Survey on Relevance Calculation Methods

Uma Gajendragadkar
Research Scholar
COEP, University of Pune, India

Sarang Joshi, Ph.D
Professor
PICT, University of Pune, India

ABSTRACT
Relevance is one of the most important concepts in information retrieval. This paper discusses various approaches for relevance calculation available in literature. In Information Retrieval, relevance is how well a retrieved document or set of documents meets the information need. The paper summarizes, what relevance is and a survey of different methods of calculating relevance.

Our goal is to study existing methods and to identify usefulness of existing methods for multilingual data and to understand the prerequisites required for these methods while using mime types likes webpages, urls, xml apart from plain text.

General Terms
Information Retrieval, Pattern Recognition, Algorithms

Keywords
Relevance calculation, Information Retrieval

1. INTRODUCTION
Huge amount of documents are generated as the Internet is growing at a very high rate and digital document repositories mushroom. Searching for relevant information in these huge collections is a very difficult task. The goal of Information Retrieval Systems is to find relevant documents with respect to given query thus relevance is a central concept of information retrieval.[1][2] Most of the documents that are retrieved using information retrieval (IR) systems are found to be irrelevant. This has led to renewed interest in the concept of relevance, which is regarded as the "fundamental and central concept" in information sciences. Objective and system-derived relevance are now being either extended or replaced, at least theoretically, by subjective relevance concepts such as situational relevance and psychological relevance. In general, relevance is now regarded as a subjective, multidimensional, dynamic, and measurable concept. [3] Relevance can be stated in various ways: User relevance, System relevance etc. [4] In Information retrieval, a document or a piece of information is said to be relevant if it satisfies users' information need.
Relevance is a multidimensional cognitive concept whose meaning is largely dependent on users' perceptions of information and their own information need situations. [4]

There are various approaches for calculating relevance. These methods are summarized below. The 20 Newsgroups dataset is used to experiment with the relevance calculation methods. For experimenting in multilingual data, a Marathi graffiti dataset is constructed from graffiti in 'Sakal', a Marathi newspaper. [5] Marathi is spoken by about 70 million people in India and it uses Devanagari script.

2. TERM FREQUENCY - TF
Tf stands for term frequency. Term frequency in a document tells the measure of the importance of a term within the document. More the number of times a term appears in a document, the higher is the term frequency for that term. [6] [7] [8][9][10] Term frequency is widely used for relevance calculation.

In the Table1 we can see various term frequencies in the documents. It is observed from Table 1 that for the term 'God', document d24 is more relevant than document d14 or d13.

The term frequency is given by

$$tf = \frac{\ln(num)}{\ln(terms)}$$

where *terms* = number of terms in document *d*
num = number of occurrences of term *t* in document *d*

Equation 1 Term Frequency

In case of Marathi Graffiti 'कान पकडल्याशिवाय माझं एकही काम होत नाही … …. चहा पिण्याचंही ….' the term frequency for word 'कान ' is 1 . In this document the term has two meanings associated with it. First meaning is 'ear' and second implied meaning is the handle of a cup. Only term frequency is not adequate to reflect the relevance of the term in this case. Another example is 'मी पोहायला घाबरतो …. लोक मला पाण्यात पाहतील म्हणून.' Word to word translation is 'I'm afraid to swim….as people will see me under water' and implied meaning is 'I'm afraid to swim….as people will be jealous with me'. If we find term frequency of word 'पाण्यात' (in water) it is 1 which doesn't reflect the second relevance 'to be jealous with'.

Table 1 Term Frequency

Term/ Document	d 1	d 2	d 3	d 4	d 5	d 6	d 7	d 10	d 11	d 12	d 13	d 14	d 16	d 17	d 20	d 21	d 23	d 24	d 25
apple								**11**	**13**							**12**			
account	**2**			**3**					**1**	**2**									

