VFAST Transactions on Software Engineering
http://vfast.org/journals/index.php/VTSE@ 2018 ISSN(e): 2309-3978;ISSN(p): 2411-6246
Volume 6, Number 1, January-December, 2018

pp. 07-14

COMPARISON AND EVALUATION OF INFORMATION RETRIEVAL MODELS

AGHA AZEEM REHMA[1], MAZHAR JAVED AWAN*[1], ILYAS BUTT[1]
[1]Department of Computer Science, University of Management and Technology, Lahore, Pakistan
Email: mazhar.awan@umt.edu.pk

ABSTRACT. *Recently data is growing day by day in the internet . Data is in the form of Structured , unstructured and semi structured in nature. Information Retrieval is the field which is regarding the study of retrieval of unstructured or semi structured documents. For every aspect IR is being used in there are different models being used in them. There are so many models in so many applications,each having some relation to one another.In this paper we will evaluate and compare various IR model techniques and algorithms and see which model excels in which field of application.*
Keywords LSI(Latent Semantic Indexing); SVD(Singular Value Decomposition); BIM(Binary Interdependence Model); PRB(Probilistic); DAG(Directed Acyclic Graph);IR (Information retrieval).

1. **Introduction.** In the world of technology,retrieving information from one place to another is a key ingredient of the software world which is the base of all technology,as there is no such technology that doesn't use information.**Information retrieval** is the field at which the user sets a query and the system that is in use has to obey that query and retrieve information as required.Theprocess only initiates when the query is made.For it to do so,there are multiple ways in retrieving information with multiple algorithms and techniques.Not all techniques and models excel in every query processing.Eachis better than the other in their own field of query processing.We will now discuss about these models and see how they work and what kind of algorithm and techniques are being used in them.Information Retrieval Models can be categorized in 3 models based on their technique of query processing. These are **Set-Theoretic Models**,**Probabilistic Models** and **Algebraic Models.**

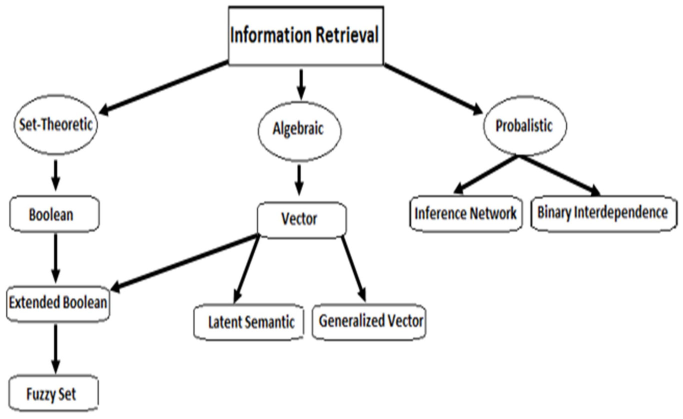

Figure-1: Taxonomy.Shows the different models we will discuss and their relation between one another.

1.1 Set Theoretic Model. These type of models all heavily depend on the term matching and the order of words which they are placed,these models are the basic models which you can see in many applications of IR.As these models rely on term matches,after the matches they use Mathematical logic Set Theory which include subset's,intersection,union and many other operations.The documents in these type of models are representated as terms or words or set of phrases.But the problem with these type of models is that they rely on term matches and the number of term occurrences in the documents judges the relevancy of the documents,this can lead to unwanted documents retrieval or less document retrieval.We will discuss the following Set Theoretic Models:-

- Boolean Model
- Extended Boolean Model
- Fuzzy Model

1.2 Probabilistic Model. If we have a corpus of known related and unrelated documents than we can start finding that a particular term is present in the relevant document or not if "t" is the term then by p (t | R=1) (*R=1 shows that we are finding probability of the relevance. And R=0 shows that we are finding probability of non relevance*) we can find the probability of the term to be present in the document. This is the basic classifier which tells us whether the document is relevant or not. An IR system has a query as an input from the user which is converted into a representation and documents which are also converted into a specific representation. On the basis of query representation and document representation IR systems try to identify which documents are relevant. Other models i.e. Boolean or vector space follow traditional method of indexing the terms for matching. But given a query any search engine is quite uncertain that which are the relevant documents. Here comes the probability theory which helps us to find the relevant documents under such uncertainty. A probabilistic model ranks the retrieved documents in descending order [8]. There are some challenges in this model which are as follows:

- Probability of relevance is based upon presence of term in query and documents.
- We need to start with initial approximate state and then need to improve on the basis of feedback.
- We need to make some simplifying assumptions.

