NTNU

Norwegian University of Science and Technology

TDT4310, Lab session 2

Back to basics: Tokenization, language modeling and word representations

Tollef Jørgensen, January 30, 2024.

Outline

- Tokenization
- Some basics of language modeling
- Word representations

The assignments and blackboard

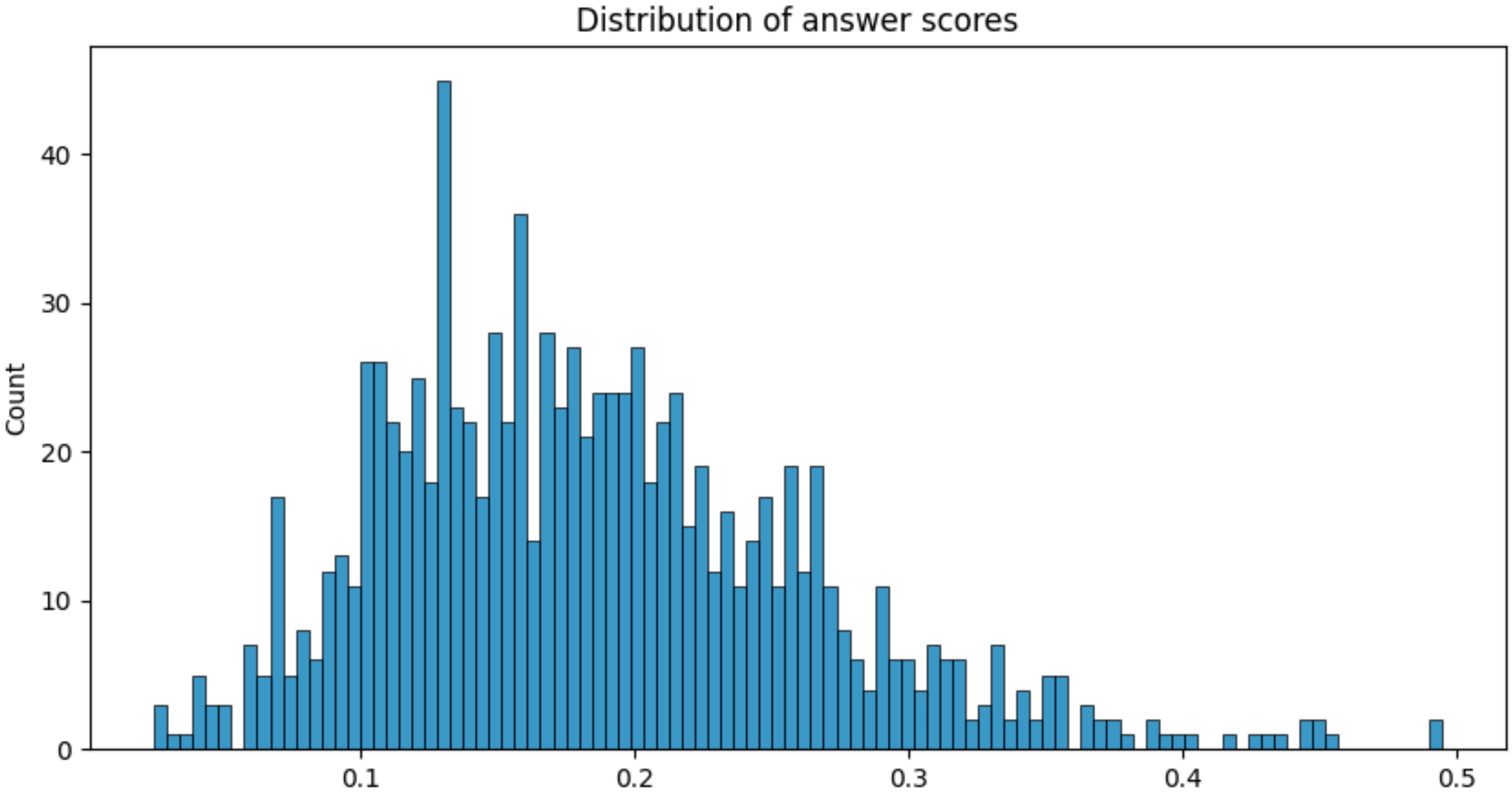
- Information about the course and assignments:
  - **Please mainly use the blackboard forum**
    - Can post anonymously as well
- Sensitive information/other questions:
  - Emails (**please add [TDT4310] in the subject**)
    - If related to an assignment (sick leave, etc.)
      - Me
    - Else:
      - Björn/me + CC TA's

Reference group

- PIs
- Probably only 2 meetings total! There will be cookies

- Talk to me or email gamback@ntnu.no / tollef.jorgensen@ntnu.no :-)

