisions are consistent. A color and a font style are associated to only one kind of information (e.g. color red associated to definitions, etc.).

3.1 Integrated external software packages

The learning process is also sustained by interactive elements, such as the possibility of changing parameters for the LSA algorithm and visualizing the results, or as the integrated programs for the computational lexicons tool: ManageLex (`http://nats-www.informatik.uni-hamburg.de/view/Main/ManageLex`) and G.E.R.L. (`http://nats-www.informatik.uni-hamburg.de/view/Main/GerLexicon`). This way the students have the possibility, not only to read the theory, but also to see the impact of their modifications in an algorithm that is described in the lessons.

Due to its architecture, other such external programs can be easily integrated within ProLiV.

4 LSA Module in ProLiV

In order to have a better overview of what a module contains and how it is organized, this section presents some aspects of the LSA module.

The LSA module makes an introduction to the topic. It gives an overview of the LSA algorithm, principles, application areas, and of the main mathematical notions used in the algorithm. Initially thought for being used mostly by students from linguistics (or linguists) - due to the mathematical algorithms -, the tool can be exploited by anybody who wants to have an introductory course on LSA.

The content is organized in four Units:

1. LSA: General Knowledge - It gives the LSA definition, a short overview of the history, its semantics, and how LSA can be used in the study of cognitive processes.

2. Mathematical Fundamentals - It describes the LSA algorithm

3. LSA Applications - It presents the application areas for the LSA, LSA limitations and critics. Also a comparison with other similar algorithms is made.

4. Compendium of Mathematics - It gives the user the mathematical background: definitions, theorems, etc.

The course has also an introduction, a motivation, conclusion and references.

[1] The G.U.I. is automatically generated not only for the lessons, but also for the term dictionary associated to each module.

[2] XML = Extensible Markup Language. More details to be found on `http://en.wikipedia.org/wiki/XML`

[3] For the moment ProLiV contains lessons in German and English

The LSA module is offering not only a textual representation of the information, but also several visualization methods (as images and animations[4]). Beside the lessons, there are implemented a term dictionary and an environment for testing LSA parameters.

4.1 The LSA Test Environment

Probably the most interesting part of the LSA module is the test environment. After learning about LSA, in this environment the user has the possibility to actually see how LSA is working, and what results can be obtained when comparing the meaning of two words. The user can set several parameters of the algorithm - e.g. the analysis mode (simple/frequency based vs. advanced/entropy based), the minimum word occurrences, the analysis dimension, the similarity measure (Cosine, Euclidean, Pearson, Dot-Product), etc. - and decide which words are not considered in the analysis. The analyzed text, the initial cooccurrence matrix and the one obtained after applying the Singular Value Decomposition (SVD) algorithm are shown in the G.U.I. The similarity measure, when comparing two words, is calculated in both unreduced and reduced cases.

5 Conclusions

The paper presents a course-ware software, ProLiV. It is a collection of (interactive) multimedia tools used mainly for the consolidation of first-years courses in computational linguistics and literary computing. Its goal is to help the humanist scientists to make use of complex formal methods, and the computer specialists to understand humanist facts and interpretations.

The main feature of the system, in the context of the conference, is not the content of the lessons, but the system's extendible and adaptable architecture. Another important aspect is the way in which the information is presented to the student.

The system runs on any platform supporting Java 1.5 or newer. It was developed on Linux and tested on Windows and Mac OS X.

Being Java-based and having as input Unicode files (XML encoded information), the system can be embedded in the future in a Web environment.

More about ProLiV can be found in (Gavrila et al, 2006) or in (Gavrila et al, TBA) and on the ProLiV homepage: `http://nats-www.informatik.uni-hamburg.de/view/PROLIV/WebHome`.

Acknowledgments

We would like to thank all people that helped in the development of our software: Project Coordinator Prof. Dr. Walther v. Hahn (Computer Science Department, Natural Language Systems Group), Prof. Dr. Angelika Redder (Department of Language, Linguistics and Media Studies, Institute for German Studies I), Dr. Shinichi Kameyama (Department of Language, Linguistics and Media Studies, Institute for German Studies I), Christina von Bremen (Computer Science Department, Natural Language Systems Group), Olga Szczepanska (Computer Science Department, Natural Language Systems Group), Irina Aleksenko (Computer Science Department, Natural Language Systems Group), Svetla Boytcheva (Academy of Sciences Sofia).

References

Wilbert O. Galitz. 1997 *The Essential Guide to User Interface Design: an Introduction to GUI Design principles and Techniques*, Wiley Computer Publishing, New York.

Robert J. Gaizauskas, Peter J. Rodgers, and Kevin Humphreys. 2001 *Visual Tools for Natural Language Processing*, *Journal of Visual Languages and Computing*, Vol. 12, Number 4, p. 375-411, Academic Press

Monica Gavrila, Cristina Vertan. 2006 *Visualization of Complex Linguistic Theories*, in the Proceedings of the ICDML 2006 Conference, p. 158-163, Bangkok, Thailand, March 13-14

Monica Gavrila, Cristina Vertan, and Walther von Hahn. To be published during 2009 *ProLiV - Learning Terminology with animated Models for Visualizing Complex Linguistics Theories*, in the Proceedings of the LSP 2007 Conference, Hamburg, Germany, August,

Julie Bauer Morrison, Barbara Twersky, and Mireille Betrancourt. 2000 *Animation: Does It Facilitate Learning?*, in the Proc. of the Workshop on Smart Graphics, AAAI Press, Menlo Park, CA.

Kay L .Orr, Katharine C. Golas, and Katy Yao. 1994 *Storyboard Development for Interactive Multimedia Training*, *Journal of Interactive Instruction Development*, Volume 6, Number 3, p. 18-31

Pete Thibodeau. 1997 *Design Standards for Visual Elements and Interactivity for Courseware*, *T.H.E. Journal*, Volume 24, Number 7, p. 84-86

[4] The animations integrated are for the LSA algorithm tested on an example and for matrix multiplication