Vector Space Models

Eva Richter

March 31, 2021

Abstract

This essay discusses different aspects of simple vector space models and explores the underlying ideas and concepts. First, the historical background is explained to place the idea of vector space models in context and to give an overview of people and events linked to their development. Subsequently, the most important terms and concepts of the Vector Space Model for information retrieval are defined, followed by a description of term-document and term-term matrices and a compilation of advantages and limitations of vector space models. Finally, the importance of vector space models for current applications, requiring weighted retrieval, is highlighted. The core ideas, as well as an outlook for further developments, are presented in the conclusion.

1 Introduction

The huge amount of data and topics stored in data repositories make the retrieval of specific information a challenging task. Due to the exponentially growing unstructured volume of data on the internet and in databases, the time and resources required to filter for specific information are not feasible by humans. This problem is tackled by automatic retrieval systems that allow searching and retrieving specific bits of information from texts, pictures and sounds within seconds. Such systems can be based on various models, such as the Boolean model, the probabilistic model, the inference network model or the Vector Space Model. The latter is probably the most widely used technique for information retrieval, as its simplicity and efficiency over large collections of documents are appealing for many tasks. The Vector Space Model is a mathematical model for representing documents and queries as vectors of weights and was developed mainly based on the work of Gerard Salton in the 1970s. Many modern systems of information filtering, information retrieval, indexing and relevancy rankings are based upon the Vector Space Model. Nowadays it is the core mechanism of many search engines and serves as the reference model for applications like clustering, cross-language retrieval or automatic text summarization.

2 Historical Background

Vectors have been used in the context of Information Retrieval (IR) since the early stage of IR, mainly as a means to describe the design and implementation of a system [Melucci, 2009]. The notion of what is nowadays referred to as Vector Space Model (VSM) is inextricably linked to the person Gerard Salton, who is considered the "father of information retrieval" due to his pioneering work for term weighting, relevance feedback, document clustering, extended boolean retrieval, term discrimination value and so forth [Brochure, 2015]. Automatic Indexing principles as expressed in the Term Discrimination Value (TDV) model (Salton et al., 1975), which eventually influenced the perception of what later became known as VSM, emerged as a result of Salton's early efforts of automatic indexing, as opposed to the manual indexing approach that had prevailed until then [Dubin, 2004]. Full-text-indexing methods, where each word in each document is presented as an index term, are still the core mechanism, on which essentially all of today's search engines are based [Brochure, 2015].

An evolutionary stage in IR was initialized by the publication of Salton's article "Mathematics and Information Retrieval" in 1979 since the VSM was referred to as an IR model and orthogonality assumption were described for the first time [Dubin, 2004]. The VSM was fully described and corralled based on critiques for the first time in 1989 [Salton et al., 1989]. It was used for the first time in the SMART Information Retrieval system, a text processing system for IR, which Salton developed throughout his life and on which modern retrieval systems are based [Brochure, 2015].
Due to its "intuitive yet formal view of indexing and retrieval", the VSM has gained extensive recognition ever since, as it provides a reliable framework to retrieve various documents for distinct languages, topics and sizes [Melucci, 2009]. On account of their mathematical properties, vector spaces can be used for further development of IR modelling, for instance for Latent Semantic Indexing [Deerwester et al., 1990], which relies on Singular Value Decomposition (SVD) or to represent context by exploiting the notion of basis of a vector space [Melucci, 2009].

3 Definitions

Due to the VSM's popularity, a wide range of definitions and descriptions have developed, using different notation and emphasising different aspects of the VSM.
Documents and queries are represented as vectors of weights in the VSM. Let x_i be the weight of an index term i in a document and k representing the count of unique index terms in the document set. Then a vector can be represented as:

$$\mathbf{x} = (x_1, ..., x_k). \tag{1}$$

Weights measure the significance of an index term occurring in a document and can be calculated using various methods that have been developed over time. For an overview of different term weighting schemes, see [Salton and Buckley, 1987]. One of the most important examples is the term frequency of i in a document:

$$x_i = \mathrm{TF}_i. \tag{2}$$

Let y_i be the weight of index term i in the query. Then a query is also represented as a vector of weights, for example:

