Search Engines--Information Retrieval in Practice

tanshuqiu

Email: tsq@cqut.edu.cn

1

Retrieval Models

First : Overview of Retrieval Models

Second : Probabilistic Models

Third : Ranking Based on Language Models

Fourth : Complex Queries and Combining Evidence

Retrieval Models

Fifth : Web Search

Sixth : Machine Learning and Information Retrieval

Seventh : Application-Based Models

3

Overview of Retrieval Models

- ✓Boolean Retrieval
- ✓The Vector Space Model

Boolean Retrieval

The Boolean retrieval model was used by the earliest search engines and is still in use today. It is also known as exact-match retrieval since documents are retrieved if they exactly match the query specification, and otherwise are not retrieved.

Boolean Retrieval

The name Boolean comes from the fact that there only two possible outcomes for query evaluation (TRUE and FALSE) and because the query is usually specified using operators from Boolean logic (AND, OR, NOT).

Boolean Retrieval

There are some advantages to Boolean retrieval. The results of the model are very predictable and easy to explain to users. The operands of a Boolean query can be any document feature, not just words, so it is straightforward to incorporate metadata such as a document date or document type in the query specification.

Boolean Retrieval

Despite these positive aspects, the major drawback of this approach to search is that the effectiveness depends entirely on the user. Because of the lack of a sophisticated ranking algorithm, simple queries will not work well.

Boolean Retrieval

As an example of Boolean query formulation, consider the following queries for a search engine that has indexed a collection of news stories. The simple query:

lincoln

would retrieve a large number of documents that mention Lincoln cars and places named Lincoln in addition to stories about President Lincoln.

Boolean Retrieval

The user may attempt to narrow the scope of the search with the following query:

```
president AND lincoln
```

This query will retrieve a set of documents that contain both words, occurring anywhere in the document.

Boolean Retrieval

If enough of these types of documents were retrieved, the user may try to eliminate documents about cars by using the NOT operator, as follows:

```
president AND lincoln AND NOT (automobile OR car)
```

This would remove any document that contains even a single mention of the words "automobile" or "car" anywhere in the document.

Boolean Retrieval

If the retrieved set is still too large, the user may try to further narrow the query by adding in additional words that should occur in biographies:

```
president AND lincoln AND biography AND life AND birthplace AND get-
tysburg AND NOT (automobile OR car)
```

Unfortunately, in a Boolean search engine, putting too many search terms into the query with the AND operator often results in nothing being retrieved. To avoid this, the user may try using an OR instead:

Boolean Retrieval

```
president AND lincoln AND (biography OR life OR birthplace OR gettysburg)
AND NOT (automobile OR car)
```

This will retrieve any document containing the words "president" and "lincoln", along with any one of the words "biography", "life", "birthplace", or "gettysburg" (and does not mention "automobile" or "car").

The Vector Space Model

In this model, documents and queries are assumed to be part of at-dimensional vector space, where t is the number of index terms (words, stems, phrases, etc.). A document D_i is represented by a vector of index terms:

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{it}),$$

The Vector Space Model

$$D_i = (d_{i1}, d_{i2}, \ldots, d_{it}),$$

$$\begin{array}{lcccc} & Term_1 & Term_2 & \ldots & Term_t \\ Doc_1 & d_{11} & d_{12} & \ldots & d_{1t} \\ Doc_2 & d_{21} & d_{22} & \ldots & d_{2t} \\ \vdots & \vdots & & & \\ Doc_n & d_{n1} & d_{n2} & \ldots & d_{nt} \end{array}$$

where d_{ij} represents the weight of the jth term. A document collection containing n documents can be represented as a matrix of term weights, where each row represents a document and each column describes weights that were assigned to a term for a particular document:

D_1 Tropical Freshwater Aquarium Fish.
D_2 Tropical Fish, Aquarium Care, Tank Setup.
D_3 Keeping Tropical Fish and Goldfish in Aquariums, and Fish Bowls.
D_4 The Tropical Tank Homepage - Tropical Fish and Aquariums.

