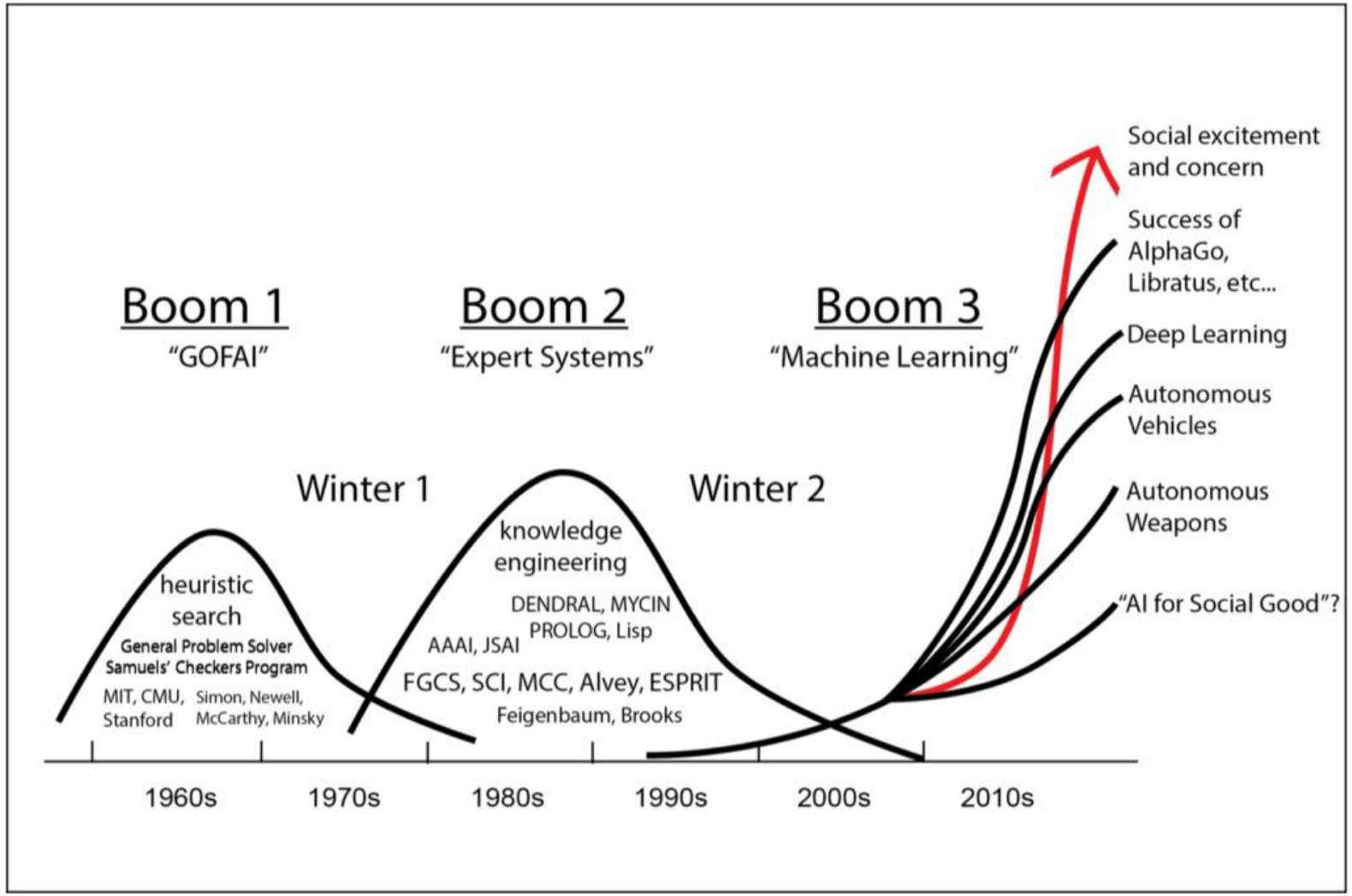
Large Language Models in Software Development

