Representation Models

HSS 510 / DS 518: NLP for HSS

Taegyoon Kim

May 13, 2025

Agenda

Things to be covered

- Static embeddings vs. contextual embeddings
- FNN
- Attention/transformer
- BERT
- BERTopic

Static Embeddings

From contexts but not contextual

- Word2Vec, GloVe, Fast2Text
  - A word’s embedding is derived from its context
- Different contexts do *not* lead to different embeddings
  → *static* embeddings
- Even if a word has the same form, its meaning differs depending on the context

Contextual Embeddings

Example

- Embeddings for the same word differ by context
- Sentence 1: *Open a* bank *account*
  $\rightarrow e_{bank_{s1}} : [0.3, 0.9, \ldots]$
- Sentence 2: *On the river* bank
  $\rightarrow e_{bank_{s2}} : [0.8, 0.1, \ldots]$

Contextual Information in Neural Networks

Recent advances in NLP leverage neural networks to generate contextual representations of language

- Early models: RNNs, LSTMs, ULMFiT, ELMo
- Transformer-based models: BERT, GPT, T5, LLaMA, etc.
  - Use attention mechanisms to model complex contextual relationships
  - More effectively and scalable

Contextual Information in Neural Networks

Broadly, we can distinguish between two types of transformer-based models

- Representation models (e.g., BERT): trained to understand language by encoding input into rich, context-aware embeddings
  - Used primarily for producing representations that are useful for downstream tasks (e.g., topic modeling, classification)
- Generative models (e.g., GPT): trained to generate coherent text sequences by predicting the next word in context
  - These models learn representations as a means to an end (generation)
- Today, we will focus on BERT, one of the most influential representation models in NLP

Attention and Transformer

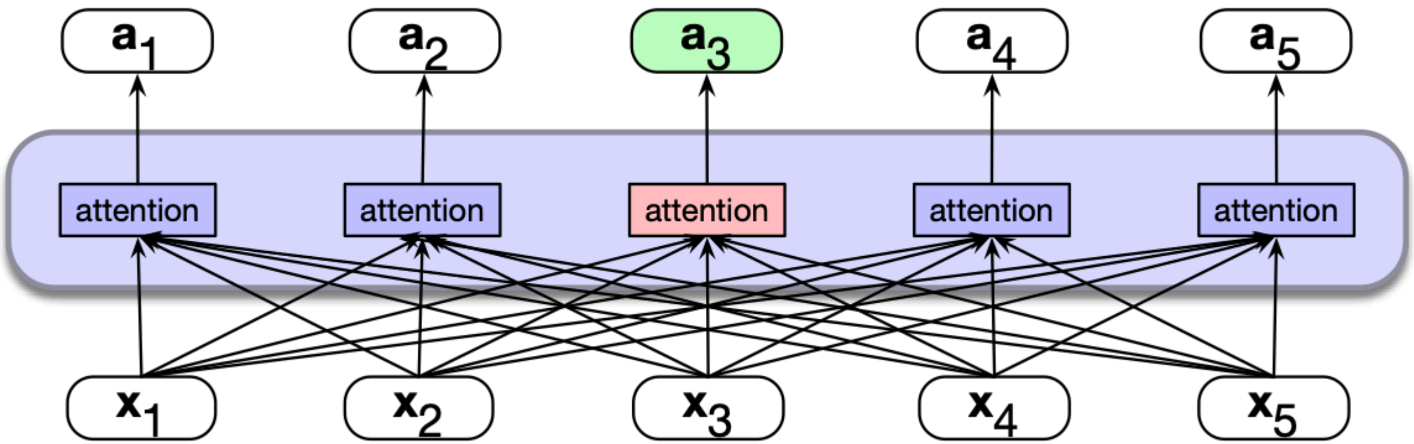
Self-attention

- The primary goal of self-attention is to generate the representations of the tokens in a sequence
- Self-attention allows each word of the input sequence to 'attend' to (or reference) all other parts of the sequence (including self)
- In simple terms, attention is quantified as weights that indicate how much focus should be put on other tokens of the sequence in generating the representation of a given token
- E.g., "I took an `umbrella` from the old cabinet because it was `raining`"

