"Language and Computers" Creating an Introduction for a General Undergraduate Audience

Chris Brew
Department of Linguistics
The Ohio State University
cbrew@ling.osu.edu

Markus Dickinson
Department of Linguistics
The Ohio State University
dickinso@ling.osu.edu

W. Detmar Meurers
Department of Linguistics
The Ohio State University
dm@ling.osu.edu

Abstract

This paper describes the creation of Language and Computers, a new course at the Ohio State University designed to be a broad overview of topics in computational linguistics, focusing on applications which have the most immediate relevance to students. This course satisfies the mathematical and logical analysis requirement at Ohio State by using natural language systems to motivate students to exercise and develop a range of basic skills in formal and computational analysis. In this paper we discuss the design of the course, focusing on the success we have had in offering it, as well as some of the difficulties we have faced.

1 Introduction

In the autumn of 2003, we created Language and Computers (Linguistics 384), a new course at the Ohio State University that is designed to be a broad overview of topics in computational linguistics, focusing on applications which have the most immediate relevance to students. Language and Computers is a general enrollment course designed to meet the Mathematical and Logical Analysis requirement that is mandated for all undergraduates at the Ohio State University (OSU), one of the largest universities in the US. We are committed to serving the average undergraduate student at OSU, including those for whom this is the first and last Linguistics course. Some of the students take the course because it is an alternative to calculus, others because of curiosity about the subject matter. The course was first taught in Winter 2004, drawing a wide range of majors, and has since expanded to three sections of up to 35 students each. In this paper we will discuss the design of the course, focusing on the success we have had in offering it, as well as some of the difficulties we have faced.

2 General Context

The Linguistics Department at OSU is the home of a leading graduate program in which 17 graduate students are currently specializing in computational linguistics. From the perspective of the graduate program, the goal of the new course development was to create more appropriate teaching opportunities for the graduate students specializing in computational linguistics. Much of the undergraduate teaching load in Linguistics at OSU is borne by graduate teaching assistants (GTAs) who receive stipends directly from the department. After a training course in the first year, most such GTAs act as instructors on the Department's "Introduction to Language," which is taught in multiple small sections. Instructors are given considerable responsibility for all aspects of course design, preparation, delivery, and grading. This works very well and produces many superb instructors, but by 2003 it was apparent that increasing competition was reducing the pool of undergraduates who want to take this general overview course.

The Ohio State University has a distribution requirement, the General Education Curricu-

Proceedings of the Second ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL, pages 15–22,
Ann Arbor, June 2005.

lum (GEC), that is designed to ensure adequate breadth in undergraduate education. The twin demands of the student's major and the distribution requirement are sufficient to take up the vast majority of the credit hours required for graduation. In practice this means that students tend to make course selections motivated primarily by the goal of completing the necessary requirements as quickly and efficiently as they can, possibly at the expense of curiosity-driven exploration. Linguistics, as an interdisciplnary subject, can create courses that satisfy both curiosity and GEC requirements.

To fill this interdisciplinary niche, the OSU Department of Linguistics has created a range of new courses such as Language and Gender, Language and the Mind, Language and the Law, and the Language and Computers course discussed in this paper. In addition to filling a distribution requirement niche for undergraduates, the courses also allow the linguistics GTAs to teach courses on topics that are related to their area of specialization, which can be beneficial both to the instructors and to those instructed. Prior to creation of the new Language and Computers course, there were virtually no opportunities for student members of the computational linguistics group to teach material close to their focus.

3 Course overview

The mission statement for our course reads:

> In the past decade, the widening use of computers has had a profound influence on the way ordinary people communicate, search and store information. For the overwhelming majority of people and situations, the natural vehicle for such information is natural language. Text and to a lesser extent speech are crucial encoding formats for the information revolution. This course will give students insight into the fundamentals of how computers are used to represent, process and organize textual and spoken information, as well as providing tips on how to effectively integrate this knowledge into their working practice. The course will cover the theory and practice of human language technology.

The course was designed to meet the Mathematical and Logical Analysis (MLA) requirement for students at the Ohio State University, which is characterized in the following way:

> A student in a B.A. program must take one course that focuses on argument in a context that emphasizes natural language, mathematics, computer science or quantitative applications not primarily involving data. Courses which emphasize the nature of correct argumentation either in natural languages or in symbolic form would satisfy this requirement, as would many mathematics or computer science courses. ... The courses themselves should emphasize the logical processes involved in mathematics, inductive or deductive reasoning, or computing and the theory of algorithms.