Terms	Documents			
	D_1	D_2	D_3	D_4
aquarium	1	1	1	1
bowl	0	0	1	0
care	0	1	0	0
fish	1	1	2	1
freshwater	1	0	0	0
goldfish	0	0	1	0
homepage	0	0	0	1
keep	0	0	1	0
setup	0	1	0	0
tank	0	1	0	1
tropical	1	1	1	2

Term-document matrix for a collection of four documents

The Vector Space Model

Document D3, for example, is represented by the vector (1, 1, 0, 2, 0, 1, 0, 1, 0, 0, 1).

Queries are represented the same way as documents. Tat is, a query Q is represented by a vector of t weights:

$$Q = (q_1, q_2, \ldots, q_t),$$

where q_j is the weight of the jth term in the query.

The Vector Space Model

Document D3, for example, is represented by the vector (1, 1, 0, 2, 0, 1, 0, 1, 0, 0, 1).

Queries are represented the same way as documents. Tat is, a query Q is represented by a vector of t weights:

$$Q = (q_1, q_2, \ldots, q_t),$$

where q_j is the weight of the jth term in the query.

The Vector Space Model

Documents could be ranked by computing the distance between the points representing the documents and the query. More commonly, a similarity measure is used (rather than a distance or dissimilarity measure), so that the documents with the highest scores are the most similar to the query.

The Vector Space Model

A number of similarity measures have been proposed and tested for this purpose. The most successful of these is the cosine correlation similarity measure. The cosine correlation measures the cosine of the angle between the query and the document vectors.

The Vector Space Model

The cosine measure is defined as:

$$Cosine(D_i, Q) = \frac{\sum_{j=1}^{t} d_{ij} \cdot q_j}{\sqrt{\sum_{j=1}^{t} {d_{ij}}^2 \cdot \sum_{j=1}^{t} {q_j}^2}}$$

The numerator of this measure is the sum of the products of the term weights for the matching query and document terms (known as the dot product or inner product).

The Vector Space Model

As an example, consider two documents D1 = (0.5, 0.8, 0.3) and D2 = (0.9, 0.4, 0.2) indexed by three terms, where the numbers represent term weights. Given the query Q = (1.5, 1.0, 0) indexed by the same terms, the cosine measures for the two documents are:

$$Cosine(D_1, Q) = \frac{(0.5 \times 1.5) + (0.8 \times 1.0)}{\sqrt{(0.5^2 + 0.8^2 + 0.3^2)(1.5^2 + 1.0^2)}}$$
$$= \frac{1.55}{\sqrt{(0.98 \times 3.25)}} = 0.87$$

$$Cosine(D_2, Q) = \frac{(0.9 \times 1.5) + (0.4 \times 1.0)}{\sqrt{(0.9^2 + 0.4^2 + 0.2^2)(1.5^2 + 1.0^2)}}$$
$$= \frac{1.75}{\sqrt{(1.01 \times 3.25)}} = 0.97$$

The second document has a higher score because it has a high weight for the first term, which also has a high weight in the query.

Probabilistic Models

- ✓Information Retrieval as Classification
- ✓The BM25 Ranking Algorithm

Probabilistic Models

One of the features that a retrieval model should provide is a clear statement about the assumptions upon which it is based. One early theoretical statement about effectiveness, known as the Probability Ranking Principle (Robertson, 1977/1997), encouraged the development of probabilistic retrieval models, which are the dominant paradigm today.

Probabilistic Models

The Probability Ranking Principle, as originally stated, is as follows:

If a reference retrieval system’ s response to each request is a ranking of the documents in the collection in order of decreasing probability of relevance to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose, the overall effectiveness of the system to its user will be the best that is obtainable on the basis of those data.

