JOHNS HOPKINS
CAREY BUSINESS SCHOOL

Lecture 6

BU.330.740 Large Scale Computing on the Cloud

Minghong Xu, PhD.
Associate Professor

Reflections

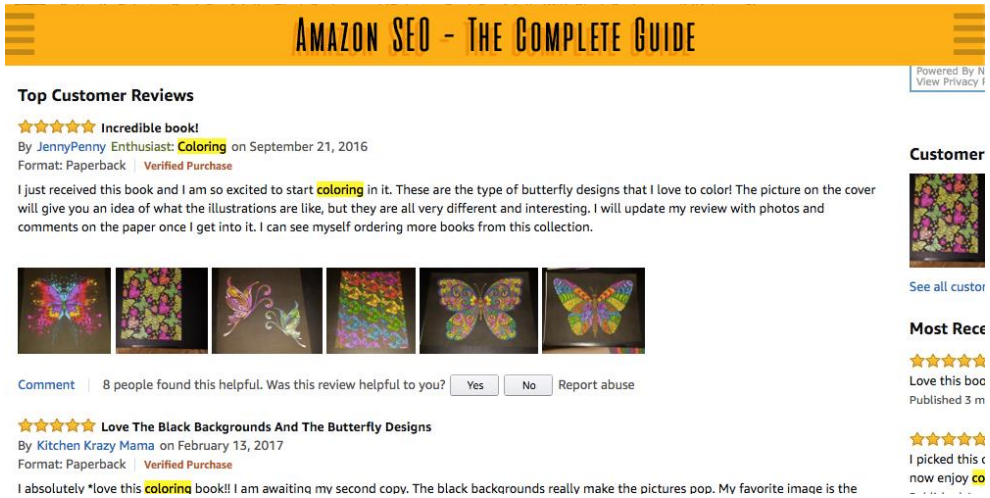
- Big data applications and computer vision applications
- AWS Rekognition
  - AutoML
  - No Code
- More code ~~examples~~?
  - SageMaker repository

Today's Agenda

- NLP tools on cloud and applications
- Project preview
- AI-assisted programming
- Lab 5: AWS Lambda and Q Developer
- Final review

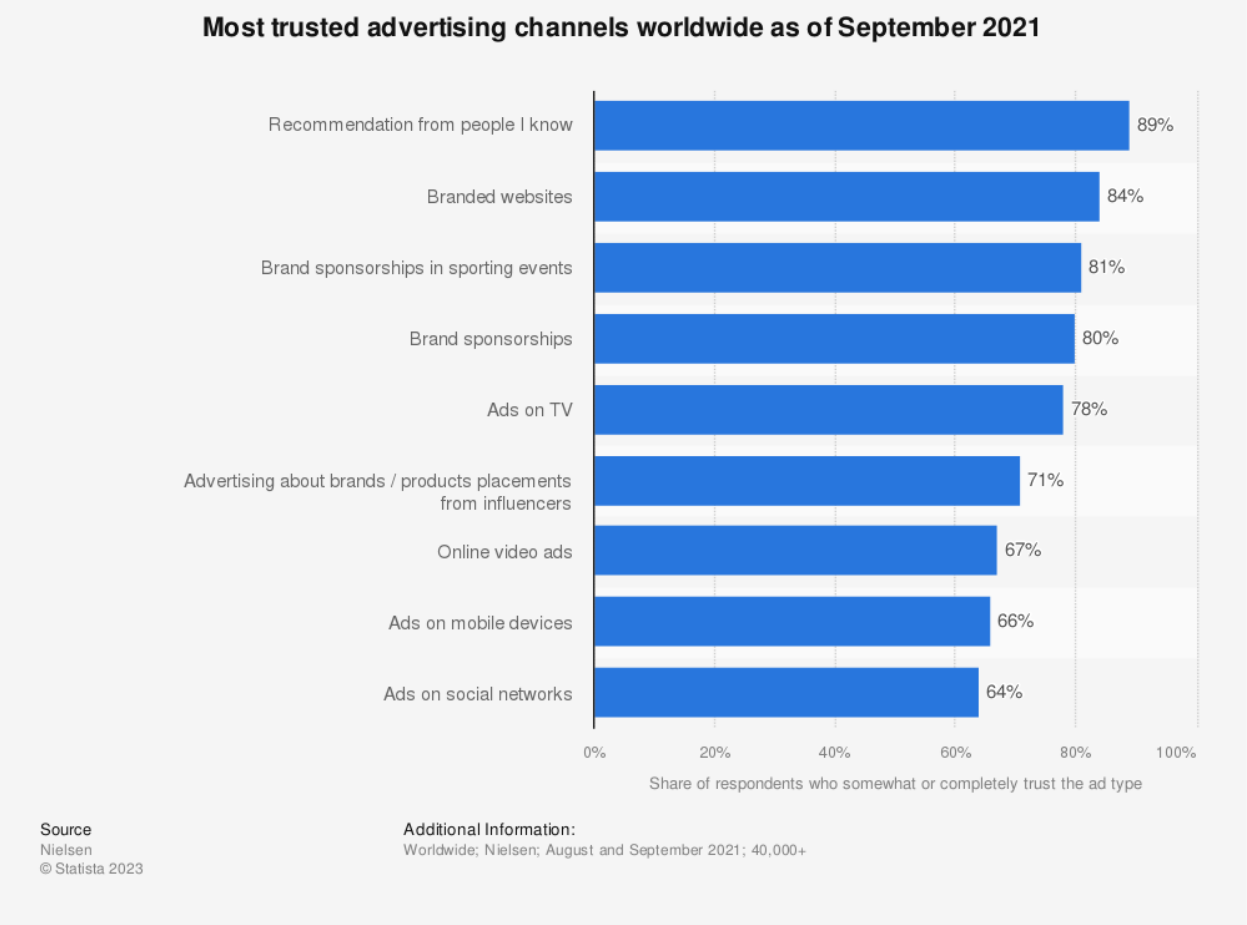
User Generated Content (UGC)