There are many probabilistic models here we will discuss the following two:

- Inference network model.
- Binary independence.

1.3. Algebraic Model. Such models that rely on Algebra and Mathematics in order for their algorithms to run all lay in this field of model.Many of the Algebraic Models use complex calculations while others are simple as a+b.Most models heavily depended on words and terms matches in order to retrieve the desired query,they all depend on the lexical matches and when the database of documents is large it can be quite a hassle to get the relevant document. This model or all the models which lay in this type uses the mathematical computation in order to get the relevancy of each document. The Algebraic Models we will discuss in this paper are the following: -

- Vector Model
- Generalized Vector Model
- Latent Semantic Indexing Model

2. Related Work

2.1.Set-Theoretic

2.1.1. Boolean Model. This model is the first and most classical model being used in IR.Itis widely being used in search engines as this simple and yet fast model is better in such type of searching techniques which considers speed over precision. The Boolean model uses binary digits in order to separate the relevant and irrelevant documents. The type of searches it does is the term matches it finds in the document. meaning that for a document to be retrieved it is necessary to have the searched term.

Now as we can see from the name "Boolean", it uses Boolean operators between terms when multiple words are used in a query. **AND** and **OR** functions are used in this model. Suppose we have documents **D** and terms **T**.Each term is considered to be an index(*the,is,be,e.t.c aredisregarded unless they are being used in a phrase*).When multiple terms are being used in a query, they are packed together in either **OR**or**AND** Boolean operators,**T1 OR T2 OR T3….TN.**The same is with the **AND**operation. Once is the query is made, it searches through the documents **D.**Each document having occurrences of that term **T** is considered to be relevant and is selected. The selection of

documents is straightforward, it is either yes or no, relevant or irrelevant,1 or 0.
Due to this binary relevance no ranking can be done. The results of **OR**query are usually overwhelmed due to the nature of **OR**operator ie.(if a document have even a single query word it is considered relevant) this produces a huge number of result documents on the other hand the results of **AND** query are heavily damped due to the nature of **AND** operator ie.(if a document is missing even one word of the query it is not retrieved).
Also only the query term is searched through the documents, related words or synonyms of that term are disregarded. So any query with the wrong term may end up giving the wrong document. Also it is quite an issue to convert a query into its Boolean expression.

