Georgetown University’s Computational Linguistics Curriculum

Nathan Schneider, January 2021

1. The CL Landscape at Georgetown **2**
Course Conventions at Georgetown 2
Departments and Programs 3
Linguistics department 3
Computer Science department 5
Other graduate programs 7
2. Course Offerings **8**
Computational Grammar Formalisms 8
LING-428 | Grammar Formalisms for Computational Research 8
Computational Corpus Linguistics 8
LING-264 | Multilingual and Parallel Corpora 8
LING-367 | Computational Corpus Linguistics 9
LING-469 | Analyzing language data with R 9
CL Fundamentals 9
LING-261 | Language and Computers 9
Natural Language Processing 10
COSC/LING-272 | Algorithms for NLP 10
LING-362 | Introduction to Natural Language Processing 10
LING-472/ANLY-521 | Computational Linguistics with Advanced Python 11
COSC/LING-572 | Empirical Methods in Natural Language Processing 11
ANLY-580 | NLP for Data Analytics 11
COSC/LING-672 | Advanced Semantic Representation 11
LING-765 | Computational Discourse Models 12
COSC-872 | Doctoral Seminar: NLP 12
Information Retrieval, Information Extraction, and Data Mining 13
COSC-285 | Introduction to Data Mining 13
COSC-488 | Information Retrieval 13
COSC-586 | Text Mining & Analysis 13
COSC-589 | Web Search and Sense-making 13
COSC-592 | Health Search and Mining 14
COSC-880 | Doctoral Seminar: Search Systems 14
COSC-882 | Doctoral Seminar: Text Mining and NLP 14
Spoken and Interactive Language Processing 14
LING-461 | Signal Processing 14
LING-463/COSC-483 | Dialogue Systems 15
Other CL Applications 15
LING-462/COSC-482 | Statistical Machine Translation 15
LING-464 | Social Factors in Comp Ling and AI 16
Machine Learning and Artificial Intelligence 16
COSC-270 | Artificial Intelligence 16
COSC-288 | Introduction to Machine Learning 16
LING-504 | Machine Learning for Linguistics 17
COSC-574 | Automated Reasoning 17
COSC-575 | Machine Learning 17
COSC-576 | Introduction to Deep Learning with Neural Nets 18
COSC-578 | Statistical Machine Learning 18
COSC-688 | Experimental AI 18
COSC-689 | Deep Reinforcement Learning 18

COSC-878 | Doctoral Seminar: Large-Scale Statistical Machine Learning 18
Data Structures and Algorithms 19
ANLY-550 | Structures and Algorithms for Analytics 19

1. The CL Landscape at Georgetown

Reflecting the interdisciplinary nature of the field of Computational Linguistics (CL) and the variety of reasons for students and scholars to come into contact with it, CL education and research at Georgetown is spread across multiple departments and degree programs. The GUCL group exists to publicize and foster interdisciplinary *research* at Georgetown via scientific talks, discussions, and a mailing list. Focusing instead on *courses*, this document consolidates the various opportunities to give a unified perspective and serve as a reference for prospective students, current students, and faculty.

This document is descriptive, not prescriptive; departmental program handbooks and directors of undergraduate/graduate studies are authoritative with respect to policies and degree requirements. Note the date above: we will make every effort to keep this document up to date, but cannot guarantee that all policy changes or new course offerings will be immediately reflected here.

Course Conventions at Georgetown

Georgetown is on the semester system. CL courses are generally not offered during the summer. Thus, for purposes of this document, the academic year consists of a fall semester and a spring semester.

CL courses are taught by a mix of full-time faculty and adjunct faculty. A full course load for graduate students is 9 credits per semester. **Most courses at Georgetown consist of 3 credits, with 2.5 hours of class time per week.** The only exceptions pertinent to this document are (a) independent study credit and (b) Computer Science doctoral seminars, which are only 2 credits and are intended primarily for students who have completed their regular coursework. Descriptions below will therefore assume 3 credits per course unless stated otherwise.

Class sizes are small: 20 students is usually considered robust enrollment for an advanced or graduate course; enrollments of 10 (or less) for specialized courses are not uncommon. Enrollment limits vary depending on the curricular demands of the course. It is not likely that any CL course would allow enrollment to exceed 30.

Courses numbered below 350 are for undergraduates[1]; those numbered between 350 and 499, a.k.a. **over/under** courses, are open to juniors and seniors as well as graduate students; and those numbered above 500 are limited to graduate students.

[1] Undergraduate courses cannot be applied toward a graduate degree except in conjunction with tutorial (independent study) credit by special arrangement with the instructor.

Departments and Programs

It should not be a surprise that the departments most directly relevant to CL are **Linguistics** and **Computer Science** (CS), both within Georgetown College. Most of the CL course offerings are from these departments (individually or cross-listed between the two), and most of the students specializing in CL are pursuing a major, minor, or graduate degree in Linguistics or CS.

Linguistics, CS, and allied departments/programs are summarized below. Linguistics and CS both offer undergraduate majors and minors, 2-year master's degrees, and Ph.D. degrees. There is also the Accelerated Master's option by which undergraduates can receive a combined bachelor's+master's degree in 5 years by double-counting courses toward the 2 degrees.

Linguistics department

Courses in the department belong to one of four *concentrations*: Theoretical Linguistics (TLI), Sociolinguistics (SLI), Applied Linguistics (ALI), and Computational Linguistics (CLI). Graduate students specialize in one of these areas, which define specific curricular requirements, or General Linguistics, if they do not anticipate focusing their studies on any one of them. Henceforth, "CLI students" refers to Linguistics MS and Ph.D. students specializing in the Computational concentration.