- Online users are creating and sharing images and words like never before
- “any form of content such as blogs, wikis, discussion forums, posts, chats, tweets, podcasts, digital images, video, audio files, and other forms of media that were created by users of an online system or service”



UGC: Generate Trust

- Consumers are more likely to trust recommendations by people they know over other forms of advertising (Nielson)
- Listing of a book on the New York Times bestseller list causes a modest increase in sales (Sorensen, 2007)
- Willingness to pay of consumers is about $4.50 greater for a top ranked app than for the same unranked app (Carare, 2012)



https://www.statista.com/statistics/222698/consumer-trust-in-different-types-of-advertising/

UGC: Big Data for Mining

- Limitless pool of content
- Businesses that use customer content on their marketing channels see higher conversion, click-through rates to product pages, and average order values

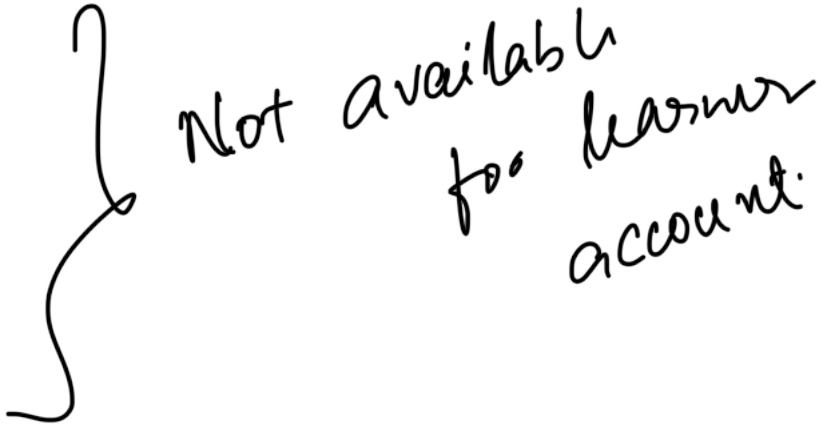
- Fake reviews detector
  - https://streetfightmag.com/2022/10/13/5-tools-for-fake-review-detection/
  - https://www.fakespot.com/

