Day 10: Language Module Introduction

Order of Topics

1. Overview of AI
2. Intro to Natural Language Processing
3. Quantifying Language
4. Word Embeddings
5. Recurrent Neural Networks (RNN)
6. Attention and Transformers

Capstone: Semantic Search

e.g.: input: “A person standing outdoors” → math model → database

- Multimodel Data
    - Images and text
    - Train your own descriptions with your own neural network (NN)

Overview of AI

- Definition: Artificial Intelligence: The ability of a machine to perform technical tasks normally performed by a human
- Limitations
    - Cant just be any type of machine
    - Technical
        - Image Generation?
        - Songwriting?

    - Understanding of human emotions/norms
    - “Tasks normally performed by a human”
- Russel and Norvig ~ *Artificial Intelligence: A Modern Approach*
    - Cognitive Modeling
        - Experimental Psychology
    - Laws of Thought
        - Formal Logic
    - Turing Test
        - Can a human distinguish the ageent from another human?
    - Rational Agent
        - In pursuit of goals in a rational/optimal manner

- Supervised Learning (data → labels, e.g. face recognition)
- Unsupervised Learning (data w/o labels, e.g. face clustering)
- Self-supervised (data → labels (data), e.g. text completion)

Timeline:

- (1956) Dartmouth AI Conference
- (1997) Deep Blue defeats Kasparov in chess
- (2002) Roomba cleans living rooms
- (2004) Spirit and Opportunity autonomously navigate Mars
- (2007) DARPA Urban Challenge
- (2011) Watson (AI) defeats Jennings in Jeopardy
- (2014) Eugene passes Turing Test
- (2016) AlphaGo defeats Lee in Go
- (2019) AlphaStar defeats TLO and MaNa in SC2
- (2022) GPT-3 defeats the school essay

- (2023) GPT-4 defeats the SATs, the bar exams, wine-tasting exams, most APs, etc.

New Turing Test

- Go make me a coffee
- Go assemble this BILLY bookcase

Natural Language Processing

- A set of techniques for making predictions or decisions about language (by a machine); e.g.
  - Q+A
  - Image captioning
  - Translation
  - Sentiment Analysis
  - Named entity recognition
  - Grammar corrections
  - Smart assistant
  - Content generation
  - Spam detection
  - Automated phone trees
  - Text summarization
- Language data (text) is plentiful, easily accessible
- Analyzing Language
  - e.g.: “A dog chasing a boy on the playground.”
    1. A sequence of characters
    2. A sequence of words
    3. A sequence of parts of speech (determiners, nouns, verbs, etc.)

4. A sequence of constituents of a sentence (noun-phrase, verb-phrase)
5. Subject-object relations (dog (animal) → chases → boy (human))

- `n-gram`: a sequence of n "chunks" of information
  - notation
    - $P(A)$ ~ probability of event A
    - $P(A, B)$ ~ probability of A and B
    - $P(A|B)$ ~ probability of A given B
    - $P(C|A, B)$ ~ probability of C given that A and B are true
    - Modeling Language
      - $P(x_n|x_{n-1}, x_{n-2}, ..., x_1)$
      - What is the probability of some x_n being the nth word given that we saw the words $x_{n-1}...x_1$ before it?
      - Example: n-gram: 5-gram
        - $P(x_n|$"the dog ate my"$)$ → $P(x_n|$"my", "ate", "dog", "the"$)$
          - $P($"homework"$|...)$
          - $P($"dinner"$|...)$
          - $P($"of"$|...)$ - low probability
          - $P($"however"$|...)$ - low probability
      - Example:
        - corpus (a body of text that we are examining/training on): "the quick black cat raced the slow black lab"
        - 2-gram: $P($"cat"$|$"black"$)$ = $P($"black cat"$)/P($"black"$) = 1/2$
        - 3-gram:
          - $P($"cat"$|$"quick black"$) = P($"quick black cat$)/P($"quick black"$) = 1/1 = 1$
          - $P($"cat"$|$"slow black"$) = P($"slow black cat"$)/P($"slow black"$) = 0/1 = 0$

- Applications:
  - autocomplete
  - language ID
  - language translation
- Markov Property ~ evolution of the Markov process in the future depends only on the present state and does not depend on past history
  - E.g.: “the quick black cat chased” - $P(\text{"quick"}|\text{"the"}) * P(\text{"black"}|\text{"quick"})...(\text{"chased"}|\text{"cat"})$

Generative AI

- `n-gram`: 4
  1. $argmax_x P(x|?,?,?)_{\rightarrow}???$ the
  2. $argmax_x P(x|?,?,the)_{\rightarrow}??$ the black
  3. $argmax_x P(x|?,black,the)_{\rightarrow}?$ the black cat

Modelling Documents

- How can we build a numerical model for a document
  - Clustering
  - Classification
  - E.g.
    - images → pixels → arrays (CNNs → spacial inductive bias)
    - documents → words? sentences? → ?
  - Bag of Words model
    - Represent documents as an inventory of words
    - Counting word occurrences
    - Ignore word order

- Example:
  - `doc_0`: `"I am a dog. A dog am I."`
  - `doc_1`: `"I am a cat."`

vocab	index
a	0
am	1
cat	2
dog	3
I	4

  - Word Frequency
    - `doc_0`: `[1 2 0 2 2]`
    - `doc_1`: `[1 1 1 0 1]`

- Term freqeuncy, inverse document-frequency encodings
  - General flow
    - Corpus → remove “stop words” (e.g: `[”a”, “or”, “the”…]`) → retain only the top k most common words across documents → final vocab
  - Term-frequency vector
    - For document `d` and term `t`:
      - $f_t^{(d)} = \frac{C_t^{(d)}}{\sum_{t \in vocab} C_t^{(d)}}$
      - * Frequency is normalized so document length does not matter
      - `doc_0`: `[1 2 0 2 2]` → `1/7 2/7 0 2/7 2/7]`
      - `doc_1`: `[1 1 1 0 1]` → `[1/4 1/4 1/4 0 1/4]`
  - Inverse document frequency:
    - Measured across documents
    - IDF for term $t = log_{10} \frac{N_{doc}}{n_t}$ where N_{doc} is the total number of documents and n_t is the number of documents containing term t

- Example:
    - “I am a dog. A dog am I”
    - “I am a cat”

  $\rightarrow log_{10}[\frac{2}{2}, \frac{2}{2}, \frac{2}{1}, \frac{2}{1}, \frac{2}{2}]$

- TF-IDF: $f_t^{(d)} log_{10} \frac{N_{doc}}{n_t}$
    - for document d : $[f_0 log_{10} \frac{N_{doc}}{n_0}, f_1 log_{10} \frac{N_{doc}}{n_1}, ..., f_{n-1} log_{10} \frac{N_{doc}}{n_{n-1}}]$