Enhancing Accuracy of a Search Output: a Conceptual Model for Information Retrieval

Wijewickrema, P. K. C. M.[1] and Ratnayake, A. R. M. M.[2]

Abstract

Providing relevant information through the determination of priorities for a user request is important as it saves both labour and time of the inquirer as well as the information provider. Therefore, the development of information retrieval models to compute these priorities as numerical representations of their relevancies is becoming a major task of the modern information retrieval arena. This paper introduces a new model for information retrieval. This model appears as a vector multiplication of the distances among the terms in the query with the distances among the same terms in the document. Therefore, it determines the relevancies among the query and the documents by matching the distances between each two terms in the query with the distances between the same terms in each document of the corpus. Also this paper uses an example to show that the new model is able to distinctly identify two or more documents according to the given query which cannot be done by using the vector space model when the documents include same term frequencies and same number of total terms.

Keywords: Information Retrieval, Vector Space Model, tf-idf, Distance Model

Introduction

Information Retrieval

Simply, Information Retrieval (IR) is the process of searching for relevant documents, based on a query represented by a user (Salton, & McGill, 1983). An IR system does not inform the user on the subject that searching for but it only notifies on the existence and location of documents relating to the request. In the early days, documents were indexed

Corresponding author:
[1] Assistant Librarian, Sabaragamuwa University of Sri Lanka, Sri Lanka.
Email: manju@sab.ac.lk

[2] Senior Assistant Librarian, Sabaragamuwa University of Sri Lanka, Sri Lanka.
Email: mano@sab.ac.lk

manually using titles, authors, keywords and subject classifications. Then the retrieving process of them was also done manually using indexes or card catalogues. But today, automatic IR systems are playing a major role in the modern information world and provide an alternative method for the conventional manual IR methods. Rapid growth of the information in digitized format is one of the reasons to increase the need of developing proper IR tools. Because, the manual indexing and retrieving of excessive amount of modern documents are not practical and they may waste time and labour intensively.

In general, automatic IR systems compare the user need (known as query) with the content of a set of documents (information source) and retrieve the matching documents according to their relevancy with the given query. This process makes it convenience for the user to extract information from the documents according to their relevancy against the user request. Moreover, these systems prioritise the retrieved documents according to their relevancies to the given query. Therefore, simply it is a listing of priorities according to the relevancies.

Usually, the key element of an IR system is formalized as a mathematical model. In fact, it is the entity which decides how to compare a query with the documents in the corpus. Therefore, the development of mathematical models to compute these priorities as numerical representations of their relevancies is becoming a major task of the present information retrieval systems. To meet these goals, distinct researchers have established various kinds of information retrieval models and their implementations. However, this paper specially focuses on one of the foremost IR model called the Vector Space Model (VSM), its limitations and a new solution to overcome the problem.

Vector Space Model

The VSM is one of the leading IR models which highly used in the modern IR arena (Salton, Wong, & Yang, 1975). It is a kind of algebraic model which represents documents and

queries in a specific mathematical form called vectors. So that, VSM uses vector operations to compare the documents with the given queries.

If a user submits a query *q* to a document corpus which contains the documents $d_1, d_2, d_3, \ldots\ldots\ldots, d_k, \ldots\ldots\ldots, d_n$, then VSM uses the equation (1) to represent the relevancies between the given query and each of the corpus documents.

$$Sim(q, d_k) = \frac{\sum_i^m w_{t_i,d_k} w_{t_i,q}}{\sqrt{\sum_i^m w_{t_i,d_k}^2} \cdot \sqrt{\sum_i^m w_{t_i,q}^2}} \qquad (1)$$

Here, $Sim(q, d_k)$ numerically denotes the matching score for the relevancy between the given query q and the document d_k in the corpus. This value is also known as the similarity value between the query *q* and the document d_k. Moreover, w_{t_i,d_k}, $w_{t_i,q}$ denote the term weights for the term t_i in document d_k and query q respectively. Basically, term weight represents a measure for the frequency of occurrence of the terms in each corpus document. Moreover, the VSM uses these weights to transform the query and the text into vectors. These weights are determined by using the *tf-idf* (term frequency-inverse document frequency) weight scheme given by the equation (2) (Tasi, Huang, Liu, & Huang, 2012).