The main CLI courses are: **Computational Corpus Linguistics** ("Corpus"), **Analyzing Language Data with R** ("R"), and the following NLP courses: **Intro to NLP** ("INLP"), **Computational Linguistics with Advanced Python** ("PyCL"), **Algorithms for NLP** ("ANLP"), and **Empirical Methods in NLP** ("ENLP"). Corpus, R, INLP, and PyCL are **over/under** courses, i.e. open to juniors and seniors as well as graduate students. ANLP an undergrad-only course. The NLP courses all use Python. Corpus focuses on corpus annotation and corpus linguistics using existing tools; it does not require or teach programming skills beyond regular expressions.

The reason for several NLP courses is that INLP serves students who are new to programming—it teaches Python; ANLP serves undergraduates who are experienced programmers; and ENLP serves graduate students who are experienced programmers; and PyCL bolsters programming skills as a bridge between INLP and ENLP. Many CLI students take INLP (fall), PyCL (spring), and then ENLP (subsequent spring). ENLP overlaps substantially with ANLP, so an undergraduate student cannot take both. In addition to Python, CLI students may benefit from additional languages, especially R (taught in the R course) and Java (not taught in Linguistics, but used in the machine learning course in CS).

A variety of other courses offered in the Linguistics Department cover complementary or more advanced topics such as speech processing, meaning representations/parsing, discourse representations/parsing, and machine translation.

Beginning in Fall 2018, we expect to generally follow the following cycle for courses, though some semesters will vary:

Fall	**Spring**
Annual: Computational Corpus Linguistics (**Corpus**; 367, Zeldes) Intro to NLP (**INLP**; 362, Zeldes)	*Annual:* Empirical Methods in NLP (**ENLP**; 572, cross-listed with CS, Schneider) Computational Linguistics with Advanced Python (**PyCL**; 472, cross-listed with Analytics, *Liz Merkhofer*)
Even years: Dialogue Systems (463, cross-listed with CS, *Matthew Marge*)	*Odd years:* Analyzing Language Data with R (**R**; 469, Zeldes)—skipping Spring 2021 Social Factors in Comp Ling & AI (464, *Shabnam Tafreshi*)
Odd years: Signal Processing (461, *Corey Miller*)	*Even years:* Statistical Machine Translation (462, cross-listed with CS, *Achim Ruopp*) Machine Learning for Linguistics (504, Zeldes) Computational Discourse Models (765, Zeldes)
Roughly every other year, including Spring 2017, Fall 2018, Fall 2020: Advanced Semantic Representation (SemRep, 672, cross-listed with CS, Schneider) *Roughly every 3 years, including Spring 2017:* Parallel Corpora (264, Zeldes) *Roughly every 3–4 years, including Fall 2018, Fall 2022:* Grammar Formalisms for Computational Research (428, Portner) *Occasionally, including Fall 2017:* Algorithms for NLP (**ANLP**, 272, cross-listed with CS, Schneider) *Occasionally, including Spring 2020:* Language and Computers (261, *Emma Manning*)	

Other courses will be offered on an ad hoc basis. Because CLI courses offer practical skills, many non-CLI students enroll in them, sometimes with little or no prior programming experience. CLI students are required to become proficient programmers; all take Corpus as well as INLP and/or ENLP, and many take CS courses. There are also department-wide distribution requirements for courses in Sound (phonology), Form (syntax), and Meaning (semantics/pragmatics). CLI students are required to take at least one seminar (typically numbered above 600). Linguistics graduate course loads are as follows:

- **Ph.D.** students take a full course load (3 courses) throughout their first 3 years.
  - A Ph.D. student with a master's degree from another institution may be eligible for **Advanced Standing**, which reduces their Georgetown course requirements by up to 6 courses (1 year's worth).

- **CLI MS** students pursuing the **thesis** option take 8 courses in 2 years, with the final semester devoted to thesis research: i.e. 3+3+2+0 + thesis research.
- **CLI MS** students pursuing the **MRP** (master's research paper—as opposed to thesis) option take 10 courses in 2 years: 3+3+3+1.
- **Non-computational MS** students take a full course load for 2 years: 3+3+3+3, regardless of whether they choose the thesis or MRP option.

For more information: *https://linguistics.georgetown.edu/graduate/handbook*

Computer Science department

One of the major areas of research in the department is "Information Systems" or "Data-Centric Computing". Faculty in this area teach courses in artificial intelligence (AI), machine learning, natural language processing (NLP), information retrieval (IR), text mining, and data science. Students do not formally specialize in an area.