arthritis										**1**				**1**					
believe	**6**	**1**	**1**		**3**	**3**	**1**			**2**	**2**	**2**			**2**		**7**	**9**	
bible	**18**		**5**	**2**	**3**	**1**	**4**	**1**		**4**							**4**	**23**	
cancer				**2**		**2**								**37**					
card								**14**								**3**			
christ	**21**	**16**	**8**	**40**	**17**	**12**	**10**			**3**							**9**	**68**	
cross				**2**	**1**	**2**			**2**	**4**							**2**		
emphasis					**1**													**21**	
father	**1**	**1**		**2**						**3**					**15**		**1**	**69**	
gain									**1**					**2**	**1**				
glory	**2**			**4**			**1**			**1**								**2**	
god	**56**	**10**		**76**	**3**	**16**	**30**			**8**	**2**	**2**			**3**		**5**	**116**	
health														**28**				**6**	
heart	**4**			**3**		**1**	**2**			**7**	**5**	**6**	**6**	**12**	**5**		**1**	**3**	**4**
MRI																			**8**
patient						**2**								**2**					
pray	**2**			**10**	**1**		**2**											**6**	
rational											**1**	**1**							
thought	**1**								**1**	**9**	**1**	**1**			**6**				**1**
world	**10**	**1**	**1**	**1**	**1**	**1**	**1**	**3**	**4**	**9**								**3**	

3. INVERSE DOCUMENT FREQUENCY- IDF

idf stands for inverse document frequency. It has a high score for rarer terms than for common terms or frequently occurring terms.

$$idf = ln\frac{D}{n}$$

where *D* = number of documents in the collection
n = number of documents containing the term *t*

Equation 2 Inverse Document Frequency

The inverse document frequency measures that if a term is frequently occurring term or it rarely occurs in all documents. It is division of the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient. [11] [12][7][8][10]

From Table 2, it is seen that the term 'MRI' is the rarest in the sample documents.

In case of Marathi Graffiti 'चुकून केलेल्या चुकीलाही चूक म्हणण्याची चूक चुकीचीच नाही का ? ' the inverse document frequency for word 'चूक ' is 1.69 .

Table 2 Inverse Document Frequency

Term	IDF
apple	**2.42**
account	**2.6**
arthritis	**3.12**
believe	**1.5**
bible	**1.73**
cancer	**2.6**
card	**3.12**
christ	**1.73**
cross	**2.27**
emphasis	**3.12**
father	**2.02**
gain	**2.6**
glory	**2.42**
god	**1.5**
health	**2.8**
MRI	**3.5**

4. TF-IDF TERM WEIGHTING

The Term Frequency and Inverse Document Frequency (Tf-Idf) is one of the ways to calculate relevance. To calculate the tf-idf weight of a term in a particular document, we need to calculate tf and idf values first and multiply tf by idf. The tf-idf weight of a term is the product of its tf weight and its idf weight. It is the best known weighting scheme in information retrieval. For scoring the documents, a Tf-Idf score is used. This score increases with the number of occurrences within a document and increases with the rarity of the term in the collection.

If the term frequency for a term is high in a document and it occurs less number of times in the complete collection of documents then it leads to high value of Tf-idf. So frequently occurring common terms are filtered out by Tf-idf. One can derive different forms of Tf-idf from a probabilistic retrieval model that resembles human relevance decision making. [12][7][13][14]

From Table 3, it can be seen that the document d24 is more relevant with the query term 'Bible' or 'Christ' or 'God' whereas d17 is more relevant with the query term 'cancer' or 'heart' etc.

In case of Marathi Graffiti 'चुकून केलेल्या चुकीलाही चूक म्हणण्याची चूक चुकीचीच नाही का ? ' the Tf-Idf for word 'चूक ' is 3.38 .

Table 3 Tf- Idf weighting

Term/ Doc	d1	d4	d5	d6	d7	d8	d 17	d24
apple	0	0	0	0	0	24.2	0	0
bible	31.1	3.46	5.19	1.73	6.92	0	0	39.8
cancer	0	5.2	0	5.2	0	0	96.2	0
christ	36.3	69.2	29.4	20.8	17.3	0	0	118
cross	0	4.54	2.27	4.54	0	0	0	0
father	2.02	4.04	0	0	0	0	0	139
gain	0	0	0	0	0	0	5.2	0
glory	4.84	9.68	0	0	2.42	0	0	4.84
god	84	114	4.5	24	45	0	0	174
heart	5.52	4.14	0	1.38	2.76	0	0	4.14

5. COSINE SIMILARITY

In cosine similarity, documents and queries are converted into representative vectors. One dimension is allocated for each separate term. The term weight represents the occurrence of a term in the document using a non-zero value. he term weights can be calculated in many ways. Using Tf-idf to compute term weights is one of the popular methods.

