# Natural Language Processing for Computer Scientists and Data Scientists at a Large State University

**Casey Kennington**
Department of Computer Science
Boise State University
caseykennington@boisestate.edu

## Abstract

The field of Natural Language Processing (NLP) changes rapidly, requiring course offerings to adjust with those changes, and NLP is not just for computer scientists; it's a field that should be accessible to anyone who has a sufficient background. In this paper, I explain how students with Computer Science and Data Science backgrounds can be well-prepared for an upper-division NLP course at a large state university. The course covers probability and information theory, elementary linguistics, machine and deep learning, with an attempt to balance theoretical ideas and concepts with practical applications. I explain the course objectives, topics and assignments, reflect on adjustments to the course over the last four years, as well as feedback from students.

## 1 Introduction

Thanks in part to a access to large datasets, increases in compute power, and easy-to-use programming programming libraries that leverage neural architectures, the field of Natural Language Processing (NLP) has become more popular and has seen more widespread adoption in research and in commercial products. On the research side, the Association for Computational Linguistics (ACL) conference—the flagship NLP conference—and related annual conferences have seen dramatic increases in paper submissions. For example, in 2020 ACL had 3,429 paper submissions, whereas 2019 had 2,905, and this upward trend has been happening for several years. Certainly, lowering barriers to access of NLP methods and tools for researchers and practitioners is a welcome direction for the field, enabling researchers and practitioners from many disciplines to make use of NLP.

It is therefore becoming more important to better equip students with an understanding of NLP to prepare them for careers either directly related to NLP, or which leverage NLP skills. In this paper, I reflect on my experience setting up and maintaining a class in NLP at Boise State University, a large state university, how to prepare students for research and industry careers, and how the class has changed over four years to fit the needs of students.

The next section explains the course objectives. I then explain challenges that are likely common to many university student populations, how NLP is designed for students with Data Science and Computer Science backgrounds, then I explain course content including lecture topics and assignments that are designed to fulfill the course objectives. I then offer a reflection on the three times I taught this course over the past four years, and future plans for the course.

## 2 Course Objectives

Most students who take an NLP course will not pursue a career in NLP proper; rather, they take the course to learn skills that will help them find employment or do research in areas that make use of language, largely focusing on the medium of text (e.g., anthropology, information retrieval, artificial intelligence, data mining, social media network analysis, provided they have sufficient data science training). My goal for the students is that they can identify aspects of natural language (phonetics, syntax, semantics, etc.) and how each can be processed by a computer, explain the difference between classification models and approaches, be able to map from (basic) formalisms to functional code, and use existing tools, libraries, and data sets for learning while attempting to strike a balance between theory and practice. In my view, there are several aspects of NLP that anyone needs to grasp, and how to apply NLP techniques in novel circumstances. Those aspects are illustrated in Figure 1.

No single NLP course can possibly account for a level of depth in all of the aspects in Figure 1, but a student who has taken courses or has experience in at least two of the areas (e.g., they have taken a

*Proceedings of the Fifth Workshop on Teaching NLP*, pages 115–124
June 10–11, 2021. ©2021 Association for Computational Linguistics

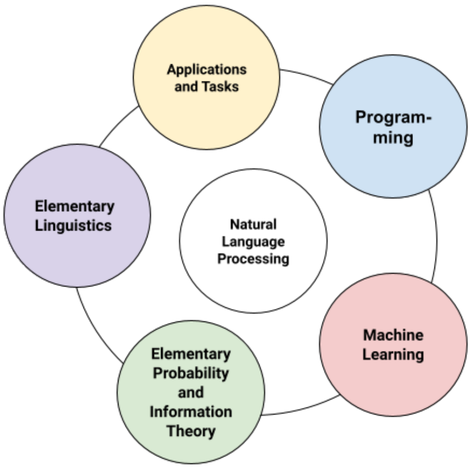
Applications and Tasks
Program-ming
Elementary Linguistics
Natural Language Processing
Machine Learning
Elementary Probability and Information Theory

Figure 1: Areas of content that are important for a course in Natural Language Processing.

statistics course and have experience with Python, or they have taken linguistics courses and have used some data science or machine learning libraries) will find success in the course more easily than those with no experience in any aspect.