Text Mining

- Text analytics, natural language processing (NLP)
- Mine knowledge/information from huge amount of text (unstructured) data
- Many NLP systems are trained on very large collections of text (also called *corpora*) such as the Wikipedia corpus

AWS Demo

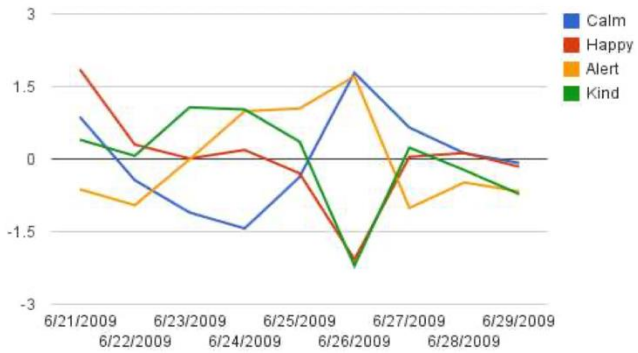
- AWS Comprehend
- AWS Polly
- AWS Transcribe
- AWS Lex



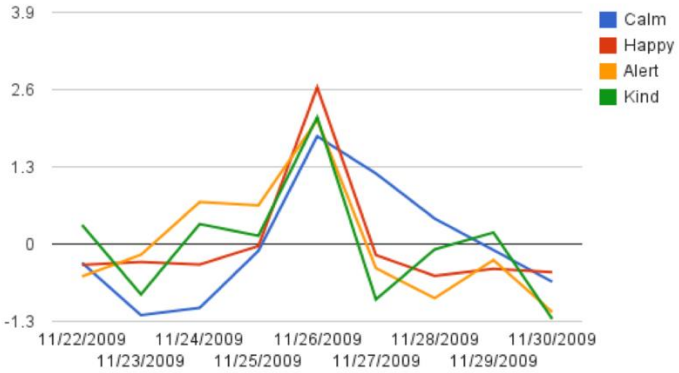
- These services are not available to learner accounts

Sentiment Analysis Use Cases

- Stock Prediction Using Twitter Sentiment Analysis (Mittal and Goel, 2012)
  - use twitter data to predict public mood
  - use the predicted mood and previous days' DJIA values to predict the stock market movements



(a) Various moods after Michael Jackson's death on 25 June 2009



(b) Various moods on Thanksgiving day on 26 November 2009

Sentiment Analysis Use Cases

- A system for real-time twitter sentiment analysis of 2012 US presidential election cycle (Wang, Hao, et al. 2012. *Proceedings of the ACL 2012 system demonstrations*)