Documents are converted into representative vectors in term space. Terms are usually stems. Binary vectors of terms are used to represent the documents. Queries are also converted into binary vectors. Ranking of retrieved documents is done using the vector distance between document and the query. Similarity between query and document is based on length and direction of their vectors. [7]

Cosine similarity is given by

$$\cos(d_i, q_j) = \frac{\sum_{k=1}^{n}(term_{ik}.qterm_{jk})}{\sqrt{\sum_{k=1}^{n}(term_{ik})^2.\sum_{k=1}^{n}(qterm_{jk})^2}}$$

Equation 3 Cosine Similarity

From Table 4, one can conclude that document d5 is more similar to d12, d6 and very less similar to d8, d9 etc.

For Marathi illustration, let us consider two graffiti text documents. First is 'चुकून केलेल्या चुकीलाही चूक म्हणण्याची चूक चुकीचीच नाही का ? ' and second is 'मी पोहायला घाबरतो …. लोक मला पाण्यात पाहतील म्हणून. '. The value of cosine similarity between the two is 0.0 . They are not similar.

Table 4 Cosine Similarity between documents

Doc	d4	d5	d6	d7	d8	d9	d 10	d 11	d 12
d1	0.87	0.72	0.84	0.74	0.46	0.38	0.58	0.65	0.86
d2	0.8	0.69	0.79	0.56	0.41	0.33	0.5	0.55	0.78
d3	0.79	0.72	0.76	0.55	0.49	0.39	0.58	0.63	0.81
d4		0.8	0.86	0.72	0.46	0.39	0.6	0.66	0.89
d5			0.79	0.56	0.43	0.4	0.6	0.67	0.8
d6				0.64	0.5	0.42	0.64	0.71	0.86
d7					0.35	0.33	0.45	0.54	0.72
d8						0.34	0.49	0.53	0.51
d9							0.46	0.52	0.44
d 10								0.72	0.65
d 11									0.71

6. CG AND DCG

These two metrics are used to calculate graded relevance of documents in a result set. Here document relevance degrees are considered. These are used to measure effectiveness of web search engine algorithms or information retrieval systems. [15] [16]

6.1 CG (cumulative Gain) [15]

The document IDs are replaced by their relevance values thus converting the ranked document lists into gained value lists. Let us consider that 0 - 3 are used as relevance values where 3 denotes high value and 0 no value.

For example: T'=< 1, 2, 2,3, 0, 0,2,2,3,0 . . . . >

The cumulated gain at ranked position *i* is calculated by adding from position *1 to i* when *i* ranges from 1 to 200. The position *i* in the gain vector T is denoted by T[i]. The cumulated gain vector TG is defined recursively as the vector TG where:

$$TG[i] = \begin{cases} T[1], & \text{if } i = 1 \\ TG[i-1] + T[i], & \text{otherwise} \end{cases}$$

Equation 4 Cumulative Gain

For example, from T' we obtain TG' = < 1, 3, 5, 8, 8, 10, 12, 15, 15, ... >. The cumulated gain at any rank may be read directly, e.g., at rank 6 it is 10.

6.2 DCG (Discounted cumulative Gain) [15]

It measures the usefulness of document based on its position in the result set.

