Teaching Language Technology at the North-West University

Suléne Pilon
sktsp@puk.ac.za

Gerhard B van Huyssteen
ntlgbvh@puk.ac.za

Bertus van Rooy
ntlajvr@puk.ac.za

Research Focus Area: Languages and Literature in the South African Context North-West University,
Potchefstroom
2531
South Africa

Abstract

The *BA Language Technology* program was recently introduced at the North-West University and is, to date, the only of its kind in South Africa. This paper gives an overview of the program, which consists of computational linguistic subjects as well as subjects from languages, computer science, mathematics, and statistics. A brief discussion of the content of the program and specifically the computational linguistics subjects, illustrates that the *BA Language Technology* program is a vocationally directed, future oriented teaching program, preparing students for both future graduate studies and a career in language technology. By means of an example, it is then illustrated how students and researchers alike benefit from working side by side on research and development projects by using a problem-based, project-organized approach to curriculum design and teaching.

1 Introduction

A new undergraduate teaching program, *BA Language Technology,* was recently introduced at the Potchefstroom Campus of the North-West University (NWU). The introduction of this program was motivated by two factors:
(a) a need within the Faculty of Arts to develop teaching programs that are relevant, vocationally directed, and future-oriented; and
(b) a need in the South African higher education system for capacity building in the field of in language technology (PanSALB & DACST, 2000).

To date, the BA Language Technology program is the only one of its kind in South Africa. It has therefore remained imperative that the program equips students adequately to fill positions in the emerging South African language technology industry. At the same time, students should be able to continue with graduate studies, and therefore the program had to be designed in such a way that students receive an academic training that incorporates a solid theoretical component alongside the need to get enough practical experience. These two imperatives are reflected in the program structure, and also in the project-based learning approach that we adopted.

2 Program Structure

After wide consultation with international and local role players and experts, a program was designed that combines language subjects and natural sciences (mainly computer science, mathematics and statistics) with a core group of computational linguistic and language technology subjects. This section offers an overview of the *BA Language Technology* program. An example of a typical program will be given and the modules which form part of the program will be discussed briefly.

The program has a basic core of compulsory modules, but allows some room for students to take modules based on personal interest and ability. A student who excels in computer programming can choose to take additional modules from that field after completing the compulsory modules. Students may also choose to take more language or mathematics modules after completing their compulsory modules. There are also a number of general formative subjects that all students at the University must take, which are not being discussed here. The basic course structure is presented in Table 1.

YEAR 1	YEAR 2	YEAR 3	YEAR 4
First semester	**First semester**	**First semester**	**First semester**
Modules	**Modules**	**Modules**	**Modules**
Computer Science (programming)	**Language Technology: Introduction**	**Introduction to NLP**	**Language technology: Internship**
2 Languages	1 Language	1 CHOICE	
Statistics (introduction)	Computer Science (programming)	2 General formative modules	
Mathematics	1 CHOICE		
Applied Mathematics			
2 General formative modules			
Second semester	**Second semester**	**Second semester**	**Second semester**
Modules	**Modules**	**Modules**	**Modules**
Computer Science (programming)	**Language Technology: Linguistics for language technology students**	**Language Technology: Speech applications**	**Advanced NLP**
1 Language	1 CHOICE	**Language Technology: Text applications**	**Language Technology Project**
Statistics (Inferential)	2 General formative modules	1 CHOICE	
1 CHOICE			

Table 1: *BA Language Technology* compulsory modules with choices

The various general formative modules offered at the university include academic literacy, study skills, computer literacy and information skills, philosophy and academic and scientific writing courses. The elective modules from which the students can choose are mathematics, computer science and languages. The languages from which the students can choose are Afrikaans, English or Setswana (regular university courses) or introductory courses (foreign language level) in two South African languages, Setswana and isiZulu and two foreign European languages, German and French.

Students are encouraged to take at least one South African language. This is motivated in part by trends in the macro-political environment. In government policy documents, such as the final language policy presented to cabinet, language technology is principally regarded as a means to promote multilingualism and increase access of information in a country with eleven official languages. In the context of the program itself, it is expected that students acquire and/or improve their proficiency in the various languages; students are also expected to develop basic knowledge of the structure of the particular languages. This basic knowledge is then developed further in the module “Linguistics for language technology students” (second year, first semester). The module includes components of phonetics, morphology and syntax, to enable students to learn how to do detailed linguistic data analysis.

In the first semester of the second year, students are introduced to Language Technology. An overview of the field of study is given and it is indicated how the knowledge students gained in the modules they have completed, should be put to use within the field. The course also focuses on the relationship between a more practical language technology orientation and a more theoretical natural language processing (NLP) orientation, to enable students to see the broader picture and develop a sense for the coherence of the teaching program.

Language technologies are the subject of two modules in the second semester of the third year. They spend equal amounts of time on speech-based technologies and text-based technologies. The focus of these courses is specific language technology applications. At any given time, there are a number of ongoing projects at the university. Students are involved in these projects, learning to develop the specific applications, but also developing general skills for other types of applications, within the framework of project-based learning, as will be outlined later in this paper. Students are expected to participate in ongoing projects on various levels, ranging from annotating corpora to intricate programming – depending on their aptitude and preferences.

