Beginning with Vector Embeddings

1. Introduction to Vector Embeddings
- Definition: Vector embeddings are numerical representations that convert objects, concepts, or entities into fixed-length vectors within a high-dimensional space. Each vector is comprised of real numbers, where each dimension corresponds to a specific attribute or feature of the entity.

- Types of Embeddings:
    - Traditional Sparse Embeddings : Characterized by a high-dimensional space where many dimensions are zero, typically used for representing categorical data such as words in a text corpus.
    - Dense Embeddings : These embeddings have lower dimensionality and consist of non-zero values across all dimensions, capturing richer semantic information.
    - Learned Sparse Embeddings : An innovative type that integrates the benefits of traditional sparse embeddings with the semantic richness of dense embeddings.

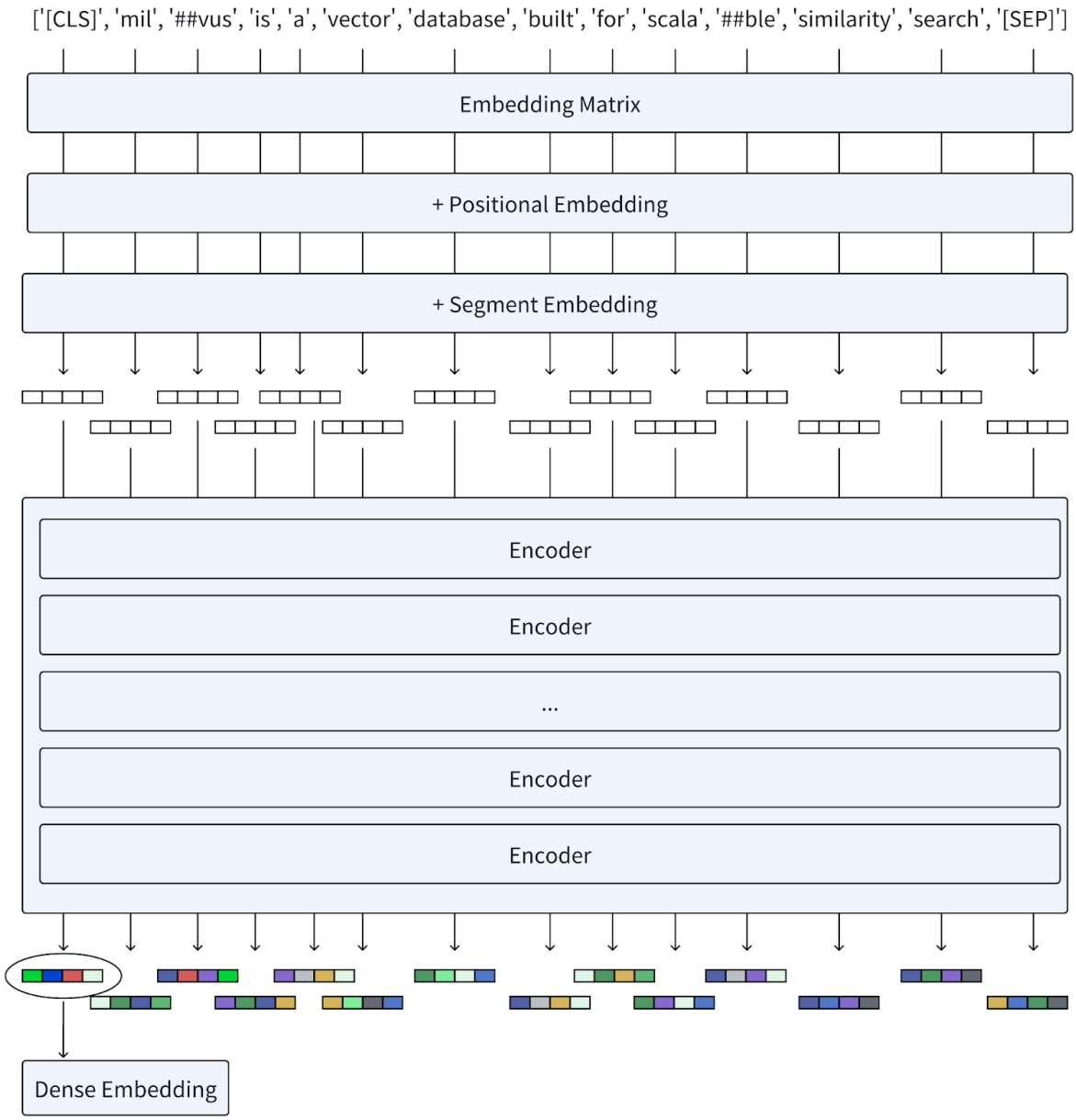
2. Traditional Sparse Embeddings
- Description : Commonly used in information retrieval tasks, these embeddings can represent tokens (words) across multiple languages or documents, with non-zero values indicating the relative importance of each token in a specific context.

