Proceedings of the Workshop on Effective Tools and Methodologies for Teaching
Natural Language Processing and Computational Linguistics, Philadelphia,
July 2002, pp. 46-53. Association for Computational Linguistics.

A niche at the nexus: situating an NLP curriculum interdisciplinarily

Deryle Lonsdale
Department of Linguistics
Brigham Young University
Provo, UT, USA, 84602
lonz@byu.edu

Abstract

This paper discusses the establishment and implementation of a curriculum for teaching NLP. At the core are two classes which involve some theoretical background, extensive hands-on experience with state-of-the-art technologies, and practical application in the form of an intensive programming project. Issues involving interdisciplinary coordination, curriculum design, and challenges in teaching this discipline are discussed.

1 Institutional context

Our university initiated teaching of an NLP curriculum in 1998. At the core are two classes which both include some theoretical background, extensive hands-on experience with state-of-the-art technologies, and practical application in the form of an intensive programming project. They are meant to be comparable in quality and scope to the best NLP courses taught elsewhere. Each semester that these classes were taught, the university administered an anonymous survey to students to gauge their satisfaction with the course and its contents. Issues involving interdisciplinary coordination, curriculum design, student satisfaction, and challenges unique to teaching this discipline are presented in this paper. First, though, necessary relevant background on the campus and student demographics is presented.

The university is one of the largest private universities in the U.S. with an enrollment of about 30,000 (with 2,000 graduate students). Uncharacteristically, almost three-fourths of the student body speaks a non-English language.

The College of Humanities houses Linguistics, Philosophy, and several language and literature departments. Over 60 languages are taught within the college to students from all disciplines. The Linguistics Department offers undergraduate and master degrees (but no Ph.D.); over 150 undergraduate majors are currently enrolled. The richness of the linguistic environment in this college has a great bearing on the results discussed in this paper. The College of Physical and Mathematical Sciences houses the Computer Science (CS) Department (and several others teaching the “hard” sciences). CS faculty strive to meet the demands of flourishing undergrad and graduate programs, and by necessity offer courses primarily in the core areas of CS.

Until 1998 there was almost no interaction between the CS and Linguistics departments, no courses were taught in NLP or CL, and no research brought together students across college boundaries. Recently, however, the situation has improved. For several reasons, NLP classes were initiated at the university in 1998: faculty hires into Linguistics held such interests, campus research projects needed students with this type of experience, an NLP infrastructure was necessary for internal work of the university and its sponsor, and the improvement of job prospects for Linguistics students was targeted. Creating learning opportunities in NLP would help on all of these accounts. This paper discusses the two new NLP classes that have been developed and taught during the last four years in response to these needs.

Both courses are referred to as NLP classes rather than as CL classes. One commonly expressed distinction between the two areas is that CL is more theoretical and cognitive in orientation, whereas NLP has a more engineering and applied focus[1]. By this criterion the two classes discussed in this paper are clearly NLP-oriented.

2 NLP for the humanities

For several years a program called Computers and the Humanities (CHum) in the College of Humanities has provided a computer background for humanities undergrads; this is necessary because CS courses have traditionally been unavailable to non-CS majors. The CHum track can lead to a minor specialization for any humanities degree and attracts students from all corners of the college. The CHum offerings include a dozen classes meant to take students from the most basic contexts (computer end-user) through computer research tools use, basic programming, and programming projects. More recently the programming language taught has been Visual Basic (though see below). In the last ten years enrollment in this popular program has increased tenfold.

Computer support for humanities students is laudable. Several labs are available, supplied with good computational resources and adequate supervision by student consultants. Many students also choose to use their own computers at home or at work to fulfill assignments. Occasionally off-campus computer use for homework is not possible when proprietary software or corpora acquired by the university for classroom use cannot be distributed off-campus.