Linguistics 384 responds to this specification by using natural language systems to motivate students to exercise and develop a range of basic skills in formal and computational analysis. The course combines lectures with group work and in-class discussions, resulting in a seminar-like environment. We enrol no more than 35 students per section, often significantly fewer at unpopular times of day.

The course philosophy is to ground abstract concepts in real world examples. We introduce strings, regular expressions, finite-state and context-free grammars, as well as algorithms defined over these structures and techniques for probing and evaluating systems that rely on these algorithms. This meets the MLA objective to emphasize the nature of correct argumentation in symbolic form as well as the logical processes involved in computing and the theory of algorithms. These abstract ideas are embedded in practical applications: web searching,

spelling correction, machine translation and dialogue systems. By covering the technologies behind these applications, the course addresses the requirement to sharpen a student's ability to reason critically, construct valid arguments, think creatively, analyze objectively, assess evidence, perceive tacit assumptions, and weigh evidence.

Students have impressions about the quality of such systems, but the course goes beyond merely subjective evaluation of systems and emphasizes the use of formal reasoning to draw and argue for valid conclusions about the design, capabilities and behavior of natural language systems.

In ten weeks, we cover eight topics, using a data projector in class, with copies of the slides being handed out to the student before each class. There is no textbook, and there are relatively few assigned readings, as we have been unable to locate materials appropriate for an average student without required background who may never take another (computational) linguistics class. The topics covered are the following, in this order:

- Text and speech encoding
- (Web-)Searching
- Spam filtering (and other classification tasks, such as language identification)
- Writers' aids (Spelling and grammar correction)
- Machine translation (2 weeks)
- Dialogue systems (2 weeks)
- Computer-aided language learning
- Social context of language technology use

In contrast to the courses of which we are aware that offer computational linguistics to undergraduates, our Language and Computers is supposed to be accessible without prerequisites to students from every major (a requirement for GEC courses). For example, we cannot assume any linguistic background or language awareness. Like Lillian Lee's Cornell course (Lee, 2002), the course cannot presume programming ability. But the GEC regulations additionally prohibit us from requiring anything beyond high school level abilities in algebraic manipulation. We initially hoped that this meant that we would be able to rely on the kind of math knowledge that we ourselves acquired in secondary school, but soon found that this was not realistic. The sample questions from Lee's course seem to us to be designed for students who actively enjoy math. Our goal is different: we want to exercise and extend the math skills of the general student population, ensuring that the course is as accessible to the well-motivated dance major as it is to the geekier people with whom we are somewhat more familiar. This is hard, but worthwhile.

The primary emphasis is on discrete mathematics, especially with regard to strings and grammars. In addition, the text classification and spam-filtering component exercise the ability to reason clearly using probabilities. All of this can be achieved for students with no collegiate background in mathematics.

Specifically, Linguistics 384 uses non-trivial mathematics at a level at or just beyond algebra 1 in the following contexts:

- Reasoning about finite-state automata and regular expressions (in the contexts of web searching and of information management). Students reason about relationships between specific and general search terms.

- Reasoning about more elaborate syntactic representations (such as context-free grammars) and semantic representations (such as predicate calculus), in order to better understand grammar checking and machine translation errors.

- Reasoning about the interaction between components of natural language systems (in the contexts of machine translation and of dialog systems).

- Understanding the basics of dynamic programming via spelling correction (edit distance) and applying algebraic thinking to algorithm design.

- Simple probabilistic reasoning (in the context of text classification).

There is also an Honors version of the course, which is draws on a somewhat different pool of students. In 2004 the participants in Honors 384 were equally split between Linguistics majors looking for a challenging course, people with a computer background and some interest in language and people for whom the course was a good way of meeting the math requirement at Honors level. Most were seniors, so there was little feed-through to further Linguistics courses.

The Honors course, which used to be called Language Processing Technology, predates Language and Computers, and includes more hands-on material. Originally the first half of this course was an introduction to phonetics and speech acoustics through Praat, while the second was a Prolog-based introduction to symbolic NLP. We took the opportunity to redesign this course when we created the non-honors version. In the current regime, the hands-on aspect is less important than the opportunities offered by the extra motivation and ability of these students. Two reading assignments in the honors version were Malcolm Gladwell's book review on the Social Life of Paper (Gladwell, 2001) and Turing's famous paper on the Imitation Game (Turing, 1950). We wondered whether the eccentricity and dated language of the latter would be a problem, but it was not.