In coming semesters, we expect to generally follow the following cycle for courses, though some semesters will vary:

Fall	**Spring**
Annual: Information Retrieval (488, Goharian) Text Mining (586, Goharian)	*Annual:* Empirical Methods in NLP (**ENLP**; 572, cross-listed with Ling, Schneider)
Even years: Dialogue Systems (483, cross-listed with Ling, *Matthew Marge*)	
Odd years: Artificial Intelligence (270, Maloof) Automated Reasoning (574, Maloof)	*Even years:* Statistical Machine Translation (482, cross-listed with Ling, *Achim Ruopp*) Intro to Deep Learning with Neural Nets (576, *Joe Garman*)
Roughly every other year, including Spring 2021: Intro Machine Learning (288, Maloof) *Roughly every other year, including Fall 2018:* Machine Learning (575, Maloof) *Roughly every other year, including Spring 2017, Fall 2018, Fall 2020:* Advanced Semantic Representation (SemRep, 672, cross-listed with Ling, Schneider) *Roughly every other year, including Spring 2017:* Health Search/Mining (592, Goharian)	

Roughly every other year, including Spring 2017 and Spring 2018:
Web Search and Sense-Making (589, Yang)
Roughly every other year, including Fall 2017 and Spring 2019:
Intro Data Mining (285, Goharian)
Roughly every 3 years, including Fall 2018:
Doctoral Seminar: Large-Scale Statistical Machine Learning (878, Yang)
Roughly every 3 years, including Spring 2019:
Deep Reinforcement Learning (689, Yang)
Roughly every 3 years, including Spring 2021:
Statistical Machine Learning (578, Yang)
Experimental AI (688, Yang)
Roughly every 3 years, including Spring 2019 and Fall 2021:
Doctoral Seminar: NLP (872, Schneider)
Roughly every 3 years, including Spring 2016 and Spring 2020:
Doctoral Seminar: Text Mining & NLP (882/883, Goharian)
Occasionally, including Fall 2017:
Algorithms for NLP (**ANLP**, 272, cross-listed with Ling, Schneider)
Occasionally, including Fall 2012 and Fall 2020:
Doctoral Seminar: Search Systems (880, Frieder)

Other courses will be offered on an ad hoc basis. CS course loads are as follows:

- **CS MS & Ph.D. students** take 10 regular courses (30 credits) in their first 2 years (typically 3+3+3+1—this "front-loaded" schedule is required for international students).
  - **CS MS students** pursuing the **thesis** option enroll in COSC-999 in place of 2 regular elective courses.
  - A **Ph.D. student** with a master's degree from another institution may be eligible for **Advanced Standing**, which reduces their Georgetown course requirements.
- **CS Ph.D. students** are additionally required to complete
  - 3 doctoral seminars (2 credits each, pass/fail), typically in years 3–5, and
  - 2 workshops in the Apprenticeship in Teaching program, typically in years 4–5 (these are non-credit-bearing).

For more information: https://cs.georgetown.edu/graduate-program-handbook/

Other graduate programs

The Department of Spanish and Portuguese offers master's and Ph.D. degrees in **Spanish Linguistics**. Students in the program typically take several courses in Linguistics. Some core linguistics topics are cross-listed between the Spanish and Portuguese department and Linguistics.

An interdisciplinary doctoral concentration in **Cognitive Science**, housed within the Graduate School, is an alternate degree path available to Ph.D. students in relevant departments

(Linguistics, Spanish Linguistics, Psychology, Philosophy, Neuroscience, Biology; Computer Science TBD). Doctoral students pursuing the CogSci concentration take the usual amount of coursework for students in their home departments but with 3–5 of those courses from other departments, including two 3-credit doctoral CogSci seminars.

The **MS in Analytics** program, housed within the Graduate School, offers students training in "computational, mathematical, and statistical methods to prepare them for careers in data science and analytics." Most of the core Analytics courses are taught by faculty in the CS and Math/Statistics departments. Programming proficiency is required, though many of the students have stronger backgrounds in mathematics/statistics than CS. Analytics students are often interested in taking CL courses as electives. The Analytics courses most relevant to CL are:

- Structures and Algorithms for Analytics (ANLY-550, Thaler)

  Offered every spring by Justin Thaler of the CS Dept. It is not intended for CS students, nor does it discuss natural language, but Linguistics students who have achieved programming proficiency may find it valuable to supplement their CS knowledge.

- Natural Language Processing for Data Analytics (ANLY-580, adjunct)

  This was taught for the first time in Fall 2017 (by Dan Loehr) after it became clear that there was high demand for NLP from Analytics students. It introduces NLP fundamentals with an emphasis on practical uses rather than on linguistic or algorithmic details; thus it is not designed for CS or Linguistics students. At least one Analytics student who enjoyed this course decided to enroll in ENLP in the following semester.

CCT: "**Communication, Culture and Technology** is an interdisciplinary Master of Arts Program focusing on challenges posed by new technologies in a range of fields, including research, government, politics, arts, media, communication, business, health, and medicine." Technology is addressed by disciplines ranging from art, media, and design to culture, society, policy, and business. The program is directed by David Lightfoot (a member of the Linguistics department); Evan Barba, one of the CCT faculty, has an affiliation with the CS Department. Some CCT students have expressed interest in CL courses.

2. Course Offerings

Below are descriptions of the courses mentioned above, organized thematically for convenience. The CL courses are also listed by semester on the GUCL website (http://gucl.georgetown.edu/).

Computational Grammar Formalisms

LING-428 | Grammar Formalisms for Computational Research

Linguists have developed a large number of formally precise syntactic theories, and many of them have been important tools for computational research. In this course, we will study five such systems with the goal of understanding both their perspective on syntax and its relation to parsing, production, and semantics, and will work to gain sufficient skill in using the formal systems to make them useful for computational work. The five systems we will discuss, along with classic early references, are the following:

1. HPSG (Head-driven Phrase-structure Grammar: Pollard and Sag 1994; Sag, Wasow, and Bender 1999)
2. CCG (Combinatory Categorial Grammar: Steedman 2000)
3. LFG (Lexical Functional Grammar: Kaplan and Bresnan 1982, Dalrymple 2001)
4. TAG (Tree Adjoining Grammar: Joshi 1987)
5. Minimalist Grammars (Stabler 2001)

We will spend most of our time on HPSG (with its semantic theory Minimal Recursion Semantics, MRS) and CCG. HPSG is is both widely used in computational research and influential as a framework for studying syntax. CCG is an important modern version of the classical framework of categorial grammar and supports a direct syntax-semantics interface. We will also do brief one-week overviews of LFG and TAG, and will take a look at Minimalist Grammars because they represent a formalization of the Minimalist syntax familiar to many Linguists.