This is followed by a six-month internship in the first semester of the fourth year, at an approved company or higher education or research institution. Apart from extending their training in the development of language technology applications, the internship is intended to let students get a “real world" experience in the language technology industry before they have to make career decisions.

In their final semester, students have to complete a supervised project, which fits in with current research at the university. It is important that students should be positive about this project and therefore students are consulted when project topics are chosen. In this stage of the program, students have very little class in order to enable them to work on their projects on a full-time base, which provides for more practical experience.

Students are introduced to Natural Language Processing in the first semester of the third year. This course focuses mainly on statistical techniques for the analysis of the kinds of phonetic, morphological and syntactic data that were introduced in the second semester of the second year. The logic is that students must be able to analyze data manually as linguists first, in order to develop an appreciation for the capabilities, power and limitations of statistical NLP methods.

An advanced NLP course is offered in the second semester after students have completed their internship and while they are working on their own projects. This course is tailored to the individual interests and needs of the students. The specific NLP techniques relevant to their projects, as well as problems they encountered during their internships, serve as guiding principles for the selection of content. At the same time, we incorporate a selection of hot topics in NLP research and some techniques for dealing with semantic data.

As computational linguistics is a relatively new field of study in South Africa, students and lecturers/researchers have to learn together, even by making mistakes and taking 'wrong' sidetracks during the learning process. In order to facilitate these circumstances, a problem-oriented and project-organized approach, based on the educational system developed at the Aalborg University, Denmark, since 1974 (Kjersdam & Enemark, 1994), was taken in the design of the curricula of the Language Technology and NLP modules. This means that the content of some of the Language Technology and NLP subjects vary from year to year, depending on the current project(s) being conducted at the university. However, by working alongside each other on research and development projects, both students and lecturers engage in active learning, proving to yield excellent results in the acquiring of knowledge in the field. The next section describes how various research and development projects are integrated in the undergraduate and graduate teaching programs, in order to facilitate hands-on, outcome-based learning.

3 Teaching Approach: Problem-Based, Project-Organized Learning

Problem-Based Learning (PBL; also called problem-oriented education) can be defined as learning "based on working with unsolved, relevant and current problems from society/real life… By analyzing the problems in depth the students learn and use the disciplines and theories which are considered to be necessary to solve the problems posed, i.e. the problem defines the subjects and not the reverse" (Kjersdam & Enemark, 1994: 16; cf. Schwartz *et al*., 2001; Macdonald, 2002). This approach is successfully implemented world-wide in the teaching of specifically more applied sciences, such as, *inter alia*, medicine (Albanese & Mitchell, 1993; Barrows and Tamblyn, 1980; Moore *et al*., 1994), and engineering (De Graaf & Kolmos, 2003; Fink, 2002).

Within the context of computational linguistics, this "applied-teaching approach" maintains a dynamic triangular equilibrium between training, research and product development, serving researchers, students, and the industry alike. A project-organized approach offers lecturers an opportunity to align course material with their research projects, while students are enabled to gain "comprehensive knowledge of the development of theoretical and methodological tools" (Kjersdam and Enemark, 1994: 17). Therefore, after completion of their formal studies, students should be able to contribute to research and the development of original paradigms to solve new and complex problems in the future.

In the *BA Language Technology* program, PBL is incorporated with project-organized education in two ways. On the one hand various *project-based modules* are included in the curriculum. For instance in the third year of study, the modules "Language Technology: Speech Applications" and "Language Technology: Text Applications" are introduced, where students have to develop various small modules for both speech and text technological applications (e.g. a simple rule-based stemmer). In the final year of study, the largest part of the year is spent on independent project work, which is conducted either at the university, or while doing an internship elsewhere. These projects are on a much larger scale than the third year projects, with the possibility to build on the work of the previous year (e.g. to develop a more sophisticated stemmer, using more advanced NLP techniques).

On the other hand, some of the other modules (e.g. the "Natural Language Processing" modules) are more *project-driven*, since they are organized around existing research projects. Students are mostly drawn in on a so-called "design-oriented" level, i.e. where they have to deal with "know-how problems which can be solved by theories and knowledge they have acquired in their lectures" (Kjersdam & Enemark, 1994: 7). After the project and the problems related to the project are explained to students, they get involved by collecting data, identifying possible/different solutions, formulating rules and algorithms, analyzing data, evaluating different components, etc. In this way they get know-how and experience in theoretical, methodological, and implementation issues.

This can be illustrated by a recent example, where work on a spelling checker project was integrated in the curricula of various modules. In this project, involving the development of spelling checkers for five different South African languages, a variety of NLP techniques were implemented in the various spelling checkers, depending on the orthographical complexity of and resources available for a specific language. For instance, languages such as Tswana and Northern Sotho have a relatively simple orthographical structure (in the sense that it is more disjunctive), and a straightforward lexicon-based approach to spelling checking therefore suffice for these languages. In contrast, Afrikaans, Zulu and Xhosa are orthographically more complex languages, requiring a spelling checking approach based on morphological analysis or decomposition, which is of course more interesting from a computational linguistic perspective. For all of these languages, almost no resources were available at the start of the project, posing a huge but interesting challenge (e.g. could available technologies for other languages, such as a Porter stemmer, be adapted for these languages?).