Practical assignments in the laboratory are possible in the honors course, because the class size can be limited. One such assignment was a straightforward run-through of the clock tutorial from the Festival speech synthesis system and another a little machine translation system between digits and number expressions. Having established that they can make a system that turns 24 into 'twenty four', and so on, the students are challenged to adapt it to speak "Fairy Tale English": that is, to make it translate 24 into 'four and twenty', and vice-versa.

[1]

[1]For a complete overview of the course materials, there are several course webpages to check out. The webpage for the first section of the course (Winter 2004) is at `http://ling.osu.edu/~dickinso/384/wi04/`. A more recent section (Winter 2005) can be found at `http://ling.osu.edu/~dm/05/winter/384/`. For the honors course, the most recent version is located at `http://ling.osu.edu/~cbrew/2005/spring/H384/`. A list of weblinks to demos, software, and on-line tutorials currently used in connection with the course can be found at `http://ling.osu.edu/~xflu/384/384links.html`

4 General themes of the course

Across the eight different topics that are taught, we try to maintain a cohesive feel by emphasizing and repeating different themes in computational linguistics. Each theme allows the students to see that certain abstract ideas are quite powerful and can inform different concrete tasks. The themes which have been emphasized to this point are as follows:

- There are both statistical and rule-based methods for approaching a problem in natural language processing. We show this most clearly in the spam filtering unit and the machine translation unit with different types of systems.
- There is a tension between developing technology in linguistically-informed ways and developing technology so that a product is effective. In the context of dialogue systems, for example, the lack of any linguistic knowledge in ELIZA makes it fail quickly, but an ELIZA with a larger database and still no true linguistic knowledge could have more success.
- Certain general techniques, such as n-gram analysis, can be applied to different computational linguistic applications.
- Effective technology does not have to solve every problem; focusing on a limited domain is typically more practical for the applications we look at. In machine translation, this means that a machine translation system translating the weather (e.g., the METEO system) will perform better than a general-purpose system.
- Intelligent things are being done to improve natural language technology, but the task is a very difficult one, due to the complexities of language. Part of each unit is devoted to

showing that the problem the technology is addressing is a complex one.

5 Aspects of the course that work

The course has been a positive experience, and students overall seemed pleased with it. This is based on the official student evaluation of instruction, anonymous, class specific questionnaires we handed out at the end of the class, personal feedback, and new students enrolling based on recommendations from students who took the course. We attribute the positive response to several different aspects of the course.

5.1 Topics they could relate to

Students seem to most enjoy those topics which were most relevant to their everyday life. On the technological end, this means that the units on spam filtering, web searching, and spell checking are generally the most well-received. The more practical the focus, the more they seem to appreciate it; for web searching, for instance, they tend to express interest in becoming better users of the web. On the linguistic end, discussions of how dialogue works and how language learning takes place, as part of the units on dialogue systems and CALL, respectively, tend to resonate with many students. These topics are only sketched out insofar as they were relevant to the NLP technology in question, but this has the advantage of not being too repetitive for the few students who have had an introductory linguistics class before.

5.2 Math they can understand

Students also seem to take pride in being able to solve what originally appear to be difficult mathematical concepts. To many, the concept and look of a binary number is alien, but they consistently find this to be fairly simple. The basics of finite-state automata and boolean expressions (even quite complicated expressions) provide opportunities for students to understand that they are capable of learning concepts of logical thinking. Students with more interest and more of an enjoyment for math are encouraged to go beyond the material and, e.g., figure out the nature of more complicated finite-state automata. In this way, more advanced students are able to stay interested without losing the other students.

More difficult topics, such as calculating the minimum edit distance between a word and its misspelling via dynamic programming, can be frustrating, but they just as often are a source of a greater feeling of success for students. After some in-class exercises, when it becomes apparent that the material is learnable and that there is a clear, well-motivated point to it, students generally seem pleased in conquering somewhat more difficult mathematical concepts.

5.3 Interactive demos

In-class demos of particular software are also usually well-received, in particular when they present applications that students themselves can use. These demos often focus on the end result of a product, such as simply listening to the output of several text-to-speech synthesizers, but they can also be used for understanding how the applications works. For example, some sections attempt to figure out as a class where a spelling checker fails and why. Likewise, an in-class discussion with ELIZA has been fairly popular, and students are able to deduce many of the internal properties of ELIZA.

