Understanding and Training Language Models: **Introduction**

Erik-Jan Senn

Faculty of Mathematics and Statistics, University of St. Gallen

CSH Autumn School at University of Hohenheim
September/October 2024

Roadmap

Hello

Course Structure

Language Modelling Pipeline

Introduction

Hello

Hello from you!

- Who are you?
- What do you want to learn here?
- How do you (want to) use Language Models?
- What experiences do you have in:
  - Statistical/machine learning
  - Language processing
  - Coding

Hello from me!

- Who are you?
  A PhD Candidate in Econometrics at the University of St. Gallen (Website)

- What do you want to learn here?
  Teach to learn, learn to teach. :)

- How do you (want to) use Language Models?
  Sentiment analysis for financial forecasting (tweets), accounting fraud detection (accounting filings)

- What experiences do you have in:
  - Statistical/machine learning *In my Master and PhD, I broadly speaking apply these tools to prediction problems in economics/finance.*

  - Language processing *I apply and work on LLMs in the context of research projects in sentiment analysis for finance.*

  - Coding *No a professional. Some years of experience in R/python mainly for data and modelling work. I write ugly code that sometimes works. No expert in torch / tensorflow.*

Introduction

Course Structure

Summary

What will you learn?

- understand and implement the components of the language modeling pipeline
- use and train foundation models
- apply and adapt pretrained models for specific supervised tasks (mainly sentiment analysis)

What do you have to do?

- attend
- run and write code (notebooks with exercises, project)
- participate actively (questions, comments)

Planned Schedule

Day 1: 30.09.2024, 9h00 - 17h30:

9h00 - 10h30	Lecture: Introduction, language modeling pipeline
10h30 - 11h00	Coffee Break
11h00 - 12h30	Lecture: Text preprocessing
12h30 - 13h30	Lunch Break
13h30 - 15h30	Exercise: Text preprocessing
15h30 - 16h00	Coffee Break
16h00 - 17h30	Lecture: Statistical Learning and Neural Networks

Planned Schedule

Day 2: 1.10.2024, 2024, 9h00 - 17h30:	
9h00 - 10h30	Lecture and Exercise: Training a neural network
10h30 - 11h00	Coffee Break
11h00 - 12h30	Lecture and Exercise: Self-attention and transformers
12h30 - 13h30	Lunch Break
13h30 - 15h30	Lecture: Foundation models
15h30 - 16h00	Coffee Break
16h00 - 17h30	Exercise: Training foundation models

Planned Schedule

Day 3: 2.10.2024, 2024, 9h00 - 17h30:

9h00 - 10h30	Lecture: Task-specific learning and fine-tuning
10h30 - 11h00	Coffee Break
11h00 - 12h30	Exercise: Task-specific learning and fine-tuning
12h30 - 13h30	Lunch Break
13h30 - 15h30	Exercise: Sentiment analysis project
15h30 - 16h00	Coffee Break
16h00 - 17h30	Lecture and Exercise: Generative language models, conclusion

Resources

1. Course Github https://aidaho-edu.uni-hohenheim.de/gitlab/eriksenn/llm_class_public. Copy/fork under different name. Pull daily to get updated material.
   - Slides
   - Notebooks (including exercises)
   - Data

2. JupyterHub Hohenheim https://aidaho-edu.uni-hohenheim.de/CSHAS2024. Our computing environment. Use GPUs only when necessary. Make sure you can access it and run code.

3. Main Reference: Book and codes for ”Build A Large Language Model (From Scratch)” by Sebastian Raschka. Codes (open source), Book (to purchase, not required) (Raschka, 2024), Figures.

4. Other nice references (that I partly rely on):
   - Deep learning class by Alfredo Canziani course github
   - Notebooks for neural networks from my great colleague Jonathan Chassot from this bachelor course
   - Deep learning notebooks by Andrej Karpathy github

Introduction

The Language Modelling Pipeline

Key Features of Text Data

Add examples to those features

Nice!

- Text follows certain syntactic rules and a hirarchical structure.
- There is lots of text availabe!
- Sequential / time series // context dependence: meaning of sentences / words

Not so nice!

- Contextual meaning and Ambiguity of text: e.g. meaning of words depends on sourounding words, and slight changes in wording or word order can change entire meaning of a document.
- Sparsity: many words/sequences appear very infrequently (and some very often)
- Language is domain-specific: different characters, words and rules in different languages, industries, text forms.

XX remove later: text characteirstics related to modelling choices

Always link back modelling choices to these characteristics.
Tokenzier: related to sparsity: tradeoff of being exact (different words have different meanings) vs sufficient data to understand the meaning of a word).
Token embeddings: sparsity
Pos embeddings and entire LM architecture: contextual meaning of text, ambiguity.
Try to use learn rules and exploit time series structure from the many texts that are available for this part
E-g- fine-tuning: best models are the ones estimated on very similar texts to what you want to do -¿ new tokens, restimated weights, ..