This introduces a challenge that has been explored in prior work on teaching NLP (Fosler-Lussier, 2008): the diversity of the student population. NLP is a discipline that is not just for computer science students, but it is challenging to prepare students for the technical skills required in a NLP course. Moreover, similar to the student population in Fosler-Lussier (2008), there should be course offerings for both graduate and undergraduate students. In my case, which is fairly common in academia, as the sole NLP researcher at the university I can only offer one course once every four semesters for both graduate and undergraduate students, but also students with varied backgrounds–not only computer science. As a result, this is not a research methods course; rather, it is more geared towards learning the important concepts and technical skills surrounding recent advances in NLP. Others have attempted to gear the course content and delivery towards research (Freedman, 2008) giving the students the opportunity to have open-ended assignments. I may consider this for future offerings, but for now the final project acts as an open-ended assignment, though I don't require students to read and understand recent research papers.

In the following section, I explain how we prepare students of diverse backgrounds to succeed in an NLP course for upper-division undergraduate and graduate students.

## 3 Preparing Students with Diverse Academic and Technical Backgrounds

Boise State University is the largest university in Idaho, situated in the capital of the State of Idaho. The university has a high number of non-traditional students (e.g., students outside the traditional student age range, or second-degree seeking students). Moreover, the university has a high acceptance rate (over 80%) for incoming first-year students. As is the case with many universities and organizations, a greater need for "computational thinking" among students of many disciplines has been an important driver of recent changes in course offerings across many departments. Moreover, certain departments have responded to the need and student interest in machine learning course offerings. In this section, we discuss how we altered the Data Science and Computer Science curricula to meet these needs and the implications these changes have had on the NLP course.[1]

**Data Science** The Data Science offerings begin with a foundational course (based on Berkeley's data8 content) that has only a very basic math prerequisite.[2] It introduces and allows students to practice Python, Jupyter notebooks, data analysis and visualization, and basic statistics (including the bootstrap method of statistical significance). Several courses follow this course that are more domain specific, giving the students options for gaining practical experience in Data Science skills relative to their abilities and career goals. One path more geared towards students of STEM-related majors (though not targeting Computer Science majors) as well as some majors in the Humanities, is a certificate program that includes the foundational course, a follow-on course that gives students experience with more data analysis as well as probability and information theory, an introductory machine learning course, and a course on databases. The courses largely use Python as the programming language of choice.

**Computer Science** In parallel to the changes in Data Science-related courses, the Department of Computer Science has seen increased enrollment and increased request for machine learning-related courses. The department offers several

[1] We do not have a Data Science department or major degree program; the Data Science courses are housed in different departments across campus.

[2] http://data8.org/

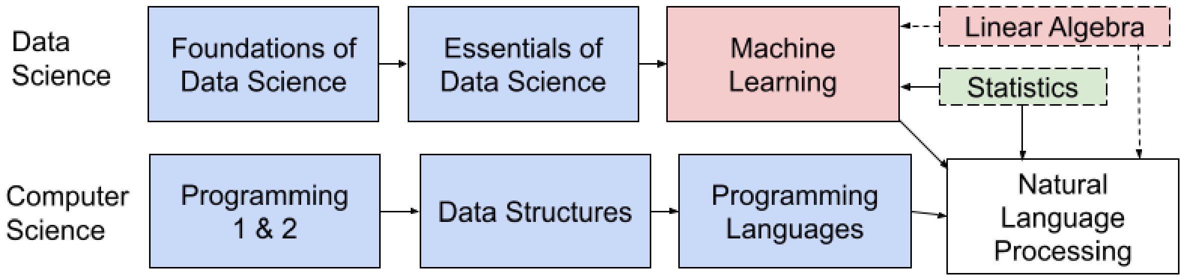

Figure 2: Two course paths that prepare students to succeed in the NLP course: Data Science and Computer Science. Solid arrows denote pre-requisites, dashed arrows denote co-requisites. Solid outlines denote Data Science and Computer Science courses, dashed outlines denote Math courses.

courses, though they focus on upper-division students (e.g., artificial intelligence, applied deep learning, information retrieval and recommender systems). This is a challenge because the main Computer Science curriculum focuses on procedural languages such as Java with little or no exposure to Python (similar to the student population reported in Freedman (2008)), forcing the upper-division machine learning-related courses to spend time teaching Python to the students. This cannot be underestimated–programming languages may not be natural languages, but they are *languages* in that it takes time to learn them, particularly relevant Python libraries. To help meet this challenge, the department has recently introduced a Machine Learning Emphasis that makes use of the Data Science courses mentioned above, along with additional requirements for math and upper-division Computer Science courses.

