Knowledge Sharing Session

14th March 2025 – 11:00 to 11:30 am

SENTENCE TRANSFORMERS

I5185

Varun Athithiya Shenbagaraj

Self-Introduction

- **2015** – Bachelor’s in Civil engineering
- **2017** – Masters in Structural Engineering
- **Till late 2021** – National Mission for Clean Ganga, Varanasi, Uttar Pradesh
- **2021 – May 2022** – Study Break
- **May 2022** – Junior Data Engineer, Indium Software
- ETL migration, Data warehousing – Python, Databricks, DBT, AWS QuickSight, SQL

History

- **1999** – Latent Semantic Analysis – Dimensionality reduction technique – SVD. (Latent – Hidden)
- **2003** – Latent Dirichlet Allocation – Probabilistic generative model. Models docs as a mix of topics based on word occurrence.
- **2013** – Word2Vec (CBOW & Skip-gram) - *First widely used neural word embedding model*
- **2014** – GloVe (Global Vectors for Word Representation)
- **2014** – Paragraph Vectors(Doc2Vec)
- **2015** – Skip-Thought Vectors – *first neural network sentence embedding model*
- **2016** – FastText (by Facebook AI)
- **2017** – Paper titled "Attention Is All You Need", Google

- - - Word level to contextual sentence-level representations - - -

- **2018** – Google – BERT – Bidirectional Encoder Representations from Transformers
- **2019** – UKP Lab (TU Darmstadt, Germany) => SBERT (faster than BERT) using cosine similarity
- SBERT => **Sentence-Transformers**

Transformer

- **A deep learning model architecture** that revolutionized NLP and is the foundation of BERT, GPT, T5
- **Before transformers** – Neural networks – sequential processing – Slow
- **After transformers** – Parallel processing by self-attention – Speed + Accuracy

Architecture

- **Self-Attention Mechanism** – focus is on important words
- **Positional Encoding** – no recurrence (like RNNs) or convolution (like CNNs) =>to comprehend word order (governed by grammar)
- **Multi-Head Attention** – parallel processing (Q, K, V). Each token computes attention scores with every other token. (subject-verb, object-adjective). Output is finally combined.
- **Feed-Forward Neural Networks (FFN)** – refines understanding using traditional RNN after attention is applied
- **Encoder-Decoder Structure** – GPT (decoder), BERT (encoder)
  - Encode the input sentence and output the similarity score
  - Decoder is used in generative tasks.

Sentence Transformer

- Search engine query – “How to cook pasta?”
- **Traditional search engines** –
  - Results are based on the number of matching tokens (“cook”, “pasta”)
- **Smart system** –
  - “Easy spaghetti recipe”
- Sentence transformers can figure out that both the queries are related to some degree.
- Input => convert the input into a vector (direction + magnitude)
- **Cosine similarity score = (-1, 1)**
  - Cosine – adjacent / hypotenuse
  - Only direction
  - Angle subtended reflects the adjacency between two entities

Difference (Transformer vs S-Transformer)

Feature	Transformer (BERT, GPT, etc.)	Sentence Transformer (SBERT, etc.)
Processing Unit	Works at **token level**	Works at **sentence level**
Embedding Type	Contextual word embeddings	Single **sentence embedding**
Pooling	No built-in pooling	Uses Mean/Max/CLS pooling
Comparison	Requires extra steps for similarity	Direct cosine similarity
Efficiency	Slow for retrieval tasks	Fast for large-scale comparisons
Use Case	Language modeling, NER, QA	Search, clustering, similarity

- Both architectures understand context.
- Transformer – context of individual words
- Sentence transformer – comprehend the relationship between words.
    - e.g., I went to the bank on a weekend <u>to deposit money (example of self attention)</u>

Metrics

Metric	Works Best For	Cons
Cosine Similarity	NLP, embeddings, search	Needs normalized vectors
Euclidean Distance	Clustering, similarity detection	Affected by magnitude
Manhattan Distance	Sparse data, feature selection	Less effective for dense embeddings
Jaccard Similarity	Keyword matching, sparse text	Not useful for dense vectors
Dot Product	Neural search, ANN retrieval	Scale-dependent
Mahalanobis Distance	Correlated high-dimensional data	Computationally expensive

What Should You Use for Sentence Transformers?

- For similarity search? → Cosine Similarity
- For clustering? → Euclidean Distance
- For retrieval (FAISS – Facebook AI Similarity Search)? → Dot Product
- For text keyword matching? → Jaccard Similarity

Models & Dimension *(show example_1)*

- Dimensions capture semantic aspects *PCA, t-SNE, and attention visualization.*

Model Name	Embedding Dimension
all-MiniLM-L6-v2	*384*
all-MiniLM-L12-v2	384
paraphrase-MiniLM-L6-v2	384
multi-qa-MiniLM-L6-cos-v1	384
all-mpnet-base-v2	*768*
paraphrase-mpnet-base-v2	768
sentence-t5-base	768
sentence-t5-large	1024
bert-base-nli-mean-tokens	768
roberta-base-nli-stsb-mean-tokens	768

Architecture Differences

- **all-MiniLM-L6-v2**
  - MiniLM (Minimal BERT) is a lightweight model designed for efficiency.
  - Uses distilled knowledge from a larger BERT model.
  - Has 6 layers and 384 hidden dimensions (hence smaller embeddings).
  - Faster inference but slightly lower accuracy.
- **all-mpnet-base-v2**
  - MPNet (Masked Permuted Network) is a more powerful model.
  - Uses a combination of BERT’s MLM (Masked Language Model) and XLNet’s
  - Permutation Language Model → captures more contextual dependencies.
  - Has 12 layers and 768 hidden dimensions (hence richer embeddings).
  - Higher accuracy in semantic similarity and clustering tasks.
- **MPNet is larger, more complex, and retains richer sentence semantics.**

Applications

- Chatbot
- Recommendation system
- Search engine
- Plagiarism detection
- Customer support portal

Way forward

- Fine tune the model for accuracy
- Visualize the mechanism of self attention
- Visualize the semantic aspects captured by the embeddings
- Build a custom model for Electronic Health Record data
  - Manually prepare a gold standard of definitions for all the jargon related to the EHR