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)

[15] ["King - man + woman = queen" is fake news, Mike Cohen's substack blog, Oct 12, 2025](https://mikexcohen.substack.com/p/king-man-woman-queen-is-fake-news)

[16] [Some Simple Effective Approximations to the 2-Poisson Model for Probabilsitic Weighted Retrieval, S.E. Robertson et al, 1994](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Some_Simple_Effective_Approximations_to_the-2_Poisson_Model_for_Probabilistic_Weighted_Retrieval_Robertson_Walker_Sigir_1994.pdf)

[17] [Probabilistic Relevance Model, Wikipedia](https://en.wikipedia.org/wiki/Probabilistic_relevance_model)

[18] [Relevance Weighting of Search Terms, SE Robertson et al, 1976](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/Relevance_Weighting_In_Search_Terms_Robertson_1976.pdf)

[19] [The Probabilistic Relevance Framework: BM25 and Beyond, SE Robertson et al, 2009](https://github.com/dimitarpg13/vector_db_intro/blob/main/articles/embeddings/The_Probabilistic_Relevance_Framework-BM25_and_Beyond_Robertson_2009.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.pdf)

[] [Vector database management systems: Fundamental Concepts, use-cases, and current challenges, Toni Taipalus, 2024](https://github.com/dimitarpg13/rag_architectures_and_concepts/blob/main/articles/vector_db/Vector_database_management_systems-Fundamental_concepts_use-acases_and_current_challenges_Taipalus_2024.pdf)

[] [Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, Patrick Lewis, Ethan Perez et al, 2021](https://github.com/dimitarpg13/rag_architectures_and_concepts/blob/main/articles/Retrieval-Augmented_Generation_for_Knowledge-Intensive_NLP_Tasks_Lewis_2021.pdf)

# Appendix

## Notes on Probabilistic Relevance Models

These notes follow the exposition in [19].

A user query is a representation of a user’s information need. *Relevance* in this context means the relevance of a document to the specific information need of the user.

The assumptions about document retrieval relevance or simply *relevance*, are

-relevance is assumed to be a property of the document given information need only, assessable without reference to other documents;

-the relevance property is assumed to be binary

Probability Ranking Principle (PRP)

The information available to the system has probabilistic nature as the information retrieval system cannot *know* the values of the relevance property of each document. One form of the Probability Ranking Principle (PRP) is:

*If retrieved documents are ordered by decreasing probability of relevance on the data available, then the system’s effectiveness is the best that can be obtained for the data.*

Notation

Consider the query which represents a single information need.

Two ranking functions and are equivalent as ranking functions if they produce the same ranking / order with respect to the set of documents. Such equivalence is denoted with the symbol :

.

In developing the model, from the probability of relevance of a document to a term-weighting and document-scoring function, we make frequent use of transformations which preserve rank order. Such transformation of a document-scoring function may be linear or non-linear, but must be strictly monotonic, so that if documents are ranked by the transformed function, they will be in the same rank order as they had been ranked by the original function.

The property of relevance is represented by the random variable with two possible values:

/\* relevant or not \*/

We will use the following short-hand notation to denote the probability that a document is relevant to the given user query :

For documents and queries we are using either set-of-words (SoW) or a bag-of-words (BoW) model. Thus, the vocabulary of terms is indexed into the set , each of which may be present or absent in the SoW model or present with some frequency in the BoW model. Thus the object of interest (either a document d or query q) is represented as a vector over the space defined by the vocabulary. For a document we have:

where represents the frequency of term in the document. We will distinguish between the random variable and its observed value in a document . The random variable will refer to the possible term frequencies for the -th term in the vocabulary.

A query can be represented in two different ways. In the first representation the query is treated as a vector similarly to a document:

The component represents the term frequencies in the query or it may represent a binary presence or absence feature.

In the analysis which follows we will need to sum or multiply variables of terms present in the query (i.e. those with . For this purpose we define the set of indices:

where

### Basic Model

Here we continue with the development of the basic model.

(prm.1)

(prm.2)

(prm.3)

(prm.4)

(prm.5)

(prm.6)

(prm.7)

(prm.8)

//TODO: finish the section on probabilistic relevance models

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

Text retrieval in response to natural language query is a core task for information retrieval systems. A popular approach to this problem adopts a two-stage pipeline, where as a first stage an initial set of documents is retrieved from the document collection by a fast retriever, and then further re-ranked by more suitable re-ranker models.

For the first-stage retrieval, neural dense representations outperform sparse methods in many NLP tasks, but this is not necessarily true in scenarios that emphasize long document retrieval and exact matching.

Moreover, for extremely large candidate collection, the dense method struggles with the efficiency vs accuracy tradeoff.

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

Question: Can we transfer the deep knowledge of the pretrained language model (PLM) to **Term**-based **Spa**rse representations, aiming to improve the representation capacity of BoW while keeping its advantages?

The proposed SparTerm comprises an importance predictor to predict the importance for each term in the vocabulary, and a gating controller to control the term activation. These two modules cooperatively ensure the sparsity and flexibility of the final text representation, which unifies the term-weighting and expansion in the same framework.

## 2-Poisson model

2-Poisson model is a statistical model used primarily in information retrieval to represent term frequencies in documents. It assumes that the frequency of a term in a document follows a mixture of two Poisson distributions – one for “elite” or highly relevant documents, and second one for “non-elite” documents capturing the different patterns in word occurrence. This model was developed to improve upon simpler methods by considering factors like within-document frequency, document length and within-query frequency.

### Definitions and assumptions

**Elite vs Non-elite terms**: the model is based on the “eliteness hypothesis” which suggests some documents are more “elite” for a given term. The model splits a document collection into two groups for a given term.

**Elite documents**: these documents have a higher mean rate () for the term’s occurrences, meaning the term appears more frequently and is more central to the document’s content

**Non-elite documents**: these documents have a lower mean rate () for the term’s occurrences, where the term appears randomly, like linguistic “glue” words or stop words.

**Mixture model**: a document is treated as a mixture of the two distributions, with parameters representing the means of each Poisson distribution and the probability of the document being in the “elite” category.

**Parameter estimation**: the model’s parameters (two means and the mixing probability) are often estimated using the Expectation-Maximization (EM) algorithm for each term in a collection.

**Document length**: Although technical the model can be used without it, it often assumes that the document length is constant. Variations have been developed to explicitly condition on document length making it a bivariate model.

## Simple Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval

This Appendix Section is a summary of the results and discussion in [16].

This paper concerns the usage of the 2-Poisson Model as a probabilistic model for information retrieval.

## Jaccard index and Jaccard distance

The *Jaccard index* is a statistic used for gauging the similarity and the lack of diversity of sample. It is defined as the ratio of two sizes (areas or volumes), the intersection size divided by the union size also called *intersection over union*.

Thus, the Jaccard index is defined as the size of the intersection divided by the size of the union of the sample sets

Obviously, . If no elements are common, then obviously . If clearly .

The *Jaccard distance* measures dissimilarity between sample sets and is complementary to the Jaccard index. It is obtained as:

.