The rank-based discount factor is used to obtain the rank of a document. As the rank increases, the smaller amount of the document value is added to the cumulated gain. As the ranked position of a relevant document increases it becomes less important for the user because user is less likely to examine such documents due to time, effort and cumulated information from documents already seen. A discounting function is used

which progressively reduces the document value as its rank increases but not too steeply to allow for user persistence in examining further documents. One of the methods of discounting with this requirement is to divide the document value by the log of its rank. For example 2log 2 = 1 and 21og1024 = 10, thus a document at the position 1024 would still get one tenth of it face value. Sharper or smoother discounts can be calculated by selecting the base of the logarithm, to show varying user behavior. If *b* denotes the base of the logarithm, the cumulated gain vector with discount DG is defined recursively as the vector DG where:

$$DG[i] = \begin{cases} T[1], & if\ i = 1 \\ DG[i-1] + \frac{T[i]}{\log_b i}, & \text{otherwise} \end{cases}$$

Equation 5 Discounted Cumulative Gain

The logarithm-based discount is not applied at rank l because blog l = 0.

For example, let b = 2. From TG' we obtain DCG' = < 3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61 . . . . >. Both the cumulated gain by document rank (TG) and the cumulated gain with discount by document rank (DCG) vectors should show the (lack of) ability of a query to rank highly relevant documents toward the top of the result list.

Table 5 CG and DCG Values

Term	cg@4	dcg@1	dgc@2	dcg@3	dcg@4
apple	**10**	**3**	**6**	**7.26**	**8.26**
account	**3**	**1**	**2**	**2.63**	**2.63**
arthritis	**2**	**1**	**2**	**2**	**2**
believe	**9**	**3**	**6**	**7.89**	**8.39**
bible	**11**	**3**	**6**	**7.89**	**8.89**
cancer	**6**	**3**	**4**	**4.63**	**5.13**
card	**4**	**3**	**4**	**4**	**4**
christ	**10**	**3**	**6**	**7.27**	**8.27**
cross	**11**	**3**	**5**	**6.27**	**7.27**
emphasis	**4**	**3**	**4**	**4**	**4**
father	**9**	**3**	**5**	**6.27**	**6.77**
gain	**9**	**3**	**5**	**6.27**	**7.27**
glory	**8**	**3**	**5**	**6.27**	**6.77**
god	**11**	**3**	**6**	**7.89**	**8.89**
health	**6**	**3**	**5**	**5**	**5**
heart	**12**	**3**	**6**	**7.89**	**9.39**
MRI	**3**	**3**	**3**	**3**	**3**
patient	**6**	**3**	**6**	**6**	**6**
pray	**7**	**3**	**5**	**5.63**	**6.13**
rational	**5**	**3**	**4**	**4.63**	**4.63**
thought	**7**	**3**	**5**	**5.63**	**6.13**
world	**11**	**3**	**6**	**7.27**	**8.27**

7. FIELD RELEVANCE SCORE

Field relevance is calculated with respect to the fields present in the document. For example, consider a query 'sloan apple sticky mouse button 1993' issued for an email collection. Here the user trying to recall some e-mail sent in '1993' by 'sloan' about 'apple's sticky mouse button' issue as subject. Obviously '1993' is expected to appear in date field, 'sloan' in from field while rest of the words may appear in subject field. However, 'sloan', '1993' and 'apple' may also appear in body. Thus a query term may have multiple matches in the fields that user did not intend, (e.g., 'apple' can be found in body field and 'sloan' can be found in body as signature), but term scores from such fields should be considered less important since those do not match with the user's structural intent. Here structural components of a collection are considered in modelling relevance. Suppose a query $Q = (q_1; :::, q_m)$ is composed of m words, and the collection *C* contains n field types $F = (F_1; :::; Fn)$, and that every document *D* in the collection is composed of fields *(F1; :::; Fn)*.

Field Relevance Calculation: Given a query $Q = (q_1; :::, q_m)$, field relevance $P(F_j \mid q_i, R)$ is the distribution of per-term relevance over document fields.

Field Relevance Model Based on field relevance estimates $P(F_j \mid q_i, R)$, the field relevance model combines field-level scores $P(q_i \mid F_j, D)$ for each document using field relevance as weights. [17]