Information Retrieval as Classification

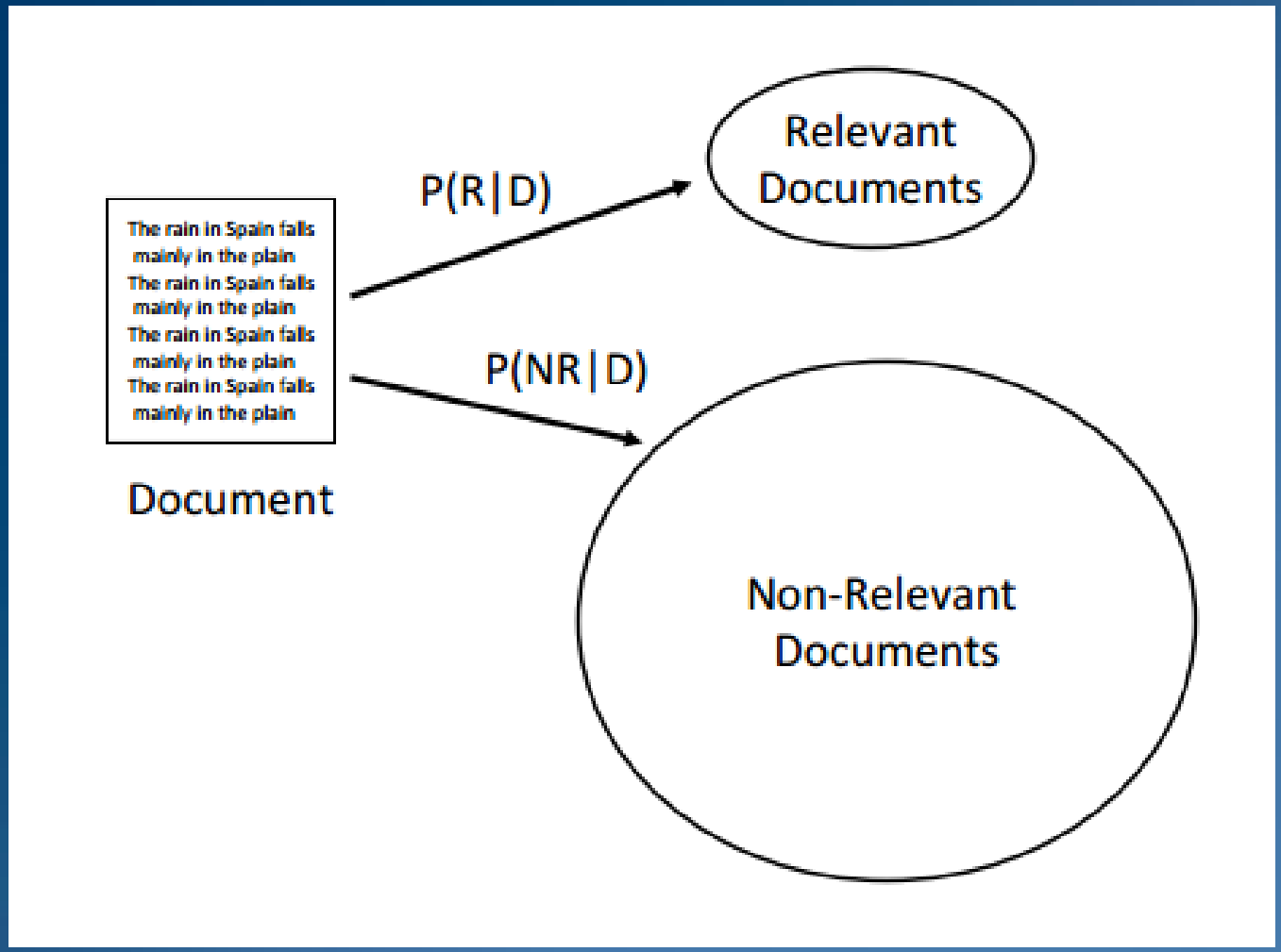
In any retrieval model that assumes relevance is binary, there will be two sets of documents, the relevant documents and the non-relevant documents, for each query. Given a new document, the task of a search engine could be described as deciding whether the document belongs in the relevant set or the non-relevant set.

Information Retrieval as Classification

Given some way of calculating the probability that the document is relevant and the probability that it is non-relevant, then it would seem reasonable to classify the document into the set that has the highest probability.