2.1.2. Extended Boolean Model. We have discussed above about the Boolean model on how simple it is yet having great flaws, we now try to improve it, to improve it we need to make **AND**a little less "Andy" and **OR** a little less "Ory" ie(we need to damp the effects of Boolean operators and make them more flexible)
So we designed the Extended Boolean Model. The improvements that have been made is that it uses partial matching of terms in documents, so even though there is less occurrences of a query term in 1 document than the other, it will be considered as relevant.
Term weights are being used which were not in the **Standard Boolean Model,** the documents are considered as vector's, here the model not only is heavily based on the previous **Boolean model**, it is also based on the **Vector Space Model** as well, it combines both the characteristics of the models. [1]
So for every document vector, there will be term weights attached with it, we can say that **D1={w1,w2,w3,w4…..wn}** where ***D1*** is the document and ***w*** is the weight of the term[2],the weights will help to show relevancy according to that term.
If we have a query **Q** and the query contains terms **T,**we need to find the relevant documents **D**,firstly we will use a Boolean operators and with that check the weights of the terms **T** in every document D.Once we check the weights,we will gather the documents with the best weights and retrieve them.Eventually we will have our desired output.
2.1.3. Fuzzy Model. Fuzzy Logic Model or Fuzzy Retrieval is a type of model that it based on the searches of documents on their weights or scores that is given to the documents. The varying weights that is given to the documents helps to optimize searches and to retrieve relevant documents. This model is actually based on both **Boolean Extended model** and **Fuzzy SetTheory** (*used in Mathematics*).
Fuzzy Retrieval differs from conventional models because of its searches on the documents weights or scores. Conventional model use either 1 or 0, yes or no to determine whether the document is relevant or irrelevant, whereas in the **Fuzzy Retrieval** model the documents relevancy ranges from 0 to 1, this also includes relevant documents (given value between 0 and 1) and irrelevant documents (given value 0).The varying value (0 ~ 1) that are given to documents is the base at which the better relevant documents is retrieved.
The fuzzy retrieval model uses the following concepts/key features in order to perform of retrieving documents which are as following:-
Functionf -> Performs and calculates the fuzzy set for each term for each document. Also known as weight of the term in document.
Fuzzy Sets - > it shows the fuzziness of the term which determines the relevancy in the documents
Now firstly we need to find a fuzzy function to create the fuzzy sets, the fuzzy function can either be as simple as **a+b** or complex as using **Integral of Cosines function**[2]. The Function ***f*** is determined first and a formula is used in them, where the formula can be any form of calculation, keeping in mind once the function f is determined and created, this must be used for the calculation of all the fuzzy sets that needs to be calculated for the terms at hand. That way the weights of the terms can be calculated in reference to that function.
After the function f is determined, we calculate the fuzziness/weight for each term, any number of parameters can be passed and the output would be the fuzziness of term, it is better if the output weight be in the range of 0 to 1.For each weight associated with the document, this will be later used for query processing. The queries generated for the fuzzy model is in either sets of **OR** queries or **AND** queries, combination of both is also used but later it is breakdown to the two following categories.
For **OR** queries Union is used between the fuzzy set of the two subject terms. Suppose we have two terms **t1** and **t2** and have document **d1**, we need to calculate the query **t1 OR t2** in **doc d1**.
First we calculate the fuzzy set of both terms in relevance to document d1 using the function***f***(determined by either the user or it will be given).
So we get $\boldsymbol{f1 = \{d1, w1\}}$[2].Where ***f1*** is the Fuzzy Set, ***d1*** is the document in reference, here it is document 1, ***w1*** is the weight associated to that term in that document, in this case it is term 1 **(*t1*)**
The same is done with ***t2*** and with the ***OR*** query we use union and we get $\boldsymbol{f1 \cup f2 =}$

$\boldsymbol{max}\{(\boldsymbol{d1},\boldsymbol{w1}),(\boldsymbol{d1},\boldsymbol{w2})\}$[2]. Where the Max() function determines the maximum value between the two weights
If we have an **AND** query between two terms we simply use intersection between the fuzzy sets of both terms $\boldsymbol{f1} \cap \boldsymbol{f2} = \boldsymbol{min}\{(\boldsymbol{d1},\boldsymbol{w1}),(\boldsymbol{d1},\boldsymbol{w2})\}$**.** Where the **min ()** function determines the minimum weight in between the two terms[2].

In this case we only used 1 document, where there are **N** documents, all the weights associated with each documents must be used in fuzzy set in order to determine the fuzziness of that term.

Fuzzy Model is based on the **Extended Boolean Model** which is in the Set-Theoretic Representation of Information Retrieval so we will compare it to that model.
We have seen that this model uses weights to determine the relevance of the term at hand to the document, this is also done in **Extended Boolean Model**, but what differs is the degree of weight it is given, it's given varying values which determines the relevancy of a term which gives a more precise relevance value, unlike in the **Boolean Model** which was given either 1 or 0 the fuzzy model takes an edge over it by its weight calculating technique.
But there is a problem as the fuzzy model does not take in concern over the number of occurrence of the term in the document, it only knows the weight of that term.
Fuzzy model is good compared to other models when it comes to the decision of relevancy when only the difference of fraction of weights is concerned. This gives a better judgment of queries. It's performance matches that of **vector space model** as the weights are ranked through a function and its simplicity derived from the **Booleanmodel** which we can say that the good attributes of both models are combined in one which is known as the **Fuzzy Model.**

2.2. Probabilistic

2.2.1. Inference network model. Networks always had been a important part of information retrieval. Now a days when there is a requirement of such systems which should be able to understand the content of query so that they could give more relevant documents. In this perspective information retrieval is a inference which require the mechanism of reasoning to find out the requirement of user. And network representation provides the same mechanism. One more additional point for this model is that it provides us with the same results for different representations. It also provide the facility to combine results from different queries. It provides the facility that a description of information need can be used to generate several query strategies and each with different information need and result provided to each would be according to its information requirements.

