# Generative AI in computer science courses

- general intro to the topic
- how generative AI works
- some ways AI can help you
- some of the risks/problems associated with AI use
- general guidelines for safe/ethical AI use
- guidelines for some courses I teach (e.g. 159, 161, 265)

# Generative AI

- obviously AI use is everywhere around us: creating/editing images and audio, writing/editing code/documents, searching for content, answering questions, etc
- it is an evolving area: its capabilities (and the associated benefits and risks) are changing rapidly
- everyone needs to be aware of how it can be used to their benefit, but also to be aware of the problems associated with it
- this is especially true for those of us in the field of Comp Sci: we play a heavy role in its development, in educating others on the risks and benefits, and in advising and development of appropriate safeguards

# How generative AI works
## a(n over-)simplified view

- consumes vast quantities of sample input (training data)
- can be trained on general content (e.g. as much as it can scrape from the public internet) or in very specific areas (e.g. genome analysis)
- tries to store/analyze/represent the training data as a vast pool of patterns of tokens (words, symbols)
- also reads your queries as a pattern of tokens

# Answering a query

- a huge collection of hardware is configured to process your query + the patterns collected from its training
- based on your query and its training, it tries to generate tokens, one by one, that it thinks fit an appropriate response pattern
  - hmmm ... a response to this query pattern would probably start with an X, then the next token would probably be a Y, then the next token would probably be a Z, ... and so on

# Effectiveness of generative AI

- because of the tremendous processing power and huge training sets, the patterns it generates as an 'answer' are often pretty good
- don't make mistake of thinking it understands what it's saying, it's all just patterns/probability
- the likelihood of its response pattern being wrong increases as the complexity of your query increases, and increases if your query isn't well represented by the training data

# The weakness in the responses

- it's often very very good at generating response patterns that ***sound*** right, rather than responses that ***are*** right
- in fact they can be either completely or (sometimes worse) very subtlely false, incorrect, misleading, or dangerous
- its responses are only as good as its training data: if it was poorly trained it will give poor results
- you can't take any statement made by a generative AI as fact or truthful: it's just a collection of words that it thinks approximate the pattern of what a valid response might look like

# Risks and issues with gen AI

- wide range of ethical risks/issues
- risks/issues with the correctness of the results
- key limitations on the kinds of things genAI does well, some important things it is still very weak at
- as mentioned earlier, it is only as good as its training: an AI trained on a poor data set (or poor for your queries) is going to give low quality results!

# Ethical issues with genAI

- built in discrimination/bias (racial, gender, religious, etc) stemming from bias in its training data
- privacy issues: you're giving away everything you tell it about yourself, your content, your family/friends, etc
- work loss: is the AI replacing work that a human would have been paid for?  is this affecting someone's ability to make a living?
- intellectual property rights:
  - if you embed AI-generated content into your product then can the AI company claim some ownership rights to your product?
  - if you put content from your product into an AI, can you be certain that content won't appear as part of a response to someone else's query?

# Correctness issues with genAI

- **token limits:** most AI (especially free versions) have limits on how many tokens it keeps track of in a query/response, so can forget/ignore things at the beginning of a complex query or long conversation
- **hallucinations**: it can generate fictional content and represent it as fact (because the fictional content ***looks*** like it has the right pattern)
- **subtle flaws:** the answer might be generally correct but contain one or more parts that aren't actually correct ... *using the response without recognizing these can be disastrous (and you have to fully analyze and understand its response to catch them)*
- **embedded risks:** if the relevant content in the training data contained security flaws/weaknesses then so may its responses

# What genAI doesn't do well

- limitations in the number of tokens, processing power, and training data mean that it doesn't handle complex queries well:
  - much of what we do in CS revolves around the design of large complex systems: the AI can often only help us with bits and pieces of the whole
- training data (so far) hasn't found a good way to emulate the human experience and human responses:
  - it isn't very good at predicting or recognizing when something will look/feel good to a human user, making it weak at designing new software that people will enjoy using
- ***want to avoid being AI-redundant in our field?***
  - Be good at those things: learn how to design/develop large complex systems, learn how to figure out what people actually want and will enjoy using