LLM usage detection for Lab 1



LLM usage detection for Lab 1

- Approach: gather outputs from 6 models for the questions given in the Lab, then rank each answer from students on the following:
  - Sentence embeddings from BERT models to create a single vector for sentence representations (**related to this lab!**)
    - Compute cosine similarity between the LLM answer and your answers
  - Weighted by Jaccard similarity (set of overlapping words)
- Results: none seemed to rely solely on LLMs, although some answers were definitely *inspired*

Tokenization

- The task of splitting a text into smaller pieces (tokens)
- Can be words, characters, subwords, ...
- Some methods
  - Whitespace
  - Regex (e.g., match URLs, hashtags)
  - Rules (such as entities, e.g. "New York")
  - Contractions (don't/do n't/don 't/don ' t)
  - Expressions (multi-word exp. tokenizer)
  - ... There's a whole lot

Tokenization

- But limiting tokenization to just words is also problematic.
- What happens when you observe an out-of-domain token?
  - Your system should be able to handle *any* input

Tokenization

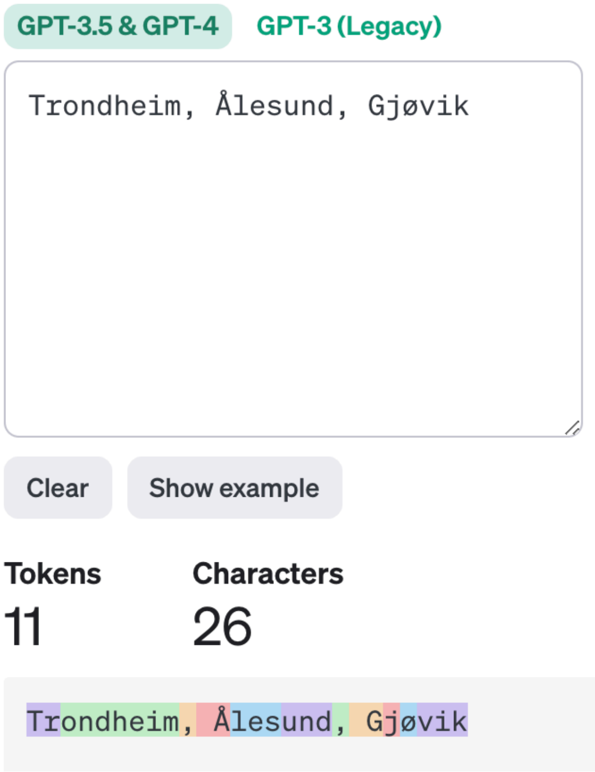
- Sentence: "cats and rats"

- Byte pair encoding
    - Merge from tokenization to commonly observed subword units
        - Better suited for large corpora
    - Add frequent pairs (for a certain amount of iterations)
        - c+a: 1, a+t: 2, t+s: 2, ... => c + ats, and, r + ats
- WordPiece (used to pre-train BERT)
    - gather vocabulary (characters)
    - split on units and add a special ## prefix for joined tokens
    - "cats" -> "c","##a", "##t", "##s". Calculate the frequency of each pair
            - here the frequency of ##t and ##s would be 2, and receive a higher score
                - merge and repeat: "c" "##a" "ts"

Tokenization

- Going back to LLMs and *context length*
  - *number of tokens in context*

- If a word is commonly used, such as "cat", this is likely one token.
  - But what about other languages and rarely occurring strings?
    - A single japanese character (such as cat, 猫) counts as 3 tokens!



Tokenization - Implementations in NLTK

- https://www.nltk.org/api/nltk.tokenize
- Chapter 3: https://www.nltk.org/book/ch03.html

Language modeling

- Not as fancy as what we explored the first time around...

- Focused on n-grams and frequency distributions
  - bigrams, trigrams, ...
- The point is to get you to think about how can we use these simple methods to model language

Language modeling

- Using co-occurrences in text will always have limitations
  - No context
  - Ambiguous words
    - running a marathon
    - the ink is running
    - she's running the show
- Given what you will hopefully learn about word representations, we can learn how these words appear together!

