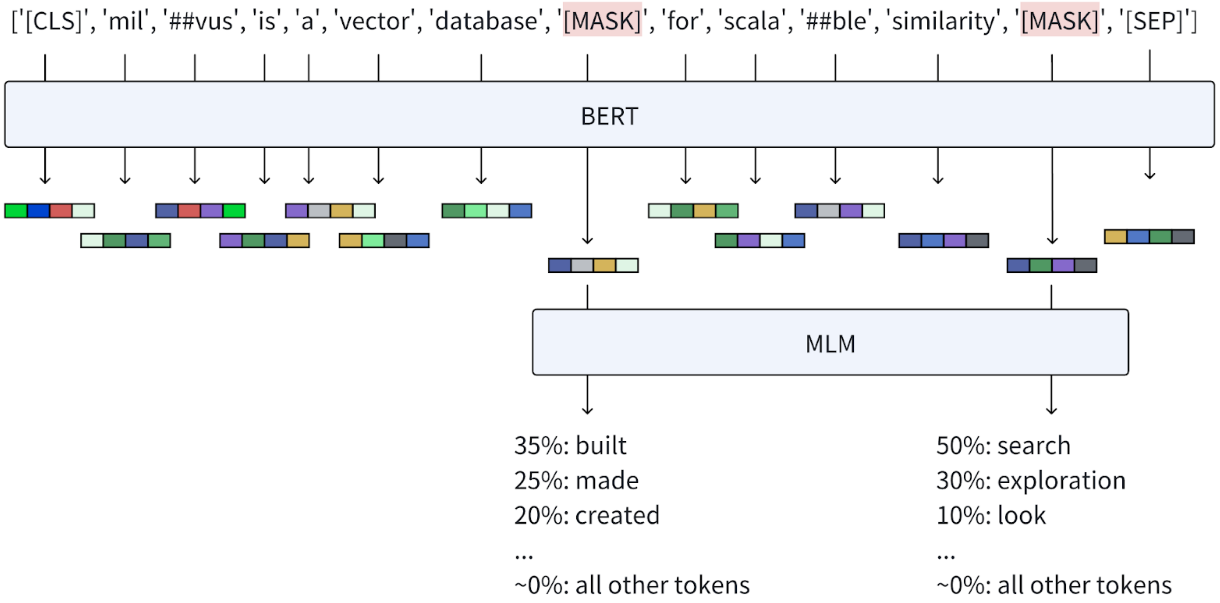
SPLADE Embedding Model

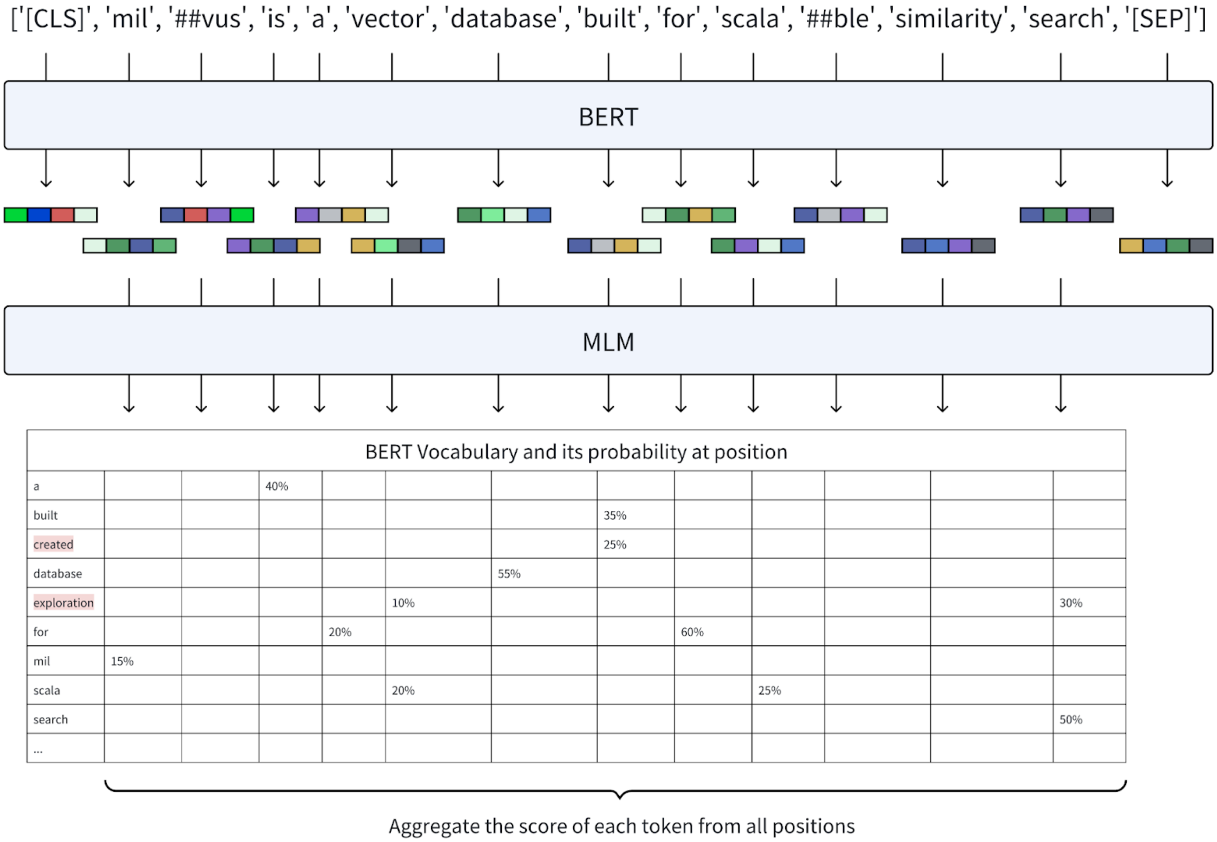
Splade model itself is another embedding model that has its base built up from the bert model as its foundation which is topped of by its unique methodology to convert the dense embeddings further into learned sparse embeddings which involves one of the pre-training metodology implemented on the bert model: MLM(Masked Language Modelling) where we hide tokens randomly within the created tokens from the data to better generate the model's linguistic comprehension and structural awareness of language, as it depends on adjacent tokens to fill the gaps with accurate predictions.

The below is the pictorial way of MLM working:


['[CLS]', 'mil', '##vus', 'is', 'a', 'vector', 'database', '[MASK]', 'for', 'scala', '##ble', 'similarity', '[MASK]', '[SEP]']
BERT
MLM
35%: built
25%: made
20%: created
...
~0%: all other tokens
50%: search
30%: exploration
10%: look
...
~0%: all other tokens

Now the pre-training of this model happens with is for every hidden slot in that of the model is filled by utilizing the contextualized embedding from bert to output a probablity distribution where the specific vocabulary token with the highest probability from bert occupies the hidden position. This output vector which has the size of bert's extensive vocabulary serves as a pivotal learning signal for refining predictions given by the model for tasks as next word generation.

The above explanation is represented below:


['[CLS]', 'mil', '##vus', 'is', 'a', 'vector', 'database', 'built', 'for', 'scala', '##ble', 'similarity', 'search', '[SEP]']
BERT
MLM
Aggregate the score of each token from all positions

BERT Vocabulary and its probability at position												
a			40%									
built							35%					
created							25%					
database						55%						
exploration					10%							30%
for				20%				60%				
mil	15%											
scala					20%				25%			
search												50%
...												

{'a': 0.4, 'built': 0.35, 'created': 0.25, 'database': 0.55, 'exploration': 0.1, 'for': 0.8, 'mil': 0.15,
'scala': 0.45, 'search': 0.5, ...}

The unique way of applying splade is due to the mlm applied at the encoding phase, Initially when the embeddings from the bert are generated splade applies mlm across all token positions and it calculates probability of each token by every word in the bert vocabulary across all positions, applying a regularization method to promote sparsity by implementing a log saturation effect. The resulting weights refer to the relevance of each token to the input tokens and thus creating laearned sparse vectors.

Thus creating a unique advantage over other models as it has its inherent technique for term expansion as it can identify and relevant terms and expand the sentence in ways possible thus having good term matching capabilities in retrieval tasks and improving the model's precision.