In equation (1), *m* denotes the number of terms in the given query and the summation of the right hand side of the equation takes up to *m* in order to compare each term in the query and the relevant terms in the corpus document.

$$tf - idf\ weight\ (w_{t_i,d_k}) = \frac{n_{t_i,d_k}}{N_{d_k}} . ln \frac{|D_{tot}|}{|D_{t_i}|} \qquad (2)$$

Here n_{t_i,d_k} N_{d_k} , D_{tot} and D_{t_i} denote the number of occurrences of term t_i (considered term) in document d_k , the number of occurrences of all terms in document d_k , the total

number of documents in the corpus and the number of documents where term t_i appears respectively.

This model normally determines the relevancy between the query and the document by comparing the number of occurrences of the query terms (term frequencies) with the similar terms (to the query terms) of each of the document in the corpus. For example, if the query is 'Library Science', then this model looks for how many times the terms 'Library' and 'Science' occur in each of the document in the corpus. According to that, it determines the *tf-idf* value for each document using the equation (2) and corresponding similarity values (using equation (1)) for the query and each of the corpus documents. Finally, the document which corresponds to the highest similarity value is considered as the most relevant corpus document to the given query.

The Problem

In general, VSM works well for when there are corpus documents with different frequencies for the same terms or at least corpus documents with different number of total terms in them. However, when the corpus includes two or more documents with the same number of query terms and the same number of total terms, the above VSM is unable to distinctly identify them from each other. In that case, the equation (1) gives the same similarity value for all those corpus documents. Hence, one cannot use this IR model as a general solution for IR. Figure 1 illustrates the problem more clearly.

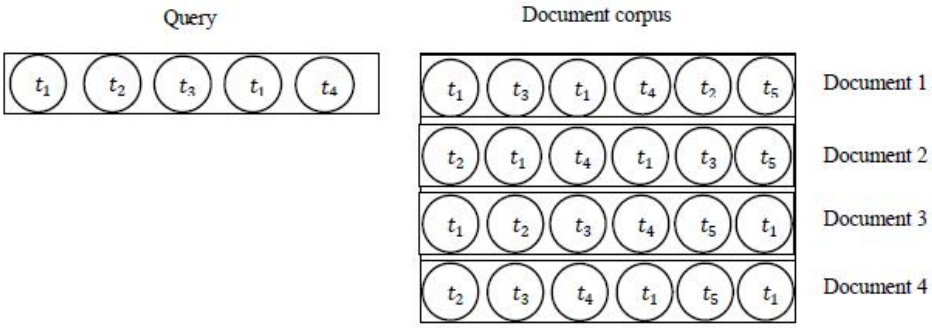


Figure 1. Problem of the conventional VSM

In the above figure, the query is consisted with five words and two of them are similar. As all the corpus documents are consisted with the same number of total terms and the same term frequencies, equation (1) calculates same similarity values for all the four documents. Therefore, although the four corpus documents are related to query in four different ways, there is no proper way to distinctly identify them from each other.

Following is another example to elaborate the problem furthermore.

Consider two corpus documents which are consisted with the same number of occurrences of the words 'Library' and 'Science'. Also if they include the same number of total terms, then the equation (2) gives the same *tf-idf* weight for both these documents as all the values for the variables of equation (2) are same for these two documents. As a result, the similarity values calculated by equation (1) also gives the same values for both these documents. Therefore, there is no any way to determine which document is mostly related to the given query 'Library Science'.

Related Works

The VSM (Salton, Wong, & Yang, 1975) considers the index representations and the query as vectors embedded in a geometric space where specific axioms and definitions are applied. Although this model is not so difficult to implement, it has certain limitations too. Becker and Kuropka (2003) as well as Manwar, Mahalle, Chinchkhede and Chavan (2012) emphasize that the VSM assumption of independent terms leads to problems with synonyms (different words with similar meanings) or strongly related terms. As a result, VSM is unable to identify some relevant corpus documents with the query just because they do not share the same terms. In addition, according to Khemani (n.d.), VSM requires a lot of computational time to process a request as it has to perform large amount of calculations. Moreover, it is necessary to recalculate all vectors at each time a new term is added into the term space. Therefore, some attempts have already been taken to eradicate the drawbacks of the existing VSM.