5.4 Fun materials

In many ways, we have tried to keep the tone of the course fairly light. Even though we are teaching mathematical and logical concepts, these concepts are still connected to the real world, and as such, there is much opportunity to present the material in a fun and engaging manner.

Group work One such way to make the learning process more enjoyable was to use group work. In the past few quarters, we have been refining these exercises. Because of the nature of the topics, some topics are easier to derive group exercises for than others. The more mathematical topics, such as regular expressions, suit themselves well for straightforward group work on problem sets in class; others can be more

creative. The group exercises usually serve as a way for students to think about issues they already know something about, often as a way to introduce the topic.

For example, on the first day, they are given a sheet and asked to evaluate sets of opposing claims, giving arguments for both sides, such as the following:

1. A person will have better-quality papers if they use a spell checker.

   A person will have worse-quality papers if they use a spell checker.

2. An English-German dictionary is the main component needed to automatically translate from English to German.

   An English-German dictionary is not the main component needed to automatically translate from English to German.

3. Computers can make you sound like a native speaker of another language.

   Computers cannot make you sound like a native speaker of another language.

To take another example, to get students thinking about the social aspects of the use of language technology, they are asked in groups to consider some of the implications of a particular technology. The following is an excerpt from one such handout.

> You work for a large software company and are in charge of a team of computational linguists. One day, you are told: "We'd like you and your team to develop a spell checker for us. Do you have any questions?" What questions do you have for your boss?
>
> ...
>
> Somehow or another, the details of your spell checker have been leaked to the public. This wouldn't be too bad, except that it's really ticked some linguists off. "It's just a big dictionary!" they yell. "It's like you didn't know anything about morphology or syntax or any of that good stuff." There's a rumor that they might sue you for defamation of linguistics. What do you do?

Although the premise is somewhat ridiculous, with such group work, students are able to consider important topics in a relaxed setting. In this case, they have to first consider the specifications needed for a technology to work (who will be using it, what the expectations are, etc.) and, secondly, what the relationship is between the study of language and designing a product which is functional.

Fun homework questions In the homeworks, students are often instructed to use a technology on the internet, or in some way to take the material presented in class a step farther. Additionally, most homework assignments had at least one lighter question which allowed students to be more creative in their responses while at the same time reinforcing the material.

For example, instructors have asked students to send them spam, and the most spam-worthy message won a prize. Other homework questions have included sketching out what it would take to convert an ELIZA system into a hostage negotiator—and what the potential dangers are in such a use. Although some students put down minimal answers, many students offer pages of detailed suggestions to answer such a question. This gives students a taste of the creativity involved in designing new technology without having to deal with the technicalities.

6 Challenges for the course

Despite the positive response, there are several aspects to the course which have needed improvement and continue to do so. Teaching to a diverse audience of interests and capabilities presents obstacles which are not easily overcome. To that end, here we will review aspects of the course which students did not generally enjoy and which we are in the process of adapting to better suit our purposes and our students' needs.

6.1 Topics they do not relate to

For such a range of students, there is the difficulty of presenting abstract concepts. Although we try to relate everything to something which students actually use or could readily use, we sometimes include topics from computational linguistics that make one better able to think logically in general and which we feel will be of future use for our students. One such topic is that of regular expressions, in the context of searching for text in a document or corpus. As most students only experience searching as part of what they do on the web, and no web search engine (to the best of our knowledge) currently supports regular expression searching, students often wonder what the point of the topic is. In making most topics applicable to everyday life, we had raised expect. In this particular case, students seemed to accept regular expressions more once it they saw that Microsoft Word has something roughly analogous.

Another difficulty that presented itself for a subset of the students was that of using foreign language text to assist in teaching machine translation and computer-aided language learning. Every example was provided with an English word-by-word gloss, as well as a paraphrase, yet the examples can still be difficult to understand without a basic appreciation for the relevant languages. If the students know Spanish, the example is in Spanish and the instructor has a decent Spanish accent, things can go well. But students tend to blame difficulties in the machine translation homework on not knowing the languages used in the examples. Understanding the distinction between different kinds of machine translation systems requires some ability to grasp how languages can differ, so we certainly must (unless we use proxies like fairy-tale English) present some foreign material, but we are in dire need of means to do this as gently as possible

6.2 Math they do not understand

While some of the more difficult mathematical concepts were eventually understood, others continued to frustrate students. The already mentioned regular expressions, for example, caused trouble. Firstly, even if you do understand them, they are not necessarily life-enhancing, unless you are geeky enough to write your papers in a text editor that properly supports them. Secondly, and more importantly, many students saw them as unnecessarily abstract and complex. For instance, some students were simply unable to understand the notion that the Kleene star is to be interpreted as an operator rather than as a special character occurring in place of any string.