Information Retrieval as Classification

We would decide that a document D is relevant if P (R|D) > P (NR|D), where P (R|D) is a conditional probability representing the probability of relevance given the representation of that document, and P (NR|D) is the conditional probability of non-relevance. This is known as the Bayes Decision Rule, and a system that classifies documents this way is called a Bayes classifier.



Classifying a document as relevant or non-relevant

Information Retrieval as Classification

To start with, let's focus on P (R|D). It's not clear how we would go about calculating this, but given information about the relevant set, we should be able to calculate P (D|R). So how does calculating P (D|R) get us to the probability of relevance? It turns out there is a relationship between P (R|D) and P (D|R) that is expressed by Bayes' Rule:

Information Retrieval as Classification

To start with, let’ s focus on P (R|D). It’ s not clear how we would go about calculating this, but given information about the relevant set, we should be able to calculate P (D|R).

Information Retrieval as Classification

So how does calculating P (D|R) get us to the probability of relevance? It turns out there is a relationship between P (R|D) and P (D|R) that is expressed by Bayes' Rule:

$$P(R|D) = \frac{P(D|R)P(R)}{P(D)}$$

where P (R) is the a priori probability of relevance (in other words, how likely any document is to be relevant), and P (D) acts as a normalizing constant.

Information Retrieval as Classification

Given this, we can express our decision rule in the following way: classify a document as relevant if $P(D|R)P(R) > P(D|NR)P(NR)$.

The BM25 Ranking Algorithm

BM25 extends the scoring function for the binary independence model to include document and query term weights. The extension is based on probabilistic arguments and experimental validation, but it is not a formal model.

The BM25 Ranking Algorithm

BM25 is an effective ranking algorithm derived from a model of information retrieval viewed as classification. Tis model focuses on topical relevance and makes an explicit assumption that relevance is binary.

Ranking Based on Language Models

Language models are used to represent text in a variety of language technologies, such as speech recognition, machine translation, and handwriting recognition. The simplest form of language model, known as a unigram language model, is a probability distribution over the words in the language. This means that the language model associates a probability of occurrence with every word in the index vocabulary for a collection.

