Before going through the BGE-M3 model lets discuss briefly as to the purpose of using these models instead of directly relying on BERT for creation of dense embeddings: Vector embeddings, or vector representations, are numerical representations of objects, concepts, or entities within a high-dimensional vector space. Each entity is represented by a vector comprising real numbers, usually fixed in length, with each vector dimension representing a distinct attribute or feature of the entity. There are typically three main embedding types: (traditional) sparse embeddings, dense embeddings, and "learned" sparse embeddings.

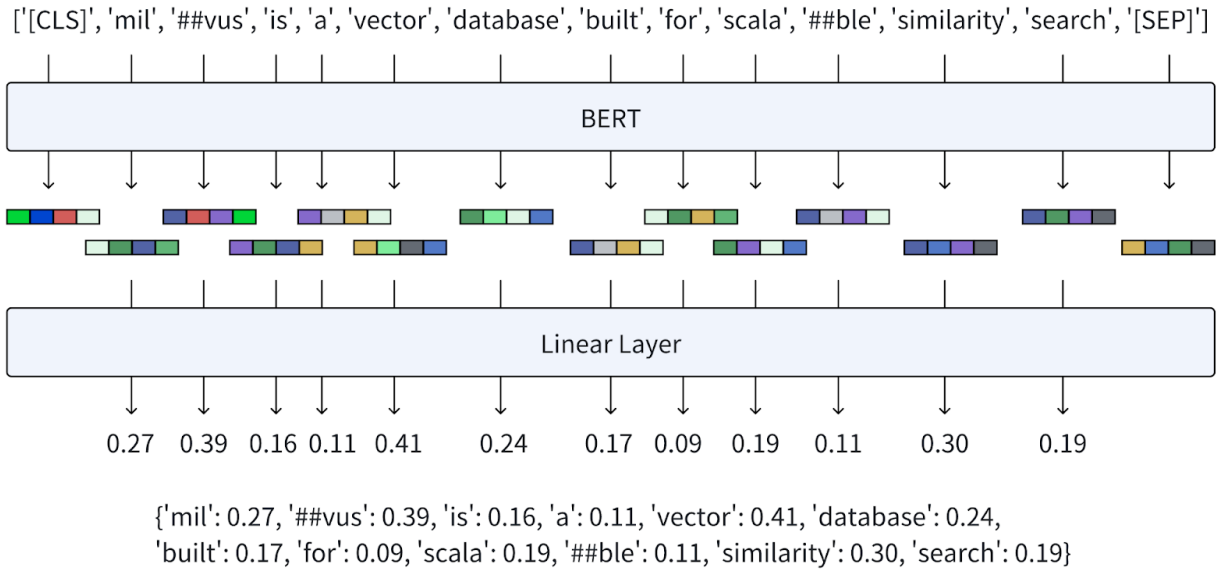
Traditional sparse embeddings, typically used in language processing, are high-dimensional, with many dimensions containing zero values. These dimensions often represent different tokens across one or more languages, with non-zero values indicating the token's relative importance in a specific document. Sparse embeddings, such as those generated by the BM25 algorithm—which refines the TF-IDF approach by adding a term frequency saturation function and a length normalization factor—are perfect for keyword-matching tasks.

In contrast, dense embeddings are lower-dimensional but packed with information, with all dimensions containing non-zero values. These vectors are often produced by models like BERT and used in semantic search tasks, where results are ranked based on the closeness of the semantic meaning rather than exact keyword matches.

“Learned” sparse embeddings are an advanced type of embedding that combines the precision of traditional sparse embeddings with the semantic richness of dense embeddings. They enhance the sparse retrieval approach by incorporating contextual information. Machine learning models like Splade and BGE-M3 generate learned sparse embeddings. They can learn the importance of related tokens that may not be explicitly present in the text, resulting in a 'learned' sparse representation that effectively captures all relevant keywords and classes.

Now lets go in detail on the BGE-M3 model :

BGE-M3 is a sophisticated machine-learning model that builds upon BERT's functionality. It aims to improve text representation by emphasizing Multi-Functionality, Multi-Linguisticity, and Multi-Granularity. Unlike just producing dense embeddings, it also creates learned sparse embeddings, effectively balancing semantic meaning with lexical accuracy, which is especially beneficial for detailed information retrieval.



From tokens to sparse embeddings

BGE-M3 enhances the traditional process by adopting a more detailed method for assessing the significance of each token:

1. Token Importance Estimation: Rather than relying solely on the representation of the [CLS] token (H[0]), BGE-M3 analyzes the contextualized embeddings of each token (H[i]) within the sequence.

2. Linear Transformation: A linear layer is added to the output from the encoder stack, which calculates the importance weights for each token. This allows BGE-M3 to derive a set of weights (W_{lex}) by processing the token embeddings through this layer.

3. Activation Function: A Rectified Linear Unit (ReLU) activation function is applied to the product of (W_{lex}) and (H[i]) to determine the term weight (w_{t}) for each token. ReLU ensures that the term weight remains non-negative, contributing to the sparsity of the embedding.

4. Learned Sparse Embedding: The final output is a sparse embedding, where each token is linked to a weight that reflects its importance within the context of the entire input text.

This approach deepens the model's comprehension of language subtleties and customizes the embeddings for tasks where both semantic and lexical factors are essential, such as in search and retrieval across large databases. This represents a significant advancement in creating more accurate and efficient methods for navigating and understanding extensive textual information.