2.1 A parsing class

In recent years CHum course content has involved increasingly more language processing. Twice now, an intermediate-level class has been taught in natural-language parsing and grammar engineering. The class was taught the Unix and Emacs environments, basic LISP programming, and lexicon and phrase-structure grammar development techniques; all of these were new areas of focus for a CHum class. One text was used for this class (Allen, 1995), and the associated parsers available from its author were actively used by the students in their coursework. The textbook was followed rather closely and was quite accessible even to humanities students at this level. Instruction involved two 80-minute lecture classes per week and was given in a technologically enhanced (TELE) classroom featuring an ethernet-connected instructional computer, overhead LCD projection, multimedia devices, and a touch-sensitive projection wall (smart-screen). Lectures were delivered primarily via PowerPoint, running parsing engines, and accessing commercial and research project websites for materials and discussion.

CHum parsing course evaluations were very good to excellent (around 5.5-6.0 on a 7-point scale), exceeding college and department means across-the-board in 30 categories. Lowest (i.e. average satisfaction) ratings went to the text book used. The only critical comments mentioned a desire for more in-class lab time (a carry-over from prerequisite classes which are held in computer lab classrooms with a computer for every student). Whereas in lower classes the focus was on learning particular programming constructs, the parsing class content was more abstract and required less classroom-time programming demonstrations.

2.2 A speech programming class

Though the parsing class was popular and successful, it has not been taught in the last year and a half. Instead, the intermediate-level CHum class has focused on teaching speech applications programming, again to meet infrastructure and pedagogical needs. In this class[2] the first third of the semester includes intensive instruction in phonetics, phonology, and speech phenomena (using (J.M.Pickett, 1999)), as well as in TCL programming. The balance of the semester involves instruction on manipulating a speech toolkit, developing and leveraging associated data resources (lexicons and corpora, phrase-structure grammars, discourse moves), and understanding applications implementation (file formats, speaker adaptation, interaction, agents, v-commerce, speech translation).

[1] See, for example, Hans Uszkoreit's website at www.coli.uni-sb.de/˜hansu/what_is_cl.html.

[2] See humanities.byu.edu/classes/chum387dl/homepage.html.

Sample homework assignments include: running, writing, and debugging simple TCL programs; developing and extending speech corpora and lexicons; manipulating speech data with toolkits (e.g. PRAAT (Boersma and Weenink, 1996) and SFS[3]); extending a simple recognizer; and creating new speech-based dialogue scenarios. For the latter tasks the OGI toolkit (Cole, 1999) has proven particularly helpful, and the students find the environment stimulating, interesting, and versatile enough for a wide array of experimentation from both the linguistics and toolkit programming sides.

A final programming project is required; its deliverables include a proposal, interim report on progress, final presentation and class demonstration, and a final write-up including commented source code and a paper discussing related work, the approach taken, and results. Sample final projects include speech interfaces for ubiquitous task-specific access (e.g. intelligent kitchen appliances, automobile navigation, and large XML-marked document access), a call-center application for a dentist's office, and a spoken-language identifier.

2.3 Future CHum prospects

Computers and the Humanities course offerings are dynamic and flexible, and more NLP-related content is being offered beyond the parsing and speech technologies classes already mentioned. For example, currently an advanced seminar class is (for the first time) teaching XML and the annotation of linguistic data.

It should be noted that recent CHum efforts have been attracting CS students as well, who don't currently have an outlet for NLP-related coursework in their home department. This introduces a slight challenge for the instructor since an asymmetry exists between humanities and CS students' programming abilities. To date, though, this has not proved very problematic since the CS students were willing to rely on their own initiative to apply classroom concepts to more complex programming projects than their humanities peers could attempt. Presumably in the future if this becomes a problem a speech class might be offered in the CS department, or a in higher section in CHum.

An important and very recent development will ensure further strengthening of NLP-related courses within the College of Humanities. With the introduction of a new campus-wide interdisciplinary undergraduate minor in Computing and Information (CpIn), the CS department has secured several new tenure-track slots to be hosted externally by departments across campus. The College of Humanities has been allocated one of these slots, and the new faculty member will be housed within the Linguistics Department. More course offerings at the CS/Linguistics nexus will be possible in the near future as a result. In turn, these classes will serve as electives for CpIn students.