Attention and Transformer

Self-attention

- Each token's representation is updated based on its relationship to all other tokens in the sequence
- Attention computes a weighted sum of the other token vectors, where the weights indicate relevance



Attention and Transformer

Self-attention

$$a_i = \sum_{j=1}^{n} \alpha_{ij} \cdot x_j$$

Where:

- a_i is the updated representation for token i
- x_j is the input representation of token j
- α_{ij} is the attention weight from token i to token j

Attention and Transformer

Self-attention

- The weights are computed using similarity scores between tokens
- The dot product is used to measure how similar tokens i and j are

$$\text{score}(x_i, x_j) = x_i \cdot x_j$$

- Normalize these scores across all tokens using softmax
- This results in a probability distribution over all tokens j, indicating how much attention token i pays to each

Attention and Transformer

Self-attention

- This is a simplified illustration of self-attention
- In practice, transformer models (including generative models as well as representation models) use attention heads (Query, Key, and Value (QKV) projections)

Attention and Transformer

Another example

- *The chicken didn't cross the road because* `it` *was too tired.*
- *The chicken didn't cross the road because* `it` *was too wide.*

$e_{it_{s1}}$ and $e_{it_{s2}}$ will have different representations after passing through attention layers

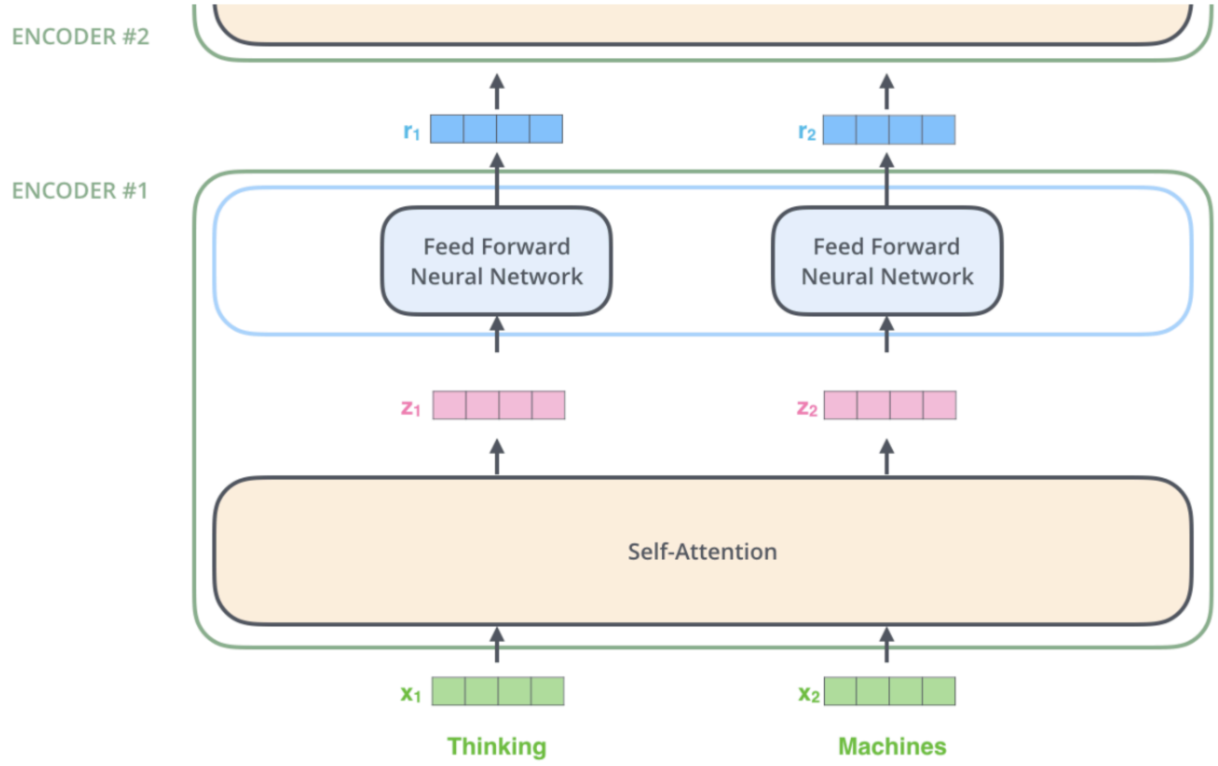
Attention and Transformer

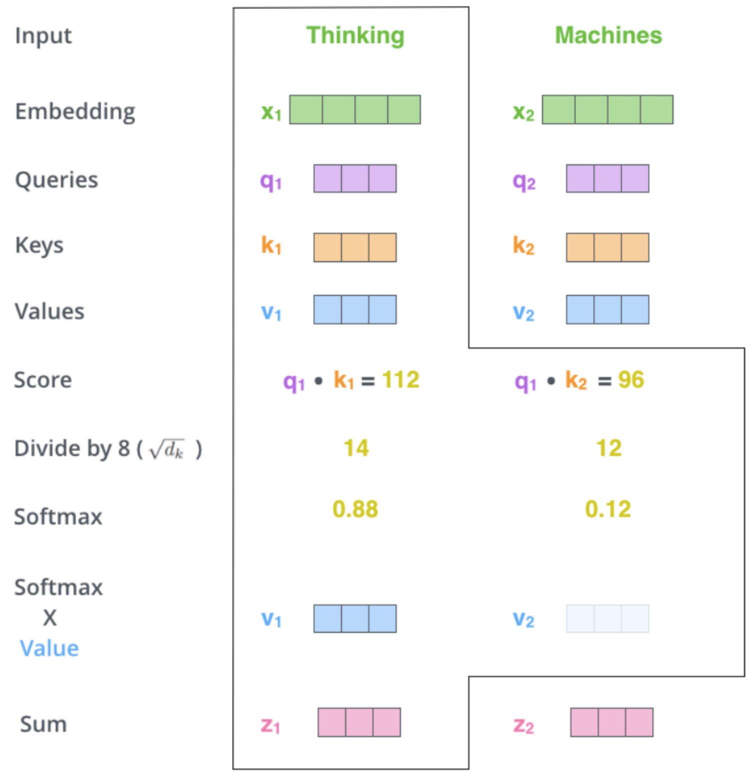
What is Transformer

- Transformer is a type of neural network architecture with self-attention mechanisms
  - Introduced in 2017: Vaswani et al. (2017)
  - Cited 179,358 times as of May 12 2025
- Transformers consist of a stack of (encoder/decoder) layers main consisting of self-attention and feed forward neural networks

Attention and Transformer

Transformer encoder





BERT

Bi-directional **E**ncoder **R**epresentations from **T**ransformers

- A form of transformer trained as a language model
  - Two tasks: masked token prediction, next sentence prediction
  - Introduced in 2019: Devlin et al. (2018)
  - Cited 130,271 times as of May 12 2025
- Pre-trained on huge data sets (BookCorpus and Wikipedia)
- Two versions of the model introduced in the paper
  - BERT BASE (12 layers)
  - BERT LARGE (24 layers)

