NLP Module: Description for Teachers

Overview This document describes an outreach module intended to introduce high school students to basic concepts from natural language processing (NLP), focusing on how ChatGPT works. NLP is a discipline around building systems to solve problems involving human language input, such as dialogue systems (like ChatGPT), machine translation (like Google Translate), question answering, and more. Specifically, we focus on describing language models, which are the core technology underlying ChatGPT. Better understanding of these models enables students more critically think about what ChatGPT can and can't do.

We discuss (at a high level) ideas from machine learning, including the general paradigm, how models are learned from data, what parameters are, and more. The main mathematical idea we explore in the context of language modeling is the idea of predicting next words by placing probability distributions over them (including conditional probability distributions) and estimating probabilities from data. These are described with examples and **do not rely on students having seen probability before.**

The module content is available at `https://www.cs.utexas.edu/~gdurrett/courses/nlp-module/`

Goals

1. Expose students to basic concepts from natural language processing and machine learning, including principles of basic probability and learning models on training data
2. Give students a guided tour of ChatGPT, including how best to use it and how to think critically about it
3. (optional) Give students a programming challenge involving querying data structures, basic loop constructs, and mathematical operations
4. Give students pointers to other resources to follow up on these topics

How to use this module

This module can be tailored to contain different amounts of coding and be suitable for students at different experience levels.

Version 1: no coding The main series of videos is meant to be appropriate for students with any level of computer science background. There may be some mathematical concepts that students may not have seen, but these are explained with examples and do not rely

Version 2: with coding There are two additional videos and a coding exercise (in either Java or Python) that allow students to explore these concepts more deeply. This exercise involves implementing a *bigram* language model, which models the probability distribution $P(\text{next word} \mid \text{current word})$. For instance, if you see the word *to*, maybe *France* has probability 0.1, *Spain* has probability 0.08, and so on and so forth. Such models can be used for tasks like predictive text on a phone. Implementing and querying this model largely uses standard math and data structures, but the data structures to store all the quantities needed (particularly counts of neighboring pairs of words) may be a little tricky for students just starting out.

We recommend using this version of the module in AP Computer Science A. It fits most naturally into Unit 7, Topics 7.1-7.3.[1] The two coding parts feature incremental construction of an ArrayList and iteration through an ArrayList with a stopping condition based on the contents. Exercise 0 also has a conceptual question about 2D arrays (Unit 8).

Detailed Description

The main content consists of 11 videos (approx. 64 minutes), plus exercises and discussion. The programming extension includes an additional 2 videos (approx. 15 minutes) and a coding exercise which may take 45 minutes.

We now describe each video in detail so you know what to expect and how to address questions that come up in the exercises. Note that all of the exercises are optional and do not interact, so you can feel free to skip any that you want and it won't impact the rest of the module.

Intro to ChatGPT and NLP This video introduces high-level ideas of what NLP is and shows some examples of ChatGPT.

Exercise: The video ends with an exercise asking the following: *Try a few things in ChatGPT and share the results with others! Try: (a) asking it about a fact; (b) having it help you brainstorm about something; (c) solve a math problem*

You should have students navigate to `chat.openai.com`, where they should see a web interface where they can ask these questions. For (a), the intention is to see whether facts are correct or not. They most likely will be for anything simple that students try, but it is definitely possible to find mistakes or to make questions that mislead the system. For (b), students should see that ChatGPT is often quite creative and can easily come up with 10 suggestions for whatever you ask it about, although the suggestions may vary in quality. For (c), simple math problems are often within its capabilities, but if you ask it to multiply two big numbers, it will usually get it wrong if you compare it to a calculator.

Machine Learning This video explains the paradigm of machine learning, which involves learning how to do a process like translation from data, rather than programming it in a "top-down" fashion.

Language Modeling This video explains language modeling and the idea of "next word prediction", which is what ChatGPT does, by analogy with predictive text.

Exercise (mid-video): The video asks: *Suppose we have the context "I want to ____". Lots of words can come next and form sensible sentences. Think about a few words that can come next; what do these have in common?*

The expected answer is that most things are verbs, but adverbs are also possible (*I want to quickly go to...*). What the video then describes is how it's not just any verb that can come next: verbs that you want to do are more likely. So even this short context becomes very complex when thinking about what can come next!

The second question is: *Can you think of a context (a start of a sentence like "I want to ____" ending in a blank) where the next word has to be one word in particular for it to be correct?*

[1] `https://apcentral.collegeboard.org/media/pdf/ap-computer-science-a-course-and-exam-description.pdf` at the time of this writing in August 2022

The video provides two examples. Usually the best example is a "fill-in-the-blank" style question that only has one answer. For example, "The movie "Titanic" was directed by ____". See the video itself for some more examples.