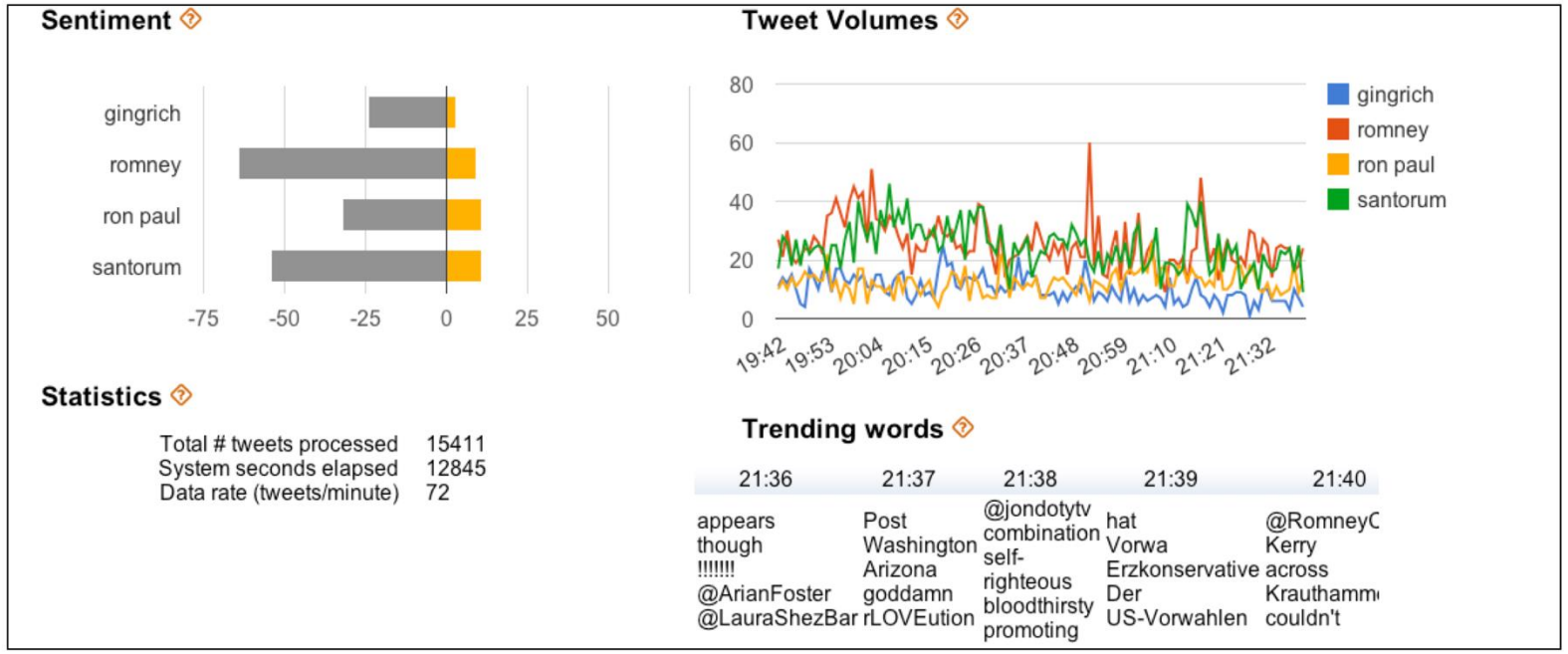


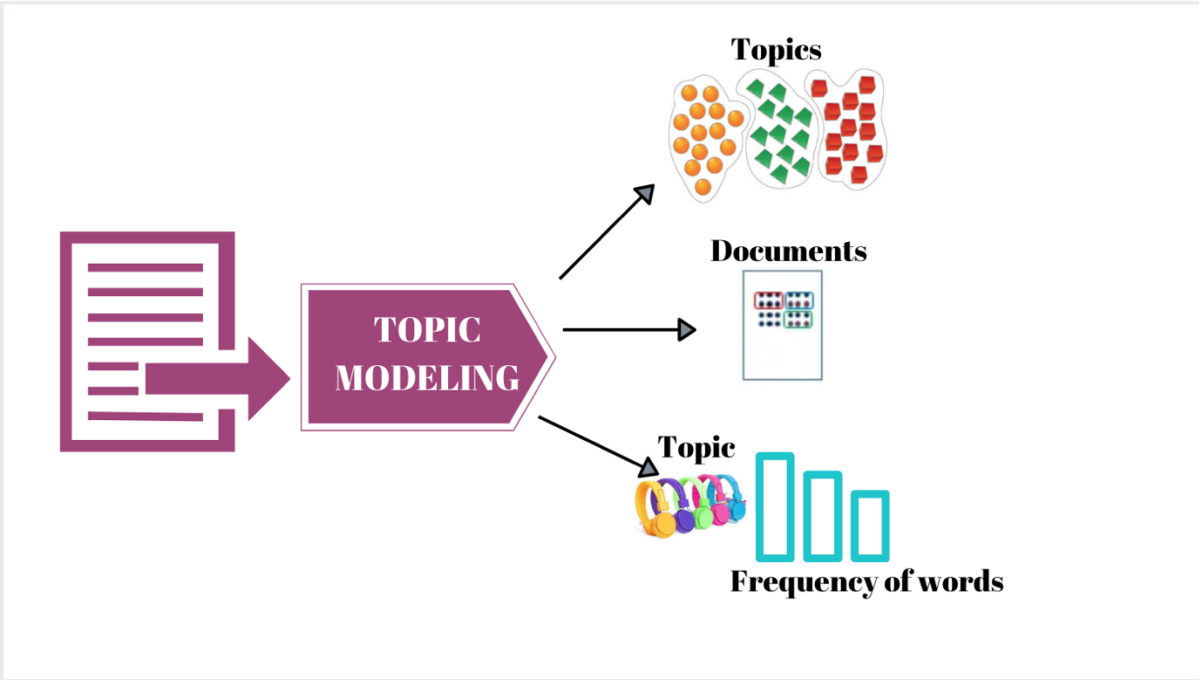
Figure 3. Dashboard for volume, sentiment and trending words