$$\mathbf{y} = (y_1, ..., y_k). \tag{3}$$

As with documents, various definitions for index term weights have been developed for queries, an important example of which is IDF, i.e. Inverse Document Frequency:

$$y_i = \mathrm{IDF}_i. \tag{4}$$

Let n_i be the number of documents indexed by i and N the total number of documents in the set. Then IDF can be defined as:

$$\mathrm{IDF}_i = \begin{cases} \log \frac{N}{n_i} & \text{if it occurs in the query} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The documents are ranked by the cosine of the angle between the document vectors and the query vector. In order for the documents to be returned by the system by decreasing cosine, the cosine of the angle is calculated for each document and query:

$$\cos = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \quad ||\mathbf{y}||}. \tag{6}$$

TF_i IDF_i refers to the TF-IDF weighting scheme. Accordingly, when term frequency and inverse document frequency are considered for weighting terms in the document and in the query, the inner product between document and query vectors is defined as follows:

$$\sum_{i=1}^{k} \text{TF}_i \text{ IDF}_i. \tag{7}$$

In case different documents are taken into account, the notations $\mathbf{x}_n = (x_{1,n}, ..., x_{k,n})$ and $\text{TF}_{i,n}$ IDF_i are displayed to draw a distinction between documents $n = 1, ..., N$ [Melucci, 2009].

4 Words and Vectors

Vector models usually rely on co-occurrence matrices, that present a distribution of co-occurring words. Two well-known matrices that will be discussed in the following are the term-document matrix and the term-term matrix.

4.1 Term-Document Matrices

The term-document matrix was defined in the context of Salton's VSM for the purpose of determining the similarity of documents for document IR. As IR refers to "the task of finding the document d from the documents D in some collection that best matches a query q" [Jurafsky and Martin, 2009], a query by a vector, which has length $|V|$ as well, has to be represented and compared.

	document 1	document 2	document 3	document 4
term 1	1	7	6	0
term 2	2	8	7	3
term 3	1	10	12	2
term 4	0	5	9	1

Table 1: Term-document matrix with each cell containing the number of times a (row) word occurs in a (column) document.

A term-document matrix consists of rows, each representing a term from the vocabulary and columns, each representing a document from a document collection [Jurafsky and Martin, 2009]. For presentation reasons, the term-document matrix, that is shown in Table 1, is limited to four words, with each cell of the matrix indicating how often a word defined by a row occurs in one of the four documents defined by a column. Real term-document matrices are naturally much larger, more precisely they are $|V|$-dimensional, since the size of the vocabulary comprises in general several thousands of words and the collection of documents D is usually quite big as well [Jurafsky and Martin, 2009].
The arrangement of the numbers in a vector space indicates differently significant dimensions on which the documents depend in each case [Jurafsky and Martin, 2009]. Accordingly, the first dimension of the four vectors chosen in this example indicates the occurrence of the four words also chosen here. From the comparison of the dimensions it can be seen that, for example, term 1 occurs in document 1 and document 4 with similar frequency (1 time and 0 times), while the values for the document 2 and 3 are much higher.
Documents and queries can be thought of as vectors or points in a $|V|$-dimensional vector space, in which each term has its dimension [Ogheneovo and Japheth, 2016]. Since $|V|$-dimensional vector spaces can hardly be visualized, Fig. 1 shows the visualization of a VSM for four terms, four documents, and one query. As can be seen from the numbers in the example term-document matrix, vectors 1 and 4 are significantly more similar to each other than to vectors 2 and 3, indicating a greater similarity