The score is given by

$$S(D,Q) = \prod_{i=1}^{m} \sum_{j=1}^{n} \hat{P}(F_j|q_i,R)P(q_i|F_j,D)$$

Equation 6 Field Relevance Score

8. DATASPARK SEARCH METHODS

Let D be a document consisting of sections such that D = {d_0, d_1, d_2, d_3 ..... d_n} where d_0 is title, d_1 is head, d_2 is body etc. Let Q be a query consisting of words such that Q = {q_0, q_1, q2... q_n}. Let C be a factor under consideration such that it maintains the count and relative distance of some $q_i \in Q$. Hence W = {C, f ($di + \Delta d$)…} is a set of number of occurrences of query word di and relative distance Δd in the document. Let T be a template document. The query word qi has occurred in all sections of this document. Let there be a function f (Q, T) such that it returns the relative distance between the occurrences of Q in T.

$$f(Q,T) = \begin{cases} \Delta di : success \\ 0 : failure \end{cases}$$

A full method of relevance calculation [18]

Let $P = W0d_0 + W1d_1 + W2d_2 + \ldots$ be the weighted sum of all section. All sections have weight equal to 1. R is the weighted sum of differences between additional factors of document found and corresponding values of ideal document. PR is the weighted sum of sections where at least one query word has been found. Then value of relevance of documents is calculated as:

$$0.5\ (P + PR)/(P + R)$$

Equation 7 Relevance Score using full method

A fast method of relevance calculation

Consider P is number of bits used for weighted values of all sections. All sections have weight equal to 1. R is the

weighted sum of differences between additional factors of document found and corresponding values of ideal document. And PR is the number of bits where weighted values of sections of ideal document are different to weighted value of sections of document found. The formula calculations for relevance of documents is as below:

$$(P - PR)/(P + R)$$

Equation 8 Relevance Score using fast method

9. BAYES THEOREM [6] [19] [8]

Relevance of a document to a given query can be estimated by knowing the distribution of terms in the document collection. Relevant and irrelevant documents have different term distributions. [20] Term distribution gives information about 'probability of relevance' of a document to a query.

The probabilistic model is based on estimating the *probability* that a document will be relevant to a user, given a particular query. The higher this estimated probability, the more likely the document is to be relevant to the user. The estimated probability of relevance can be expressed as $Pq(R \mid x)$, the probability of relevance given a document x and a query q. [6] [11] [7] The probability $Pq(R \mid x)$ can't be estimated directly given the binary representation of document so Bayes theorem can be used to calculate it. [14]

$$P_q(R|x) = \frac{P_q(x|R)P_q(R)}{P(x)}$$

Equation 9 Bayes Theorem

Calculation of $Pq(R| x)$ through Baysian theorem
$Pq(R)$ is the prior probability that any document in the collection is relevant to q.
$Pq(x \mid R)$ is the probability of observing document x given relevance information
$P(x)$ is the probability of observing document x irrespective of relevance

10. CLUSTERPOINT SERVER METHOD

The relevance of the document according to the search request is calculated as follows: [19]

Each section has an associated weight. The weights of all search terms in a document are added. Relevance is calculated by multiplying the total weight with a value that represents the distance between the search terms in the document: the greater the distance, the smaller this value.

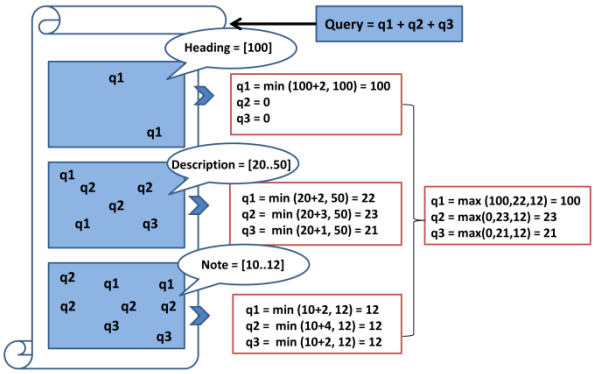

Figure 1: Calculating weight for each document

For example, a document consists of three document parts: heading, description, and note. Each document part contains words q_1, q_2, and q_3 and has its own weight interval, as described in Figure 1.

