Proceedings of the Workshop on Effective Tools and Methodologies for Teaching
Natural Language Processing and Computational Linguistics, Philadelphia,
July 2002, pp. 39-45. Association for Computational Linguistics.

Design and Evolution of a Language Technologies Curriculum

Robert E. Frederking, Eric H. Nyberg, Teruko Mitamura, and Jaime G. Carbonell
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA

Abstract

The Language Technologies Institute (LTI) of the School of Computer Science at Carnegie Mellon University is one of the largest programs of its kind. We present here the initial design and subsequent evolution of our MS and PhD programs in Language Technologies. The motivations for the design and evolution are also presented.

1 Introduction

The Language Technologies Institute (LTI)[1] of the School of Computer Science at Carnegie Mellon University is one of the largest programs of its kind. At Carnegie Mellon, Language Technologies is the focus of an entire department, while Computer Science is the focus of an entire school or college. As a result, the language technology courses are not limited to a few specialized electives in a more general program; rather, the LTI MS and PhD programs are dedicated to teaching language technologies and related disciplines in-depth.

Although some aspects of the LTI programs result from these unique circumstances, we feel that sharing our design process and evolutionary experience should be helpful to those interested in designing programs with similar goals in other universities.

[1]See http://www.lti.cs.cmu.edu/ for more information.

2 Initial Curriculum Design

In this section we describe the overall goals of the LTI degree programs and discuss the initial curriculum that was put in place when the LTI programs commenced in 1996.

2.1 Student Population and Initial Curriculum Design

The LTI MS and PhD programs are open to all qualified students, including those with primarily a linguistics background, or primarily a computer science background. Therefore the curriculum was initially designed to meet the educational needs of students with diverse backgrounds. To address these needs, the curriculum includes fundamental courses in three areas: linguistics, computer science, and machine learning. In addition to fundamentals, the curriculum includes courses on various application areas, such as machine translation, information retrieval, speech processing, etc. The curriculum was also designed to contain a significant "hands-on" or "learning by doing" component. We decided to offer a series of laboratory courses, which correspond to and augment particular lecture courses by offering a set of programming tasks intended to illustrate the material learned in the primary course. The curriculum also contains a two-course sequence on software engineering and project management, which provides a mixture of planning and management activities with hands-on system design, implementation and testing.

Originally, the MS was envisioned as primarily a professional degree for students wishing to

enhance their skills for employment in language-related technology fields. As a result, the MS program did not include any breadth requirements – students could concentrate their course work on a particular application area. We designed the MS curriculum in such a way that students could graduate after one year of intensive coursework, if desired, although the normal (expected) pace was 1.5 to 2 years. In contrast, the PhD degree was envisioned as preparation for a career in academia or industrial research, and included a breadth requirement as a way to ensure well-rounded graduates. This breadth requirement is defined in terms of sets of courses called "Focus Areas"; these are called the Linguistic Focus, Computer Science Focus, Learning/Statistical Focus, and Task Focus. Each PhD student must take at least one course from each of these Focus Areas.

2.2 Initial Curriculum

The LTI's first incoming class of students entered in Fall of 1996. Based on the design goals listed above, we created the set of core courses shown in Figure 1. (Note that labs count for 1/2 course credit.) Figures 2 and 3 present the initial set of degree requirements for the MS and PhD, respectively.

Type/Number	Name
Linguistic Focus:	
11-711	Algorithms for NLP
11-721	Grammar and Lexicon
11-723	Formal Semantics
CS Focus:	
15-681	Machine Learning
15-750	Algorithms
15-780	Artificial Intelligence
Learning Focus:	
11-761	Language and Statistics
15-681	Machine Learning
15-88x	Neural Networks
Task Focus:	
11-731	Machine Translation
11-741	Information Retrieval
11-751	Speech Recognition
Labs:	
11-712	Laboratory in NLP
11-732	Laboratory in MT
11-742	Laboratory in IR
11-754	Speech Laboratory
Other courses:	
11-713	Principles of Translation
11-791	Soft. Eng. for LT I
11-792	Soft. Eng. for LT II

Figure 1: **Initial** set of LTI courses

2.3 Initial Course Descriptions

We will very briefly describe here the main focus of each of our initial courses; we list them in numerical order:

- **11-711 Algorithms for NLP:** An introductory graduate-level course on the computational properties of natural languages and the fundamental algorithms for processing natural languages.
- **11-712 Laboratory in NLP:** This lab is intended to complement the 11-711 lecture course by providing a chance for hands-on, in-depth exploration of various NLP paradigms.
- **11-713 Principles of Translation:** Focuses on the principles and methodology of accurate **human** translation, including machine-aided translation, with practical examples.
- **11-721 Grammar and Lexicon:** An introductory graduate-level course on linguistic data analysis and theory, focusing on methodologies that are suitable for computational implementations. The course covers major syntactic and morphological phenomena in a variety of languages. The emphasis is on examining both the diversity of linguistic structures and the constraints on variation across languages.
- **11-723 Formal Semantics:** An introductory graduate-level course on formal linguistic semantics: Given a syntactic analysis of a natural language utterance, how can one assign the correct meaning representa-

- Ten (10) senior or graduate courses
  - Eight (8) of the 10 must be LTI core courses
  - One (1) of the 10 can be directed research
  - One (1) elective

Figure 2: **Initial** LTI MS requirements

- Courses:
  - Eight (8) LTI graduate core courses
  - At least one in each Focus Area
  - At least two labs (count as 1/2 each)
- Proficiencies:
  - Writing
  - Presentation
  - Programming
  - Teaching
- Thesis Proposal and Defense

Figure 3: **Initial** LTI PhD requirements

tion to it, using a formal logical system?

- **11-731 Machine Translation:** An introductory graduate-level course surveying the history, techniques, and research topics in the field of Machine Translation.
- **11-732 Laboratory in MT:** This lab is intended to complement the 11-731 lecture course by providing a chance for hands-on, in-depth exploration of various MT paradigms.
- **11-741 Information Retrieval:** This course studies the theory, design, and implementation of text-based information systems.
- **11-742 Laboratory in IR:** This lab is intended to complement the 11-741 lecture course by providing a chance for hands-on, in-depth exploration of various IR research topics.
- **11-751 Speech Recognition:** This course provides an introduction to the theoretical foundations, essential algorithms, major approaches, experimental strategies and current state-of-the-art systems in speech recognition.
- **11-754 Speech Laboratory:** This course teaches participants how to implement a complete spoken language system while providing opportunities to explore research topics of interest in the context of a functioning system.
- **11-761 Language and Statistics:** This course introduces some of the central themes and techniques that have emerged in statistical methods for language technologies and natural language processing.
- **11-791/792 Software Engineering for Language Technologies I/II:** This two course sequence combines classroom material and assignments in the fundamentals of software engineering (11-791) with a self-paced, faculty-supervised directed project (11-792). The two courses cover all elements of project design, implementation, evaluation, and documentation.
- **15-681 Machine Learning:** This is the core Computer Science Department (CSD) course concerned with the theory and practice of computer programs that automatically improve their performance through experience. We cover topics such as learning decision trees, neural network learning, statistical learning methods, genetic algorithms, Bayesian learning methods, explanation-based learning, and reinforcement learning.
- **15-750 Algorithms:** This is the core CSD graduate-level course on algorithm analysis

and design.

- **15-780 Artificial Intelligence:** This is the core CSD graduate-level course on artificial intelligence: The course will focus on the AI algorithms and techniques to build a full intelligent agent with cognition, action, and perception.

- **15-88x Neural Networks:** This is one of a set of three rotating courses involving different aspects of real neural systems from a computer science perspective.

3 Evolution of the Curriculum

Having presented the initial design of our set of courses and requirements, we will now describe several significant changes to the LTI's curriculum over its first seven years. These changes have resulted in the current curriculum and requirements, which are shown in Figures 4, 5, and 6.

3.1 Optional Masters Thesis

The actual population of MS students that we have acquired has differed in at least one significant respect from that which we had envisioned. As mentioned above, we had planned our MS degree as a professional degree, with professionals in the language technology field taking one or two years to enhance their careers with a Masters degree.

However, the large majority of our MS students are in fact academically-oriented, planning to go on to a PhD, either here or elsewhere. One concrete effect of this difference is that there was student demand for a Masters Thesis, to produce and document a tangible, individual research effort, in order to enhance the students' applications to PhD programs.

Accordingly, we added a Masters Thesis option to our MS program. One concern we had was to ensure that the Masters Thesis was more significant than a one semester project, but not as large in scope as a PhD thesis. We therefore structured it as a two course sequence: a Directed Research course (typically in the Fall of the second year) that should culminate in a written proposal being approved by the Masters Thesis committee, followed by a course called "Masters Thesis" (11-929). The student selects a committee consisting of three faculty members, one of whom is typically the student's advisor. Although the Masters Thesis is listed as a course, it is more open-ended than a normal course; the student revises the Masters Thesis until the committee judges it to be finished.

3.2 Changes to MS Degree Requirements

Other changes to the MS degree resulted from new school-wide rules adopted within the School of Computer Science (SCS). To ensure a consistent level of achievement in a set of "core competencies", each MS program in SCS was required to ensure (via a set of required courses) that students graduating from the program can demonstrate a set of "core competencies":

- Analytical Competency
- Requirements Competency
- Design Competency
- Implementation Competency
- Correctness Competency
- Extra-technical Competency

Following our principle of trying to leave the MS program as unconstrained as possible, we mapped these competencies[2] into three essential requirements, as shown in Figure 5. "Algorithms for NLP" and "Software Engineering for IT 1" were already being taken by most of our MS students, and between them accounted for most of the core competencies. So all our MS students must now take both of these courses. In addition, there is a need to make certain the MS student acquires some hands-on implementation experience; we satisfied this by requiring either a lab course, "Software Engineering for IT 2" (where the design from "SE/IT 1" is implemented), or a project-oriented Masters Thesis.

[2] See the following URL for a detailed draft definition of these Competencies: http://www.lti.cs.cmu.edu/MS/scs-requirements.html

Finally, to ensure that some specific subject area is understood in depth, there is a requirement for one "Task Focus" course.

3.3 Core course concept removed

As described above, and shown in Figures 1-3, when the curriculum was initially designed, there was a concept of a list of "LTI Core Courses". Students were required to select courses from this list, which included relevant courses in the Computer Science Department (CSD) (courses with a 15-xxx number), and did not include some LTI courses that were seen as more elective or special-purpose in nature.

One concern driving this distinction was that if the students were spread between too many different courses, there would not be a large enough student population in any one course (this was especially important in the first several years of the program). Another concern was to ensure that our students took courses that were really relevant to language technologies, but were also able to take relevant courses in CSD. (While taking elective courses is a perfectly acceptable possibility, some students felt that the required courses were already a sufficient course load.)

Individual students would occasionally request that some course that they were interested in be added to the list of core courses. This would usually result in the Chair of the programs explaining the nature of the core course list, and the possibility of taking the course in question as an elective. But eventually the students as a group raised the issue of whether the distinction caused more problems than it solved. The students were asked by the faculty to draft a proposed modification, and in fact produced two alternative proposals.

In the course of the ensuing discussion, a new possibility emerged: simply discard the concept of a core course. Given the increased size of the LTI student body (currently about 50 students total, of whom about 35 are taking classes[3]), the minimum-class-size issue was no longer so critical. To ensure appropriate language technolo-

[3]Current class-taking students consist of 11 MS students, 14 first-year PhDs, and 9 second-year PhDs.

Type/Number	Name
Masters Courses:	
11-682	Language Technologies
Linguistic Focus:	
11-713	Principles of Translation
11-721	Grammar and Lexicon
11-723	Formal Semantics
CS Focus:	
11-711	Algorithms for NLP
11-791	Soft. Eng. for IT I
15-681	Machine Learning
15-750	Algorithms
15-780	Art. Intelligence
Learning Focus:	
11-761	Language & Statistics
15-681	Machine Learning
15-88x	Neural Networks
Task Focus:	
11-731	Machine Translation
11-741	Information Retrieval
11-751	Speech Recognition
11-752	Speech: PPPS
11-792	Soft. Eng. for IT II
Labs:	
11-712	Laboratory in NLP
11-732	Laboratory in MT
11-742	Laboratory in IR
11-754	Speech Laboratory
Other courses:	
11-716	Dialogue Processing
11-717	LT for Computer-Aided Language Learning
11-743	Adv. IR Seminar/Lab
11-929	Masters Thesis

Figure 4: **Current** set of LTI courses

- Ten (10) senior or graduate courses
  - Six (6) of the 10 must be LTI courses
  - Two (2) of the 10 must be SCS courses
  - One (1) of the 10 may be directed research
  - One (1) elective
- School-level MS requirements:
  - Both 11-711 and 11-791
  - Either lab, 11-792, or Thesis project
  - One Task Focus course

Figure 5: **Current** LTI MS requirements

- Courses:
  - Six (6) LTI graduate courses
    - At least one in each Focus Area
    - At least two labs (count as 1/2 each)
  - Two (2) SCS graduate courses
- Proficiencies:
  - Writing
  - Presentation
  - Programming
  - Teaching
- Thesis Proposal and Defense