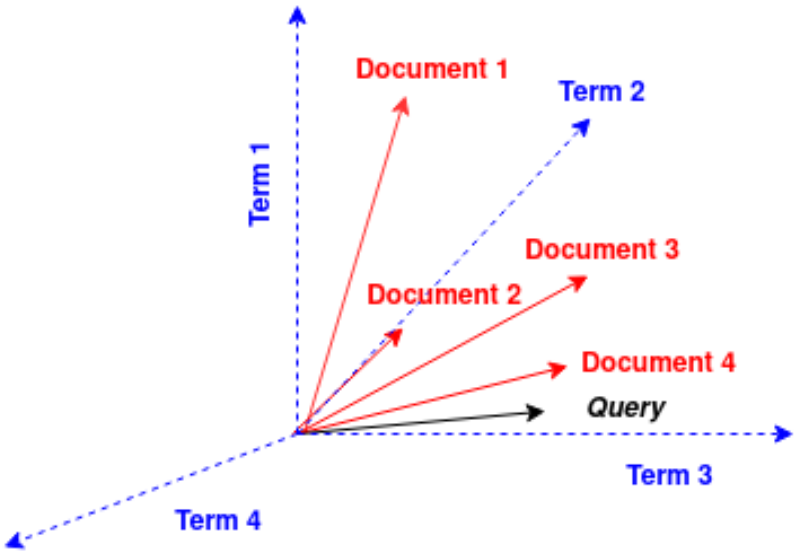

Figure 1: Visualization of a VSM for 4 documents, 4 terms, and 1 query.

between these documents. This shows that since similar documents have similar vectors, they have similar words and since similar words have similar vectors they appear in similar documents. Thus, term-document matrices are a method of representing the meaning of words based on the documents in which they appear [Jurafsky and Martin, 2009].

4.2 Term-Term Matrices

Another possibility to represent documents as vector counts using term-document matrices (also word-word or word-context matrices) are term-term matrices. In contrast to the term-document matrix, in the term-term matrix the columns are not classified as documents but as words, according to which the dimensionality is $|V| \times |V|$ [Jurafsky and Martin, 2009].

	term 1	term 2	term 3	term 4
term 1	1	7	6	0
term 2	2	8	7	3
term 3	1	10	12	2
term 4	10	15	9	11

Table 2: Term-term matrix showing the number of times the row (target) word and the column (context) word co-occur in a context.

Table 2 shows how each cell captures the count of how often a target word, i.e. a row word and a context word (a column word) co-occur in a context in the training data. According to the numbers in the table, term 1 and 2 are more similar to each other than to the terms 3 and 4, i.e. they are closer together in the vector space, as Fig. 2 illustrates.

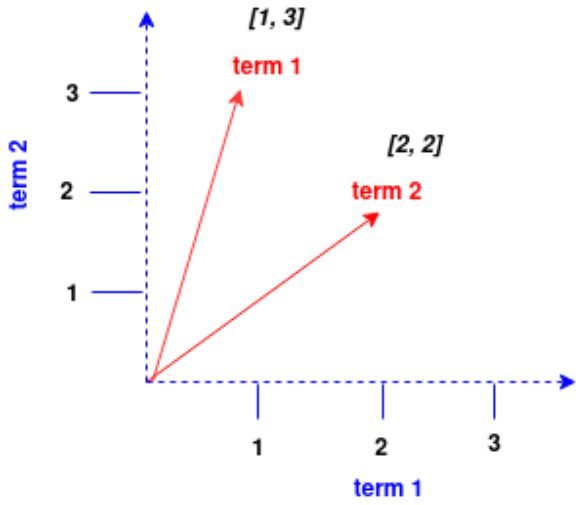

Figure 2: Visualization of (context) word vectors for two dimensions, corresponding to two (target) words.

In most cases, the context is chosen to be relatively small, so that it includes only a few preceding and following words of the word in question, though the document as a whole can also form the context [Jurafsky and Martin, 2009]. In the following, examples from [Jurafsky and Martin, 2009] are shown in which four words before and after a context word (the column word) are considered as context:

traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

4.3 Cosine for measuring similarity

Similarity measurement between two vectors is not inherent in the VSM, wherefore a method that measures similarity between two vectors is needed to determine the similarity between two target words v and w. The most popular metric, the cosine of the angle between two vectors, is based on the dot product. Since the dot product has the property of becoming higher with increasing vector length, and since more frequent words have longer vectors because they occur more frequently in conjunction with other words, the raw dot product is higher for more frequent words than for less frequent words. This poses a problem in that the similarity metric is expected to measure the similarity of words independent of their frequency of occurrence. For this reason, the dot product is normalised by dividing each of the two vectors so that it ends up equal to the cosine of the angle between two vectors. The cosine similarity can then be calculated as follows:

$$\cos(\mathbf{v}, \mathbf{w}) = \frac{\mathbf{v} \cdot \mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2} \sqrt{\sum_{i=1}^{N} w_i^2}} \tag{8}$$

In some cases, each vector is pre-normalised, i.e. divided by its own length, to create a unit vector of length 1 [Jurafsky and Martin, 2009].