Complex Queries and Combining Evidence

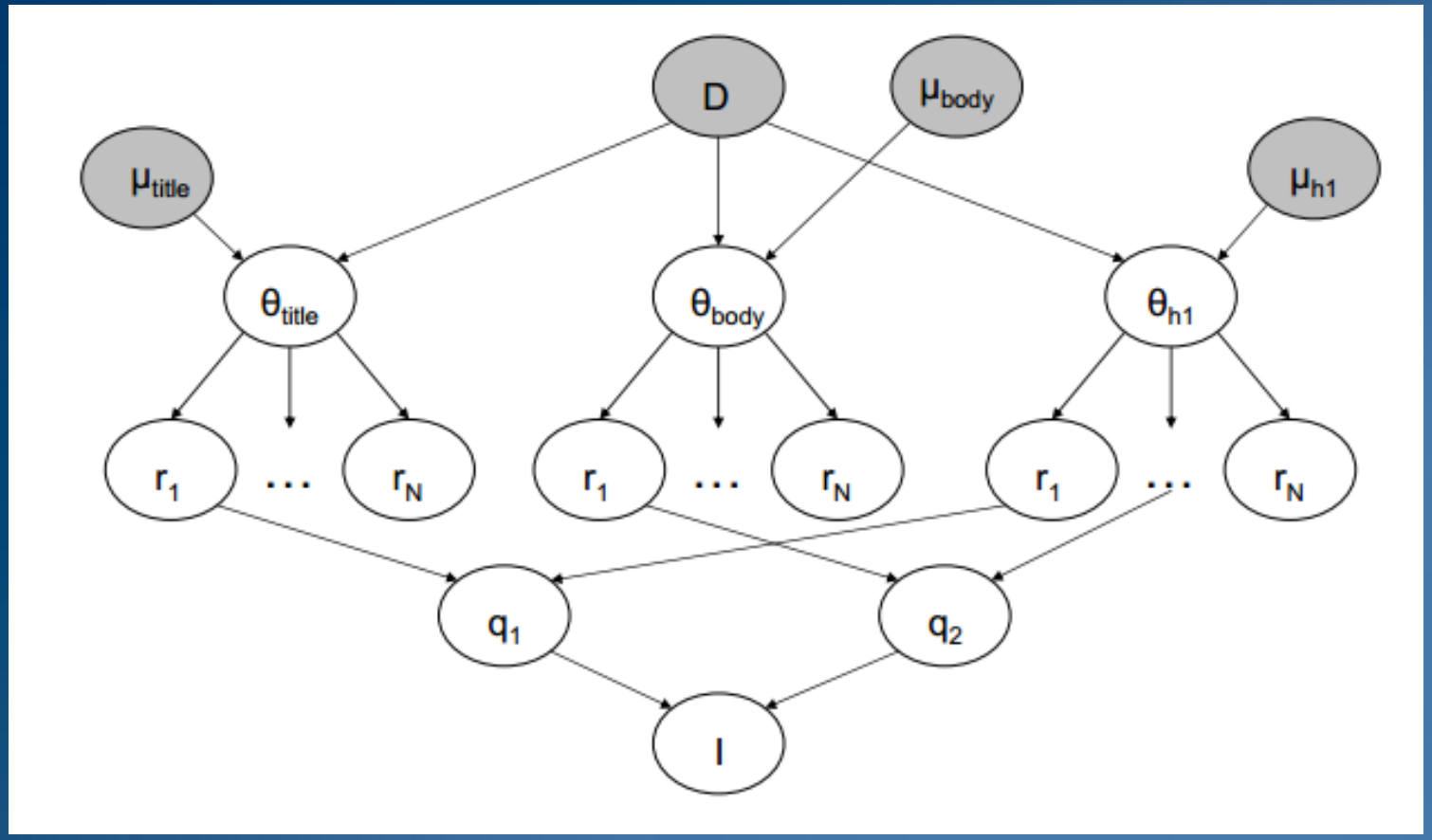
Effective retrieval requires the combination of many pieces of evidence about a document’ s potential relevance. In general, however, there can be many other types of evidence that should be considered. Even considering words, we may want to take into account whether certain words occur near each other, whether words occur in particular document structures, such as section headings or titles, or whether words are related to each other.

The Inference Network Model

Effective retrieval requires the combination of many pieces of evidence about a document’ s potential relevance. In general, however, there can be many other types of evidence that should be considered. Even considering words, we may want to take into account whether certain words occur near each other, whether words occur in particular document structures, such as section headings or titles, or whether words are related to each other.

The Inference Network Model

A Bayesian network is a probabilistic model that is used to specify a set of events and the dependencies between them. The networks are directed, acyclic graphs (DAGs), where the nodes in the graph represent events with a set of possible outcomes and arcs represent probabilistic dependencies between the events.



Example inference network model

The Inference Network Model

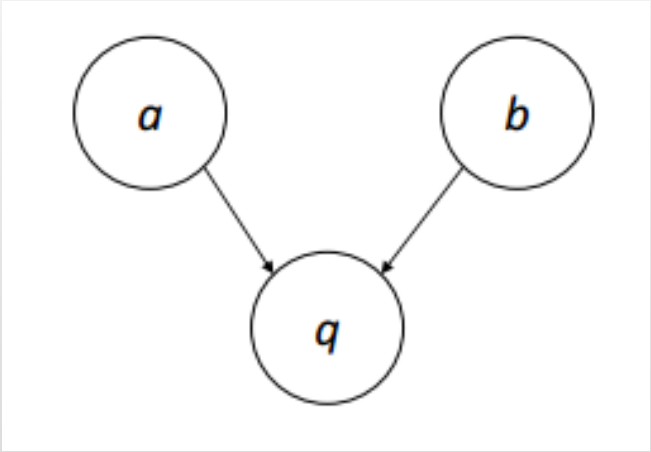
This picture shows an inference net where the evidence being combined are words in a web page’ s title, body, and <h1> headings. In this figure, D is a document node. This node corresponds to the event that a document (the web page) is observed. There is one document node for every document in the collection, and we assume that only one document is observed at any time.