Prerequisite: Syntax 1 or a course which includes basic formal language theory

Computational Corpus Linguistics

LING-264 | Multilingual and Parallel Corpora

Parallel and multilingual corpora are collections of natural language data in several languages, constructed using principled design criteria, which contain either aligned translations of the same texts, or distinct but comparable texts. As such, they are vital resources for comparative linguistics, translation studies, multilingual lexicography and machine translation. This course sets out to explore the theoretical problems raised by translated language, as well as practical issues and empirical patterns found in multilingual data. This includes questions such as exploring similarities and differences between closely related languages, such as different Romance or Slavic languages, or very distant ones, such as English and Japanese. The focus of the course is on the study of actual examples of parallel and comparable corpora using computational methods. Students will have access to search engines indexing aligned translations and will learn some of the basics of building a parallel corpus. The course introduces some fundamental computational methodology, but does not have a programming requirement. However, familiarity with at least one language other than English is required to complete coursework.

LING-367 | Computational Corpus Linguistics

Digital linguistic corpora, i.e. electronic collections of written, spoken or multimodal language data, have become an increasingly important source of empirical information for theoretical and applied linguistics in recent years. This course is meant as a theoretically founded, practical introduction to corpus work with a broad selection of data, including non-standardized varieties such as language on the Internet, learner corpora and historical corpora. We will discuss issues of corpus design, annotation and evaluation using quantitative methods and both manual and automatic annotation tools for different levels of linguistic analysis, from parts-of-speech, through syntax to discourse annotation. Students in this course participate in building the corpus described here: https://corpling.uis.georgetown.edu/gum/

LING-469 | Analyzing language data with R

This course will teach statistical analysis of language data with a focus on corpus materials, using the freely available statistics software 'R'. The course will begin with foundational notions and methods for statistical evaluation, hypothesis testing and visualization of linguistic data which are necessary for both the practice and the understanding of current quantitative research. As we progress we will learn exploratory methods to chart out meaningful structures in language data, such as agglomerative clustering, principal component analysis and multifactorial regression analysis. The course assumes basic mathematical skills and familiarity with linguistic methodology, but does not require a background in statistics or R.

CL Fundamentals

LING-261 | Language and Computers

Science fiction has promised us intelligent robots like C3P0 and HAL, but instead we're stuck with Siri. What happened? Why has getting computers to understand language proven so difficult?

In this course, we'll look at this question through the history of computational linguistics and natural language processing: what approaches have researchers taken over the last 60 years of computational linguistics. In what ways did those approaches succeed? In what ways did they fail?

Topics will include:

- The Goals and Applications of Computational Linguistics
- Pre-statistical Approaches to Natural Language Processing (NLP)
- Modern, Statistical Approaches to NLP

Students will also learn:

- Basic Theoretical Linguistics and Sociolinguistics
- Intuitions of what's possible with NLP and computers
- To address critically over-hyped claims of machine intelligence

The class will focus on the concepts behind these topics rather than implementing them, so no programming experience is required.

No prerequisites, though many concepts will overlap with those in Introduction to Language (LING-001).

Natural Language Processing

COSC/LING-272 | Algorithms for NLP

Human language technologies increasingly help us to communicate with computers and with each other. But every human language is extraordinarily complex, and the diversity seen in languages of the world is massive. Natural language processing (NLP) seeks to formalize and unpack different aspects of a language so computers can approximate human-like language abilities. In this course, we will examine the building blocks that underlie a human language such as English (or Japanese, Arabic, Tamil, or Navajo), and fundamental algorithms for analyzing those building blocks in text data, with an emphasis on the structure and meaning of words and sentences. Students will implement a variety of core algorithms for both rule-based and machine learning methods, and learn how to use computational linguistic datasets such as lexicons and treebanks. Text processing applications such as machine translation, information retrieval, and dialogue systems will be introduced as well.

This course is designed for undergraduates who are comfortable with the basics of discrete probability and possess solid programming skills, including the ability to use basic data structures and familiarity with regular expressions. COSC-160: Data Structures is the prerequisite for CS students, and LING-001 is the prerequisite for Linguistics students. Students that are new to programming or need a refresher are directed to LING-362: Introduction to NLP. The languages of instruction will be English and Python.

LING-362 | Introduction to Natural Language Processing

This course will introduce students to the basics of Natural Language Processing (NLP), a field which combines insights from linguistics and computer science to produce applications such as machine translation, information retrieval, and spell checking. We will cover a range of topics that will help students understand how current NLP technology works and will provide students with a platform for future study and research. We will learn to implement simple representations such as finite-state techniques, n-gram models and basic parsing in the Python programming language. Previous knowledge of Python is not required, but students should be prepared to invest the necessary time and effort to become proficient over the course of the semester. Students who take this course will gain a thorough understanding of the fundamental methods used in natural language understanding, along with an ability to assess the strengths and weaknesses of natural language technologies based on these methods.