5 Weighting Schemes

The great efficiency that makes VSM so common is mainly due to the term weighting applied to the term of the document vectors [Ogheneovo and Japheth, 2016].
Depending on whether a single document is considered or a collection of documents, IR systems can assign local $L_{1,2}$ or global $G_{1,2}$ weights to index terms in documents and queries, so that the weight of an index term i present in a document j can be computed as:

$$w_{i,j} = L_{i,j} G_i. \tag{9}$$

The simplest weighting model is a binary model (BNRY). The model only considers whether a term occurs in a document or not, but it does not consider the frequency with which it occurs or global weights. Since it is a low precision model, which is not able to distinguish between relevant and irrelevant results, it tends to find vocabulary-rich documents just because they happen to contain query terms, which makes the model easy to trick by automatically generating documents with often queried terms [Garcia, 2016b].

An improvement of the model was achieved in the Term Count Model, which, in contrast to the binary model, assumes that a more frequent occurrence of a term in a document is associated with likely high relevance of the document to that term. In order to increase the precision of queries, local weights are represented as a linear function of term frequencies in the Term Count Model:

$$w_{i,j} = L_{i,j} = f_{i,j}. \tag{10}$$

Nevertheless, even this model does not provide an optimal matching approach, since it can easily be manipulated by simple repetitions of terms [Garcia, 2016b].
Both models presented consider only local weights. Adding global weights resulted in the *TF-IDF* weighting, one of the most popular and robust weighting methods, which is described in more detail below.

5.1 *TF-IDF* Weighting

Inverse Document Frequency (IDF) was first introduced in 1972 by Karen Spärck-Jones and describes a measure of the level of precision at which an index term represents a given concept. The intuition behind this was mainly to solve the paradox that words that appear with higher frequency in several documents are more important than words that hardly appear, but words that appear too frequently are also unimportant [Jurafsky and Martin, 2009]. Words that appear frequently in many documents, like function words, are low-*IDF* terms since they are not good discriminators in that they are not document-specific and thus cannot be used to distinguish between documents and topics, whereas technical or scientific terms are usually high-*IDF* terms due to their higher discriminatory power. Accordingly, low-*IDF* terms should be given less weight than high-*IDF* terms [Garcia, 2016c].
Due to its high robustness as well as its heuristic nature, *TF-IDF* is the core of most ranking methods used in search engines today [Robertson and Spärck-Jones, 2000]. Furthermore *TF-IDF* forms an integral part of many weighting schemes and has prevailed over other models, also in language processing techniques for other purposes apart from text retrieval [Robertson and Spärck-Jones, 2000].

Based on probabilities, the global weight of an index term can be set as $G_i = \log(p_i^{-1})$, where $p_i = \frac{d_i}{D}$ is the probability of an index term i being included in a document from a collection of D documents, with d_i counting in how many documents from D the index term i is present. Since probabilities are always between 0 and 1, in order to avoid negative values of the global weight, the inverse of the probability p_i is taken, while the logarithm is used so even extremely small and large p_i can be appropriately compared. This global weight G_i is dubbed *inverse document frequency (IDF)*, and the

weight $w_{i,j}$ becomes

$$w_{i,j} = L_{i,j}G_i = f_{i,j} \log\left(\frac{D}{d_i}\right) = f_{i,j}IDF_i, \tag{11}$$

the *Term Frequency-Inverse Document Frequency (TF-IDF) Model*.

Since probabilities p_1 and p_2 of independent events are known to yield the joint probability p_1p_2, the logarithmic definition of the global weight implies that for independent terms, the IDF of sequences of terms is simply the sum of the IDFs of the individual terms, for instance $IDF_{12} = IDF_1 + IDF_2$. However, this is often not the case, as related terms are usually found within a document dealing with a certain topic [Garcia, 2016c].