3 An advanced NLP course

The linchpin of the NLP curriculum is the advanced NLP course[4]. Hosted in the Linguistics department, it also has been cross-listed as a CS course. It is intended primarily for graduate students, though it is open to advanced undergrads with requisite background. Proficiency is assumed in at least one programming language; in addition, a background in algorithms, data structures, and some basic discrete math is also required. Linguistics students and CHum students with a solid background in linguistics and good programming skills are accepted with the instructor's approval.

3.1 Course goals and student demographics

The course's goals are: to teach how computational techniques are used successfully in various areas of NLP; to demonstrate by hands-on experience how to use NLP tools in the performance of linguistically interesting tasks; to demonstrate the application of a novel, nontrivial approach to solving some aspect of NLP-related computation; and to read, understand, and assess current research literature and trends in the relevant areas. The class is by design very broad in scope, trying to address as many areas of NLP as possible in a semester. The breadth of coverage entails that the depth in any one area cannot be addressed fully; still, each topic is addressed at some nontrivial level of detail. The topics are sequenced in such a way as to build upon previously introduced topics. For example, units on part-of-speech tagging

[3] See www.phon.ucl.ac.uk/resource/sfs/.

[4] See humanities.byu.edu/classes/ling581dl/homepage.html.

and lexicons precede those on parsing, which in turn precede those on speech understanding.

The class has been taught four times so far, with an average of ten students per semester (plus interested faculty who sat in on classes without registering). Each class had an equal three-way balance of students from CS, Linguistics, and other areas of campus (physics, engineering, and even modern languages). Half of the students are undergrads and half are graduates. Without exception, every student had knowledge of at least one non-English language. One of the challenges, but also unique opportunities, of this class is to bring their disparate backgrounds together in class discussions. For example class discussion, homework assignments, and final projects often center around the students' linguistic knowledge and their application of principles learned to the processing of non-English languages.

3.2 Course content

Materials: Class lectures, discussions, and demonstrations are based primarily on the content of two NLP texts (Manning and Schütze, 1999; Cole et al., 1997) and several supplementary readings from the Web[5]. The class is held thrice weekly in one-hour sessions; it too is held in a TELE room. Each student is required to "adopt" a lecture topic from the curriculum: researching intensively this particular field, preparing a lecture in consultation with the instructor, and teaching the class. Often students choose an area that reflects the strengths of their background, and as a result their lectures, materials and discussions are of high quality.

Coursework: Students are generally free to do their homework in any of the labs on campus or on their own machines elsewhere. In some cases, however, this is not possible due to licensing constraints on software needed for work in the course: several resources require that the data or programs only be used on an on-campus computer licensed to the CS and/or Linguistics departments. For this reason a Unix server has been acquired by the Linguistics department and set up with the requisite software to act as a classwork project server. Students can also access the machine remotely to do their work within these constraints. Students from the CS department have access to CS and Linguistics servers where class-related resources can be used. Students also have access to the campus supercomputer when necessary for NLP projects, under the instructor's supervision.

Sample non-trivial hands-on and programming assignments are given weekly. They include such topics as: work with various corpus manipulation and annotation tools, use of various POS taggers and their comparison (Brill, 1992; Tufis and Mason, 1998), development of morphophonological rules in PC-Kimmo (Antworth, 1990), understanding and manipulating content from WordNet databases (Fellbaum, 1998), aligning bitext, using and evaluating a machine translation system, developing a phrase-structure grammar for syntactic and then semantic chart parsing, experimenting with information retrieval, working with a speech toolkit to develop a simple application, or developing knowledge for a text generation engine (Tomita and Nyberg, 1988). Tutorials are provided for for any necessary remedial work that the student might need or desire in such topics as using the Emacs editor, using Unix shell scripts, or writing Perl or Tcl scripts.