LING-472/ANLY-521 | Computational Linguistics with Advanced Python

This course teaches advanced topics in programming for linguistic data analysis and processing using the Python language. A series of assignments will give students hands-on practice implementing core algorithms for linguistic tasks. By the end of the course, students will be able to transform pseudocode into well-written code for algorithms that make sense of textual data, and to evaluate the algorithms quantitatively and qualitatively. Linguistic tasks will include edit distance, semantic similarity, authorship detection, and named entity recognition. Python topics will include the appropriate use of data structures; mathematical objects in numpy; exception handling; object-oriented programming; and software development practices such as code documentation and version control.

Prerequisites: Basic Python programming skills are required (for example satisfied by LING-362, Intro to NLP)

COSC/LING-572 | [Empirical Methods in Natural Language Processing]

Systems of communication that come naturally to humans are thoroughly unnatural for computers. For truly robust information technologies, we need to teach computers to unpack our language. Natural language processing (NLP) technologies facilitate semi-intelligent artificial processing of human language text. In particular, techniques for analyzing the grammar and meaning of words and sentences can be used as components within applications such as web search, question answering, and machine translation.

This course introduces fundamental NLP concepts and algorithms, emphasizing the marriage of linguistic corpus resources with statistical and machine learning methods. As such, the course combines elements of linguistics, computer science, and data science. Coursework will consist of lectures, programming assignments (in Python), and a final team project. The course is intended for students who are already comfortable with programming and have some familiarity with probability theory.

Prerequisite: A data structures course, INLP (PyCL recommended), or equivalent experience

ANLY-580 | NLP for Data Analytics

This course will cover the major techniques for mining and analyzing textual data to extract interesting patterns, discover knowledge, and support decision-making. In this course, the students will learn the main concepts and algorithms in Natural Language Processing and their applications in data science. These include search and information retrieval, document clustering and classification, topic modeling, sentiment analysis, and deriving meaning from unstructured narratives. In addition to traditional techniques in machine learning such as regression, decision trees, and Naive Bayes algorithms, the course will also examine the latest approaches in Deep Learning. The students will be given the opportunity to develop hands-on experience in building foundational tools and machine learning algorithms that can be applied to real analytics problems. The data obtained from textual content can be used to augment numerical data for the purposes of building predictive models, identifying emerging issues, detecting opinion, and determining important relationships.

COSC/LING-672 | [Advanced Semantic Representation]

Natural language is an imperfect vehicle for meaning. On the one hand, some expressions can be interpreted in multiple ways; on the other hand, there are often many superficially divergent ways to express very similar meanings. Semantic representations attempt to disentangle these two effects by exposing similarities and differences in how a word or sentence is interpreted. Such representations, and algorithms for working with them, constitute a major research area in natural language processing.

This course will examine semantic representations for natural language from a computational/NLP perspective. Through readings, presentations, discussions, and hands-on exercises, we will put a semantic representation under the microscope to assess its strengths and weaknesses. For each representation we will confront questions such as: What aspects of meaning are and are not captured? How well does the representation scale to the large vocabulary of a language? What assumptions does it make about grammar? How language-specific is it? In what ways does it facilitate manual annotation and automatic analysis? What datasets and algorithms have been developed for the representation? What has it been used for? Representations covered in depth will include FrameNet (http://framenet.icsi.berkeley.edu), Universal Cognitive Conceptual Annotation (http://www.cs.huji.ac.il/~oabend/ucca.html), and Abstract Meaning Representation (http://amr.isi.edu/).

Term projects will consist of (i) innovating on the representation's design, datasets, or analysis algorithms, or (ii) applying it to questions in linguistics or downstream NLP tasks.

Prerequisites: Corpus Linguistics or Empirical Methods in NLP

LING-765 | Computational Discourse Models

Recent years have seen an explosion of computational work on higher level discourse representations, such as entity recognition, mention and coreference resolution and shallow discourse parsing. At the same time, the theoretical status of the underlying categories is not well understood, and despite progress, these tasks remain very much unsolved in practice. This graduate level seminar will concentrate on theoretical and practical models representing how referring expressions, such as mentions of people, things and events, are coded during language processing. We will begin by exploring the literature on human discourse processing in terms of information structure, discourse coherence and theories about anaphora, such as Centering Theory and Alternative Semantics. We will then look at computational linguistics implementations of systems for entity recognition and coreference resolution and explore their relationship with linguistic theory. Over the course of the semester, participants will implement their own coding project exploring some phenomenon within the domain of entity recognition, coreference, discourse modeling or a related area.

COSC-872 | Doctoral Seminar: NLP

This course will expose students to current research in natural language processing and computational linguistics. Class meetings will consist primarily of student-led reading discussions, supplemented occasionally by lectures or hands-on activities. The subtopics and reading list will be determined at the start of the semester; readings will consist of research papers, advanced tutorials, and/or dissertations.

Prerequisites: Familiarity with NLP using machine learning methods (for example satisfied by COSC-572, Empirical Methods in NLP)

Information Retrieval, Information Extraction, and Data Mining

COSC-285 | Introduction to Data Mining

This course covers concepts and techniques in the field of data mining. This includes both supervised and unsupervised algorithms, such as naive Bayes, neural network, decision tree, rule based classifiers, distance based learners, clustering, and association rule mining. Various issues in the pre-processing of the data are addressed. Text classification, social media mining, and recommender systems will be addressed. The students learn the material by building various data mining models and using various data pre-processing techniques, performing experimentation and provide analysis of the results.