5.2 *TF-IDF* Based Models

Many different weighting schemes based on TF-IDF models have been devised by altering $L_{i,j}$ and/or G_i. For instance,

$$L_{i,j} = \begin{cases} \frac{f_{i,j}}{\max f_{i,j}} & \text{if} \quad f_{i,j} > 0 \\ 0 & \text{if} \quad f_{i,j} = 0 \end{cases} \tag{12}$$

$$L_{i,j} = \begin{cases} 0.5 + 0.5\frac{f_{i,j}}{\max f_{i,j}} & \text{if} \quad f_{i,j} > 0 \\ 0 & \text{if} \quad f_{i,j} = 0. \end{cases} \tag{13}$$

Here, the term frequency is normalized in regards to the frequency of the most frequent term in document j, and then augmented so that for the local weights $0.5 < L_{i,j} \leq 1$. This so-called augmented normalized frequency model of ATF1 yields to two revised TF-IDFs by inserting $L_{i,j}$ into the previous TF-IDF:

$$w_{i,j} = \frac{f_{i,j}}{\max f_{i,j}} \log\left(\frac{D}{d_i}\right) = \frac{f_{i,j}}{\max f_{i,j}} IDF_i. \tag{14}$$

$$w_{i,j} = \left(0.5 + 0.5\frac{f_{i,j}}{\max f_{i,j}}\right) \log\left(\frac{D}{d_i}\right) = \left(0.5 + 0.5\frac{f_{i,j}}{\max f_{i,j}}\right) IDF_i. \tag{15}$$

Document and query weighting schemes can also be combined. Which combination is ultimately used depends on various factors, such as the search behavior of the users, the length of the documents and queries, and the type of databases [Garcia, 2016c].

5.3 Pointwise Mutual Information

The second well-known weighting function apart from *TF-IDF* is Pointwise Mutual Information (PMI), a useful tool for identifying strongly associated words. This method plays a role for term-term matrices, i.e. when the vector dimensions correspond to words and not documents.
PMI measures the number of occurrences of an event x and an event y compared to their expected occurrences, assuming they were independent. Pointwise mutual information between a target word w and a context word c can thus be defined as follows:

$$PMI(w, c) = \log_2 \frac{P(w, c)}{P(w)P(c)} \tag{16}$$

While the numerator here indicates how often two words have occurred in conjunction, the denominator indicates how often one would expect the occurrence of these words, assuming they occur independently, in conjunction. Accordingly, the ratio indicates approximately how many times more often two words appear in conjunction than one would expect by chance.
The PMI values can be positive or negative. Negative values mean that words occur less frequently than expected in conjunction, which is problematic in that enormous corpora would be required. For this reason, Positive PMI (PPMI) is used, i.e. zeros are entered instead of negative values [Jurafsky and Martin, 2009].

6 Advantages and Limitations

The initially used Boolean models for IR have many disadvantages compared to VSM for IR, for instance that the ranking of documents does not play a decisive role and it is difficult for users to make good search requests. VSM-based IR systems have the advantage over Boolean systems that they assign a numerical score to documents and rank the documents based on this score [Singhal, 2001].
However, one drawback of the VSM, is its assumption that the terms spanning the vector space are orthogonal to each other, which is not necessarily the case [Gudivada and Rao, 2018]. A solution to this limitation was found by [Wong et al., 1987], who resolved the term orthogonality assumption in their Generalised Vector Space Model (GVSM), modelling information by using vector spaces. On the other hand, Dubin criticises that while the GVSM is a justifiable proposal for using word co-occurrence data in an IR system, it is problematic to represent this as a formal model for vector correspondence and orthogonality in IR [Dubin, 2004].
Further criticism pointed out by [Gudivada and Rao, 2018], among others, relates to the notion of similarity, which doesn't always correspond to relevance.
Nevertheless, the VSM has the advantage over many other models that it is a framework, which does not specify how the dimensions of the vector space are defined, how the terms for documents and queries are weighted and how similarity between a document and query vectors is measured. In this sense, the VSM represents a heuristic model, that forms a solid basis for experimentation and evaluation of other retrieval models [Gudivada and Rao, 2018]. As already indicated, the model has prevailed over other models, particularly on its robustness, simplicity and efficiency, and is the underlying technology of most IR systems [Ogheneovo and Japheth, 2016], [Melucci, 2009].