**Preqrequisites** Unlike Agarwal (2013) which attempted and introductory course for all levels of students, this course is designed for upper-division students or graduate students. Students of either discipline, Computer Science or Data Science, are prepared for this NLP course, albeit in different ways. Computer Scientists can think computationally, have experience with a syntactic formalism, and have experience with statistics. Data Science students likewise have experience with statistics as well as Python, including data analysis skills. Though the backgrounds can be quite diverse, my NLP course allows two prerequisite paths: all students must take a statistics course, but Computer Science students must take a Programming Languages course (which covers context free grammars for parsing computer languages and now covers some Python programming), and the Data Science students must have taken the introductory machine learning course. Figure 2 depicts the two course paths visually.

## 4 NLP Course Content

In this section, I discuss course content including topics and assignments that are designed to meet the course objectives listed above. Woven into the topics and assignments are the themes of *ambiguity* and *limitations*, explained below.

### 4.1 Topics & Assignments

**Theme of Ambiguity** Figure 3 shows the topics (solid outlines) that roughly translate to a single lecture, though some topics require multiple lectures. One main theme that is repeated throughout the course, but is not a specific lecture topic is *ambiguity*. This helps the students understand differences between natural human languages and programming languages. The *Introduction to Linguistics* topic, for example, gives a (very) high-level overviews of phonetics, morphology, syntax, semantics, and pragmatics, with examples of ambiguity for each area of linguistics (e.g., phonetic ambiguity is illustrated by hearing someone say *it's hard to recognize speech* but it could be heard as *it's hard to wreck a nice beach*, and syntactic ambiguity is illustrated by the sentence *I saw the person with the glasses* having more than one syntactic parse).

**Probability and Information Theory** This course does not focus only on deep learning, though many university NLP offerings seem to be moving to deep-learning only courses. There are several reasons not to focus on deep learning for a university like Boise State. First, students will not have a depth of background in probability and information theory, nor will they have a deep understanding of optimization (both convex and non-convex) or error functions in neural networks (e.g., cross entropy). I take time early in the course to explain discrete and continuous probability, and information theory. Discrete probability theory is straight forward as it requires counting, something that is intuitive when working with language data represented as text strings. Continuous probability theory, I have found, is more difficult for students to grasp as it relates to machine learning or NLP, but building on students' understanding of discrete probability theory seems to work pedagogically. For example, if we use continuous data and try somehow to count values in that data, it's not clear what should be counted (e.g., using binning), highlighting the importance of continuous probability functions that fit around the data, and the importance of estimating parameters for those continuous functions–an important concept for understanding classifiers later in the course. To illustrate both discrete and continuous probability, I show students how to program a discrete Naive Bayes classifier (using ham/spam email classification as a task) and a continuous Gaussian Native Bayes classifier (using the well-known iris data set) from scratch. Both classifiers have similarities, but the differences illustrate how continuous classifiers learn parameters.

**Sequential Thinking** Students experience probability and information theory in a targeted and highly-scaffolded assignment. They then extend their knowledge and program, from scratch, a part-of-speech tagger using counting to estimate probabilities modeled as Hidden Markov Models. These models seem old-fashioned, but it helps students gain experience beyond the standard machine learning workflow of mapping many-to-one (i.e., features to a distribution over classes) because this is a many-to-many sequential task (i.e., many words to many parts-of-speech), an important concept to understand when working with sequential data like language. It also helps students understand that NLP often goes beyond just fitting “models" because it requires things like building a trellis and decoding a trellis (undergraduate students are required to program a greedy decoder; graduates are required to program a Viterbi decoder). This is a challenging assignment for most students, irrespective of their technical background, but grasping the concepts of this assignment helps them grasp more difficult concepts that follow.