A vector based approach called Topic-based Vector Space Model (TVSM) enables to determine the document similarities within a relational-database (Becker, & Kuropka, 2003). In contrast to the VSM, TVSM has the advantage of not assuming independence of the terms which leads to full integration of stop words, stemming and thesaurus into the model. Silva, Souza and Santos (2004) have developed an extension to the VSM to reflect the dependence semantics among the terms. A Generalized Vector Space Model (GVSM) is introduced by Tsatsaronis and Panagiotopoulou (2009) in order to expand the VSM by incorporating the semantic information to the model. Furthermore, a few adaptations and applications of VSM are also available among the literature. Castells, Fernandez and Vallet (2007) have adapted the classic VSM to build an ontology based IR system. This system has used an ontology to improve the search capabilities over large document repositories. Another VSM based application called Enhanced Topic-based Vector Space Model (eTVSM) has been developed by Santos, Laorden, Sanz and Bringas (2012) to develop a model for semantics-aware spam filtering. In addition to this, Zeng, Lu and Gu (2008) have formulated an another VSM based approach for email classification. This approach intends to increase the accuracy of classification even under certain limitations.

The above mentioned numerous attempts reveal that the existing model of VSM has not yet been sufficiently developed. Out of number of negative aspects, most of the studies concentrate to solve the problems arising due to its basic assumption of term independence. Therefore, so far there is no enough studies have carried out to address the difficulty emphasize by this paper.

Besides the VSM approaches for IR, there are other approaches as well. Out of number of other existing models, the Boolean model (Salton, Fox, & Wu, 1983), Region model (Burkowski, 1992; Clarke, Cormack, & Burkowski, 1995), Probabilistic approaches (Fuhr, 1989), Bayesian network models (Turtle, & Croft, 1991; Metzler, & Croft 2004) and Language models (Ponte, & Croft, 1998; Miller, Leek, & Schwartz, 1999) are the most known systems in the field.

Methodology

The goal of this paper is, to introduce a new IR component which can be used to discriminate two or more documents those which cannot distinctly identify by using the VSM given by the equation (1). However, the proposed new IR component compares the distance between each and every two terms of the query with the distance between the same terms in each of the document in the corpus. Figure 2 has been illustrated this setting. In the figure 2, circles labeled with "t_1", "t_2", etc. are the occurrences of query terms in the document, while those labeled with "x" are the other terms that are not appearing in the query.

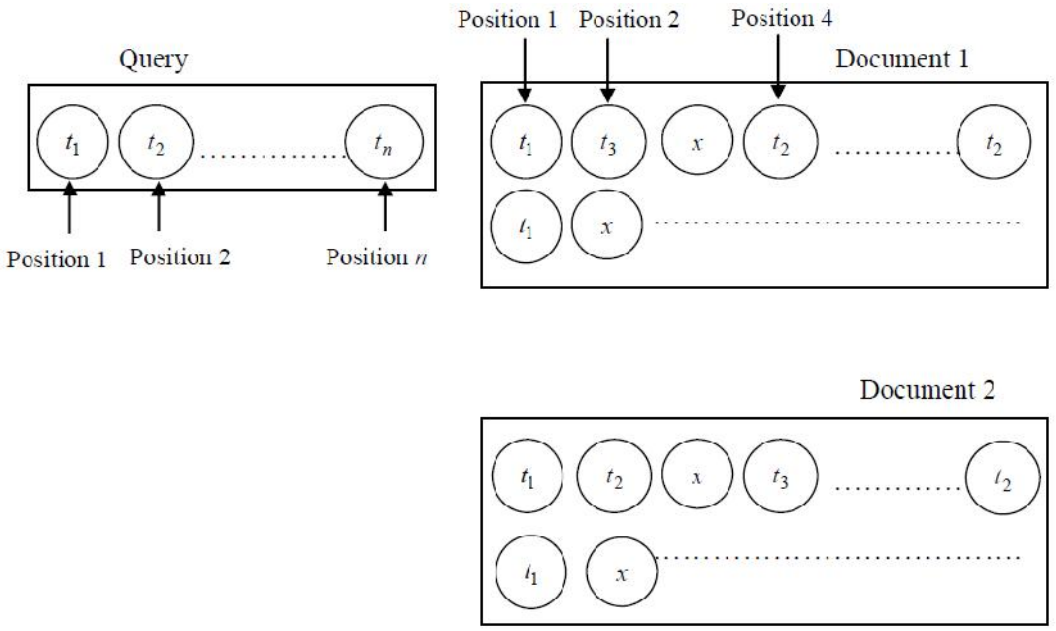


Figure 2. Original appearance of the terms in query and document

In the query, t_1 expresses the first term of the query and t_2 denotes the second term. To make this to a real world example, assume that t_1 (appears at position 1 of the query) and t_2 (appears at position 2 of the query) correspond to the words "Library" and "Science". Therefore in the query, these two terms apart with distance 1 (difference between the