Ulf Leser

Artifical Intelligence and AI Winters



https://jaylatta.net/history-of-ai-from-winter-to-winter/

Language Models

- A language model solves the following task: Given a prefix of a text (sentence), predict the next word
  - For conversations? Prefix of conversation, e.g., question
- Auto-regressive, creating a text word-by-word
  - While prefix increases in length
- Requires obeying rules of syntax, semantics, pragmatics
  - Syntax: Das Fahrrad ist trinken
  - Semantics: Das Fahrrad geht nicht an
  - Pragmatics: The way how humans interact by language

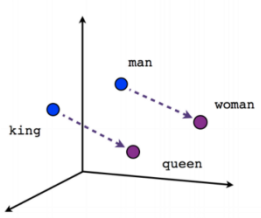
Restrictions of Early LMs

- Considered only (suffixes of) prefixes of 4-5 words
  - Counted relative probability of n-grams of size 5-6 in a large corpus
  - Then always choose the most frequent
  - Larger prefixes: Combinatorial explosion in number of n-grams required too large corpora
- Words were only equal or unequal
  - No consideration of word meaning: Buch != bücher, Arzt != Ärztin, Topf != fahren, …
  - Synonyms and homonyms
- No technical means to obtain and handle very large corpora

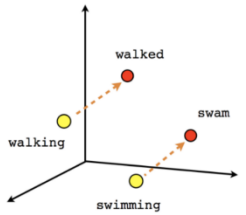
Things that Happened since then

- Web, web crawls, web tech companies: Extremely large corpora, extremely large funds, extremely large compute clusters
- GPUs: Ability to perform thousands of computations in parallel

- Word embeddings: Represent words as vectors such that semantically similar words have similar vectors