COSC-488 | Information Retrieval

Information retrieval is the identification of textual components, be them web pages, blogs, microblogs, documents, medical transcriptions, mobile data, or other big data elements, relevant to the needs of the user. Relevancy is determined either as a global absolute or within a given context or view point. Practical, but yet theoretically grounded, foundational and advanced algorithms needed to identify such relevant components are taught.

The Information-retrieval techniques and theory, covering both effectiveness and run-time performance of information-retrieval systems are covered. The focus is on algorithms and heuristics used to find textual components relevant to the user request and to find them fast. The course covers the architecture and components of the search engines such as parser, index builder, and query processor. In doing this, various retrieval models, relevance ranking, evaluation methodologies, and efficiency considerations will be covered. The students learn the material by building a prototype of such a search engine. These approaches are in daily use by all search and social media companies.

COSC-586 | Text Mining & Analysis

This course covers various aspects and research areas in text mining and analysis. Text may be a document, query, blog, tag description, etc. The structure of the course is a combination of lectures & students' presentations. The lectures will cover Text/Web/query classification, information extraction, word sense disambiguation, opinion mining & sentiment analysis, query log analysis, ontology extraction and integration, and more. The students are assigned a related topic in the field for further study and presentation in the class.

COSC-589 | Web Search and Sense-making

The Web provides abundant information which allows us to live more conveniently and make quicker decisions. At the same time, the growth of the Web and the improvements in data creation, collection, and use have lead to tremendous increase in the amount and complexity of the data that a search engine needs to handle. The increase of the magnitude and complexity of the data has become a major drive for new data analytics algorithms and technologies that are scalable, highly interactive, and able to handle complex and dynamic information seeking tasks in the big data era. How to effectively and efficiently search for the documents relevant to our information needs and how to extract the valuable information and make sense out from “big data” are the subjects of this course.

The course will cover Web search theory and techniques, including basic probabilistic theory, representations of documents and information needs, various retrieval models, link analysis, classification and recommender systems. The course will also cover programming models that allow us to easily distribute computations across large computer clusters. In particular, we will teach Apache Spark, which is an open-source cluster computing framework that has soon become the state-of-the-art for big data programming. The course is featured in step-by-step weekly/bi-weekly small assignments which composes a large big data project, such as building Google’s PageRank on the entire Wikipedia. Students will be provided knowledge to Spark, Scala, Web search engines, and Web recommender systems with a focus on search engine design and "thinking at scale”.

COSC-592 | Health Search and Mining

This course will be a combination of lectures and students' presentations. After providing a review of information retrieval and data mining, the lectures will cover health text processing on scientific literature, clinical notes, and social media, among others. The Students will present and discuss research literature. This includes: review of current literature on specific topic, and experimental results and evaluation of a proposed approach. Students are expected to have the knowledge of data structures.

COSC-880 | Doctoral Seminar: Search Systems

This doctoral seminar surveys the recent literature in search systems.

COSC-882 | Doctoral Seminar: Text Mining and NLP

This doctoral seminar consists of readings/discussions/potential research efforts on topics related to text mining and utilization of NLP in text mining.

Spoken and Interactive Language Processing

LING-461 | Signal Processing

This course will survey speech processing technology from a computational linguistic perspective. Speech processing technology is a component of human language technology that focuses on the processing of audio data. The audio data can be either the input or output of speech processing. When speech serves as the output, the technology is known as speech synthesis or text-to-speech (TTS). Additional technologies to be examined include spoken language identification (SLID), speaker verification and identification and speech diarization, which is the parsing of audio data into individual speaker segments.

Particular attention will be paid to the linguistic components of speech technology. Phonetics and phonology play an important role in both TTS and STT. In addition, morphology, syntax and pragmatics are important both in authentic modeling of TTS and in constraining possible STT output. Semantics plays a role in the interpretation of STT output, which can feed into text-based natural language processing (NLP).

The algorithms underlying contemporary speech technology approaches will be discussed. Despite the focus on the linguistic aspects of the technology, it is important for students to have sufficient understanding of the algorithms used in order to grasp both where linguistics fits in and the possible constraints on its incorporation into larger systems.

The course will examine freely available TTS and STT packages so that students can build their own engines and experiment with the construction of the components. For assignments and projects, students will be encouraged to pick a language or dialect of their choice in order to build a synthesizer or recognizer for that variety. It would be most interesting to focus on languages or varieties that do not generally receive attention in commercial applications, such as African American or accented varieties of English.

Students from a variety of backgrounds are encouraged to take this course. Helpful background includes: natural language processing, phonetics, phonology and sociolinguistics. While not required, helpful technical background includes familiarity with speech analysis software such as PRAAT, Linux, shell scripting and coding/scripting in languages like Python, Java, C++, etc.

LING-463/COSC-483 | Dialogue Systems

Nearly all of us interact with dialogue systems -- from calling up banks and hotels, to talking with intelligent assistants like Siri, Alexa, or Cortana, dialogue systems enable people to get tasks done with software agents using language. Since the interaction is bi-directional, we must consider the fundamentals of how people engage in conversation so as to manage users' expectations and track how information is exchanged in dialogue. Dialogue systems require an array of technologies to come together for them to work well, including speech recognition, natural language understanding, dialogue management, natural language generation, and speech synthesis. This course will explore what makes dialogue systems effective in commercial and research applications (ranging from personal assistants and chatbots to embodied conversational agents and language-directed robots) and how this contrasts with everyday human-human dialogue.