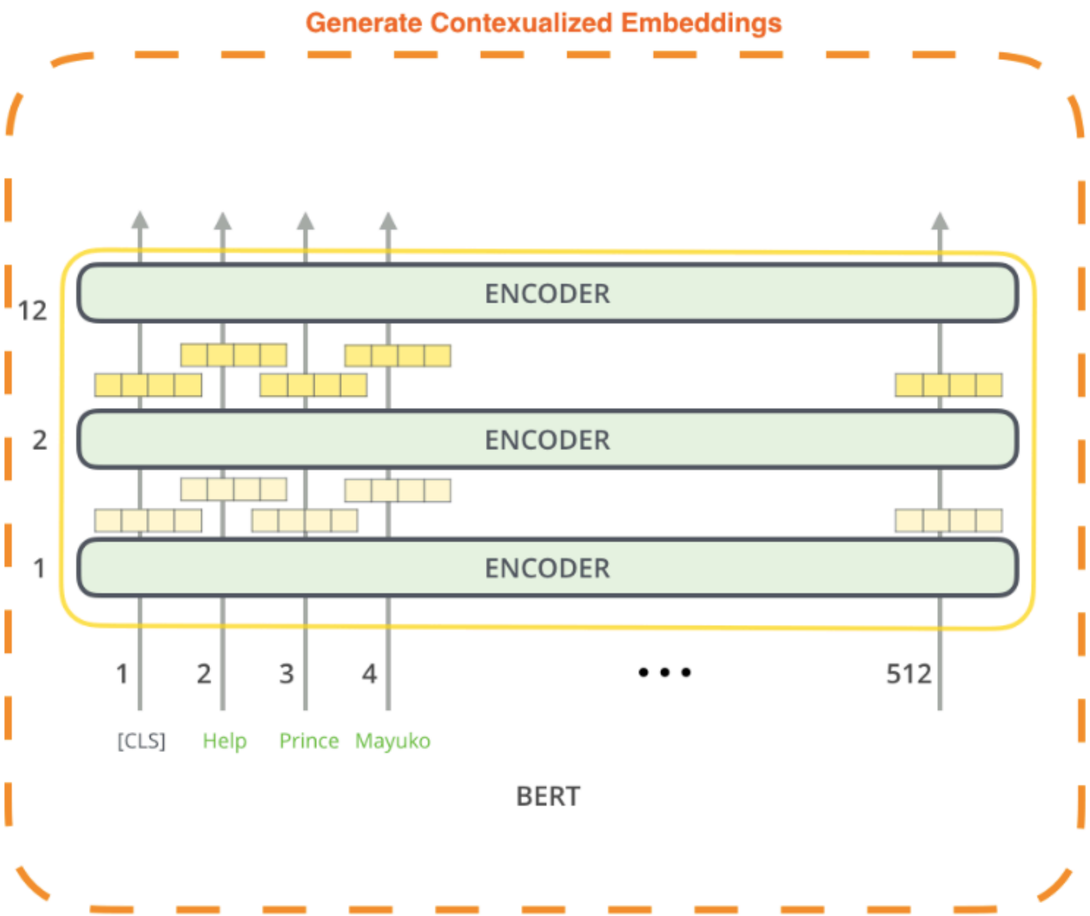
BERT

Peak at under the hood

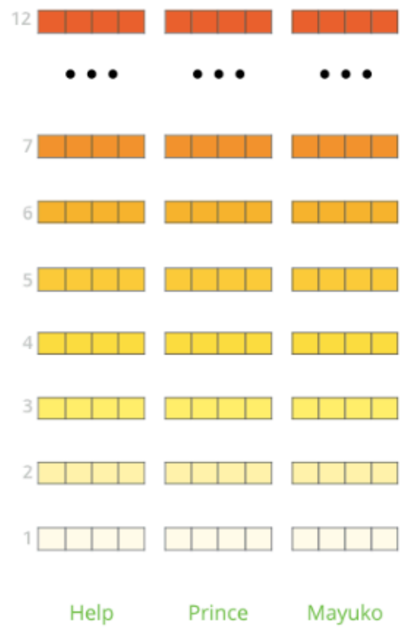
- BERT generates its own embeddings (from scratch) as part of its training process
- Stack of Transformer encoder layers involving “multi-head” self-attention
- Each layer passes its results through a feed-forward network, and then hands it off to the next encoder in the stack
- Each position outputs a vector of size 768 (BASE) or 1024 (LARGE)

BERT

Extracting embeddings



The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

Topic Modeling with Contextual Embeddings

(Some of) practical usage in applied research

- Fine-turning for text classification
  - E.g., idenfiying statements and stance on immigration (p. 7 *Classification* in Card et al. 2022)
- The pre-trained model (and embeddings) is of interest
  - E.g., identifying dehumanizing metaphors against immigrants (p. 8 *Measuring Dehumanization* in Card et al. 2022)
- Topic models
  - E.g., BERTopic

