I got some inspiration when I was doing a natural language processing project. I was trying to implement an AutoCodeCompletion project where the model would predict the next word or next type given the input JS code. We used the Word2Vec mechanism where we created an embedding table where each row was a vector/tensor representation of each distinct word, the smaller the Euclidean distance, the stronger the connectivity. This reminds me of a previous article on unstructured data management [The one on the master branch], their main goal was to connect related information as much as possible and their approaches were focusing on the semantic analysis and lexical analysis. Furthermore, the principle of vector databases is similar to that, they transform the human-readable words/code[input] to vector so that it would be way easier for finding the connectivity.

Thus, I was thinking if we could apply Word2Vec in our query clustering process[unsupervised clustering]. CMU group used the arrival rate for similarity features search, I was thinking if it could be an alternative methodology for query clustering.

One of the drawbacks for Word2Vec[skipgram] is the order, for example, if we train a sentence “I am Roy”, the distance between “I” and “am ” is the same as the distance between “Roy” and “am ”[We lost the order of information].

~~~~~~~~~~~~~~~~~~~~~~~~~~~~1/12/2021 update ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Order does matter, suppose we perform select queries in two brand new databases separately, and there are no previous or simultaneous queries, and the storage would be SSD and we ignore the difference caused by SSD wear. In this case, the workload should be the same, however, it would be a different story if there are any previous or simultaneous queries, even worse, what if a delete query is executing for the same table that the user is trying to select? One way could be looking up the previous K queries or elements as the helpers, just like we could easily figure out the point of this paragraph without looking back to the first paragraph, that comes to the sequence processing and prediction.[Given the previous input I have, what kind of problems will the current query encounter?].
~~~~~~~~~~~~~~~~~~~~~~~~~~~~