positions of these two terms). However in document 1, these two terms apart with the distance 3 (for this explanation we consider only first appearances of these two terms) while them apart with the distance 1 in the document 2. Therefore, the query should much relevant to the document 2 as the phrase "Library Science" in the query appears in the same way (i.e. with the same distance between the two terms) in the document 2. However, these two terms occurs in document 1 with having dissimilar distance to the corresponding distance of the query. Therefore, perhaps the document 1 may about a subject like 'Science', 'Library' or 'Use of library for science' rather than it is about 'Library science'. Considering this situation, we build a new IR model to determine the similarity between a query and a document by concerning the distance between each two terms of the query with the distance between the same terms in each of the document in the corpus. The order of their appearances is also considered in addition to the distance.

To determine the relevancy in between the query and the documents, this model has been used the *cosine* similarity technique (Singhal, 2001).

In mathematics, cosine similarity is a measure of relevancy between two vectors. It measures the *cosine* of the angle between them. If the two vectors are ***A*** and ***B***, then the *cosine* similarity of them is defined as follows (Becker, & Kuropka, 2003).

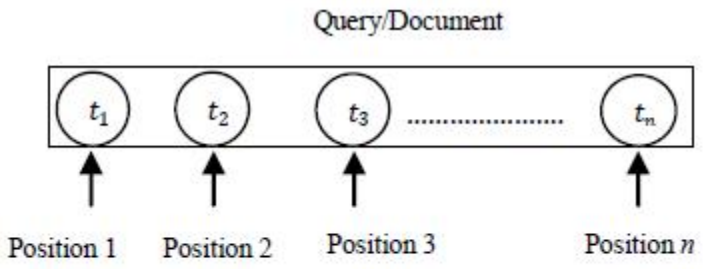
Cosine similarity = $$\frac{\boldsymbol{A}.\boldsymbol{B}}{\|\boldsymbol{A}\|\,\|\boldsymbol{B}\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Here, the two vectors ***A*** and ***B*** have *n* number of components and *i* denote each of these components.

In order to apply above *cosine* similarity into our problem, we obtain the distances between the terms of the query and the distances between the same terms of the documents as vectors. Therefore, *cosine* similarity is obtained as the vector multiplication of the distances among the terms in the query with the distances among the same terms in the document, in their numerical form. In this case, sequential positions of terms in the

query (or document) are considered as their distance from the initial term of the query (or document). Therefore, the distance between two terms become as the difference between the corresponding positions of them.

Figure 3 shows the way that we have calculated the distances between the terms of the query or the documents.



Distance between term t_3 and t_1 = position 3-position 1
= 3-1
= 2

Distance between term t_n and t_2 = position n-position 2
= n-2

Figure 3. Calculating distances between words

Using above simple representations for the distances between the terms, a much advanced weight component is developed to determine the distance between each two terms of the query. This is given by the equation (3).

For i = 1,2,3,……., n and j = 1,2,3,……., n ; where i and j denotes the positions of the query terms and n is the position of the last term in the query.

$$QL_{i,j} = \frac{\sum_{l \in S_i} \sum_{m \in S_j} \frac{1}{(m-l)}}{d_{i,j}} \quad ; \text{ for } m > l, Q_{t_i} \neq Q_{t_e} \text{ for } e < i \text{ and } \quad (3)$$

$$Q_{t_j} \neq Q_{t_f} \text{ for } f < j$$

Here the terms Q_{t_i} and $QL_{i,j}$ denote the i^{th} term in the query and the weight component for distance between i^{th} and j^{th} query terms respectively.

Here set S_r is defined as,

$$S_r = \{v : Q_{t_v} = Q_{t_r}\} \quad ; \text{for } r = 1,2,3,\ldots\ldots,n$$

$d_{i,j}$ denotes the number of occurrences of v such that each $v \in S_i < v \in S_j$.