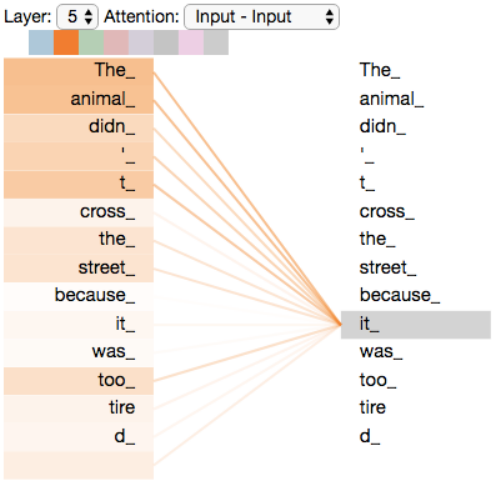


Male-Female



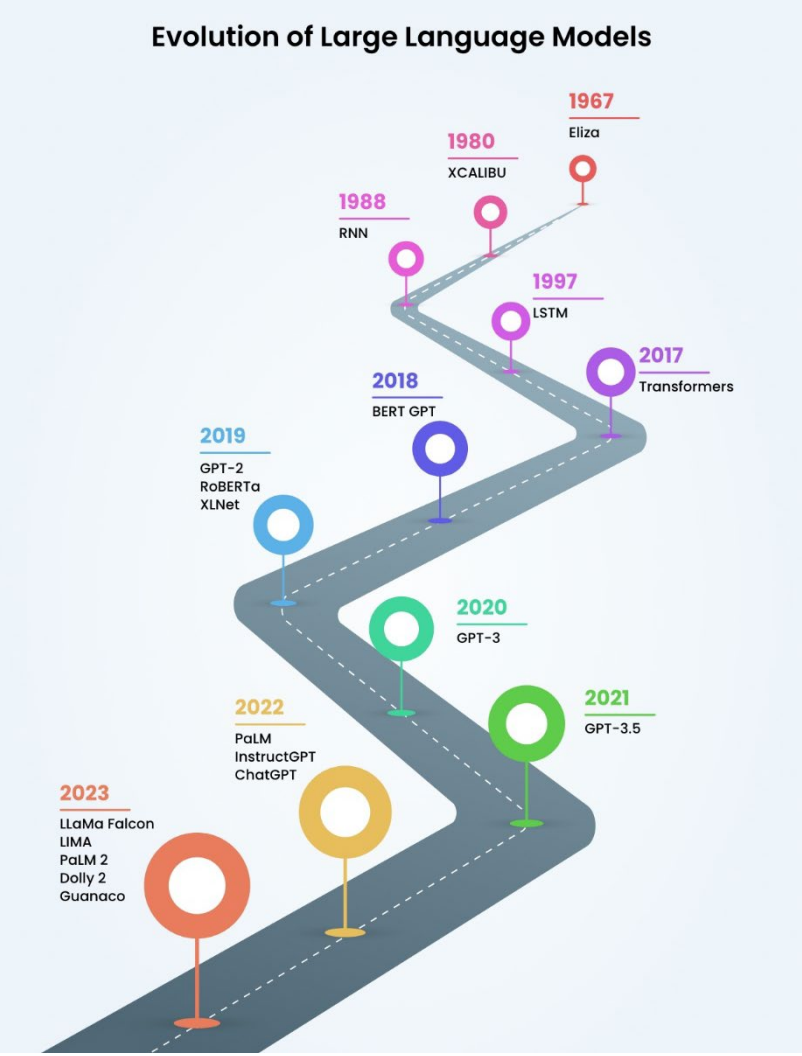
Verb tense

- Transformer: Learn to pay attention even to words far away in the prefix (in an efficient manner)

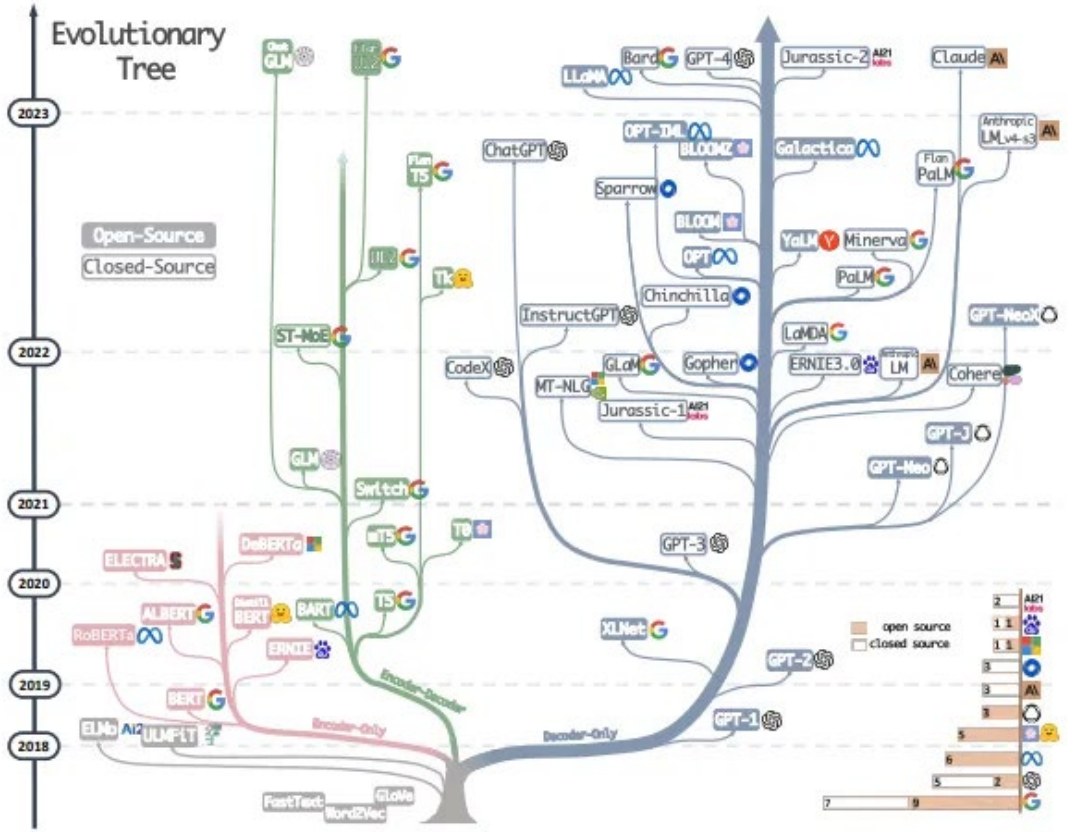


https://jalammar.github.io/illustrated-transformer/

Large Language Models



https://www.appypieagents.ai/blog/evolution-of-language-models



https://medium.com/@avijitsur_8327/the-evolution-of-large-language-models-from-gpt-to-multimodal-mastery-2e38b23cf1a3