7 Applications

The VSM is widely used in information filtering, information retrieval, indexing and relevancy rankings. It is the most widely used method for IR and is extremely popular over other models due to its simplicity and efficiency over large document collections. Its efficiency is due in particular to the term weighting (see chapter 5) applied to the terms of the document vectors [Ogheneovo and Japheth, 2016].
Numerous evaluations of the VSM, particularly by the Text Retrieval Conference (TREC), have shown that the VSM is the reference model for many applications where best-match and weighted retrieval are sought. Examples of those applications include clustering, cross-language retrieval, text summarization and personalization to name. The first and best-known application of the VSM is its integration into the SMART system for IR, developed by Salton and his research group, representing one of the first text search engines that were implemented for experiments in text-search on the basis of the *TF-IDF* weighting scheme [Melucci, 2009].

8 Further Directions

A major existing challenge in the field of IR is to develop a model that manages the complexity of the relationships between the constituents of a system and its end users [Melucci, 2009]. Van Rijsbergen's work "the Geometry of Information Retrieval" [van Rijsbergen, 2004] represents a significant step in this direction and leads to a new conception of vector spaces. More specifically, he explores the application of Hilbert space mathematics and linear operators to IR: a document can be represented as a vector in Hilbert space and an observable quantity such as *relevance* or *relatedness* can be represented by a Hermitian operator.
Another key issue is the disclosure of similarity in the application of Hilbert space mathematics to IR and quantum mechanics, since quantum mechanics offers a "ready-made interpretation of [the mathematical] language" [van Rijsbergen, 2004] that can also be applied to describe IR. The main concepts of quantum mechanics such as state vector, observable, uncertainty, superposition and so on

can be transferred to IR, making the theorems of quantum mechanics also accessible as theorems in IR[van Rijsbergen, 2004].

9 Conclusions

The previous chapters have shown that the VSM is the most common technique for IR, and that it is also of great importance for information filtering, indexing and relevancy rankings, as well as beyond their boundaries. In this work, however, the focus was on Salton's VSM for IR, which has a wide range of applications, ranging from text summarisation to cross-language retrieval and search engines.
The VSM's great success is related to its efficiency over large document collections, which the model has been able to maintain over other models over the past decades.
Despite some weaknesses and drawbacks, given its simplicity and robustness, the VSM will probably continue to play a major role in the future. In the meantime, many methods have been developed that are based on the VSM or adapt and further develop it for various purposes. With his work "the Geometry of Information Retrieval", Rijsbergen introduces a new path for future developments of the VSM and leads to a new conception of vector spaces by transferring the most important concepts of quantum mechanics to IR in the context of Hilbert space mathematics.

References

Brochure. The father of information retrieval. 2015.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the Association for Information Science and Technology*, 41(6):391–407, 1990.

D. Dubin. The most influential paper gerard salton never wrote. *Libr. Trends*, 52(4):748–764, 2004.

E. Garcia. The Binary and Term Count Models. 2016b.

E. Garcia. The Classic TF-IDF Vector Space Model. 2016c.

V. Gudivada and C.R. Rao. *Handbook of Statistics (38): Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*. North-Holland, 2018.

D. Jurafsky and J.H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, N.J., 2009.

M. Melucci. *Vector-Space Models. In: Encyclopedia of Database Systems*. Springer, Boston, MA, United States, 2009.

E.E. Ogheneovo and R.B. Japheth. Applications of Vector Space Model to Query Ranking and Information Retrieval. *International Journal of Advanced Research in Computer Science and Soft-ware Engineering*, 6(5):42–47, 2016.

S.E. Robertson and K. Spärck-Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 2000.

G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing Management*, 24(5):513–523, 1987.

G. Salton, A. Wong, and C. Yang. *Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer*. Addison-Wesley, Boston, MA, United States, 1989.

A. Singhal. Modern information retrieval: A brief overview. *IEEE Database Engineering Bulletin*, 24(4):35–43, 2001.

C. van Rijsbergen. *The Geometry of Information Retrieval*. Cambridge University Press, Cambridge, UK, 2004.

S. K. M. Wong, W. Ziarko, V.V. Raghavan, and P. C. N. Wong. On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems*, 12(2), pages = 299–321, 1987.