Topic Modeling with Contextual Embeddings

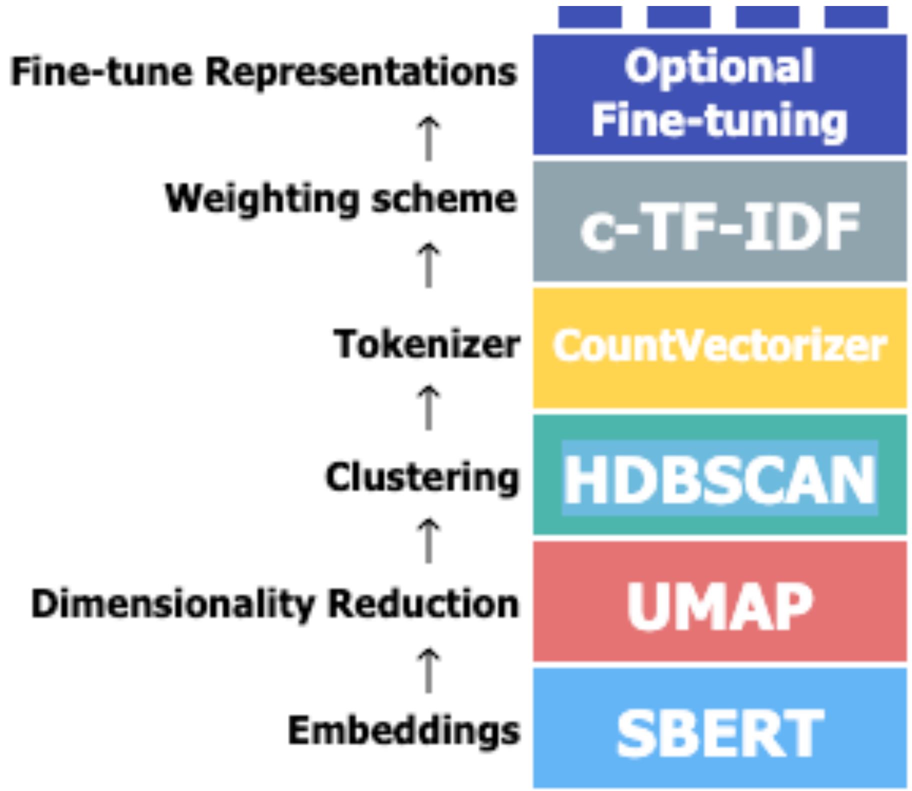
Drawbacks for conventional models

- LDA, CTM, STM, DTM, etc.
- These are bag-of-words models
- Lack of capturing contextual information

Topic Modeling with Contextual Embeddings

Major steps in BERTopic

- Document embeddings: SBERT
- Dimensionality reduction: UMAP
- Clustering: HDBSCAN
- Topic representations: class-based TF-IDF



Topic Modeling with Contextual Embeddings

Characteristics of BERTopic

- Assumes no generative process
  - Unlike LDA, CTM, STM, etc.
- Semantically rich: cab be good for short texts
- With embeddings form proprietary models, it is not quite transparent
- Guide coding involves detailed discussions of BERTopic in general and its modules, but ***is modularity good?***

Summary

- Modern NLP models based on transformer layers excel at capturing contextual information
    - Representations models are designed to understand and represent language
    - Generative models aim to generate coherent language
- Transformer layers powered by self-attention mechanisms form the core of these models’ effectivenes
- Representation models are versatile
    - The embeddings themselves are valuable outputs
    - Useful for modeling the thematic structure of a corpus (i.e., topic modeling)
    - Applicable to downstream tasks like classification

Guided Coding

- LDA basics: BERTopic overview
- LDA and BERTopic (Week 12): LDA/BERTopic on Stack Overflow data in Python