The Inference Network Model

The r_i or representation nodes are document features (evidence), and the probabilities associated with those features are based on language models θ estimated using the parameters μ. There is one language model for each significant document structure (title, body, or headings).

The Inference Network Model

Given a simple network for a query node q with two parent nodes a and b, as shown in Figure



Inference network with three nodes

The Inference Network Model

$P(q = \text{TRUE} \mid a, b)$	a	b
0	FALSE	FALSE
0	FALSE	TRUE
0	TRUE	FALSE
1	TRUE	TRUE

Conditional probabilities for example network

Web Search

There are some major differences between web search and an application that provides search for a collection of news stories, for example. The primary ones are the size of the collection (billions of documents), the connections between documents (i.e., links), the range of document types, the volume of queries (tens of millions per day), and the types of queries.

Web Search

There are a number of different types of search in a web environment. One popular way of describing searches was suggested by Broder (2002). In this taxonomy, searches are either informational, navigational, or transactional.

Web Search

An informational search has the goal of finding information about some topic that may be on one or more web pages. Since every search is looking for some type of information, we call these topical searches in this book.

Web Search

A navigational search has the goal of finding a particular web page that the user has either seen before or assumes must exist.

A transactional search has the goal of finding a site where a task such as shopping or downloading music can be performed.

Machine Learning and Information Retrieval

- ✓Learning to Rank
- ✓Topic Models and Vocabulary Mismatch

Learning to Rank

The best-known of the approaches used to learn a ranking function for search is based on the Support Vector Machine (SVM) classifier. The input to the Ranking SVM is a training set consisting of partial rank information for a set of queries

$$(q_1, r_1), (q_2, r_2), \ldots, (q_n, r_n)$$

Learning to Rank

$$(q_1, r_1), (q_2, r_2), \ldots, (q_n, r_n)$$

where qi is a query and r_i is partial information about the desired ranking, or relevance level, of documents for that query. This means that if document d_a should be ranked higher than d_b, then $(d_a, d_b) \in r_i$.

Learning to Rank

Let’ s assume that we are learning a linear ranking function $\vec{w}.\vec{d}_a$ where $\vec{w}$ is a weight vector that is adjusted by learning, and $\vec{d}_a$ is the vector representation of the features of document d_a .

Learning to Rank

If a document is represented by three features with integer values $\vec{d}$ = (2, 4, 1) and the weights $\vec{w}$ = (2, 1, 2), then the score computed by the ranking function is just:

$$\vec{w}.\vec{d} = (2, 1, 2).(2, 4, 1) = 2.2 + 1.4 + 2.1 = 10$$

Topic Models and Vocabulary Mismatch

One of the important issues in general information retrieval is vocabulary mismatch. Tis refers to a situation where relevant documents do not match a query, because they are using different words to describe the same topic.

Topic Models and Vocabulary Mismatch

TREC experiments have shown that topical queries produce better results using query expansion. Query expansion (using, for example, pseudo-relevance feedback) is the standard technique for reducing vocabulary mismatch, although stemming also addresses this issue to some extent.

Topic Models and Vocabulary Mismatch

TREC experiments have shown that topical queries produce better results using query expansion. Query expansion (using, for example, pseudo-relevance feedback) is the standard technique for reducing vocabulary mismatch, although stemming also addresses this issue to some extent. A different approach would be to expand the documents by adding related terms.

Application-Based Models

Ranking algorithms that work well in web search engines often do not produce the best rankings in other applications. Customizing a ranking algorithm for the application will nearly always produce the best results.

Application-Based Models

The first step in doing this is to construct a test collection of queries, documents, and relevance judgments so that different versions of the ranking algorithm can be compared quantitatively.

The next step is to identify what evidence or features might be used to represent documents.

Application-Based Models

Simple terms and proximity terms are almost always useful. Significant document structure—such as titles, authors, and date fields—are also nearly always important for search. In some applications, numeric fields may be important. Text processing techniques such as stemming and stop words also must be considered.

Application-Based Models

Having identified the various document features and other evidence, the next task is to decide how to combine it to calculate a document score. the combination and weighting of evidence can be expressed in the query language and many variations can be tested quickly.