Another set P is defined as,

$$P = \{\{i, j\} : \text{for } m > l,\ Q_{t_i} \neq Q_{t_e} \text{ for } e < i \text{ and } Q_{t_i} \neq Q_{t_f} \text{ for } f < j\}$$

The set *P* is defined with a view to simplify the notations of equation (5).

Here (*m-l*) represents the distance between two query terms and the reciprocal values of them (i.e. $\frac{1}{(m-l)}$) have been used in order to minimize the effects which can be badly affected in the comparison of short distance query terms with similar type of long distance document terms. This can be seen in equation (3) as well as equation (4). The set S_r is defined to categorize positions of the (similar) terms of the query and it makes easy to compute the values for $QL_{i,j}$. In equation (3), summation over $l \in S_i$ and $m \in S_j$ are taken to measure the reciprocal values of the distances between each couple of terms and to add together all such distances if there are multiple similar terms in the query. Moreover, $d_{i,j}$ is used to normalize the $QL_{i,j}$ whenever there are multiple similar terms in the query and since it is needed to consider all their distances to obtain $QL_{i,j}$. However, other

constraints are added to the equation (3) in order to avoid calculating the same distance twice.

Following the same way, a similar type of formula (to equation (3)) can be developed for the document representation too. But here, the terms and positions should come from the domain of the corresponding documents. The equation (4) shows the distance comparison among the document terms relevant to i^{th} and j^{th} terms of the query.

$$DL_{i,j} = \frac{\sum_{l \in S'_i} \sum_{m \in S'_j} \frac{1}{(m-l)}}{d'_{i,j}} \text{ ; for } m > l, \quad Q_{t_i} \neq Q_{t_e} \text{ for } e < i \text{ and} \qquad (4)$$

$$Q_{t_j} \neq Q_{t_f} \text{ for } f < j$$

With the equation (4), S'_r can be defined as,

$$S'_r = \{v : D_{t_v} = Q_{t_r}\} \quad \text{; where } r = 1,2,3,\ldots\ldots\ldots, n$$

Here the terms D_{t_v} and $DL_{i,j}$ denote the v^{th} term in the document and the weight for distance between document terms which are equal to i^{th} and j^{th} query terms respectively.

$d'_{i,j}$ denotes the number of occurrences of v such that each $v \in S'_i < v \in S'_j$.

Now it is possible to use the *cosine* similarity measure to obtain a new IR model to determine the distance similarities among the terms of the query and the documents. This newly developed IR model is given by the equation (5).

$$Sim\ (q, d_k) = \frac{\sum_{\alpha} QL_{\alpha} . DL_{\alpha}}{\sqrt{\sum_{\alpha \in P} QL_{\alpha}^{2}} . \sqrt{\sum_{\alpha \in P} DL_{\alpha}^{2}}} \qquad \text{Here } \alpha \text{ denotes } \{i, j\}. \qquad (5)$$

Evaluation and Results

Evaluation of the new model was done by using the following example.

We assumed that the query q has been given as 'Ceylon Library Research' and there are three documents (d_1, d_2 and d_3) in the corpus with the following natural structures. This each document has the total number of 600 terms.

Document d_1:

Positions of the term 'Ceylon' - 1, 25, 88, 191, 226, 351, 476

Positions of the term 'Library' - 2, 38, 89, 192, 277, 386, 477, 521

Positions of the term 'Research' - 39, 90, 228, 290, 315, 415, 478, 535, 576

Document d_2:

Positions of the term 'Ceylon' - 1, 34, 71, 186, 271, 347, 490

Positions of the term 'Library' - 2, 20, 72, 115, 187, 348, 390, 491

Positions of the term 'Research' - 3, 21, 36, 116, 188, 273, 349, 492, 519

Document d_3:

It does not contain either one of the above terms.

Now it has been used the VSM and the new model to obtain the similarity values for the documents d_1 , d_2 and d_3.

Using the equation (1), following similarity values are obtained for the documents d_1, d_2 and d_3.

$Sim(q, d_1) = 0.994835$

$Sim(q, d_2) = 0.994835$

$Sim(q, d_3) = 0$

These three similarity values are calculated based on the term frequencies while the following similarity values are based on the distances among the terms.

Using the equation (5), following similarity values are obtained for the same three documents.