Word Representations

- Much higher flexibility than working with tokens of words directly
- Encodes some form of representation for each word
- Simple examples:
  - One-hot encoding
  - Bag-of-words (BoW)
    - Mapping of word:count
  - The above are *sparse* representations. They don't scale well with infinitely many words!

```
word_vector_map = {
    "cat": [0.1, 0.2, 0.3, 0.4],
    "dog": [0.2, 0.3, 0.4, 0.5],
    "mouse": [0.3, 0.4, 0.5, 0.6],
    ...
}
```

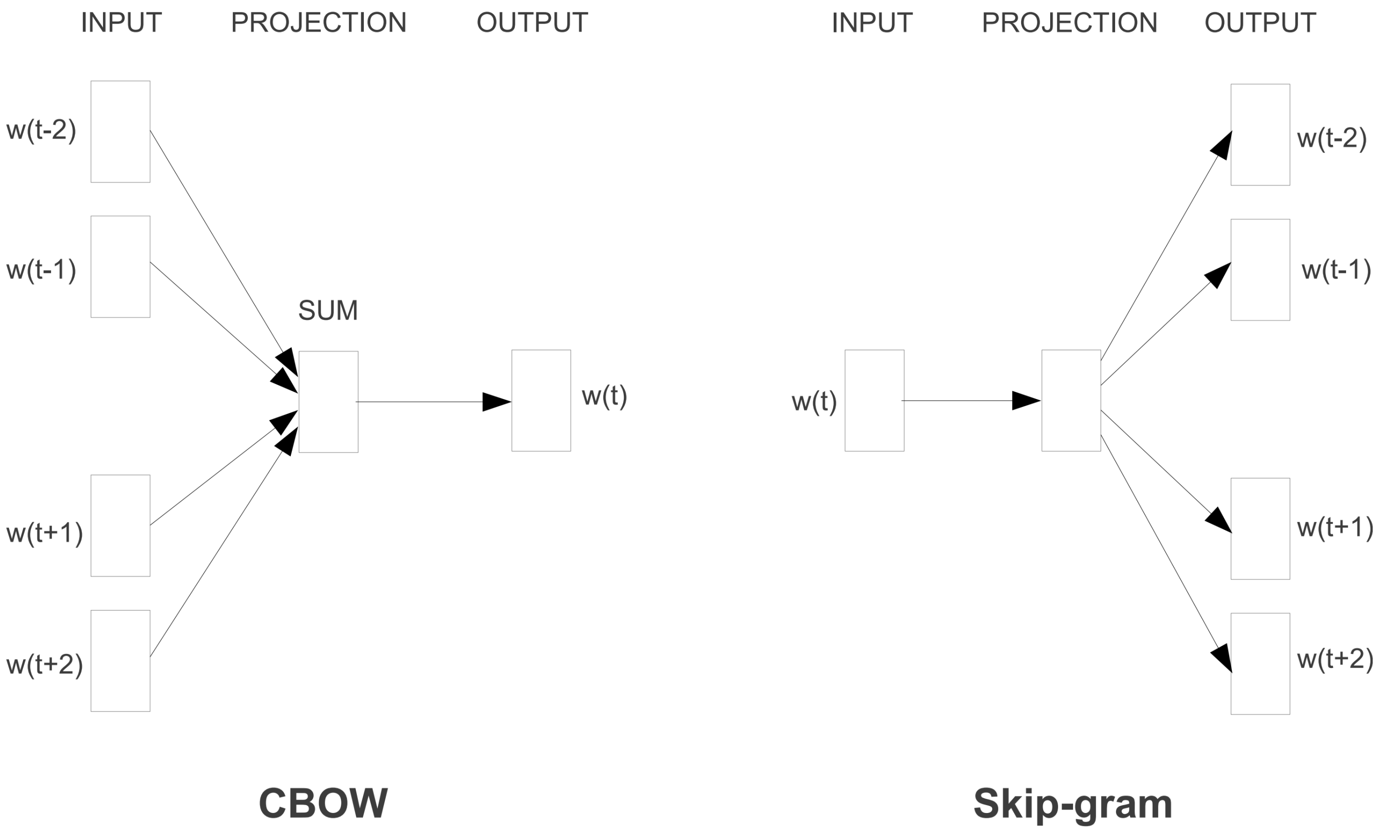
$$vec_{cat} = [1, 0, 0]$$

$$vec_{dog} = [0, 1, 0]$$

$$vec_{mouse} = [0, 0, 1]$$

Word Representations

- Distributional hypothesis
  - *You shall know a word by the company it keeps (Firth, 1957)*
- Heavily represented in word embeddings
  - E.g., Word2Vec
- Two separate objectives (word2vec)
  - Continuous bag of words
    - Given a context, predict a word
  - Skip-gram
    - Given a word, predict context
- Suggested: https://arxiv.org/pdf/1301.3781.pdf


INPUT
PROJECTION
OUTPUT
w(t-2)
w(t-1)
SUM
w(t)
w(t+1)
w(t+2)
CBOW
INPUT
PROJECTION
OUTPUT
w(t)
w(t-2)
w(t-1)
w(t+1)
w(t+2)
Skip-gram