Sentiment Analysis Use Cases

- Word power: A new approach for content analysis (Jegadeesh and Wu. 2013. *Journal of financial economics*)
  - 10-Ks filed from January 1995 through December 2010 from the SEC's Electronic Data Gathering, Analysis, and Retrieval (EDGAR) database
  - Find a significant relation between measure of the tone of 10-Ks and market reaction for both negative and positive words
- Evidence on the information content of text in analyst reports (Huang, Allen H., et al. 2014. *The Accounting Review*)
  - Investors react more strongly to negative than to positive text, suggesting that analysts are especially important in propagating bad news
  - Analyst report text is shown to have predictive value for future earnings growth in the subsequent five years.

Topic Modeling

- Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently



Latent Dirichlet Allocation (LDA) Model

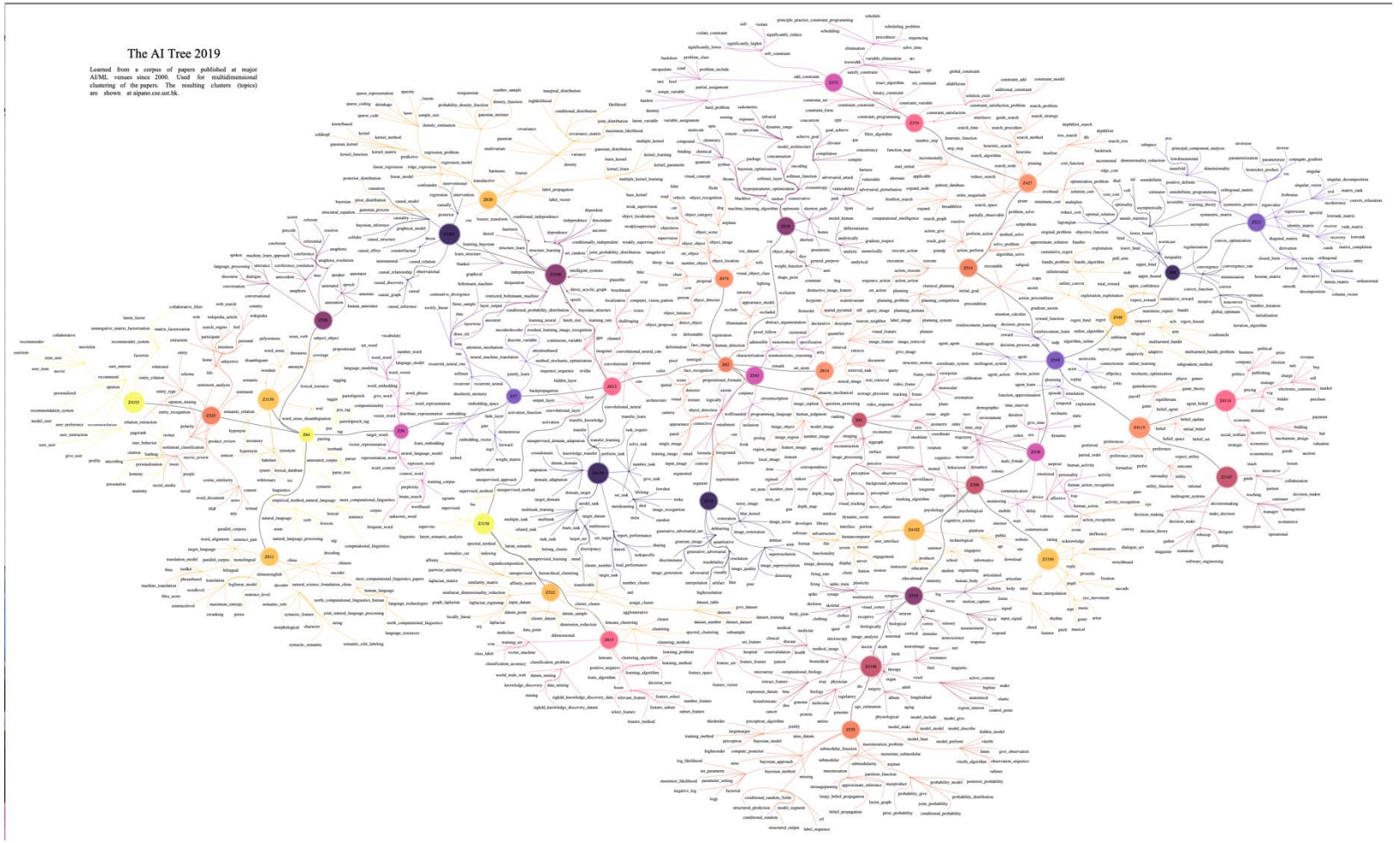
- Traditional topic model
- Documents cover a small number of topics and that topics often use a small number of words
- Basis for other topic models

