Notes on Vector Retrieval

Compiled by D. Gueorguiev, 10/18/2025, following Sebastian Bruch’s *“Foundations of Vector Retrieval”*

# Introductory Notes

A diagram of a number

AI-generated content may be incorrect.Let us consider a text document in English. Strip the document of grammar and word order and we end up with *a set* of words a.k.a. “bag of words” (BoW).

Figure: Vector representation of a piece of text by adopting BoW approach: a text document, when stripped of grammar and word order, is represented by a vector where each coordinate represents a term in our vocabulary and its value records the frequency of that term or represent some function of the frequency. The resulting vectors are *sparse* – that is , they have few non-zero coordinates.

Transformer-based models ([5], [6]) brought about vector representations that beyond the elementary formation above. The resulting vector representation is referred to as an *embedding*, instead of a “feature vector”, though the underlying concept is the same – an object is encoded as a real -dimensional vector, a point in .

Question: how the embedding of a text document differs from its representation as a frequency-based feature vector?

In the lexical BoW representation if a coordinate is non-zero, that implies that the corresponding term is present in the document and its value indicates its frequence-based feature. In the embedding representation our embedding algorithm *learns* to turn coordinates on or off and when a coordinate is turned on , its value must predict the significance of the corresponding term based on semantics and contextual information. For example, the absent synonyms of a present term may get a non-zero value, and terms that offer little discriminative power in the given context become 0 or close to it.

This idea has been explored by many recent embedding models such as those discussed in [7], [8], [9], [10], [11], [12], [13], and [14].

Similarity in vector space must imply similarity between objects. So, as we engineer features to be extracted from an object or design a protocol to learn a model to produce embeddings of data, we must choose the dimensionality of the target space (a subset of ) along with a distance function . Together, these define an inner product of metric space.

Consider the lexical representation of a text document where d is the size of the English vocabulary

# References

[1] [Foundations of Vector Retrieval, S. Bruch, 2024](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Foundations_of_Vector_Retrieval_Bruch_2024.pdf)

[2] [Distributed Representations of Words and Phrases and their Compositionality, Thomas Mikolov et al, Google, 2013](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/embeddings/Distributed_Representations_of_Words%20and_Phrases_and_their_Compositionality_Mikolov_2013.pdf)

[3] [Efficient Estimation of Word Representations in Vector Space, Thomas Mikolov et al, Google, 2013](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/embeddings/Efficient_Estimation_of_Word_Representations_in_Vector_Space_Mikolov_2013.pdf)

[4] [GloVe: Global Vectors for Word Representation, Jeffrey Pennington et al, 2014](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/embeddings/GloVe-Global_Vectors_for_Word_Representation_Pennington_2014.pdf)

[5] [Attention is all you need, Ashish Vaswani, Google Brain, 2017](https://github.com/dimitarpg13/transformers_intro/blob/main/articles_and_books/Attention-is-all-you-need-NIPS-2017.pdf)

[6] [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin et al, 2019](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/embeddings/BERT-Pre-training_of_Deep_Bidirectional_Transformers_for_Language_Understanding_Devlin_2019.pdf)

[7] [Sparterm: Learning term-based sparse representation for fast text retrieval, Y. Bai et al, 2020](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/SparTerm-Learning_Term-based_Sparse_Representation_for_Fast_Text_Retrieval_Bai_2020.pdf)

[8] [Splade: Sparse lexical and expansion model for first stage ranking, T. Formal et al, 2021](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Splade-Sparse_lexical_and_expansion_model_for_first_stage_ranking_Formal_2021.pdf)

[9] [Fast Passage Re-ranking with Contextualized Exact Term Matching and Efficient Passage Expansion, S. Zhuang et al, 2021](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Fast_Passage_Re-ranking_with_Contextualized_Exact_Term_Matching_and_Efficient_Passage_Expansion_Zhuang_2021.pdf)

[10] [Context-aware term weighting for first stage retrieval, Z. Dai et al, 2020](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Context-Aware_Passage_Term_Weighting_For_First_Stage_Retrieval_sigir20-Zhuyun_Dai.pdf)

[11] [COIL: Revisit Exact Lexical Match in Information Retrieval with Contextualized Inverted List, L. Gao et al, CMU, 2021](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/COIL-revisit_exact_lexical_match_in_information_retrieval_with_contextualized_inverted_list_Gao_2021.pdf)

[12] [Learning Passage Impacts for Inverted Indexes, A. Mallia et al, NYU, 2021](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Learning_Passage_Impacts_for_Inverted_Indexes_Mallia_2021.pdf)

[13] [From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing, H. Zamani et al, U Mass Amherst, 2018](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/From_Neural_Re-Ranking_to_Neural_Ranking-Learning_a_Sparse_Representation_for_Inverted_Indexing_Zamani_UMassAmherst_2018.pdf)

[14] [A Few Brief Notes on DeepImpact, COIL, and a Conceptual Framework for Information Retrieval Techniques, J. Lin et al, 2021](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/A_Few_Brief_Notes_on_DeepImpact_COIL_and_a_Conceptual_Framework_for_Information_Retrieval_Techniques_Lin_2021.pdf)

[] [Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality, Piotr Indyk et al, Stanford, 1998](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Approximate_Nearest_Neighbors-Towards_Removing_the_Curse_of_Dimensionality_Indyk_1998)

[]

# Appendix

## Notes on SparTerm model for Learning Term-based Sparse Representation for Fast Text Retrieval

The paper proposing SparTerm model and related algorithm is [7].