$Sim(q, d_1) = 0.968595$

$Sim(q, d_2) = 0.998475$

$Sim(q, d_3) = 0$

According to the above results, it is clear that if we use the VSM, there is no way to distinctly identify the documents d_1 and d_2 since their term frequencies are equal. But it is possible to overcome this problem by using the equation (5) instead of (1). Moreover, similarity calculations with equation (5) indicate the query has a higher relevancy with the document d_2 than the document d_1.

Discussion

By using the equation (5) and following the similar way, it is possible to obtain the similarity values $Sim(q, d_1),\ Sim(q, d_2),\ Sim(q, d_3),$ etc. for the documents $d_1, d_2, d_3,$ etc. which are in a corpus. Since the *cosine* similarity is less than 1, the values obtain for the above equation should not exceed the value 1 and document priorities should be assigned from the highest value to the lowest (i.e. the document which achieves the highest value for $Sim(q, d_k)$, is the most suited document for the given query).

This work mainly focused on obtaining an alternative method to overcome one major problem which comes with the use of VSM. Even though there exist some extensions for the vector space model like topic-based vector space model (Becker, & Kuropka, 2003), enhance topic-based vector space model (Polyvyanyy, & Kuropka, 2007) and methods such as rank reductions which increase the expecting of relevant document returns (Berry, Drmac, & Jessup, 1999), viewing the problem from a different kind of view point has the

same important as going deeper from traditional perspectives. The Gravitation-Based Model (Shi, Wen, Yu, Song, & Ma, 2005) is a very good example for such kind of specific perspective. Even though there is a wide difference between IR and gravitation, it is remarkable of using the Newton's model for gravitation to build up an IR model.

In this paper, with a view to determine the *tf-idf* weight function, it has been used the term frequency (*tf*) in the form of $\frac{n_{t_i,d_k}}{N_{d_k}}$ and inverse document frequency (*idf*) as $\ln\frac{|D_{tot}|}{|D_{t_i}|}$.

Even if it is the case, there is the possibility of using various types of *tf* and *idf* functions instead of the forms which have been used here.

As the future work of this research, there still remain some major tasks to be achieved. One of them is that, it should have to be checked that whether the new model is also some kind of indicator of the term frequencies too. If it is not, then it is necessary to modify the new similarity function in a way that it can evaluate the effect of term frequencies as well. In addition to these vital tasks, implementation of this model in a freely available IR search engine or the development of a simple search engine in order to use this new model is also too important for the progress of this work.

Conclusion

The given example has been clearly showed that the newly developed similarity function (equation (5)) earns the distinct values 0.968595 and 0.998475, while the VSM earns the similar value 0.994835 for the same set of documents. Therefore, using VSM, there is no way to distinctly identify the document d_1 from d_2 since their term frequencies are equal and as they have the same number of terms in the documents. To get rid from this, we can use equation (5) instead of (1). Furthermore, these results imply that the highest similarity value 0.998475 is earned due to the higher relevancy of the document d_2 to the given query. Document d_1 has been earned its similarity value as 0.968595 and implies its fewer relevancies (relative to the document d_2) to the query. Since the similarity value of the document d_3 is zero, it has the lowest relevancy with the given query. Therefore

finally, the new model provides an alternative way to discriminate two or more documents from each other according to the given query when they have similar term frequencies and same number of total terms.

References

Becker, J., & Kuropka, D. (2003). Topic-based Vector Space Model. *Proceedings of Business Information Systems 2003.* Retrieved from http://www.kuropka.net/files/TVSM.pdf

Berry, M. W., Drmac Z., & Jessup, E. R. (1999). Matrices, Vector Spaces, and Information Retrieval. *SIAM Review,* 41(2), 335-362. Retrieved from http://delab.csd.auth.gr/~dimitris/courses/ir_spring07/papers/Matrices_vector_spaces_and_information_retrieval.pdf

Burkowski, F. (1992). Retrieval activities in a database consisting of heterogeneous collections of structured texts. *Proceedings of the 15th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92)*, 112-125.

Castells, P., Fernandez, M., & Vallet, D. (2007). An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval. *Knowledge and Data Engineering, IEEE Transactions*. 19 (2), 261-272. Retrieved from http://ir.ii.uam.es/~search/publications/semantic-search-tkde07.pdf

Clarke, C., Cormack, G., & Burkowski, F. (1995). Algebra for Structured Text Search and a Framework for its Implementation. *The Computer Journal*, 38 (1), 43-56.