**Syntax** The Syntax & Parsing assignment also deserves mention. The students use any parser in NLTK to parse a context free grammar of a fictional language with a limited vocabulary.[3] This helps the students think about structure of language, and while there are other important ways to think about syntax such as dependencies (which we discuss in the course), another reason for this assignment is to have the students write grammars for a small vocabulary of words in a language they don't know, but also to create a non-lexicalized version of the grammar based on parts of speech, which helps them understand coverage and syntactic ambiguity more concretely.[4] There is no machine learning or estimating a probabilistic grammar here, just parsing.

**Semantics** An important aspect of my NLP class is semantics. I introduce them briefly to formal semantics (e.g., first-order logic), WordNet (Miller, 1995), distributional semantics, and grounded semantics. We discuss the merits of representing language “meaning" as embeddings and the limitations of meaning representations trained only on text and how they might be missing important semantic knowledge (Bender and Koller, 2020). The Topic Modeling assignment uses word-level embeddings (e.g., word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)) to represent texts and gives them an opportunity to begin using a deep learning library (*tensorflow* & *keras* or *pytorch*).

We then consider how a semantic representation that has knowledge of modalities beyond text, i.e., images is part of human knowledge (e.g., what is the meaning of the word *red*?), and how recent work is moving in this direction. Two assignments give the students a deeper understanding of these ideas. The transfer learning assignment requires the students to use convolutional neural networks pre-trained on image data to represent objects in images and train a classifiers to identify simple object types, tying images to words. This is extended in the Grounded Semantics assignment where a binary classifier (based on the words-as-classifiers model introduced in Kennington and Schlangen (2015), then extended to work with images with "real" objects in Schlangen et al. (2016)) is trained

[3] I opt for Tamarian: https://en.wikipedia.org/wiki/Darmok

[4] I have received feedback from multiple students that this was their favorite assignment.

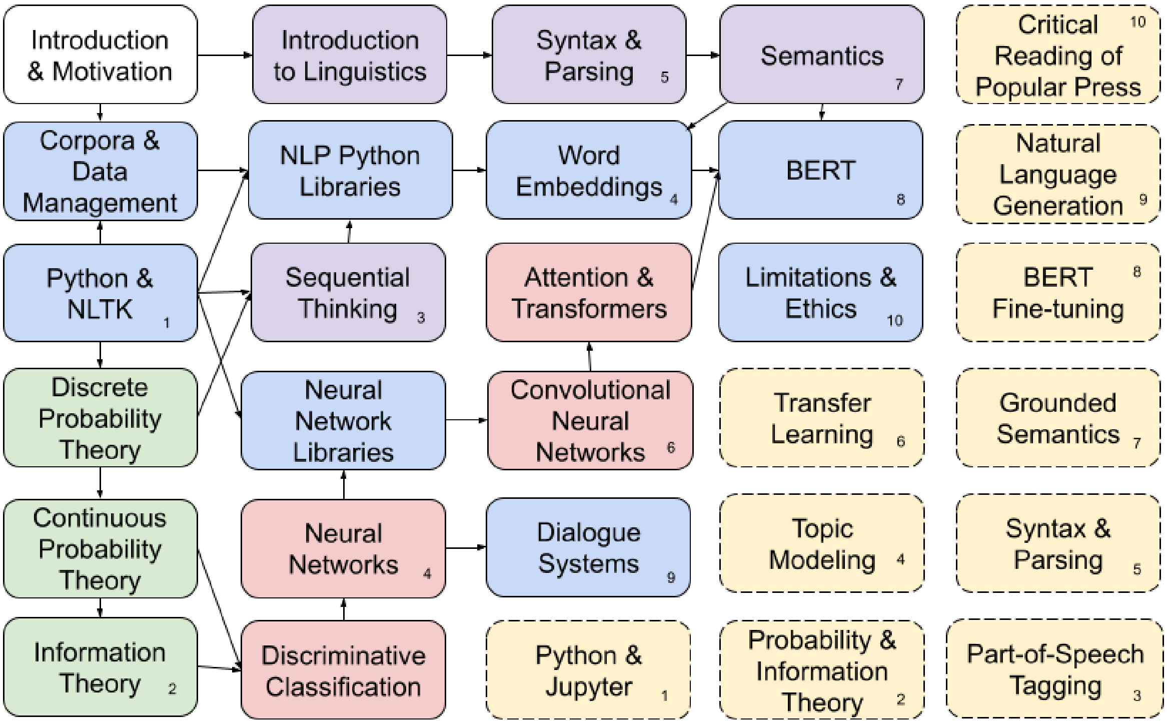

Figure 3: Course Topics and Assignments. Colors cluster topics (green=probability and information theory, blue=NLP libraries, red=neural networks, purple=linguistics, yellow=assignments). Courses have solid lines; topic dependencies are illustrated using arrows. Assignments have dashed lines. Assignments are indexed with the topics on which they depend.

for all words in referring expressions to objects in images in the MSCOCO dataset annotated with referring expressions to objects in images (Mao et al., 2016). Both assignments require ample scaffolding to help guide the students in using the libraries and datasets, and the MSCOCO dataset is much bigger than they are used to, giving them more real-world experience with a larger dataset.

**Deep Learning** An understanding of deep learning is obviously important for recent NLP researchers and practitioners. One constant challenge is determining what level of abstraction to present neural networks (should students know what is happening at the level of the underlying linear algebra, or is a conceptual understanding of parameter fitting in the neurons enough?). Furthermore, deep learning as a topic requires and understanding of its limitations and at least to some degree how it works “under the hood" (learning just how to use deep learning libraries without understanding how they work and how they “learn" from the data is akin to giving someone a car to drive without teaching them how to use it safely). This also means explaining some common misconceptions like how neurons in neural networks “mimic" real neurons in human brains, something that is very far from true, though certainly the idea of neural networks is inspired from human biology. For my students, we progress from linear regression to logistic regression (illustrating how parameters are being fit and how gradient descent is different from directly estimating parameters in continuous probability functions; i.e., maximum likelihood estimation vs convex and non-convex optimization), building towards small neural architectures and feed-forward networks. We also cover convolutional neural networks (for transfer learning and grounded semantics), attention (Vaswani et al., 2017), and transformers including transformer-based language models like BERT (Devlin et al., 2018), and how to make use of them; understanding how they are trained, but then only assigning fine-tuning for students to experience directly. I focus on smaller datasets and fine-tuning so students can train and tune models on their own machines.

**Final Project** There is a "final project" requirement. Students can work solo, or in a group of up to three students. The project can be anything

NLP related, but projects generally are realized as using an NLP or machine/deep learning library to train on some specific task, but others include methods for data collection (a topic we don't cover in class specifically, but some students have interest in the data collection process for certain settings like second language acquisition), as well as interfaces that they evaluate with some real human users. Scoping the projects is always the biggest challenge as many students initially envision very ambitious projects (e.g., build an end-to-end chatbot from scratch). I ask students to consider how much effort it would take to do three assignments and use that as a point of comparison. Throughout the first half of the semester students can ask for feedback on project ideas, and at the halfway point in the semester, students are required to submit a short proposal that outlines the scope and timeline for their project. They have the second half of the semester to then work through the project (with a "checkpoint" part way through to inform me of progress and needed adjustments), then they write a project report on their work at the end of the semester with evaluations and analayses of their work. Graduate students must write a longer report than the undergraduate students, and graduate students are required to give a 10-minute presentation on their project. The timeline here is critical: the midway point for beginning the project allows students to have experience with classification and NLP tasks, but have enough time to make adjustments as they work on their project. For example, students students attempt to apply BERT fine-tuning after the BERT assignment even though it wasn't in their original project proposal.

**Theme of Limitations** As is the case with ambiguity, *limitations* is theme in the course: limitations of using probability theory on language phenomena, limitations on datasets, and limitations on machine learning models. The theme of limitations ties into an overarching ethical discussion that happens at intervals throughout the semester about what can reasonably be expected from NLP technology and whom it affects as more practical models are deployed commercially.

The final assignment *critical reading of the popular press* is based on a course under the same title taught by Emily Bender at the University of Washington.[5] The goal of the assignment is to learn to critically read popular articles about NLP. Given an article, they need to summarize the article, then scrutinize the sources using the following as a guide: can they (1) access the primary source, such as original published paper, (2) assess if the claims in the article relate to what's claimed by the primary source, (3) determine if experimental work was involved or if the article is simply offering conjecture based on current trends, and (4) if the article did not carry out an evaluation, offer ideas on what kind of evaluation would be approrpriate to substantiate any claims made by the article's author(s). Then students should relate the headline of the article to the main text and determine if reading the headline provides an abstract understanding of the article's contents, and determine to what extent the author identified limitations to the NLP technology they were reporting on, what someone without training in NLP might take away from the article, and if the authors identified the people who might be affected (negatively or positively) by the NLP technology. This assignment gives students experience in recognizing the gap between the reality of NLP technology, how it is perceived by others, whom it affects, and its limitations.

We dedicate an entire lecture to ethics, and students are also asked to consider the implications of their final projects, what their work can and cannot reasonably do, and who might be affected by their work.[6]

**Discussion** Striking a balance between content on probability and information theory, linguistics, and machine learning is challenging for a single course, but given the diverse student population at a public state school, this approach seems to work for the students. An NLP class should have at least some content about linguistics, and framing aspects of linguistics in terms of ambiguity gives students the tools to think about how much they experience ambiguity on a daily basis, and the fact that if language were not ambiguous, data-driven NLP would be much easier (or even unnecessary). The discussions about syntax and semantics are especially important as many have not considered (particularly those who have not learned a foreign

[5] The original assignment definition appears to no longer be public.

[6] Approaching ethics from a framework of limitations I think helps students who might otherwise be put off by a discussion on ethics because it can clearly be demonstrated that NLP models have not just technical limitations and those limitations have implications for real people; an important message from this class is that *all* NLP models have limitations.

language) how much they take for granted when it comes to understanding and producing language, both speech and written text. The discussions on how to represent meaning computationally (symbolic strings? classifiers? embeddings? graphs?) and how a model should arrive at those representations (using speech? text? images?) is rewarding for the students. While most of the assignments and examples focus on English, examples of linguistic phenomena are often shown from other languages (e.g., Japanese morphology and German declension) and the students are encouraged to work on other languages for their final project.

Assignments vary in scope and scaffolding. For the probability and information theory and BERT assignments, I provide a fairly well-scaffolded template that the students fill in, whereas most other assignments are more open-ended, each with a set of reflection and analysis questions.

### 4.2 Content Delivery

Class sizes vary between 35-45 students. Class content is presented largely either as presentation slides or live programming using Jupyter notebooks. Slides introduce concepts, explain things outside of code (e.g., linguistics and ambiguity or graphical models), but most concepts have concrete examples using working code. The students see code for Naive Bayes (both discriminative and continiuous) classifiers, I use Python code to explain probability and information theory, classification tasks such as spam filtering, name classification, topic modeling, parsing, loading and prepossessing datasets, linear and logistic regression, sentiment classification, an implementation of neural networks from scratch as well as popular libraries.

While we use NLTK for much of the instruction following in some ways what is outlined in Bird et al. (2008), we also look at supported NLP Python libraries including *textblob*, *flair* (Akbik et al., 2019), *spacy*, *stanza* (Qi et al., 2020), *scikitlearn* (Pedregosa et al., 2011), *tensorflow* (Abadi et al., 2016) and *keras* (Chollet et al., 2015), *pytorch* (Paszke et al., 2019), and *huggingface* (Wolf et al., 2020). Others are useful, but most libraries help students use existing tools for standard NLP pre-processing like tokenization, sentence segmentation, stemming or lemmatization, part-of-speech tagging, and many have existing models for common NLP tasks like sentiment classification and machine translation. The *stanza* library has models for many languages. All code I write or show in class is accessible to the students throughout the semester so they can refer back to the code examples for assignments and projects. This of course means that students only obtain a fairly shallow experience for any library; the goal is to show them enough examples and give them enough experience in assignments to make sense of public documentation and other code examples that they might encounter.

The course uses two books, both which are available free online, the NLTK book,[7] and an ongoing draft of Jurafsky and Martin's upcoming 3rd edition.[8] The first assignment (Python & Jupyter in Figure 3) is an easy, but important assignment: I ask the students to go through Chapter 1 and parts of Chapter 4 of the NLTK book and for all code examples, write them by hand into a Jupyter notebook (i.e., no copy and pasting). This ensures that their programming environments are setup, steps them through how NLTK works, gives them immediate exposure to common NLP tasks like concordance and stemming, and gives them a way to practice Python syntax in the context of a Jupyter notebook. Another part of the assignment asks them to look at some Jupyter notebooks that use tokenization, counters, stop words, and n-grams, and asks them questions about best practices for authoring notebooks (including formatted comments).[9]

Students can use cloud-based Jupyter servers for doing their assignments (e.g., Google colab), but all must be able to run notebooks on a local machine and spend time learning about Python environments (i.e., anaconda). Assignments are submitted and graded using okpy which renders notebooks and allows instructors to assign grading to themselves or teaching assistants, and students can see their grades and written feedback for each assignment.[10]

### 4.3 Adjustments for Remote Learning

This course was relatively straightforward to adjust for remote delivery. The course website and okpy (for assignment submissions) are available to the students at all times. I decided to record lectures live (using Zoom) then make them available with

[7] http://www.nltk.org/book_1ed/

[8] https://web.stanford.edu/~jurafsky/slp3/

[9] I use the notebooks listed here for this part of the assignment https://github.com/bonzanini/nlp-tutorial

[10] https://okpy.org/

| semester | Python | Ling | ML | DL |
|---|---|---|---|---|
| Spring 2017 | none | none | 10% | none |
| Spring 2019 | 50% | 30% | 30% | 10% |
| Spring 2021 | 80% | 30% | 60% | 20% |

Table 1: Impressions of preparedness for Python, Linguistics (Ling), Machine Learning (ML), and Deep Learning (DL).

transcripts to the students. This course has one midterm, a programming assignment that is similar in structure to the regular assignments. During an in-person semester, there would normally be a written final, but I opted to make the final be part of their final project grade.

## 5 Reflection on Three Offerings over 4 years

Due to department constraints on offering required courses vs. elective courses (NLP is elective), I am only able to offer the NLP course in the Spring semester of odd years; i.e., every 4 semesters. The course is very popular, as enrollment is always over the standard class size (35 students). Below I reflect on changes that have taken place in the course due to the constant and rapid change in the field of NLP, in our undergraduate curriculum, and the implications those changes had on the course. These reflections are summarized in Table 1. As I am, to my knowledge, the first NLP researcher at Boise State University, I had to largely develop the contents of the course on my own, requiring adjustments over time as I better understand student preparedness. At this point, despite the two paths into the course, most students who take the course are still Computer Science students.

**Spring 2017** The first time I taught the course, only a small percentage of the students had experience with Python. The only developed Python library for NLP was NLTK, so that and scikit-learn were the focus of practical instruction. I spent the first three weeks of the course helping students gain experience with Python (including *Jupyter, numpy, pandas*) then used Python as a means to help them understand probability and information theory. The course focused on generative classification including statistical n-gram language modeling with some exposure to discriminative models, but no exposure to neural networks.

**Spring 2019** Between 2017 and 2019, several important papers showing how transformer networks can be used for robust language modeling were gaining in momentum, resulting in a shift towards altering and understanding their limitations (so called BERTology, see Rogers et al. (2020) for a primer). This, along with the fact that changes in the curriculum gave students better experience with Python, caused me to shift focus from generative models to neural architectures in NLP and to shift to cover word-level embeddings more rigorously. I spent the second half of the semester introducing neural networks (including multi-layer perceptrons, convolutional, and recurrant architectures) and giving students assignments to give them practice in tensorflow and keras. After the 2017 course, I changed the pre-requisite structure to require our programming languages course instead of data structures. This led to greater parepraredness in at least the syntax aspect of linguistics.

**Spring 2021** In this iteration, I shifted focus from recurrant to attention/transformer-based models and assignmed a BERT fine-tuning assignment on a novel dataset using *huggingface*. I also introduced *pytorch* as another option for a neural network library (I also spend time on *tensorflow* and *keras*). This shift reflects a shift in my own research and understanding of the larger field, though exposure to each library is only partial and somewhat abstract. I note that students who have a data science background will likely appreciate tensorflow and *keras* more as they are not as object-oriented than *pytorch*, which seems to be more geared towards students with Computer Science backgrounds. Students can choose which one they will use (if any) for their final projects. More students are gaining interest in machine learning and deep learning and are turning to MOOC courses or online tutorials, which has led in some degree to better preparation for the NLP course, but often students have little understanding about the limitations of machine learning and deep learning after completing those courses and tutorials. Moreover, students from our university have started an Artificial Intelligence Club (the club started in 2019; I am the faculty advisor), which has given the students guidance on courses, topics, and skills that are required for practical machine learning. Many of the NLP class students are already members of the AI Club, and the club has members from many academic disciplines.

### 5.1 Student Feedback

I reached out to former students who took the class to ask for feedback on the course. Specifically, I asked if they use the skills and concepts from the NLP class directly for their work, or if the skills and concepts transferred in any way to their work. Student responses varied, but some answered that they use NLP directly (e.g., to analyze customer feedback or error logs), while most responded that they use many of the Python libraries we covered in class for other things that aren't necessarily NLP related, but more geared towards Data Science. For several students, using NLP tools helped them in research projects that led to publications.

## 6 Conclusions & Open Questions

With each offering, the NLP course at Boise State University is better suited pedagogically for students with some Data Science or Computer Science training, and the content reflects ongoing changes in the field of NLP to ensure their preparation. The topics and assignments cover a wide range, but as students have become better prepared with Python (by the introduction of new prerequisite courses that cover Python as well as changing some courses to include assignments in Python), more focus is spent on topics that are more directly related to NLP.

Though I feel it important to stay abreast of the ongoing changes in NLP and help students gain the knowledge and skills needed to be successful in NLP, an open question is *what* changes need to be made, and a related question is *how soon*. For example, I think at this point it is clear that neural networks are essential for NLP, though it isn't always clear what architectures should be taught (e.g., should we still cover recurrant neural networks or jump directly to transformers?). It seems important to cover even new topics sooner than later, though a course that is focused on research methods might be more concerned with staying up-to-date with the field, whereas a course that is more focused on general concepts and skills should wait for accessible implementations (e.g., huggingface for transformers) before covering those topics.

With recordings and updated content, I hope to flip the classroom in the future by assigning readings and watching lectures before class, then use class time for working on assignments.[11]

Much of my course materials including notebooks, slides, topics, and assignments can be found on a public Trello board.[12]

## Acknowledgements

I am very thankful for the anonymous reviewers for giving excellent feedback and suggestions on how to improve this document. I hope that it followed those suggestions adequately.

## References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283.

Apoorv Agarwal. 2013. Teaching the basics of NLP and ML in an introductory course to information science. In *Proceedings of the Fourth Workshop on Teaching NLP and CL*, pages 77–84, Sofia, Bulgaria. Association for Computational Linguistics.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Emily M Bender and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Association for Computational Linguistics*, pages 5185–5198.

Steven Bird, Ewan Klein, Edward Loper, and Jason Baldridge. 2008. Multidisciplinary instruction with the natural language toolkit. In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, pages 62–70, Columbus, Ohio. Association for Computational Linguistics.

François Chollet et al. 2015. Keras. https://keras.io.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv*.

[11] This worked well for the Foundations of Data Science course that I introduced to the university; the second time I taught the course, I flipped it and used class time for students to work on assignments, which gave me a better indication of how students were understanding the material, time for one-on-one interactions, and fewer issues with potential cheating.

[12] https://trello.com/b/mRcVsOvI/boise-state-nlp

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.

Reva Freedman. 2008. Teaching NLP to computer science majors via applications and experiments. In *Proceedings of the Third Workshop on Issues in Teaching Computational Linguistics*, pages 114–119, Columbus, Ohio. Association for Computational Linguistics.

C. Kennington and D. Schlangen. 2015. Simple learning and compositional application of perceptually groundedword meanings for incremental reference resolution. In *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, volume 1.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What we know about how BERT works. *arXiv*.

David Schlangen, Sina Zarriess, and Casey Kennington. 2016. Resolving References to Objects in Photographs using the Words-As-Classifiers Model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1213–1223.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.