Works Surprisingly Well

- Steepest increase in number of users ever
- People fall in love with LLMs
- Texts generated by LLM cannot be discerned from human texts anymore
- LLMs start to replace humans in many dialogue-oriented tasks
- Within 4 years, AI companies became among the most valuables in the world

- Everybody in this room used and uses it – more and more
- You will use LLM to find and summarize research papers
- You will be tempted to let an LLM write your seminar thesis
- You will be tempted to let an LLM design your experiment
  - And perform it?

You should!

You should not!

… also in Software Development

- Code is just text that can be created token by token
- LLMs integrated in most (all?) popular IDEs
- LLMs are said to have reached the level of junior software developers
- Number of job openings has decreased substantially
- AI skills are among the most wanted competencies in job ads

Seminar Motivation

- How good are LLMs for software development, really?
- How can LLMs be used in software development?
- Give you an overview on use cases and state-of-the-art

- Application focus: This is not a course on the technology behind LLMs

Seminar Idea

- We build groups of two
- You pick a topic from the list
  - Or you propose a new topic
- You research the topic, present it in talks, write a seminar thesis
- You design, perform and evaluate a (small) practical experiment
- You learn from other students presentations

Who should be here

- Monobachelor Informatik, IMP, Kombi-Bachelor
- Ability to read English papers
- Good knowledge in statistics, programming, software development processes
  - Some knowledge in NLP would be great as well
- Willingness and ability to work independently in teams
  - You develop the topic on your own
  - Questions always welcome

How it will work

- Today: Presentation and choice of topics per group
- 17.11.25: Send an outline of your topic (next slide)
- Before Christmas: Present your topic in 5min talk
- 31.01.26: Meet your advisor to discuss slides
- February: Present your topic in a Blockseminar (dates tba)
- 31.03.26: Write seminar thesis (~15 pages)

The outline

- Topics are very general
- Find yourself and read a set of suitable papers
  - A specific focus is allowed and welcome
  - Extract the most important information
- Write into an outline of your thesis
  - Abstract: Roughly 20 lines – what is the topic, what will the thesis describe?
  - Structure: Chapters and sections
  - 1-2 sentences per section to describe the content
  - List of app. 10 important references
    - Mark your top-3 references – those that most likely will form the basis of your work
- Experiment: One chapter must contain design of an experiment
  - Research hypothesis, data set, evaluation metrics, experiment design

The 5-min flash talk

- Focus on marketing – tell a good story to sell your topic well
  - What is the topic?
  - Why is it challenging?
  - Why is it cool?
  - What are important applications?
  - What will your talk be about?
- At most 5 slides
- Focus on figures & examples; omit details or algorithms

Presentation

- ~30min presentations
- German or English
- Explain topic, experiment, state-of-the-art, field coverage
- Most important goal: Audience should understand what you say

ToC

- Introduction
- Topics
- Assignment
- Hints on presenting your topic and writing your thesis

Topics

Topic	Assigned to
LLM for debugging	
LLM for testing	
LLM for program verification	
LLM for requirements engineering	
LLM for creating web-based systems	
LLM and software architectures	
LLM for performance optimization	
LLM for code integration	
LLM in SQL (text-to-code)	
LLM and scientific workflow systems	
LLM for documentatoin	
LLM for programming (auto-complete etc.)	

Exemplary Experiments

- Debugging: Choose a LLM, chose a set of programs (write some yourself), introduce errors (systematic / at random), ask LLM to correct, count fraction of detected / corrected bugs
- Requirements engineering: Choose LLM, create some user stories, ask LLM to generate models (e.g. UML) or code, count fraction of correct model elements / covered requirements

- The papers you will read will contain many experiments – copy designs
- No need for comparing 10 LLM, inter-annotator-agreements, five programming languages, extensive prompt engineering, hyper-parameter search …
- You may experiment with LLM-as-a-judge

Introductory Topics

- Finding and assessing scientific literature
- (Scientific presentations)
- Writing a (seminar) thesis