Fuhr, N. (1989). Models for Retrieval with Probabilistic Indexing. *Information processing and management*, 25 (1), 55-72.

Hiemstra, D. (n.d.). Information Retrieval Models. Retrieved from http://wwwhome.cs.utwente.nl/~hiemstra/papers/IRModelsTutorial-draft.pdf

Khemani, D. (n.d.). Vector Space Model. Retrieved from http://aidblab.cse.iitm.ac.in/cs625/10.VectorSpace-model.pdf

Manwar, A. B., Mahalle, H. S., Chinchkhede, K. D., & Chavan, V. (2012). A Vector Space Model for Information Retrieval: A MATLAB Approach. *Indian Journal of Computer Science and Engineering (IJCSE)*, 3 (2), 222-229. Retrieved from http://www.ijcse.com/docs/INDJCSE12-03-02-028.pdf

Metzler, D., & Croft, W. (2004). Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40 (5), 735-750.

Miller, D., Leek, T., & Schwartz, R. (1999). A Hidden Markov Model Iinformation Retrieval System. *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, 214-221.

Polyvyanyy, A., & Kuropka, D. (2007). A Quantitative Evaluation of the Enhanced Topic-based Vector Space Model. *Technical Report of the Hasso-Plattner-Institute*. Retrieved from http://www.hpi.uniposdam.de/fileadmin/hpi/source/Technische_Berichte/HPI_19_A_quantitative_evaluation.pdf

Ponte, J., & Croft, W. (1998). A Language Modeling Approach to Information Retrieval. *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, 275-281.

Salton, G., Fox, E. A., & Wu, H. (1983). Extended Boolean Information Retrieval. *Communications of the ACM*, 26 (11), 1022-1036. Retrieved from http://neuron.csie.ntust.edu.tw/homework/93/Fuzzy/%E6%97%A5%E9%96%93%E9%83%A8/homework_1/D9009204/Fuzzy%20Homework%20I.files/p1022-salton.pdf

Salton, G., & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.

Salton, G., Wong, A., & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, 18(11), 613-620. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.7453&rep=rep1&type=pdf

Santos, I., Laorden, C., Sanz, B., & Bringas, P. G. (2012). Enhanced Topic-based Vector Space Model for Semantics-aware Spam Filtering. *Expert Systems with Applications*, 39 (1), 437-444. Retrieved from

http://paginaspersonales.deusto.es/isantos/publications/2012/Santos_2012_ESWA_Enhanced_Topic_Based_Vector_Space_Model_For_Spam_Filtering.pdf

Shi, S., Wen, J. R., Yu, Q., Song, R., & Ma W. Y. (2005). Gravitation-Based Model for Information Retrieval. *Proceedings of the 28th Annual International ACM SIGIR Conference (SIGIR 2005)*. Retrieved from http://research.microsoft.com/en-us/um/people/jrwen/jrwen_files/publications/GBM_SIGIR05.pdf

Silva, I. R., Souza, J. N, & Santos, K. S. (2004). Dependence among Terms in Vector Space Model. *Proceedings of the Database Engineering and Applications Symposium*, 97-102. Retrieved from http://www.lbd.dcc.ufmg.br/colecoes/sbbd/2004/paper63.pdf

Singhal, A. (2001). Modern Information Retrieval: A Brief Overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24 (1), 35-42. Retrieved from http://singhal.info/ieee2001.pdf

Tasi, C. S., Huang, Y. M., Liu, C. H., & Huang, Y. M. (2012). Applying VSM and LCS to Develop an Integrated Text Retrieval Mechanism. *Expert Systems with Applications*, 39, 3974-3982.

Tsatsaronis, G., & Panagiotopoulou, V. (2009). A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness. *Proceedings of the EACL 2009 Student Research Workshop*, 70-78. Retrieved from http://aclweb.org/anthology//E/E09/E09-3009.pdf

Turtle, H., & Croft, W. (1991). Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9 (3), 187-222.

Zeng, C., Lu, Z., Gu, J. (2008). A New Approach to Email Classification Using Concept Vector Space Model. *Proceeding of the Future Generation Communication and Networking Symposia, 2008, FGCNS '08*, 162-166.