i-th row, *j*-th column: # times word *j* appears in doc *i*

Word/Topic	Weather	Food
Cold	0.3	0.1
Hot	0.7	0.3
Apple	0	0.5
Pie	0	0.1

Topic/Document	Doc 1	Doc 2
Weather	0.1	0.5
Food	0.9	0.5

The AI Tree



http://home.cse.ust.hk/~lzhang/topic/ai-tree.pdf

Term Project

Project Presentation Next Week

- Project report and peer evaluation due next week
  - via Canvas->Project
  - 12 pages PPT deck, with details in notes section
  - Zip with preliminary dataset and results
- Project presentation
  - Every team has 10 mins
  - Presentation order
- Good chance to learn from other teams

Project Topic

- Business proposal using big data, can be either:
  - Volume, big in size
  - Velocity, high in speed such as streaming
  - Variety, especially unstructured such as text and image
  - Value, timely response is important
  - Veracity, complex data preprocessing is required
- Report coverage
  1. Business opportunity/question
  2. Data
  3. Method
  4. Preliminary results and findings

Project Ideas and Tips

- You can explore almost everything, except
  - A relational database solution
  - Problems can be solved in Excel
- Any cloud tool is acceptable, such as AWS, Google CoLab, ect.

- Model performance heavily depends on data
  - Insufficient data may result in overfitting
- You can use a small preliminary dataset to demo feasibility
  - But it should be at least larger than lab/homework datasets

Skills Trained

- For project in general
  - **Ideas are important!**
  - How to apply class knowledge to business cases
  - Choose/collect relevant data
  - Apply appropriate/fit methods
  - Interpret results, generate report
- For presentation
  - Training for your interview, especially for different audience
- For Q&A
  - After interviewer talks about a sample project, you should be able to ask questions or make comments