There is a clear requirement in the world of automated handling of uncertainty. And in the field of information retrieval inference model provide us this facility. There are two models with probabilistic approach which are as follows[7]:

- Bayesian inference networks
- Dempster-Shafer theory of evidence

Dempster-Shafer theory is not purely a network model so here we will discuss the Bayesian network.
A Bayesian inference is actually an acyclic dependency graph **(DAG)** where nodes shows constants and edges represent the dependency between the nodes. If "p" is a node which causes another node "q" then it is written as "p" to "q". Here "q" will have a link matrix p (p | q)link matrix will have all the information of these two variables. The link matrix will contain all the dependency information in it. If a node has two parents then link matrix will store the child's node dependency with both the nodes which causes him[7].
The basic model of inference consists of two parts:

- Query network.
- Document network.

Document query is the collection of documents including several documents using several types of representation. Once the document collection structure has been build it remains the same throughout the query process. And query is actually a single node which represents the user information need which may be represented in one or two queries. The information need get on changing as time passes with respect to response[7].We will now see what the uses of Inference Model are:-

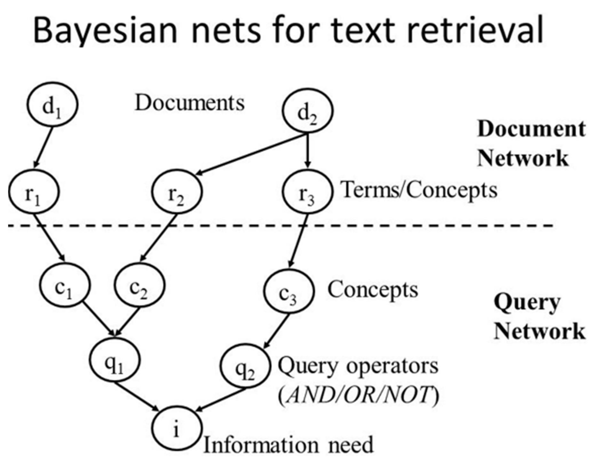

Figure-2: General representation of how the Bayesian Nets work.

- Document network is needed to build just for once for a given information set.
- It has traditional feedback method in it. As if the ranking is not correct we can add information to query network or in its structure to get improved results.
- In the set of documents we do not need to compare query with each document rather then that we will compare it with the subset of documents which provide us the highest probability of finding the query.

2.2.2. Binary independence model. Binary independence model (BIM) is used with PRB. It is a technique which involves some assumptions to for a probability function **p(R∣d,q)** here **binary** is equivalent to Boolean. It means that both documents and queries will be represented in binary incidence vector form. Document “d” is represented in vector form as x={x1,x2….,xm} if xi = 1 it means that term “t” is present in the document. And **independence** means that the terms in the documents are independent of each other there is no association between them. Actually this statement is far from correct but it works sometimes.
To get this model to work we need to know length of documents, term frequency, document frequency and other information that is needed to estimate the relevance of the documents. And we arrange the documents in the decreasing order of probability.
Here we assume that the relevance of each document is independent of the other documents. It is not a fruitful assumption because it may cause to give overlapping or approximately same documents.
By Bayesian rule [3]:-

$$P(R=1|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})}$$

$$P(R=0|\vec{x},\vec{q}) = \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})}$$

Where R=1 means it shows probability of relevance. And R=2 means it shows the probability of non-relevance. And document is represented as a vector “r”[9]. But how it is possible to compute all the probabilities every time therefore we use **p(R=1∣q) and p(R=0∣q)** as prior probabilities of retrieving a relevant or non-relevant document for a particular query.
A document is whether relevant or irrelevant the following equation must be fulfilled[9]:

$$P(R=1|\vec{x},\vec{q}) + P(R=0|\vec{x},\vec{q}) = 1$$

2.3. Algebraic Model

2.3.1. Vector Model. As we can see from the name,we are using vectors in this model to connect terms with documents.The vector model is a simple model by using vectors as a direction of relevancy of documents with the