Figure 6: **Current** LTI PhD requirements

gies focus, while allowing for relevant courses in CSD (and other SCS departments), the new requirement was for six (6) LTI courses and two (2) SCS courses. The definition of an LTI course is simply any course with a number in the LTI ("11-xxx"), while the definition of an SCS course is similarly a course in any department in the SCS. (Thus the "SCS courses" could also be LTI courses, if a student so chose.) Note that the PhD Focus Area courses are still defined by a list of specific courses[4]. This final change in requirements resulted in the current requirements for the Masters and PhD degrees, as shown in Figures 5 and 6, respectively.

3.4 Evolution of the Self-Paced Labs

The original concept of an LTI lab course was a sequence of planned, well-structured exercises to be undertaken by individual students working on their own. For example, the original NLP Lab course included tasks such as: building a bottom-up chart parser; adding semantic constraints to a syntactic unification grammar; and writing a simple generation grammar. Such structured courses have certain desirable properties – primarily, they are easy to administer from semester to semester, and it is easier to create a supportive infrastructure (software environments, tools, etc.) for well-defined content.

Nevertheless, there has been a recent trend toward more flexible, less structured lab courses. Many students find the structured approach less appealing, especially when the exercises are abstract and not related to solving a real-world problem. In many cases, a structured approach requires a pre-defined problem and a given partial solution, which students find artificially limiting. It was also felt that a certain amount of group-oriented work would improve the realism of the lab courses. Recent offerings of the lab courses have included a "team project" option – students may elect either the standard, structured exercises or an open-ended group project (e.g., "build an elementary question-answering system using available tools"). While the latter option tends to stimulate creativity and gener-

[4] Careful readers may notice that several courses changed Focus Areas in the intervening years.

ate more enthusiasm among the students, the instructor must take care to advise the students appropriately, so that the nature and scope of the project are realistically bounded.

3.5 Current Curriculum

Figure 4 lists the current (Fall 2001/Spring 2002) set of LTI courses. (The figure omits a few courses that may not be taught again.)

Figures 5 and 6 present the current requirements for the Masters and PhD degrees, respectively. In addition to refinements to the graduate curriculum, we have added several courses:

- **11-682 Language Technologies:** This introductory course is also open to undergraduates; it is intended as a "gateway" to LTI for upperclass undergrads, and also for LTI MS students without much LT background. The course will include specific computational techniques for different LT tasks, such as vector-spaces for IR, Web-spidering for IR and text mining, Hidden-Markoff-Models for speech, chart parsing and so on. There is also a synthesis task, such as combining parsing, generation and disambiguation for MT.
- **11-752 Speech: Phonetics, Prosody, Perception and Synthesis:** The goal of this course is to give the student basic knowledge from several different fields that is necessary in order to pursue research in automatic speech processing.
- **11-716 Dialogue Processing:** Dialog systems and processes are becoming an increasingly vital area of interest both in research and in practical applications. The purpose of this course is to examine, in a structured way, the literature in this area as well as learn about ongoing work.
- **11-717 LT for Computer-Aided Language Learning:** This course studies the design and implementation of CALL systems that use Language Technologies such as Speech Synthesis and Recognition, Machine Translation, and Information Retrieval. (It serves primarily as part of a new Masters in CALL offered jointly by LTI and our Modern Languages department.)
- **11-743 Advanced IR Seminar/Lab:** This is a seminar that focuses on current research in Information Retrieval. The seminar covers recent research on subjects such as retrieval models, text classification, information gathering, fact extraction, information visualization, summarization, text datamining, information filtering, collaborative filtering, question answering systems, and portable information systems.
- **11-929 Masters Thesis:** This course provides a record in the student's transcript of the Optional Masters Thesis, as described above.

4 Conclusion

We have presented here the initial design and subsequent evolution of the MS and PhD programs of the Language Technologies Institute at Carnegie Mellon University. The main lesson to be drawn is perhaps that listening to the concerns of the program's students, engaging them in frank discussions, and finding a reasonable balance between their concerns and the concerns of the faculty, has led to what we feel are definite improvements in the shape of the program. We call the final set of figures our "current" program to acknowledge the near certainty that future changes to the curriculum will occur, based on changing faculty concerns, student demands, or external factors. We hope that sharing our design process and evolutionary experience will be helpful to those interested in designing programs with similar goals in other universities.