Even though we thought we had calibrated our expectations to respect the fact that our students knew no math beyond high school, the amount that they had retained from high school was often less than we expected. For example, many students behaved exactly as if they had never seen Venn diagrams before, so time had to be taken away from the main material in order to explain them. Likewise, figuring out how to calculate probabilities for a bag of words model of statistical machine translation required a step-by-step explanation of where each number comes from. A midterm question on Bayesian spam filtering needed the same treatment, revealing that even good students may have significant difficulties in deploying the high school math knowledge they almost certainly possess.

6.3 Technology which did not work

Most assignments required students to use the internet or the phone in some capacity, usually to try out a demo. With such tasks, there is always the danger that the technology will not work. For example, during the first quarter the course was taught, students were asked to call the CMU Communicator system and interact with it, to get a feel for what it is like to interact with a computer. As it turns out, halfway through the week the assignment was due, the system was down, and thus some students could not finish the exercise. Following this episode, homework questions now come with alternate questions. In this case, if the system is down, the first alternate is to listen to a pre-recorded conversation to see how the Com-

municator works. Since some students are unable to listen to sounds in the campus computer labs, the second alternate is to read a transcript.

Likewise, students were instructed to view the page source code for ELIZA. However, some campus computer labs at OSU do not allow students to view the source of a webpage. In response to this, current versions of the assignment have a separate webpage with the source code written out as plain text, so all students can view it.

One final note is that students have often complained of weblinks failing to work, but this "failure" is most often due to students mistyping the link provided in the homework. Providing links directly on the course webpage or including them in the web- or pdf-versions of the homework sheets is the simplest solution for this problem.

7 Summary and Outlook

We have described the course Language and Computers (Linguistics 384), a general introduction to computational linguistics currently being taught at OSU. While there are clear lessons to be learned for developing similar courses at other universities, there are also more general points to be made. In courses which assume some CS background, for instance, it is still likely the case that students will want to see some practical use of what they are doing and learning.

There are several ways in which this course can continue to be improved. The most pressing priority is to develop a course packet and possibly a textbook. Right now, students rely only on the instructor's handouts, and we would like to provide a more in-depth and cohesive source of material. Along with this, we want to develop a wider range of readings for students (e.g. Dickinson, to appear) to provide students with a wider variety of perspectives and explanations for difficult concepts.

To address the wide range of interests and capabilities of the students taking this course as a general education requirement, it would be good to tailor some of the sections to audiences with specific backgrounds—but given the lack of a dedicated free time slot for all students of a particular major, etc., it is unclear whether this is feasible in practice.

We are doing reasonably well in integrating mathematical thinking into the course, but we would like to give students more experience of thinking about algorithms. Introducing a basic form of pseudocode might go some way towards achieving this, provided we can find a motivating linguistic example that is both simple enough to grasp and complex enough to justify the overhead of introducing a new topic. Further developments might assist us in developing a course between Linguistics 384 and Linguistics 684, our graduate-level computational linguistics course, as we currently have few options for advanced undergraduates.

Acknowledgements We would like to thank the instructors of Language and Computers for their discussions and insights into making it a better course: Stacey Bailey, Anna Feldman, Xiaofei Lu, Crystal Nakatsu, and Jihyun Park. We are also grateful to the two ACL-TNLP reviewers for their detailed and helpful comments.

References

Markus Dickinson, to appear. Writers' Aids. In Keith Brown (ed.), *Encyclopedia of Language and Linguistics. Second Edition*, Elsevier, Oxford.

Malcolm Gladwell, 2001. The Social Life of Paper. *New Yorker*. available from `http://www.gladwell.com/archive.html`.

Lillian Lee, 2002. A non-programming introduction to computer science via NLP, IR, and AI. In *ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*. pp. 32–37.

A.M. Turing, 1950. Computing Machinery and Intelligence. *Mind*, 59(236):433–460.