- BM25 Algorithm :
    - An enhancement over TF-IDF, BM25 applies:
    - Term Frequency Saturation : A diminishing return effect on term frequency, addressing over-representation of frequently occurring terms.
    - Length Normalization : Adjusts scores based on the length of the document, preventing bias towards longer documents.

3. Dense Embeddings
    - Key Architecture : BERT (Bidirectional Encoder Representations from Transformers)
    - Bidirectional Contextualization : Unlike traditional models that process text sequentially, BERT evaluates the entire context of words simultaneously, leading to better understanding.

- Pre-Training Tasks :
    - Masked Language Modeling (MLM) :
    - Randomly masks certain tokens in the input and trains the model to predict these based on the surrounding context.
    - This task enables BERT to understand relationships and nuances within text by considering all contextual information.

- Next Sentence Prediction (NSP) :
- Trains the model to predict if one sentence logically follows another, which helps in understanding discourse and text coherence.



From tokens to BERT dense embeddings

- Architecture Components :

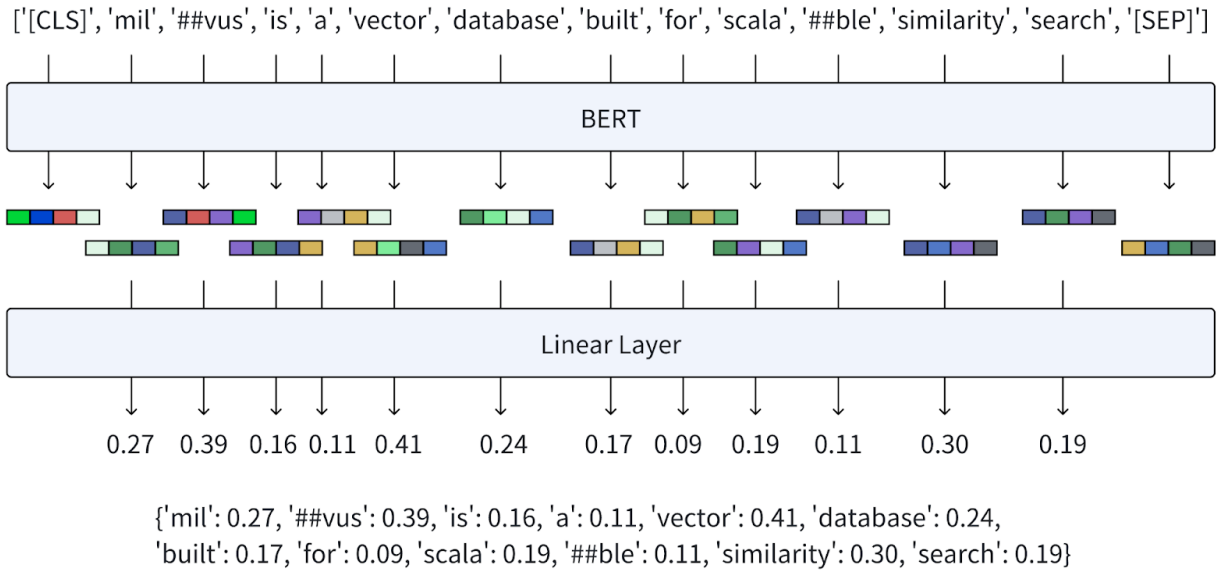
- Self-Attention Mechanism : Each encoder layer employs self-attention to determine the relevance of all words in the sentence when interpreting a specific word, allowing for nuanced understanding.

- Positional Encoding : This feature allows the model to consider the order of words in the input, essential for capturing the syntactic structure of the text.

4. Learned Sparse Embeddings

- BGE-M3 Model :

- Design Goals : Aims to enhance text representation with multi-functionality, multi-linguisticity, and multi-granularity.
- Token Importance Estimation : Instead of relying solely on a single representation (e.g., [CLS] token), BGE-M3 evaluates the contextualized embeddings of all tokens in the sequence.