# General suggestions using genAI

- check with your boss/instructor/teammates to see what their policies/opinions are for AI use in your current job/project/assignment: ***don't put content into an AI and don't use AI-generated content unless all concerned parties have explicitly given permission***
- never trust any response it gives you: treat it as a starting point for fact checking/exploration
- never put anything into an AI unless you're ok with it becoming public
- never copy anything AI generated into your product unless you're ok sharing ownership of your product with the AI company
- think of it like a mischievous genie or unreliable co-worker: it may give you exactly what you need, but don't trust it

# How generative AI *can* help

- generating practice questions/exercises
- learning to understand types of error messages
- learning correct syntax or use of language features
- getting alternative explanations for concepts/techniques
- getting suggestions for tools, features, techniquest that might be helpful
- learning about strengths/weaknesses of tools, features, techniques
- getting feedback on code/ideas (some risks to discuss)
- getting suggestions on how to review/test code/items

# Have genAI explain concepts

- If you're struggling to understand the way a prof explains a feature/concept, or with the advantages/disadvantages of using a technique, ask an AI for an explanation
  - I'm struggling to understand how pointers work in C++ or how and why I'd want to use them, could you provide an introductory explanation?

# AI-generated practice exercises

- tell it the topic area and ask it to generate things you can use for practice or self-evaluation:
  - practice quiz questions
  - practice programming exercises
  - practice essay/discussion questions

# genAI and error messages

- compilers and run-time crashes often generate cryptic error messages when syntax errors are encountered
- use the AI to investigate what kinds of things cause a specific type of error message, e.g.
  - “I'm a first year CS student writing linked list code in C++ and I keep getting 'segmentation fault' error messages when I run my program.  Could you describe the kinds of things that might be causing them?”

# Using genAI to suggest tools/features

- sometimes we need ideas for starting points when we're trying to come up with a design or implementation for something new
  - “I'm a second year CS student trying to include a database in my term project written in C++ on linux, could you outline some of the available choices and their pros/cons?”

# Learning syntax

- the first few times we use specific features in a language (or options/settings in a tool) it often helps to have more examples on correct use, e.g.
  - “I'm a first year Comp Sci student programming in C++.  Could you give examples showing the correct syntax and appropriate use of pass-by-reference?”

# Getting testing/review suggestions

- generally we want to ensure our code is of high quality before submitting it, which means carefully reviewing and testing it
  - you can ask an AI to suggest test cases to try out for the particular problem your code is solving
  - you can ask an AI to suggest things you should look for when you're reviewing your code to make sure it's “good”

# Generating boilerplate code

- sometimes there are chunks of very repetitive, mechanical code we need to generate: segment after segment that is nearly the same and entirely predictable, involving little thought but lots of typing
- for these cases it might improve efficiency to have a genAI-based tool generate the starting code
- ***this carries the risks mentioned earlier about actually embedding said code in your product/submission***

# Summary of effective AI use

- use the AI tool as a learning aid, not a substitute for learning: *ideally it teaches you something so that you don't need to ask an AI that same question again in the future*
- make sure you carefully check everything an AI generates for you
- be extremely wary of entering any of your content into an AI and of embedding any AI-generated content into your product

# Guidelines for genAI use in my lower-level courses (159, 161, 265)

- use AI as a learning aid, not as a substitue for learning
- quizzes/midterms/exams are paper/pencil with no AI to help you
- anything AI-generated (indeed, anything that wasn't designed and written by you) must be explicitly cited in the code comments and an accompanying readme ***(otherwise it is academic misconduct)***
- if an exercise is meant to demonstrate that you can design/implement something yourself and you use AI-generated code instead then no credit will be given
- ***for any content you submit for a course I reserve the right to call you in to discuss your solution and evaluate if you actually understand it, and may adjust your mark accordingly***

# My use of genAI for courses

- ***I will not*** enter information about you into a genAI
- ***I will not*** put answers/content you have generated into an AI (e.g. for marking/evaluation)
- ***I recommend you check with instructors in other departments to see if they ever put your information or your content into genAIs, and check if they are aware of the risks/issues if they say yes***
- I may periodically use it in the 'learning' approaches mentioned earlier, as one starting point for investigating tools, techniques, features, syntax, ideas, etc