Mathematics of Language Modeling This video explains how models like ChatGPT predict probability distributions over next words. These concepts are introduced as if students have not seen probability before.

Exercise: The video asks: *Find some prompts in ChatGPT that always return the same answer. Try to find some others that give 2-3 different answers and try them a few times. Can you get a sense of the probability for each answer? Hint: try asking for a random word, or a random word starting with some letter, or a random number.*

The first of these can be achieved by asking a very factual question, like those in the exercise for "Language Modeling." The second is more open to exploration. Interesting, asking ChatGPT to give a random letter, or a random word starting with "d", etc. will not give a wide variety of options. Often, just one or two distinct answers will be returned, even if you ask the model the same thing multiple times. It is not easy to explain precisely why different examples behave differently, because it depends on details of the model's data that are impossible to analyze or understand with current tools. This kind of exploration is the sort of thing that researchers in the field do as well, so students should use their creativity and see what they find.

Mathematics of Language Modeling (Advanced) This segment is optional, but recommended if students will do the programming assignment. It draws on some more advanced concepts from probability, such as conditional probability, and walks through things with a slightly higher level of mathematical sophistication.

Exercise: See "Exercise 0" on the website for pen-and-paper exercises around understanding language models and the data structures involved with them. Solutions can be found at
`https://cs.utexas.edu/~gdurrett/courses/nlp-module/exercise0-solutions.pdf`

If you are doing the coding version, this is the most natural place to stop and do that. However, you can also do it at the end.

ChatGPT: The Basics This video describes the way that ChatGPT implements a language model and is learned from data. It describes the Transformer neural network architecture that ChatGPT uses at a very high level.

ChatGPT Part 2 This video explains how ChatGPT is not just a basic language model, but has been additionally trained to produce certain types of responses.

How does ChatGPT know things? This video returns to the idea of factual queries and explores why ChatGPT may get some right and get some wrong.

Exercise: The video asks: *Try asking GPT about some questions from topics you've learned in history, science, or other classes. Do you see any mistakes in what it says? Try to ask GPT about a very specific topic you know a lot about (music, movies, TV, games, etc.). Hint: pick something obscure and consider asking "why" questions. For example, "why did [character] do [action]"? Or ask about a minor detail. See if you can find a mistake in what it says!*

Hopefully students can find some questions that "trick" the language model. Very basic questions about history or science will usually be answered correctly. As we saw in video 1, some math problems don't work. The system has surprisingly obscure knowledge about media. However, it will not know recent movies or TV shows.

Risks of Large Language Models This video explores some concrete worries that researchers today have around how language models like ChatGPT could be applied.

Exercise: The video asks *What do you think are the biggest potential harms of ChatGPT that you can imagine or you've heard about?*

It is up to you whether or not you want to have this discussion. Some possible topics beyond what's mentioned in the video are: (a) ChatGPT being used to produce disinformation; (b) ChatGPT being used to produce scripts in place of Hollywood screenwriters; (c) people becoming more reliant on these tools and not thinking critically, or accepting its information at face value.

For most of these questions, it is not known yet how effective these models might eventually be. And for some of them, it becomes an ethical issue (should we be putting humans out of work with tools like this?) with no right or wrong answer.

Future Risks of Large Language Models? This video explores some more "sci-fi" fears around language models, including the idea that these systems are advancing very rapidly and could pose a threat to human civilization.

Exercise: The video asks *Where do you think AI systems might be in 10 years? What about in 50 years?* This is largely speculative, and even experts in the field don't have good answers. Some people believe that our world will be radically transformed in even 10 years. ChatGPT is certainly far beyond where language processing technology was 10 years ago. However, robotics has progressed relatively much less in that time. It is very unlikely we will have humanoid robots walking among us in 10 years. Self-driving cars may be increasingly commonplace, though.

Where to go next This video concludes the module by giving some advice and web links to places where students can explore more of this material.

Programming

This module supports both Java and Python versions; the two videos that depend on code specifics have been recorded twice, once for each language. Code for both versions is available either as stand-alone downloads or as web-based projects through `repl.it`.

Bigram LM Code note: there are versions of this video for both Java and Python

Querying the LM note: there are versions of this video for both Java and Python

Exercise 1: Implementing bigram language models Students will use basic data structures and provided framework code to query a probabilistic model of what the next word in a sentence is likely to be. Framework code reads in data and populates the needed data structures; students primarily implement loop logic to sample sentences from the probability distribution placed by the model. This exercise follows a worksheet.