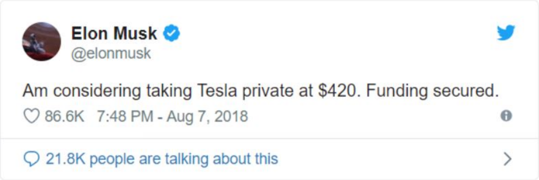
Natural Language Processing (NLP)

1. Humans communicate using a “natural language” (e.g. English), but machines (computers) use numbers.

2. Lots of relevant information exists (and is stored) in the form of human language, not numbers.

Aim of NLP: Enable machines (computers) to understand and communicate in human language.

Example Tasks in NLP



Elon Musk
@elonmusk

Am considering taking Tesla private at $420. Funding secured.

♡ 86.6K 7:48 PM - Aug 7, 2018
21.8K people are talking about this

Tweet of Elon Musk, 2018

Example NLP tasks in increasing order of difficulty:

- Does the tweet meet the (old) 140 character limit on twitter?
  Yes/No 1/0
- What is the sentiment of the tweet towards Tesla?
  Positive/Neutral/Negative 1/0/-1
- Analyze why this tweet might be related to stock market manipulation.
  {Text output}

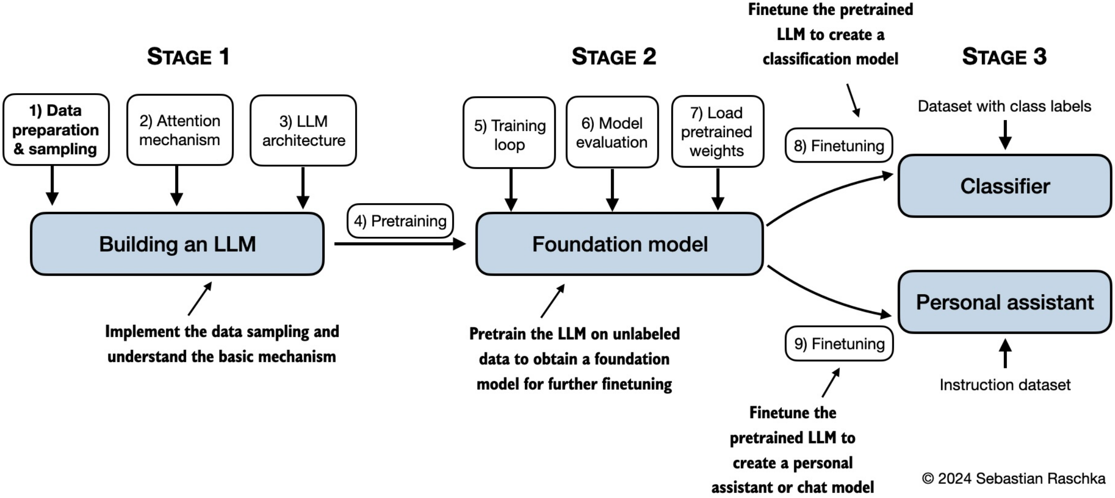
Approaches to NLP

- Symbolic/Rule-based NLP: Process human language by pre-defined rules.
  - Example 1: Identify nouns as words that follow an *the* or *a*.
  - Example 2: Use pattern matching (regular expressions) to identify dates.
- Statistical/machine-learning-based NLP: Learn human language from data.
  - *Simple* statistical models; e.g. based on word-counting (bag-of-words), Word2Vec, ...
  - Neural (deep learning based) models: (Large) Language Models (LM/LLMs)

Large Language models (LLMs)

What is part of a model: architecture, parameters, (data).
Examples of the models:
Whats large? Parameter counts, data used to train them
Capabilities of the model: Show Chatgpt example?!

The Language Modelling Pipeline



STAGE 1

1) Data preparation & sampling

2) Attention mechanism

3) LLM architecture

Building an LLM

Implement the data sampling and understand the basic mechanism

4) Pretraining

STAGE 2

5) Training loop

6) Model evaluation

7) Load pretrained weights

Foundation model

Pretrain the LLM on unlabeled data to obtain a foundation model for further finetuning

Finetune the pretrained LLM to create a classification model

8) Finetuning

9) Finetuning

Finetune the pretrained LLM to create a personal assistant or chat model

STAGE 3

Dataset with class labels

Classifier

Personal assistant

Instruction dataset

© 2024 Sebastian Raschka

Language Modelling Pipeline. Source

Disclamer

- We focus on basic concepts that appear in most LLMs to improve understanding. However, every LLM is different and we cannot tackle the newest developments in the field.

- The LLM literature follows using the principle: “Whatever works in practice is good!” (My personal opinion.)
  We will not focus much on the “Why”, because often there is no theory-guided answer.

- Mathematical rigour is often sacrificed/ignored for intuition. Sorry!

- Creating well-performing LLMs requires
  (i) a strong model architecture,
  (ii) large amounts of data,
  (iii) and lots of computing power.
  Compared to big-tech companies, we sadly lack (ii) data and (iii) computing resources. Therefore, in parallel to learning how to build sub-components of an LLM ourselves, we will also learn how to use pretrained LLM components for better performance.

Questions ?

References

Sebastian Raschka. *Build A Large Language Model (From Scratch)*. Manning, 2024. ISBN 978-1633437166. URL https://www.manning.com/books/build-a-large-language-model-from-scratch.