A Balanced and Broadly Targeted Computational Linguistics Curriculum

Emma Manning **Nathan Schneider** **Amir Zeldes**
Georgetown University
{esm76, nathan.schneider, amir.zeldes}@georgetown.edu

Abstract

This paper describes the primarily-graduate computational linguistics and NLP curriculum at Georgetown University, a U.S. research university that has seen significant growth in these areas in recent years. We discuss the principles behind our curriculum choices, including recognizing the various academic backgrounds and goals of our students; teaching a variety of skills with an emphasis on working directly with data; encouraging collaboration and interdisciplinary work; and including languages beyond English. We reflect on challenges we have encountered, such as the difficulty of teaching programming skills alongside NLP fundamentals, and discuss areas for future growth.

1 Introduction

This paper describes the computational linguistics and NLP curriculum at Georgetown University, a private research university whose computational linguistics program has grown significantly over the past 7 years. This curriculum has been developed with a high degree of collaboration between the Linguistics and Computer Science departments, as well as involvement with related programs such as the Master's in Data Science and Analytics. We reflect on several principles that underlie our curriculum, and discuss opportunities for further expansion.

2 Course Offerings

Table 1 summarizes our main graduate-level courses focusing on computational linguistics and NLP.[1] These include:

- A 3-course NLP sequence for novice programmers, which can be shortened to 1 or 2 courses for students proficient in Python (discussed in §3).
- A group of courses targeting methods for computational linguistics research: corpus design and use (§5), statistical analysis with R, and machine learning techniques.
- A selection of application-oriented speech and language technology courses encompassing speech processing, dialogue systems, and machine translation.
- Special topics courses addressing issues such as social factors and ethics in NLP, discourse parsing, grammar formalisms, and meaning representation design and parsing. These tend to be reading- and research-oriented courses, whereas the other courses place more emphasis on implementation and theory learning.

Advanced students of NLP can also take a number of related courses in the CS and Analytics departments on topics like information retrieval and machine learning for general (and not only language-oriented) purposes.[2]

Syllabi for courses taught by Nathan Schneider (instructor S in the table), including detailed schedules with course materials, can be found via http://people.cs.georgetown.edu/nschneid/teaching.html. Recent syllabi for courses taught by Amir Zeldes (instructor Z) can be found at https://corpling.uis.georgetown.edu/amir/pdf/syllabi/.

3 Interdisciplinarity

Our students include many from both the Linguistics and Computer Science departments, as well as some from other programs, such as Data Science & Analytics and language departments. We have developed a sequence of NLP courses designed

[1] All are standard fall or spring semester courses for 3 credits, with 150 minutes of instruction time per week. The total number of 3-credit courses a student takes varies considerably by graduate program: 8–10 for the CS MS; 10 for the CS Ph.D.; 8–12 for the Linguistics MS; and 18 for the Linguistics Ph.D. This includes departmental core requirements and other course options beyond computational linguistics and NLP.

[2] A full list of CL-relevant courses are described at: http://gucl.georgetown.edu/gu-cl-curriculum.pdf

Proceedings of the Fifth Workshop on Teaching NLP, pages 65–69
June 10–11, 2021. ©2021 Association for Computational Linguistics

	Course	Target audience	Frequency	Instructor
NLP	Intro NLP (INLP)	any except CS	Annual	Z
	Advanced Python for CL	Ling+Analytics	Annual	A
	Empirical Methods in NLP (ENLP)	Ling+CS	Annual	S
CL METHODS	Computational Corpus Linguistics	any	Annual	Z
	Analyzing Language Data with R	Ling	2 Years	Z
	Machine Learning for Linguistics	Ling	2 Years	Z
APPLICATIONS	Speech Processing	Ling	2 Years	A
	Dialogue Systems	any	2 Years	A
	Statistical/Neural Machine Translation	any	2 Years	A
SPECIAL TOPICS	Social Factors in CL/AI	any	2 Years	A
	Discourse Modeling	Ling+CS	2 Years	Z
	Grammar Formalisms	Ling	3–4 Years	P
	Meaning Representations	Ling+CS	2 Years	S

Table 1: Courses oriented specifically at computational linguistics or NLP and targeting graduate students (many are also open to undergraduates in their third and fourth years). The first group is the main NLP sequence that includes Python programming and fundamental algorithms, representations, and tasks; fluent Python programmers can start with ENLP. The second group focuses on computational linguistic methods. The third group focuses on application areas and associated tools. The last group consists of special topics. **Instructors**: Courses designated S or Z are taught by dedicated computational linguistics faculty, Nathan **S**chneider and Amir **Z**eldes. Grammar Formalisms is taught by Paul **P**ortner, a Linguistics professor. Other courses, designated A, are taught by **A**djunct professors (different for each course).

to accommodate these various backgrounds. Linguistics students with little or no prior programming experience are introduced to basic Python and NLP foundations in an Introduction to NLP (INLP) course;[3] they can then further develop their programming skills with Computational Linguistics with Advanced Python before taking Empirical Methods in NLP (ENLP). Students who already have strong programming skills, such as Computer Science graduate students, can begin their NLP journey in this same ENLP course, which has projects emphasizing collaboration between students of different backgrounds;[4] as discussed in Fosler-Lussier (2008), cross-disciplinary collaborations are helpful to establish respect between students from different fields and mitigate the challenges of disparate backgrounds. Many other NLP courses, such as those focusing on Dialogue Systems and Machine Translation, are also cross-listed between the Linguistics and CS departments, which, as noted in e.g. Baldridge and Erk (2008), helps these courses reach a wider audience.

Teaching NLP concepts alongside basic programming skills has been a significant challenge. INLP requires no prior programming experience, but students who enter the course with none sometimes struggle to grasp programming concepts at the speed they are taught, and many students rely on significant support from teaching assistants to successfully complete the course's programming assignments. Our experience has taught us that frequent contact and check-ins initiated by teaching assistants are very important for catching students who may fall behind before assignment submissions make problems more obvious. Use of IDEs with syntax validation and auto-complete facilities, which are freely available for academic purposes, are also very useful in this respect, and in recent years students have used PyCharm (`https://www.jetbrains.com/pycharm/`) as their first Python IDE for this purpose.

Previously, linguistics students who completed INLP were encouraged to enroll in ENLP immediately afterward. However, we found that INLP alone did not adequately prepare students for the more advanced programming assignments in ENLP—INLP assignments tend to involve making fairly limited modifications to provided starter code, while ENLP expects independent implementation of more substantial algorithms. Thus, the Advanced Python course was introduced to give

[3] Introducing these together allows linguistics students who are unsure how interested they are in NLP to get a taste of it in just one class, without requiring them to spend time on an non-language-related programming class first.

[4] Depending on class makeup, there are sometimes requirements for the composition project groups to enforce this, e.g. that each group needs to contain at least one linguist.

students more practice implementing algorithms for linguistic tasks as code. This bridges the gap between the introductory and more advanced NLP courses; however, it does mean that linguistics students who enter the program with little or no programming experience may need to take a sequence of 3 courses to gain a thorough understanding of NLP fundamentals, while students with a CS background only need to take one course. ENLP's Assignment 0 is a diagnostic of Python proficiency to help students choose the appropriate course level.[5]

4 Balancing Skills Taught

Along with coming from different academic backgrounds, we acknowledge that students studying NLP have a variety of goals: for example, they may wish to pursue NLP in academia or industry, or they may be interested in using computational methods for linguistics, or other Digital Humanities or Social Science fields. To support these varying goals, we endeavor to teach a balance of different skills and perspectives on NLP. While some courses emphasize algorithms, others focus more on computational representations of language, on creating and using resources such as corpora, or on using existing NLP tools. We are also careful to consider that not all NLP applications are realized in a Big Data context, and we therefore include units targeting low resource settings across our course offerings.

5 Focus on Data

In all courses, we emphasize working directly with language data. This is perhaps best exemplified in the Computational Corpus Linguistics course, which teaches corpus design and construction methods along with analytical Corpus Linguistics methodology and relevant readings on data and its potential pitfalls. As part of its assignment structure, the course integrates a set of five annotation projects in which each student chooses a single document from a selection of genres to annotate throughout the semester with a variety of annotations layers, including structural markup (which teaches XML basics), part-of-speech tagging, dependency treebanking, entity and coreference resolution, and finally discourse parsing. A unit on inter-annotator agreement evaluates and compares the students' own work, underscoring the subjective nature of 'ground-truth' data, the range of linguistic variation across genres, and the importance of consistency and validation. At its end, the course engages students in a 'real-world' research project, which produces valuable linguistically annotated data, which can be released to the research community under an open license as part of the Georgetown University Multilayer (GUM) Corpus (Zeldes, 2017) if students so wish.[6]

Several other courses also include practice annotation of data, including a POS tagging in-class exercise in ENLP, and annotation in three different semantic representations in Meaning Representations. Others include error analysis of NLP systems, such as a comparison of the output from statistical and neural translation systems in Machine Translation. The Discourse Modeling course teaches discourse parsing frameworks and algorithms, including introducing students to topics in annotating Rhetorical Structure Theory (Mann and Thompson, 1988), Segmented Discourse Representation Theory (SDRT, Asher and Lascarides 2003) and the Penn Discourse Treebank framework (PDTB, Prasad et al. 2014).

6 Collaboration

Our coursework emphasizes frequent collaboration among students. This includes in-class group activities, such as practicing part-of-speech tagging in small groups in ENLP, or working together as a class to create a morphological analyzer for a low resource language in INLP (an activity which literally runs in a simultaneous collaborative online code editing format). On a larger scale, students work in groups on final projects in courses such as ENLP, and have collaborated on an entry to a shared task in a our discourse parsing course, with the resulting system winning some shared task categories. For this latter project, students attempted to tackle the same task in small groups, and finally submitted an ensemble system fed by each group's model to the competition.

Some classes use wikis to maintain information about course content, such as annotation guidelines for Computational Corpus Linguistics and some seminars tackling specific topics; this allows students to collaborate not only with their classmates, but with past and future students of the same course, which also increases the sense of relevance

[5] http://people.cs.georgetown.edu/cosc572/s21/a0/

[6] https://corpling.uis.georgetown.edu/gum/

of course work, as students can see that their work may live on long after they complete the course.

7 Including Languages Beyond English

In response to an unfortunate tendency of NLP teaching and research to focus primarily on English, we try to include data and examples from other languages when possible, while keeping in mind that students cannot be expected to know these languages in detail. In ENLP, for example, each student gives a short presentation on a different language of their choosing to develop awareness of the diversity of the world's languages, and the challenges of NLP on different languages. Other assignments integrating data from other languages include a finite state transducer for Japanese morphology in INLP as well as a unit on a 'surprise' low resource language, work on multilingual discourse treebanks in our discourse parsing course, statistical analysis of non-English data in our stats-centered R course for language data, and analysis of data from other languages in Lexical Functional Grammar (LFG) and Head-driven Phrase Structure Grammar (HPSG) in Grammar Formalisms. In the past we have also offered a dedicated course on parallel and other types of multilingual corpora, which we hope to be able to offer again, based on the availability of resources.

8 Teaching Research Skills

While many of the courses in table 1 cover textbook-style fundamentals, the Special Topics courses expose students to the scientific literature in particular areas. For Ph.D. students in particular, this provides the opportunity to engage with research ideas by reading critically and developing original ideas as term papers or projects. Two courses include a mock reviewing activity simulating an ACL reviewing experience. Final projects in several courses—including Corpus Linguistics, Machine Learning for Linguistics, ENLP, and Meaning Representations—consist of an open-ended research project with an ACL-style writeup for the final report.

9 Directions for Growth

The current curriculum caters primarily to graduate students, though many of the courses are also available to advanced undergraduates, who sometimes continue on into our regular Computational Linguistics MS, or Accelerated MS programs. While we do offer a few undergrad-specific classes, such as 'Algorithms for NLP,' 'Languages and Computers,' and 'Multilingual and Parallel Corpora,' these are taught on an occasional basis; in the future, resources allowing, we would like to develop a more consistent NLP curriculum aimed at undergraduates.

We have recently introduced a course on Social Factors in NLP to address a major gap in our curriculum, which is the lack of material focusing on the impact of real world NLP applications on society, and the ways in which models reflect demographic and other types of bias. While this is a step toward teaching a better understanding of the relationship of NLP to society, we believe it is worthwhile to integrate more content on societal impact and ethical considerations into the core NLP courses as well, and are working to do so for coming years. We would also like to continue to expand our curriculum to address other topics that currently receive little coverage, such as grammar engineering and computational psycholinguistics.

Acknowledgments

We would like to acknowledge the other faculty who teach the courses described in this paper: Matthew Marge, Corey Miller, Elizabeth Merkhofer, Paul Portner, Achim Ruopp, and Shabnam Tafreshi. We thank the anonymous reviewers and members of the NERT lab for feedback on this paper, as well as the organizers of the workshop.

References

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Studies in Natural Language Processing. Cambridge University Press, Cambridge.

Jason Baldridge and Katrin Erk. 2008. Teaching computational linguistics to a large, diverse student body: Courses, tools, and interdepartmental interaction. In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, pages 1–9, Columbus, Ohio. Association for Computational Linguistics.

Eric Fosler-Lussier. 2008. Strategies for teaching "mixed" computational linguistics classes. In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, pages 36–44, Columbus, Ohio. Association for Computational Linguistics.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Rashmi Prasad, Bonnie Webber, and Aravind Joshi. 2014. Reflections on the Penn Discourse TreeBank, comparable corpora, and complementary annotation. *Computational Linguistics*, 40(4):921–950.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.