Final project: A final programming project is required, similar in scope to that described above for the humanities course: close coordination with the instructor, meeting milestones, documenting and demonstrating the final product, and producing a write-up of the significance and contributions of the result. Of course, a much higher standard is required of these advanced students. The student is free to choose any relevant project, the programming language(s) to be used, and the theoretical approach to be taken. Sample final projects cover almost as wide a range of topics as those covered in the curriculum[6]. Linguistics students often focus on the programmatic development of knowledge sources whereas CS students tend to engineer large-scale integrations of several components in novel ways to address multi-faceted issues. The most common tendency with all students is to scope their work a little too ambitiously at first; close consultation with the instructor is crucial throughout the process. Teamwork is permitted, and often a Linguistics student will pair up with a CS

[5] Particularly useful are researchers' personal and project pages worldwide, the ACL NLP/CL Universe, and the arXiv Computation and Language archive.

[6] See humanities.byu.edu/classes/ling581dl/egnlpprojs.htm.

one; this usually results in good knowledge and skill transfer for both parties.

Evaluations: A three-hour (non-programming) final exam is given which tests a knowledge of concepts, algorithms, tools, procedures, and approaches learned throught the semester. Class evaluation ratings by students have improved over time, from very good (5.1/7.0, first time offered) to exceptional (6.7/7.0, last semester). The most frequent complaints concerned amount of background that the textbook assumes, and the lack of a midterm examination to help students gauge their progress.[7]

4 Other courses

The infrastructure developed for teaching the two courses mentioned above has also been successfully applied in other classes as well. This section explains how other classes have benefited from the NLP infrastructure being put in place.

A linguistics major undergrad survey course covers all of the core areas of linguistics (phonetics, phonology, morphology, syntax, semantics, and pragmatics) as well as several application areas. Interestingly, one chapter of the textbook used in this class even contains a very cursory overview of computational linguistics (Klavans, 2001). Several already-mentioned tools supporting the NLP classes have also been used in the undergrad survey class: a speech toolkit for sound wave manipulation, WordNet for lexical semantics, and a morphology engine for English.

The Linguistics department offers a translation theory and practice class, which traditionally attracts up to 40 students with advanced capabilities in as many as 25 languages per class section. With the NLP infrastructure recently developed, more technological exercises have been added to the curriculum involving WordNet, bitext alignment, corpus and lexicography tools, software localization (l10n) and internationalization (i18n), machine-assisted translation, and machine translation systems (standalone and web-based).

Other Linguistics classes also have recently leveraged the NLP infrastructure: a graduate semantics class uses WordNet, a grad phonology class works with a speech toolkit, a grad morphology class uses a morphology engine, and a language modeling seminar uses machine learning and other exemplar-based methods. In the CS department, a series of two 90-minute lectures in l10n and i18n has been developed and is regularly presented to the grad class in human-computer interation. Finally, several foreign-language classes outside of the Linguistics/CS area have used recently-assembled tools such as corpora, part-of-speech taggers, and WordNets in their own instruction and student assignments.

5 Extracurricular opportunities

As with any field of endeavor, chances to apply NLP principles acquired in the classroom enrich greatly a student's learning experience and solidify the concepts taught. Various outlets are provided on our campus for experiencing the field.

5.1 Research opportunities

Several research projects in both the Linguistics and CS departments welcome undergrad and graduate students. Weekly meetings involve keeping abreast of current research literature, discussing project deliverables, and practicing conference paper presentations. Groups where NLP-related work is done focus on: a data-driven, exemplar-based approach to language modeling, integrating speech tools and discourse engines for task-oriented language learning, extraction and integration of web-based document content, technologies in human and machine translation, and cognitive modeling of natural-language processing. Attendance is voluntary and no course credit is given, but generally participation in project work is enthusiastic and consistent, especially since the NLP class was initiated.

One relevant change in the NLP offerings is worthy of note: at first, the advanced NLP class was offered in Winter semester, with many students treating it like a capstone class in their last semester. They would then leave the school for further graduate work or employment. Consequently the class was recently moved from Winter to Fall, keeping the students' experience on campus for at least another semester's worth of project participation.

The university sponsors a mechanism for funding undergraduate research projects. Proposals are reviewed by faculty, and the very competitive program

[7] The instructor as a general rule does not give graduate classes midterms; this is being rethought for the NLP class because of student comment.

offers cash awards to the winners, who are required to submit a written report synthesizing results obtained. NLP students have had phenomenal success in winning these awards and have gained valuable experience with such projects as: morphological engines for Cebuano and Farsi, cognitive modeling of word-sense disambiguation, modeling task-oriented discourse structure, and developing exemplar-based machine translation. One advantage for NLP students in this competition is that interdisciplinary research is more likely to win funding.