First, the weights of words are calculated in each part of the document:

q_1(heading)=min(100+2,100)=100,
q_1(description)=min(20+2,50)=22,
q_1(note)=min(10+2,12)=12,
q_2(heading)=0,
q_2(description)=min(20+3,50)=23,
q_2(note)=min(10+4,12)=12,
q_3(heading)=0,
q_3(description)=min(20+1,50)=21,
q_3(note)=min(10+2,12)=12

Then, the weights of words in the entire document are calculated:

q_{1d}=max(q_1(heading),q_1(description),q_1(note))= max(100,21,12)=100
q_{2d}=max(q_2(heading),q_2(description),q_2(note))=max(0, 23,12)=23
q_{3d}=max(q_3(heading),q_3(description),q_3(note))=max(0, 21,12)=21

Finally, the relevance of the document is calculated:

$q_{total}=q_{1d} + q_{2d} + q_{3d}$ = 100+23+21 = 144

Then relevance is calculated as

Relevance = q_{total} * d

11. CONCLUSION

The paper summarized the various relevance calculation approaches. Tf, IDF, Tf-Idf and cosine similarity are used by most information retrieval systems or search engines for relevance calculation. Field relevance calculation is useful if the relevance needs to be in a particular field of a document. For graded relevance calculation CG and DCG are the useful methods. Current IR systems return a large number of irrelevant results so there is enough scope to improve the relevance calculation methods.

12. REFERENCES

[1] Pia Borlund, The Concept of Relevance in IR in Journal of The American Society for information Science and Technology, 54(10):913–925, 2003

[2] S. Mizzaro. How many relevances in information retrieval? Interacting With Computers, 10(3):305–322, June 1998. ISSN: 0953-5438. Elsevier, The Netherlands.

[3] Wu HC, Luk RWP, Wong KF, Kwok KL (2008). "Interpreting tf–idf term weights as making relevance decisions". *ACM Transactions on Information Systems* **26** (3): 1–37.

[4] Yunjie (Calvin) Xu, Zhiwei Chen, Relevance Judgment: What Do Information Users Consider Beyond Topicality? in Journal of The American Society for information Science and Technology,57(7):961–973, 2006

[5] epaper.esakal.com

[6] Juan Ramos Using TF-IDF to Determine Word Relevance in Document Queries Tech. rep., Departament of Computer science. Rutgers University (2000)

[7] C. D. Manning, P. Raghavan and H. Schütze (2008), Introduction to Information Retrieval, Cambridge University Press.

[8] Solr relevancy FAQ in Solr wiki

[9] Salton, G. and C. Buckley, 1988 Term-weighting approaches in automatic text retrieval. Information Processing and Management *24* (5)**:** 513–523.

[10] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Modern Information Retrieval, ACM Press, 1999.

[11] Ian Ruthven Mounia Lalmas A survey on the use of relevance feedback for information access systems In: Knowledge Engineering Review, Vol. 18, No. 2, 06.2003, p. 95-145.

[12] Michael Dittenbach Scoring and Ranking Techniques - tf-idf term weighting and cosine similarity in Information Retrival Facility articles March 2010

[13] PEARL, J. 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan-Kaufmann, San Mateo, CA.

[14] G. Salton, A. Wong, and C. S. Yang (1975), "A Vector Space Model for Automatic Indexing," *Communications of the ACM*, vol. 18, nr. 11, pages 613–620.

[15] Kalervo Järvelin & Jaana Kekäläinen IR evaluation methods for retrieving highly relevant documents In: Belkin, N.J., Ingwersen, P. and Leong, M.-K. (eds.) Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY: ACM, pp. 41–48.(2000)

[16] Wang, Yining and Wang, Liwei and Li, Yuanzhi and He, Di and Liu, Tie-Yan and Chen, Wei. A Theoretical Analysis of NDCG Type Ranking Measures . InCoRR, (abs/1304.6480) Year 2013.

[17] Jin Young Kim and W. Bruce Croft A Field Relevance Model for Structured Document Retrieval in ECIR'12 Proceedings of the 34th European conference on Advances in Information Retrieval Pages 97-108 Springer-Verlag Berlin, Heidelberg ©2012

[18] Maxim Zakharov DataparkSearch at TREC-2005 in TREC-2005

[19] Relevance Ranking in Clusterpoint Server clusterpoint wiki.

[20] C. J. van Rijsbergen. Information Retrieval. Butterworths, London, 2nd edition, 1979.