AI-Assisted Programming

Andrej Karpathy

Andrej Karpathy

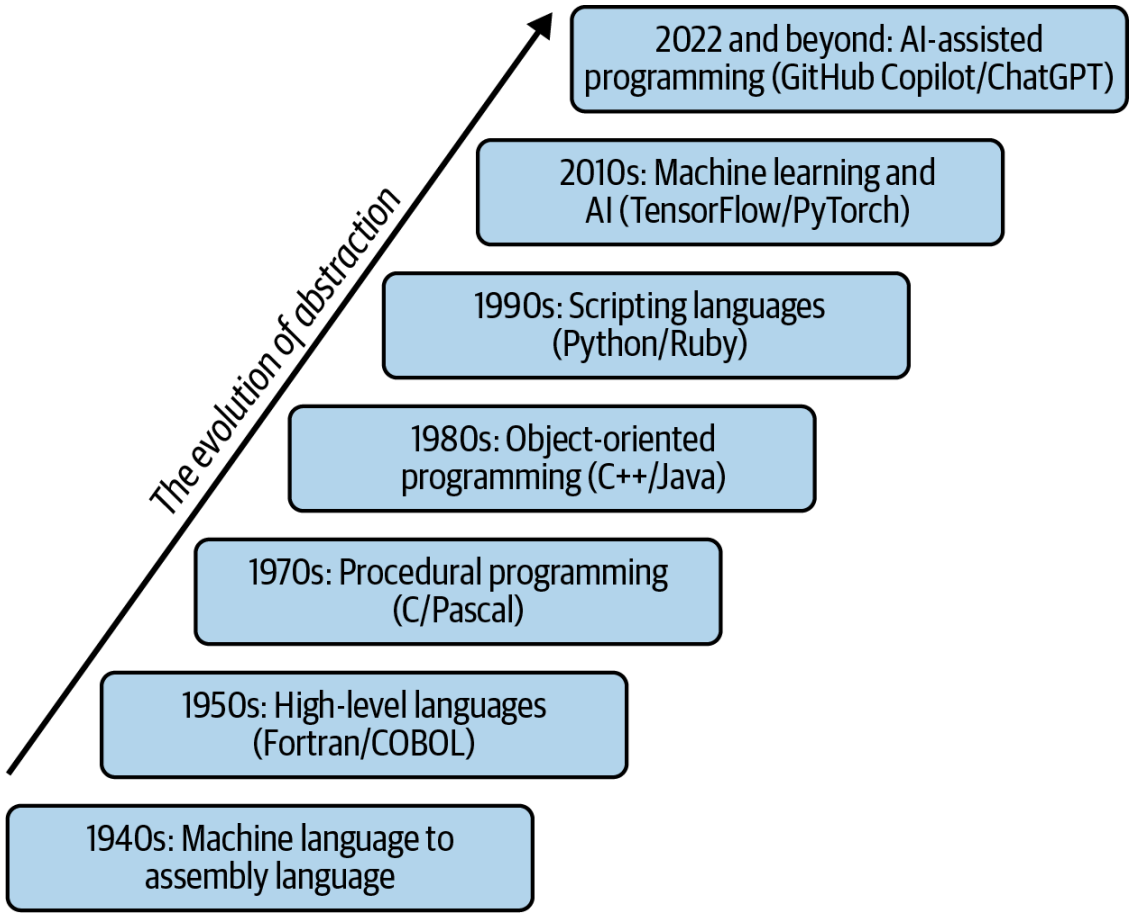
I designed and was the primary instructor for the first deep learning class Stanford - CS 231n: Convolutional Neural Networks for Visual Recognition. The class ...

Some Books and Tools

- AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment; Tom Taulli; O'Reilly Media; 1st edition (May 21, 2024)
- Learn AI-Assisted Python Programming; Leo Porter and Daniel Zingaro; Manning Publications; 2nd edition (October 2024)

- Tools
  - GitHub Copilot
  - AWS Q Developer (formerly CodeWhisperer)
  - Google Gemini
  - ChatGPT

Evolution


2022 and beyond: AI-assisted programming (GitHub Copilot/ChatGPT)
2010s: Machine learning and AI (TensorFlow/PyTorch)
1990s: Scripting languages (Python/Ruby)
1980s: Object-oriented programming (C++/Java)
1970s: Procedural programming (C/Pascal)
1950s: High-level languages (Fortran/COBOL)
1940s: Machine language to assembly language
The evolution of abstraction

Reading Code

- Lower level: understand line by line
  - Trace the value of variables
- High level: determine overall purpose of a program

- Ask the tools to explain the code for you
  - *What does the following Python code do…*
  - *Explain the following program at high level…*

Problem Decomposition

- **“Manage” the code**
- Break a large problem down into subproblems
- Each is well-defined and re-usable
- Keep each step short and understandable
  - No more than 20 lines
- Further divide if the subproblem is still too large

- Ask the tools to suggest decomposition
  - *What are the steps to build a …?*

Other Reports

- Stack Overflow 2023 developer survey
- https://survey.stackoverflow.co/2023/#overview

- Accenture case
- https://aws.amazon.com/blogs/machine-learning/how-accenture-is-using-amazon-codewhisperer-to-improve-developer-productivity/