5.2 Beyond the campus

In its short time, the NLP environment has also provided several students with the requisite skills to be placed in summer internships, most of them competitive and paid. Students have been placed in an European MT project, a U.S. industrial NLP research lab, another university's language acquisition research project, and a defense-funded research institute. Particularly appealing to their sponsors was the combination of proven foreign-language aptitudes with a computational background and an understanding of NLP techniques and tools.

Students whose project work is promising are encouraged to present their work at conferences, and several have presented their work at local, national, and international venues. Particularly noteworthy projects have also served as the foundation for peer-reviewed conference papers, undergrad honors theses, and master's theses. Successfully defended or ongoing theses in six departments (Linguistics, Spanish/Portuguese, Asian Languages, Language Acquisition, CS, and Physics) were initiated in the NLP class.

6 Reflections on issues

Naturally, communication and cooperation across department and college lines offers continual challenges for an interdisciplinary NLP curriculum. Still, both sides recognize the unique linguistic skills present in our students and the need to develop an environment fostering wider NLP expertise on-campus. Students, for their part, are attracted to such offerings and seem satisfied.

6.1 NLP and CL

One as-yet unrealized goal is to develop and offer a class in computational linguistics. Since Linguistics doesn't offer a Ph.D. degree, advanced grads are not available to the program; current students typically do not have, in this teacher's estimation, appropriate background in computational complexity, algorithmic analysis, AI techniques, formal logic, math/stats, and formal linguistics to be adequately prepared for an intensive CL class. To be sure, many students have a background in some of these areas, but not across a wide enough base to prove necessary for theoretical CL work. This may change over time, if a Ph.D. program is adopted; in the meantime, the NLP courses do fill a necessary niche.

Another promising recent development might help stimulate progress in this area: a newly hired CS faculty member with a computational linguistics background will begin teaching in that department next year. Initially, it has been decided to offer two somewhat complementary NLP classes. The CS Department will offer one class, which will resemble the advanced class discussed above, including using that textbook. The other class, hosted by Linguistics, will use a different text (Jurafsky and Martin, 2000) with its content focused more on the lexicon, morphology, speech, semantics, and deep parsing. Overlap between the two courses will be minimized as much as possible, with the goal of broadening NLP content offerings. Whether students will be attracted to a two-course sequence of NLP remains an open question.

6.2 Resource issues

A few obstacles and difficulties have been experienced in spite of the overall positive aspects of implementing an NLP curriculum mentioned in this paper. A few of these are sketched in this section.

Texts: A frequent complaint from Linguistics (but not CS) students enrolled in the advanced NLP class is that textbook discussions almost invariably focus on English. Since these students have all studied linguistics intensively, including exposure to a large variety of language typologies and nontrivial issues of language analysis and complexity, these discussions seem overly narrow and simplistic in many cases. As mentioned earlier, classroom discussion can to some extent elicit the wider-scope issues that interest them. Certainly the vast array of Web-published research findings also helps to fill this void. Still, several students have voiced the desire for a comprehensive NLP textbook that would address typologically diverse issues.

Support: One clear disadvantage to hosting an NLP class in the College of Humanities' Linguistics Department is one of comparatively limited resources. CS resources for coursework computers, research facilities, and student labs are noticeably superior. Software acquisition and licensing procedures, resolution of networking issues, and computer support are more problematic in a non-CS context on our campus. This is primarily a problem for CS students, who occasionally chafe at the need to use non-CS computers for coursework.