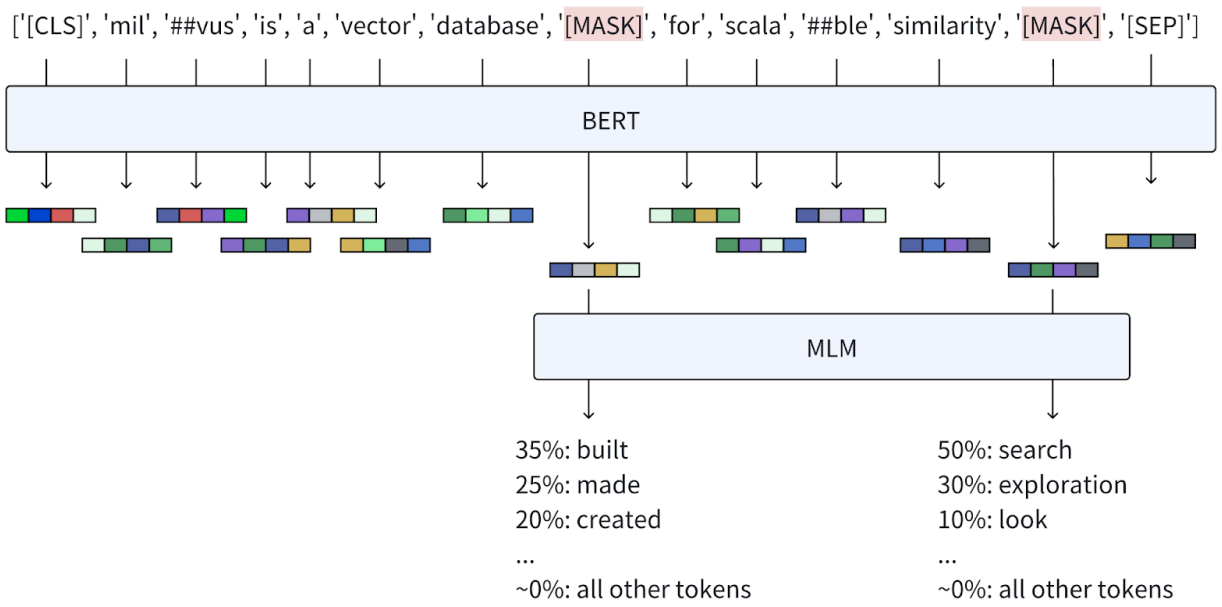
From tokens to sparse embeddings.png

- Mechanism:

- Linear Transformation: An additional linear layer is used to compute importance weights for each token based on its contextualized embedding.
- Activation Function: A Rectified Linear Unit (ReLU) is applied to ensure that the weights are non-negative, thus contributing to the sparsity of the embedding.
- Output: The result is a sparse embedding where each token has an associated weight reflecting its significance in the context of the entire input text.

- SPLADE Model:

- Innovative Approach : Builds on BERT's architecture with a focus on refining embedding sparsity.



MLM predicts the original token based on the BERT embedding of [MASK]

- Mechanism:

- Utilizes MLM during encoding, assessing the relevance of each token with respect to the entire vocabulary by generating probability distributions for masked tokens.
- Aggregation: The model aggregates probabilities across all positions to create a sparse vector that includes weights for vocabulary terms relevant to the input tokens.
- Log Saturation Effect : Regularization method used to promote sparsity, enhancing the model's capacity for term expansion.
- Term Expansion Capability : SPLADE can identify and include terms not present in the original text, enriching the context significantly. For instance, it can expand a short query into a larger representation, thereby improving retrieval accuracy.

5. Applications and Implications

- Semantic Search: Dense embeddings, particularly from BERT and its derivatives, are effective for semantic search tasks where the goal is to retrieve relevant documents based on meaning rather than exact keyword matches.

- Information Retrieval : Learned sparse embeddings (BGE-M3 and SPLADE) enhance retrieval precision by producing representations that incorporate both semantic and lexical elements, making them particularly useful for nuanced queries in large databases.

6. Conclusion

- The advancement in embedding technologies, particularly through learned sparse embeddings, represents a significant leap in natural language processing capabilities. These innovations allow for more effective retrieval and understanding of language, addressing the complexities and nuances inherent in human communication.

Key Technical Terms

- Embedding Matrix: A learned matrix that converts tokens into dense vectors.
- Contextualized Embedding: The representation of a token in relation to its context within a sentence.
- Weighting Mechanism: The process of assigning importance to tokens in the embedding output, facilitating more effective information retrieval.