Lab 5 AWS Lambda and Q Developer

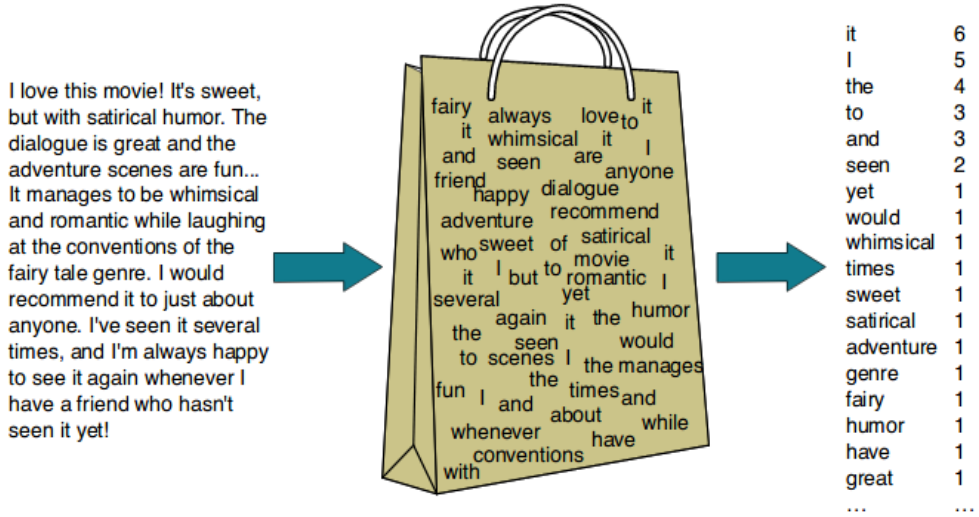
- AWS Lambda: “run code without thinking about servers”
  - Refer to Lecture 4, serverless model deployment

- AWS Q Developer: similar to Google Colab “generate with AI”
- https://docs.aws.amazon.com/amazonq/latest/qdeveloper-ug/setting-up-AWS-coding-env.html

Appendix: Traditional NLP

Bag of Words Model

- A text (such as a sentence or a document) is represented as the bag of its words
- Disregard grammar and even word order, but keep multiplicity
- Commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier



Term Frequency

- Term: token in text
  - Words, phrases, etc.
- Term frequency: how often a term occurs in a document
  - A term is more important if it occurs more frequently in a document
- *So what to do?*
  - Count the occurrence!
- $tf(t, d) = frequency\ count\ of\ term\ t\ in\ doc\ d$
- *Any issue with this approach?*

TF Normalization

- Documents have different length
  - Doc 1 has 1000 words, and 'Hadoop' appears 5 times
  - Doc 2 has 10 words, and 'Hadoop' appears 2 times
- *How to solve this?*
  - Normalization!
- $tf(t,d) = \frac{frequency\ count\ of\ term\ t\ in\ doc\ d}{total\ words\ in\ doc\ d}$
- There are other ways to do normalization, such as maximum TF normalization

Inverse Document Frequency

- Document frequency: a term is more discriminative if it occurs only in fewer document
- Inverse document frequency: assign higher weights to rare terms
- $idf(t) = \log(\frac{total\ documents}{documents\ with\ term\ t})$
- Combining *tf* and *idf*
- $tf \cdot idf = tf(t,d) \times idf(t)$

Python Packages for NLP

- Natural Language Toolkit (NLTK)
- TextBlob
- CoreNLP
- Gensim
- spaCy
- polyglot
- scikit–learn
- Pattern

Other Techniques in NLP

- Word Sense Disambiguation (WSD)
  - Words have different meanings in context
- Named Entity Recognition
  - Phrases instead of two words, such as “Johns Hopkins”
- Part-of-speech tagging
  - Distinguish nouns, verbs, adjectives…
- Sentence recognition
  - Figure out when sentences end, text reasoning

References

- AI-Assisted Programming: Better Planning, Coding, Testing, and Deployment; Tom Taulli; O'Reilly Media; 1st edition (May 21, 2024)
- Learn AI-Assisted Python Programming; Leo Porter and Daniel Zingaro; Manning Publications; 2nd edition (October 2024)