From the onset of the spelling checker project, students were involved in all aspects of the project. Using Jurafsky & Martin (2000) as a point of departure, students were introduced to the basic problems of spelling checking, relating it to the current project and specifically to the challenges posed by spelling checking for Afrikaans (e.g. productive concatenative compound formation, derivational word formation, etc.). Students were thoroughly involved in all discussions of the aims of the project, potential problems and possible solutions, as well as the general system architecture (i.e. students were involved on the design-oriented level). Students were therefore introduced to basic concepts such as tokenization, stemming, and Levenshtein Distance (for purposes of generating suggestions), within a real-world context.

After the planning and design phase, each student got involved in solving different problems of the project, e.g. developing a stemmer (using finite-state techniques) and a compound analyzer (using machine-learning techniques), the automatic generation of a lexicon, evaluating spelling checkers (within the broader context of the evaluation of NLP applications), etc. Although each student worked separately on different problems, they were forced to extend their experience by helping each other with their different tasks, thereby expanding their general knowledge and experience. In this way, students also came to learn that different problems call for different approaches: to use finite-state techniques for hyphenation in Afrikaans is simply to labor-intensive, while machine learning offers highly efficient solutions to the problem. An introduction to machine learning was therefore also introduced in the curriculum.

The advantages of this approach proved to be many: not only did the project benefit from the sub-projects of each of the students, but students got the feeling that they were involved in "important" and relevant work. They got the opportunity to apply the theoretical knowledge they acquired in the classes in a practical, hands-on environment, to improve their understanding of the theories and concepts of the study field, and to solve real-world problems. Additionally, members of staff were enabled to harmonize their research and teaching responsibilities, optimizing the quality and quantity of their outputs. Moreover, existing students were motivated to continue with their studies in computational linguistics on MA level (where they are working on more advanced problems), while undergraduate student numbers increased (which can be ascribed to a greater awareness of language technology in the community, brought about partially by media coverage of the project, focusing on the promotion of multilingualism and language empowerment).

4 Conclusion

Since its very inception, the *BA Language Technology* program at the North-West University was designed as a vocationally directed, future-oriented teaching program. A curriculum with a core of computational linguistic subjects, strengthened by a strong foundation in languages, computer science, mathematics, and statistics, equips students both with enough practical experience to start working in the industry, and with enough theoretical knowledge to continue with postgraduate studies.

By taking a problem-based, project-organized approach to curriculum design, students and researchers alike benefit from working side by side on research and development projects (as illustrated by the incorporation of a spelling checker project in the curricula of various subjects). The same approach is followed in other subjects, such as "Language Technology: Speech Applications", where students are working in collaboration with their lecturers on various speech-based projects. As new research projects are initiated, the curricula of the various subjects are adapted accordingly. For example, in 2005 a new research project on syntactic parsing commenced – consequently, new students are confronted with other problems than their predecessors, while still learning, for example, about the differences between linguistic and statistical approaches to NLP. With the help of students in the program and others involved, the program is constantly evaluated and adjusted accordingly, thereby ensuring that it delivers well-educated and informed students, prepared for the challenges of a career in language technology.

References

Albanese MA & Mitchell S. 1993. Problem-based learning: A review of literature on its outcomes and implementation issues. *Academic Medicine* 68, 52-81.

Barrows HS & Tamblyn RM. 1980. *Problem-Based Learning: An Approach to Medical Education*. New York: Springer Publishing Company.

De Graaff, E & Kolmos, A. 2003. Characteristics of Problem-Based Learning. International *Journal of Engineering Education* 19(5).

Fink, FK. 2002. Problem-Based Learning in engineering education: a catalyst for regional industrial development. *World Transactions on Engineering and Technology Education* 1(1): 29-32.

Jurafsky, D & Martin, JH. 2000. *Speech and language processing : an introduction to natural language processing*. Upper Saddle River: Prentice Hall, 2000.

Kjersdam F & Enemark S. 1994. *The Aalborg Experiment: Project Innovation in University Education*. Aalborg: Aalborg University Press.

Macdonald R. 2002. Problem-based learning: some references. Available at: [WWW:]www.ics.ltsn.ac.uk/pub/pbl [Accessed 5 May 2003].

Moore GT, Block SD, Briggs Style C & Mitchell R. 1994. The influence of the New Pathway curriculum on Harvard medical students. *Academic Medicine* 69, 983-989.

Pan South African Language Board (PanSALB) & Department of Arts, Culture, Science and Technology (DACST). 2000. The development of Human Language Technologies in South Africa: Strategic Planning. (Report by the joint steering committee). Pretoria: Government Printers. Available at: www.dac.gov.za/about_us/cd_nat_language/language_planning/hlt_strategic_plan/hlt_strategic_plan2.htm#policy [Accessed April 1, 2005].

Schwartz P, Mennin S & Webb G. 2001. *Problem-based Learning: Case Studies, Experience and Practice*. London: Kogan Page.