Tools: Tools accessibility is the greatest difficulty perceived by this author in trying to develop cohesive and coherent course content. With its admittedly fast-paced progress, the field of NLP has seen the development of a plethora of systems, tools, modules, and development environments for a wide array of tasks. Keeping track of these products, and assessing their availability and appropriateness for student use, is a daunting task. Several helpful resources assist this teacher besides those already mentioned. Listservs like LinguistList[8], Corpora[9], and LN[10] notify subscribers of new NLP software. There are even a few hierarchically-structured meta-indexes for NLP tools: among others, the ACL Universe (mentioned above), Colibri[11], and especially the Natural Language Software Registry[12]. These repositories, while helpful and greatly appreciated, tend to be spotty and idiosyncratic at best in their consideration of resources. Certainly a more systematic and comprehensive clearinghouse of NLP tools would be a boon to educators and researchers, particularly if its contents could be individually annotated for pedagogical applicability.

6.3 Teacher background

In a recent analysis of several years' worth of linguistics job announcements, Richard Sproat offered interesting conclusions relevant to employment in linguistics[13]. He notes "a prejudice that linguists are not typically well trained in computational skills", and the fact that "relatively few linguistics programs have got serious about providing their students with extensive computational training." In most current CL courses, he claims, "there is little emphasis on practical applications". If these observations hold, the NLP offerings discussed in this paper serve a valuable purpose in providing Linguistics (and CS) students much-needed practical experience in language computation technologies.

Sproat also detects a trend in linguistics showing "little effort to hire faculty members who have had extensive industrial experience", whereas in CS such experience is often desired, valued, and sought. He concludes that: "Departments thinking of building up [CL] programs would be well advised to consider people with industrial experience." The present author's 11-year experience in the NLP industry before pursuing graduate CL studies has proven invaluable in administering an NLP curriculum, facilitating such tasks as: advising students in their programming projects; directing them in job searches and internship opportunities; helping them propose and establish research agendas; collaborating with commercial and governmental sponsors; and dealing with issues of software licensing and technology transfer.

7 Conclusion

Our experience has shown that a small core of NLP classes housed in a Linguistics department and serving well-prepared students from other fields can greatly enhance the research and pedagogical infrastructure across many disciplines on-campus, while also preparing students for further grad studies or careers in the industrial NLP sector. Though challenges and issues remain, NLP courses are enjoying good enrollment, high satisfaction ratings, and appreciable learning outcomes.

[8] See www.linguistlist.org.

[9] See www.hit.uib.no/corpora/.

[10] See www.biomath.jussieu.fr/LN/.

[11] See colibri.let.uu.nl/.

[12] See registry.dfki.de.

[13] See www.research.att.com/˜rws/lingjobs.

References

James Allen. 1995. *Natural Language Understanding*. Benjamin/Cummings Publishing.

Evan Antworth. 1990. *PC-KIMMO: a two-level proecssor for morphological analysis*. Number 16 in Occasional Publications in Academic Computing. Summer Institute of Linguistics, Dallas, TX. See also www.sil.org/pckimmo/.

P. Boersma and D. J. M. Weenink. 1996. Praat, a system for doing phonetics by computer, version 3.4. Technical Report 132, Institute of Phonetic Sciences, Amsterdam. See also fonsg3.let.uva.nl/praat/.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the DARPA speech and natural language workshop*.

R. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue, editors. 1997. *Survey of the State of the Art in Human Language Technology*. Cambridge University Press.

Ron Cole. 1999. Tools for research and education in speech science. In *Proceedings of the International Conference of Phonetic Sciences*, San Francisco, CA, August. See also cslu.cse.ogi.edu/toolkit/index.html.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA. See also www.cogsci.princeton.edu/˜wn/.

J.M.Pickett. 1999. *The Acoustics of Speech Communication*. Allyn & Bacon.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice-Hall.

Judith Klavans. 2001. Computational linguistics. In William O'Grady, John Archibald, Mark Aronoff, and Janie Rees-Miller, editors, *Contemporary Linguistics: an Introduction*. Bedford/St. Martin's.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Masaru Tomita and Eric H. Nyberg, 3rd. 1988. Generation Kit and Transformation Kit: Version 3.2 user's manual. Technical Report CMU-CMT-88-MEMO, Carnegie Mellon Center for Machine Translation, October.

Dan Tufis and Oliver Mason. 1998. Tagging Romanian texts: a case study for QTAG, a language-independent probabilistic tagger. In *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, Grenada, Spain, May.