terms.The model was proposed in response to the Boolean Model as it uses binary values to determine relevancy,which was considered to be vague when retrieving documents.
Every query and every document has vector associated with them,we can say that queries Q and documents D have their respective vectors, **Q={w1,w2,w3,…wN}** and **D={w1,w2,w3….wN}.**When the query matches one of the documents,the weight of the term is calculated,meaning that whatever term is considered,it's vector will have a non-zero value,which is it's weights[4].
The vector space is also considered,the vector dimension is all the unique words or terms being used in all the documents that the corpus includes[4].
The vector model has it's advantages over the other models,as it doesn't use binary values but rather term weights which gives a more relevance of the document.Partial matching is done as well as ranking the documents accorsing to their relevance.But there are problems as well with this model,it cannot find the related context of the term with other documents,meaning it's semantic sensitivity.Also that this model does not consider the format of the terms,the order in which the terms are generated are lost or disregarded when these are represented as vectors.This model also creates an issue regarding the pairwise of terms **orthogonality assumption**[3],meaning that no relation is considered between terms,even though it is vital in a sentence and the meaning changes when the terms are independent of each other.
2.3.2. Generalized Vector Model. The vector model was considered good but it's flaws were needed to be fixed,just like **the Extended Boolean** being the better version of the **Standard Boolean** ,the **Generalized Vector Model** is the better version of the **Vector Model**,eliminating some of it'sflaws.This model solves the issue of relatedness between the terms,thepairse orthogonal assumption is eliminated,meaning that the terms now share a relation with each other,which can find better relevant documents.
There was one vector space being used in the earlier version where terms and documents have independent vectors,now in this model they are considering another vector space where the term vectors are created by one another,meaning that multiple vectors joining together,to form a dependence on each other.This solves the issue of multiple terms having the same context(*synonymous*).The new dimension would be considered as a $\boldsymbol{2^{n}Dimension}$,as the vectors in that space and created by $\boldsymbol{2^{n}Vectors}$.[3].
Since than there are two basic field of **GVM** in which many effort's are being done,first is the relatedness of terms,inshort,finding a solution for the **synonym problem** of terms.Second is the computation of frequency of co-occurrence stats in very large corpus or documents having large number or unique terms[3].
2.3.1. Latent Semantic Indexing Model. This model is based on the Vector Model.The word semantic means to understand the meaning of the word which shows us a little on how it works.LSI just does not only matches the terms in the documents but also find such relevant documents which doesn't even have the query term in it.Unlike other IR models discussed above,this model uses the context of the term (*meaning or term*) and find such relevant documents which are in context of that query term.
Before we start the reader must have prior knowledge to Eigen vector values calculation and a little bit of matrix calculation in order to fully understand the working of this model and how it searches for the relevant documents.
We will first create a matrix X with rows and columns.The columns(*represented as n*) are the documents in the server or at hand which we are searching for the term in,the rows (*represented as m*) are the totals terms that are searchable and are used in those documents.Now we will fill in the matrix or we will already have a matrix filled.We would have something like this:-

	D 1	D 2 - - - -	D n
Term 1			
Term 2 | |			
Term n			

Figure-3: General Representation of Matrix X

All terms (*Term*) from 1 to n in rows and all the documents (*D*) from 1 to n in the columns.
(*m x n Matrix X*).Once the Table is filled or already given a filled table we will now calculate the Singular Value Decomposition (*SVD*) ***X=AB(CT)***[5].X is our Matrix (*m x n*).A is a Matrix (*m x r*) where r is the minimum of (*m x n*) and it holds the **eigenvectors** of X Matrix.B is having the **singular values** of A and it is a diagonal matrix .C matrix is the document vector of matrix X and ***(CT)*** is Transpose of matrix C.
Also that matrix A and C are orthogonal matrices which means ***C(CT)=A(AT)= I (Identity Matrix)***[5].Where ***CT***

and AT are Transpose of matrix C and A respectively.The three matrix and commonly known as ,B - >***Concept Matrix***,A - >***term-concept similarity matrix***,C - >***Document Concept Similarity Concept.***

Once we have filled the three matrices A,B and C we will now put a limit on the matrix or a **threshold (k)** to a reduced dimensionality,it's value is normally 2 or is set by the person calculating the SVD.Once we have the value of k,we will reduce all the matrices to that dimension,meaning convert all matrices to a k-dimensional space.Our new matrices will look like this.***X k =A k B k (CT)k***[5]**.**

Once we transform all the matrices,we will see at matrix ***(CT)k***.These values in this matrix are the eigenvector values,basically they will represent the coordinates of each documents in vector space,hense as we said before the value of k is normally 2 because the C matrix will at the end result the x and y coordinates of each document.