This course will introduce students to the fundamentals of dialogue systems, expanding on technologies and algorithms that are used in today's dialogue systems and chatbots. There will also be emphasis on the psycholinguistic properties of human conversation (turn-taking, grounding) so as to prepare students for designing effective, user-friendly dialogue systems. The course will also include examining datasets and dialogue annotations used to train dialogue systems with machine learning algorithms. Coursework will consist of lectures, writing and programming assignments, and student-led presentations on special topics in dialogue. A final project will give students a chance to build their own dialogue system using open source and freely available software. This course is intended for students that are already comfortable with limited amounts of programming (in Python).

Other CL Applications

LING-462/COSC-482 | Statistical Machine Translation

After more than 60 years since Machine Translation (MT) research started at Georgetown, this area of Natural Language Processing (NLP) research is more active than ever. In this course we explore the data-driven approaches to translate human language with computers that supplanted rule-based approaches in the past quarter century. First, we lay foundations for the course with statistical NLP relevant to MT and corpus preparation. Next, we start exploring statistical MT (SMT) – from word-based models to phrase-based models to tree-based models. We will then cover domain-adaptation, incremental learning and how to integrate linguistic information. We will learn how to evaluate system output with automatic and human evaluation methods.

Recently, deep learning-based approaches have proven to produce superior translation quality compared to SMT. We will investigate the current state-of-the-art in neural MT (NMT) and contrast its strength and weaknesses with SMT.

Machine translation does not exist in a vacuum; it is now used to provide draft translations for human translators and is embedded in other NLP systems. With better quality, raw MT is increasingly used in in written and spoken human communication. We study the adaptation of MT for the most common applications.

Requirements: Basic Python programming skills are required (for example satisfied by LING-362, Intro to NLP)

LING-464 | Social Factors in Comp Ling and AI

Advances in technologies for processing human languages have increasingly brought computational linguistics into contact with people. As such, what language reveals about people—and how AI algorithms make decisions affecting people based on their language—is of paramount concern. At the same time, contemporary algorithms for processing language offer powerful new tools for studying people and society on a large scale. Designed for students with grounding in computational linguistics, this course will examine the intersection of people, language, and algorithms with technical precision as well as an appreciation for human context. Topics will include: computational models of conversational interaction and power dynamics; emotions, sentiment, subjectivity, and politeness; toxic language; sociolinguistic variation; detection of attributes such as race and gender; issues of privacy, ethics, bias, and fairness, with special attention to minoritized speakers, languages, and dialects; and the use of large-scale language data for studying political framing and social movements like #MeToo and Black Lives Matter.

Requirements: Basic Python programming skills are required (for example satisfied by LING-362, Intro to NLP)

Machine Learning and Artificial Intelligence

COSC-270 | Artificial Intelligence

Artificial Intelligence (AI) is the branch of computer science that studies how to program computers to reason, learn, see, and understand. The lecture portion of this class surveys basic and advanced concepts and techniques of artificial intelligence, including search, knowledge representation, automated reasoning, uncertain reasoning, and machine learning. Additional topics include the Lisp programming language, theorem proving, game playing, rule-based systems, and philosophical issues. Applications of artificial intelligence will also be discussed and will include domains such as medicine, computer security, and face detection. Students must complete midterm and final exams, and five projects using the Lisp programming language.

Prerequisite: Data Structures (COSC-160)

COSC-288 | Introduction to Machine Learning

This undergraduate course surveys the major research areas of machine learning. Through traditional lectures and programming projects, students learn (1) to understand the foundations of machine learning, (2) to implement methods of machine learning in a high-level programming language, (2) to comprehend papers from the primary literature, and (4) to design and conduct their own studies. The course compares and contrasts machine learning with related endeavors, such as statistical learning, pattern classification, data mining, and information retrieval. Topics include instance-based approaches, naive Bayes, decision trees, rule induction, linear classifiers, neural networks, support vector machines, ensemble methods, evaluation, and applications.

Prerequisite: Data Structures (COSC-160)

LING-504 | Machine Learning for Linguistics

In the past few years, the advent of abundant computing power and data has catapulted machine learning to the forefront of a number of fields of research, including Linguistics and especially Natural Language Processing. At the same time, general machine learning toolkits and tutorials make handling ‘default cases’ relatively easy, but are much less useful in handling non-standard data, less studied languages, low-resource scenarios and the need for interpretability that is essential for drawing robust inferences from data. This course gives a broad overview of the machine learning techniques most used for text processing and linguistic research. The course is taught in Python, covering both general statistical ML algorithms, such as linear models, SVMs, decision trees and ensembles, and current deep learning models, such as deep neural net classifiers, recurrent networks and contextualized continuous meaning representations. The course assumes good command of Python (ability to implement a program from pseudo-code) but does not require previous experience with machine learning.

Requirements: Intermediate Python (courses such as LING-472: Computational Linguistics with Advanced Python provide a good preparation)

COSC-574 | Automated Reasoning

This graduate lecture surveys methods of automated deductive reasoning. Through traditional lectures, programming projects, paper presentations, and research projects, students learn (1) to understand the foundations of logical and probabilistic methods of automated reasoning. (2) to implement algorithms for logical and probabilistic reasoning, (3) to comprehend, analyze, and critique papers from the primary literature, (4) to replicate studies described in the primary literature, and (5) to design, conduct, and present their own studies. Topics include propositional logic, predicate logic, resolution proof, production systems, Prolog, uncertain reasoning, certainty factors, Bayesian decision theory, Bayesian networks, exact inference, approximate inference, first-order probabilistic models, probabilistic programming languages, and applications.

Prerequisites: Students should have taken undergraduate courses in computer science through data structures; at the very least, students must be able to implement trees and graphs in a high-level object-oriented programming language. Students should have also taken undergraduate courses in mathematics, such as probability, statistics, and perhaps propositional and first-order logic.

COSC-575 | Machine Learning

This course surveys the major research areas of machine learning, concentrating on inductive learning. The course will also compare and contrast machine learning with related endeavors, such as statistical learning, pattern classification, data mining, and information retrieval. Topics will include rule induction, decision trees, Bayesian methods, density estimation, linear classifiers, neural networks, instance-based approaches, genetic algorithms, evaluation, and applications. In addition to programming projects and homework, students will complete a semester project.

Prerequisites: Students should have taken undergraduate courses in computer science through data structures; at the very least, students must be able to implement trees and graphs in a high-level object-oriented programming language. Students should have also taken undergraduate courses in mathematics, such as calculus, linear algebra, and probability and statistics.

COSC-576 | Introduction to Deep Learning with Neural Nets

Recent advances in hardware have made deep learning with neural networks practical for real-world problems. Neural networks are a powerful tool that have shown benefit in a wide range of fields. Deep learning involves creating artificial neural networks with greater layer depth or deep neural nets (DNN) for short. These DNNs can find patterns in complex data, and are useful in a wide variety of situations. In numerous fields, state-of-the-art solutions have been accomplished with DNNs and DNN systems dominate head-to-head competitions. This course will introduce the student to neural networks, explain different neural network architectures, and then demonstrate the use of these neural networks on a wide array of tasks.

COSC-578 | Statistical Machine Learning

Statistical machine learning brings together statistics and computational sciences such as computer science, system science, and optimization. The recent developments in bioinformatics, signal processing, information management, finance, and artificial intelligence have been largely influenced by statistical machine learning. With a focus on mathematical and algorithmic theories, this class offers basics in statistical methodology in dealing with applied problems in science and technology. Topics covered in the class include probability, mathematical statistics, inference, sampling, optimization, and their applications in machine learning. The class will have lectures, mathematical homework, exams, and a programming-based project.

COSC-688 | Experimental AI

This course offers opportunities for students to have an in-depth understanding and hands-on experience with practical AI systems for state-of-the-art evaluation campaigns. It includes seminar-style classroom presentations and a significant project component. Students will be guided to go through the design and implementation of AI systems in different domains. The course will review recent AI and Machine Learning publications and lead students to work in small groups to build systems. Students are expected to have strong programming skills and previous experience in machine learning, deep learning, and/or AI.

COSC-689 | Deep Reinforcement Learning

Deep Reinforcement learning is an area of machine learning that learns how to make optimal decisions from interacting with an environment. From the environment, an agent observes the consequence of its action and alters its behavior to maximize the rewards received in the long term. Reinforcement learning has developed strong mathematical foundations and impressive applications in diverse disciplines such as psychology, control theory, artificial intelligence, and neuroscience. An example is the winning of AlphaGo, developed using Monte Carlo tree search and deep neural networks, over world-class human Go players. The overall problem of learning from interaction to achieve goals is still far from being solved, but our understanding of it has improved significantly. In this course, we study fundamentals, algorithms, and applications in deep reinforcement learning. Topics include Markov Decision Processes, Multi-armed Bandits, Monte Carlo Methods, Temporal Difference Learning, Function Approximation, Deep Neural Networks, Actor-Critic, Deep Q-Learning, Policy Gradient Methods, and connections to Psychology and to Neuroscience. The course has lectures, mathematical and programming assignments, and exams.

COSC-878 | Doctoral Seminar: Large-Scale Statistical Machine Learning

This doctoral seminar studies topics in statistical machine learning in the age of big data and artificial intelligence. In the seminar, we will read both classical and recent work in supervised learning, nonparametric models, optimization, and deep reinforcement learning. In the class, we will read textbooks and survey milestone papers. Students are expected to submit questions for the readings before each class and give presentations when it is their turn. To have first-hand experience, students are also expected to do a few programming exercises in the textbooks.

Data Structures and Algorithms

ANLY-550 | Structures and Algorithms for Analytics

This course covers algorithmic techniques for solving different types of data science problems. It will cover Big O notation, data structures (arrays, stacks, queues, lists, trees, heaps, graphs), sorting and searching (binary search trees, hash tables), and algorithmic paradigms for efficient problem solving (divide and conquer, recursion, greedy algorithms, dynamic programming, etc.). It will include both theory and practice. You will learn to design, analyze, and implement fundamental data structures and algorithms. This course will provide the algorithmic background essential for further study of computer science topics.

In more detail, from this course you will learn the basic language and tools for algorithm analysis, as well as several specific problems and general paradigms for algorithm design. We will focus on the theoretical and mathematical aspects in class and on the homework assignments. But because one gains a deeper understanding of algorithms from actually implementing them, the course will include a substantial programming component. Large programming assignments can be done (but do not have to be done!) in pairs. More details will be available when the first programming assignment is given. We will be covering a great deal of material in this class. I expect the course to be challenging, both in terms of the workload and the difficulty of the material. You should be prepared to do a lot of work outside of class. The payoff will be that you will learn a lot of both useful and interesting things.

The formal prerequisites for this course are ANLY-501 and ANLY-511. The expected skills are as follows. Students should be able to program in a standard programming language (C, C++, Java, Python, etc.). Some mathematical maturity also will be expected; students should have some idea of what constitutes a mathematical proof and how to write one. Some knowledge of basic probability will also be helpful.