The query which the person at hand will establish will be in the form terms,for each term a singular matrix of 1 x n rows and columns."n" would be the total number of documents in Matrix X.The values would be either 1 or 0 depending if that term is in the document or not.So basically it would be like a binary form of 1's and 0's with each binary representing if that term is in that column document or not.

Once the query is generated we will now calculate the coordinates of the query at hand with the K threshold we used above which can be calculated by formula.***Q k =(QT) Ak (Bk)(Inverse)***[5]**.*****(QT)*** is the Transpose of Matrix of the original query we made.***(Bk)(Inverse)*** is the inverse of matrix ***B k***.At the end we will have the coordinates of the query in ***Q k.***

Now we will calculate the relevance of the query according to the documents.We will do this by calculating the query-document cosine similarities.We will use the following formula.***(q,d)*** = $\frac{(q.d)}{|q|\,|d|}$[6]**.**Where "**q**" is the query and "**d**" is the document in which we are finding the query .**(q.d)** is the dot product of "q" which is ***Q k*** Calculated above and "d" is that document vector coordinates calculated above and is stored in ***(CT)k***.**|q|** and **|d|** are determinants of the "**q**" and "**d**".

We will do this to all documents in the matrix X with the same query ***(q,d1), (q,d2), (q,d3)….. (q,dn)***.At the end we will have the values of relevance of that query in each document.Sort them in Descending order and the value which is the greatest will be the document with the most relevance value and the model will select that document for that query[6].

We see that LSI is relatively better as it provides a model which calculates the context of the query as well,this is only plausible when the query we generated has a synonym meaning and it confuses with another word and brings out the documents which we are not looking for.LSI excels in this field of synonyms terms and works well according to it.

The model also can also retrieve documents with multiple terms that put together form a different meaning,but to a limit,as the number of words related to each other in the query increases the effectiveness also decreases,so the LSI model is best kept at a limited of query terms.But there is a problem with that,the LSI model disregards the order of occurrence of the term in a sentence,meaning the order of words in a query sentence are ignored

The Only problem that we face is the computational power it needs,we saw above we need to calculate all the different matrices and further again calculate more,it's doesn't matter if the tokens and documents size are large or small,this effects on all models,the computational time and effectiveness is larger when related to the Vector Space Model.

We also see the K threshold being used here which helps in reducing the dimensionality of the terms which in return improves the time of computational calculation but this reduces the vector space and these vectors are not sparse anymore,meaning the values are close to 0.

REFERENCES

[1] Salton, G., Fox, E. A., & Wu, H. (1983). Extended Boolean information retrieval. *Communications of the ACM*, *26*(11), 1022-1036.

[2] Klir, G., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic* (Vol. 4). New Jersey: Prentice hall.

[3] Tsatsaronis, G., & Panagiotopoulou, V. (2009, April). A generalized vector space model for text retrieval based on semantic relatedness. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop* (pp. 70-78). Association for Computational Linguistics.

[4] Raghavan, V. V., & Wong, S. M. (1986). A critical analysis of vector space model for information retrieval. *Journal of the American Society for information Science*, *37*(5), 279-287.

[5] Hofmann, T. (2017, August). Probabilistic latent semantic indexing. In *ACM SIGIR Forum* (Vol. 51, No. 2, pp. 211-218). ACM.

[6] Grossman, D. A., & Frieder, O. (2012). *Information retrieval: Algorithms and heuristics* (Vol. 15). Springer Science & Business Media.

[7] Turtle, H., & Croft, W. B. (2017, August). Inference networks for document retrieval. In *ACM SIGIR Forum* (Vol. 51, No. 2, pp. 124-147). ACM.

[8] Ross, S. M. (2014). *Introduction to probability models*. Academic press.

[9] Mogotsi, I. C. (2010). Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval.