Towards an Information Retrieval Theory of Everything*

Djoerd Hiemstra
University of Twente
http://www.cs.utwente.nl/~hiemstra

Abstract

I present three well-known probabilistic models of information retrieval in tutorial style: The binary independence probabilistic model, the language modeling approach, and Google's page rank. Although all three models are based on probability theory, they are very different in nature. Each model seems well-suited for solving certain information retrieval problems, but not so useful for solving others. So, essentially each model solves part of a bigger puzzle, and a united view on these models might be a first step towards an *Information Retrieval Theory of Everything*.

1 Introduction

Many applications that handle information on the internet would be completely inadequate without the support of information retrieval technology. How would we find information on the world wide web if there were no web search engines? How would we manage our email without spam filtering? Much of the development of information retrieval technology, such as web search engines and spam filters, requires a combination of *experimentation* and *theory*. Experimentation and rigorous empirical testing are needed to keep up with increasing volumes of web pages and emails. Furthermore, experimentation and constant adaptation of technology is needed in practice to counteract the effects of people that deliberately try to manipulate the technology, such as email spammers. However, if experimentation is not guided by theory, engineering becomes trial and error. New problems and challenges for information retrieval come up constantly. They cannot possibly be solved by trial and error alone. So, what is the theory of information retrieval?

*A more extensive overview of information retrieval theory, covering eight models is given in: Djoerd Hiemstra, Information Retrieval Models. In: A. Goker and J. Davies (eds.) *Information Retrieval: Searching in the 21st Century*. John Wiley and Sons, Ltd., ISBN-13: 978-0470027622, November 2009.

There is not one convincing answer to this question. There are many theories, here called *formal models*, and each model is helpful for the development of some information retrieval tools, but not so helpful for the development of others. In order to understand information retrieval, it is essential to learn about these retrieval models. In this paper, I present three well-known probabilistic models of information retrieval in a tutorial style. But first, we will describe what exactly it is that these models model.

2 Terminology

An information retrieval system is a software program that stores and manages information on documents, often textual documents but possibly multimedia. The system assists users in finding the information they need. It does not explicitly return information or answer questions. Instead, it informs on the existence and location of documents that might contain the desired information. Some suggested documents will, hopefully, satisfy the user's information need. These documents are called *relevant* documents. A perfect retrieval system would retrieve only the relevant documents and no irrelevant documents. However, perfect retrieval systems do not exist and will not exist, because search statements are necessarily incomplete and relevance depends on the subjective opinion of the user. In practice, two users may pose the same query to an information retrieval system and judge the relevance of the retrieved documents differently: Some users will like the results, others will not.

There are three basic processes an information retrieval system has to support: the representation of the content of the documents, the representation of the user's information need, and the comparison of the two representations. The processes are visualized in Figure 1. In the figure, squared boxes represent data and rounded boxes represent processes.

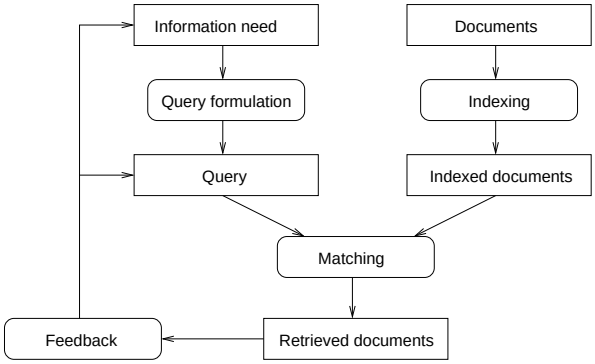

Figure 1: Information retrieval processes

Representing the documents is usually called the *indexing* process. The process takes place off-line, that is, the end user of the information retrieval system is not directly involved. The indexing process results in a representation of the document. Often, full text retrieval systems use a rather trivial algorithm to derive the index representations, for instance an algorithm that identifies words in an English text and puts them to lower case. The indexing process may include the actual storage of the document in the system, but often documents are only stored partly, for instance only the title and the abstract, plus information about the actual location of the document.

Users do not search just for fun, they have a need for information. The process of representing their *information need* is often referred to as the *query formulation* process. The resulting representation is the query. In a broad sense, query formulation might denote the complete interactive dialogue between system and user, leading not only to a suitable query but possibly also to the user better understanding his/her information need: This is denoted by the feedback process in Figure 1.

The comparison of the query against the document representations is called the *matching* process. The matching process usually results in a ranked list of documents. Users will walk down this document list in search of the information they need. Ranked retrieval will hopefully put the relevant documents towards the top of the ranked list, minimizing the time the user has to invest in reading the documents. Simple but effective ranking algorithms use the frequency distribution of terms over documents, but also other statistics, such as the number of hyperlinks that point to the document. Ranking algorithms based on statistical approaches easily halve the time the user has to spend on reading documents. The theory behind ranking algorithms is a crucial part of information retrieval and the major theme of this paper.

3 Models of Information Retrieval

There are two good reasons for having models of information retrieval. The first is that models guide research and provide the means for academic discussion. The second reason is that models can serve as a blueprint to implement an actual retrieval system.

Mathematical models are used in many scientific areas with the objective to understand and reason about some behavior or phenomenon in the real world. One might for instance think of a model of our solar system that predicts the position of the planets on a particular date, or one might think of a model of the world climate that predicts the temperature given the atmospheric emissions of greenhouse gases. A model of information retrieval predicts and explains what a user will find relevant given the user query. The correctness of the model's predictions can be tested in a controlled experiment. In order to do predictions and reach a better understanding of information retrieval, models should be firmly grounded in intuitions, metaphors and some branch of mathematics. Intuitions are important because they help to get a model accepted as reason-

able by the research community. Metaphors are important because they help to explain the implications of a model to a bigger audience. For instance, by comparing the earth's atmosphere with a greenhouse, non-experts will understand the implications of certain models of the atmosphere. Mathematics are essential to formalise a model, to ensure consistency, and to make sure that it can be implemented in a real system. As such, a model of information retrieval serves as a blueprint which is used to implement an actual information retrieval system.

The following sections will describe three models of information retrieval rather extensively. Section 4 describes the classical probabilistic retrieval model, Section 5 describes the language modeling approach, and Section 6 describes the page rank model. Section 7 describes an approach to unify these three rather distinct models in a attempt to construct a "theory of everything". Section 8 concludes this paper.

4 The probabilistic retrieval model

Several approaches that try to model matching and ranking using *probability theory*. The notion of the probability of something, for instance the probability of relevance notated as $P(R)$, is usually formalized through the concept of an experiment, where an experiment is the process by which an observation is made. The set of all possible outcomes of the experiment is called the sample space. In the case of $P(R)$ the sample space might be $\{relevant, irrelevant\}$, and we might define the random variable R to take the values $\{0, 1\}$, where $0 =$ irrelevant and $1 =$ relevant.

Let's define an experiment for which we take one document from the collection at random: If we know the number of relevant documents in the collection, say 100 documents are relevant, and we know the total number of documents in the collection, say 1 million, then the quotient of those two defines the probability of relevance $P(R = 1) = 100/1{,}000{,}000 = 0.0001$. Suppose furthermore that $P(D_k)$ is the probability that a document contains the term k with the sample space $\{0, 1\}$, ($0 =$ the document does not contain term k, $1 =$ the document contains term k), then we will use $P(R, D_k)$ to denote the *joint probability distribution* with outcomes $\{(0, 0), (0, 1), (1, 0)$ and $(1, 1)\}$, and we will use $P(R|D_k)$ to denote the *conditional probability distribution* with outcomes $\{0, 1\}$. So, $P(R=1|D_k=1)$ is the probability of relevance if we consider documents that contain the term k.

Stephen Robertson and Karen Spärck-Jones based their probabilistic retrieval model on this line of reasoning (Robertson and Spärck-Jones 1976). They suggested to rank documents by $P(R|D)$, that is the probability of relevance R given the document's content description D. Note that D is here a vector of binary components, each component typically representing a term. In the probabilistic retrieval model the probability $P(R|D)$ has to be interpreted as follows: there might be several, say 10, documents that are represented by the same D. If 9 of them are relevant, then $P(R|D) = 0.9$. To make this work in

practice, we use Bayes' rule on the probability odds $P(R|D)/P(\overline{R}|D)$, where $\overline{R}$ denotes irrelevance. The odds allow us to ignore $P(D)$ in the computation while still providing a ranking by the probability of relevance. Additionally, we assume independence between terms given relevance.

$$\frac{P(R|D)}{P(\overline{R}|D)} \;=\; \frac{P(D|R)P(R)}{P(D|\overline{R})P(\overline{R})} \;=\; \frac{\prod_k P(D_k|R)P(R)}{\prod_k P(D_k|\overline{R})P(\overline{R})} \tag{1}$$

Here, D_k denotes the k^{th} component (term) in the document vector. The probabilities of the terms are defined as above from examples of relevant documents. A more convenient implementation of probabilistic retrieval uses the following three order preserving transformations. First, the documents are ranked by sums of logarithmic odds, instead of the odds themselves. Second, the a priori odds of relevance $P(R)/P(\overline{R})$ is ignored. Third, we subtract $\sum_k \log(P(D_k = 0|R)/P(D_k = 0|\overline{R}))$, i.e., the score of the empty document, from all document scores. This way, the sum over all terms, which might be millions of terms, only includes non-zero values for terms that are present in the document.

$$\text{matching-score}(D) \;=\; \sum_{\substack{k \in \text{match-}\\ \text{ing terms}}} \log \frac{P(D_k{=}1|R)\,P(D_k{=}0|\overline{R})}{P(D_k{=}1|\overline{R})\,P(D_k{=}0|R)} \tag{2}$$

In practice, terms that are not in the query are also ignored in Equation 2. Making full use of the probabilistic retrieval model requires two things: examples of relevant documents and long queries. Relevant documents are needed to compute $P(D_k|R)$, that is, the probability that the document contains the term k given relevance. Long queries are needed because the model only distinguishes term presence and term absence in documents and as a consequence, the number of distinct values of document scores is low for short queries. For a one-word query, the number of distinct probabilities is two (either a document contains the word or not), for a two-word query it is four (the document contains both terms, or only the first term, or only the second, or neither), for a three-word query it is eight, etc. Obviously, this makes the model inadequate for web search, for which no relevant documents are known beforehand and for which queries are typically short. However, the model is helpful in for instance spam filters. Spam filters accumulate many examples of relevant (no spam or 'ham') and irrelevant (spam) documents over time. To decide if an incoming email is spam or ham, the full text of the email can be used instead of just a few query terms.

5 Language modeling

Language models were applied to information retrieval by a number of researchers in the late 1990's (Ponte and Croft 1998; Hiemstra 1998; Miller et al. 1999). They originate from probabilistic models of language generation developed for automatic speech recognition systems in the early 1980's. Automatic

speech recognition systems combine probabilities of two distinct models: the acoustic model, and the language model. The acoustic model might for instance produce the following candidate texts in decreasing order of probability: "food born thing", "good corn sing", "mood morning", and "good morning". Now, the language model would determine that the phrase "good morning" is much more probable, i.e., it occurs more frequently in English than the other phrases. When combined with the acoustic model, the system is able to decide that "good morning" was the most likely utterance, thereby increasing the quality of the system.

For information retrieval, language models are built for each document. By following this approach, the language model of the NVTI newsletter you are reading now would assign an exceptionally high probability to the word "theory" indicating that this document would be a good candidate for retrieval if the query contains this word. Language models take the following starting point: Given D – the document is relevant – the user will formulate a query by using a term T with some probability $P(T|D)$. The probability is defined by the text of the documents: If a certain document consists of 100 words, and of those the word "good" occurs twice, then the probability of "good" given that the document is relevant is simply defined as 0.02. For queries with multiple words, we assume that query words are generated independently from each other, i.e., the conditional probabilities of the terms $T_1, T_2, \cdots$ given the document are multiplied:

$$P(T_1, T_2, \cdots | D) = \prod_i P(T_i|D) \tag{3}$$

> Note that the notation $P(\ldots)$ is overloaded. Any time we are talking about a different random variable or sample space, we are also talking about a different measure P. So, one equation might refer to several probability measures, all ambiguously referred to as P. Also note that random variables like D and T might have different sample spaces in different models. For instance, D in the language modeling approach is a random variable denoting "*this* is the relevant document", that has as possible outcomes the identifiers of the documents in the collection. However, D in the probabilistic retrieval model is a random variable that has as possible outcomes all possible document descriptions, which in this case are vectors with binary components d_k that denote whether a document is indexed by term k or not.

As a motivation for using the probability of the query given the document, one might think of the following experiment. Suppose we ask one million monkeys to pick a good three-word query for several documents. Each monkey will point three times at random to each document. Whatever word the monkey points to, will be the (next) word in the query. Suppose that 7 monkeys accidentally pointed to the words "information", "retrieval" and "model" for document 1, and only 2 monkeys accidentally pointed to these words for document 2. Then, document 1 would be a better document for the query "information retrieval model" than document 2.

The above experiment assigns zero probability to words that do not occur anywhere in the document, and because we multiply the probabilities of the single words, it assigns zero probability to documents that do not contain all of the words. For some applications this is not a problem. For instance for a web search engine, queries are usually short and it will rarely happen that no web page contains all query terms. For many other applications empty results happen much more often, which might be problematic for the user. Therefore, a technique called *smoothing* is applied: Smoothing assigns some non-zero probability to unseen events. One approach to smoothing takes a linear combination of $P(T_i|D)$ and a background model $P(T_i)$ as follows.

$$P(T_1, \cdots, T_n|D) = \prod_{i=1}^{n} (\lambda P(T_i|D) + (1-\lambda)P(T_i)) \tag{4}$$

The background model $P(T_i)$ might be defined by the probability of term occurrence in the collection, i.e., by the quotient of the total number of occurrences in the collection divided by the length of the collection. In the equation, λ is an unknown parameter that has to be set empirically. Linear interpolation smoothing accounts for the fact that some query words do not seem to be related to the relevance of documents at all. For instance in the query "capital of the Netherlands", the words "of" and "the" might be seen as words from the user's general English vocabulary, and not as words from the relevant document he/she is looking for. In terms of the experiment above, a monkey would either pick a word at random from the document with probability λ or the monkey would pick a word at random from the entire collection. A more convenient implementation of the linear interpolation models can be achieved with order preserving transformations that are similar to those for the probabilistic retrieval model (see Equation 2). We multiply both sides of the equation by $\prod_i (1-\lambda)P(T_i)$ and take the logarithm, which leads to:

$$\text{matching-score}(d) \;=\; \sum_{\substack{k \in \text{match-} \\ \text{ing terms}}} \log(1 + \frac{P(T_i|D)}{P(T_i)} \cdot \frac{\lambda}{1-\lambda}) \tag{5}$$

Language models are well-suited in situations that require searching for documents that are *similar* to a query. Instead of modeling what a relevant document looks like, as done by the probabilistic model, the language modeling approach simply returns the document that is most similar to the query. As such a language modeling approach is well-suited for search systems that get ad hoc, short queries, as is for instance the case in web search.

6 Google's page rank model

When Sergey Brin and Lawrence Page launched the web search engine Google in 1998 (Brin and Page 1998), it had two features that distinguished it from other web search engines: It had a simple no-nonsense search interface, and, it used

a radically different approach to rank the search results. Instead of returning documents that closely match the query terms, they aimed at returning high quality documents, i.e., documents from trusted sites. Google uses the hyperlink structure of the web to determine the quality of a page, called *page rank*. Web pages that are linked at from many places around the web are probably worth looking at: They must be *high quality* pages. If pages that have links from other high quality web pages, for instance DMOZ or Wikipedia[1], then that is a further indication that they are likely to be worth looking at. The page rank of a page d is defined as $P(D = d)$, i.e., the probability that d is relevant, exactly as it is used in the language modeling approach of Section 5 as well. It is defined as:

$$P(D{=}d) \;=\; (1{-}\lambda)\frac{1}{\#\text{pages}} \quad + \quad \lambda \sum_{i|i \text{ links to } d} P(D{=}i)P(D{=}d|D{=}i) \tag{6}$$

If we ignore $^{(1-\lambda)}/_{\#\text{pages}}$ for the moment, then the page rank $P(D = d)$ is recursively defined as the sum of the page ranks $P(D = i)$ of all pages i that link to d, multiplied by the probability $P(D{=}d|D{=}i)$ of following a link from i to d. One might think of the page rank as the probability that a random surfer visits a page. Suppose we ask the monkeys from the previous section to surf the web from a randomly chosen starting point i. Each monkey will now click on a random hyperlink with the probability $P(D = d|D = i)$ which is defined as one divided by the number of links on page i. This monkey will end up in d. But other monkeys might end up in d as well: Those that started on another page that happens to link to d. After letting the monkeys surf a while, the highest quality pages, i.e., the best connected pages, will have most monkeys that look at it.

The above experiment has a similar problem with zero probabilities as the language modeling approach. Some pages might have no links pointing to them, so they will get a zero page rank. Others might not link to any other page, so you cannot leave the page by following hyperlinks. The solution is also similar to the zero probability problem in the language modeling approach: We smooth the model by some background model, in this case the background is uniformly distributed over all pages. With some unknown probability λ a link is followed, but with probability $1 - \lambda$ a random page is selected, which is like a monkey typing in a random (but valid) URL.

Page rank is a so-called static ranking function, that is, it does not depend on the query. It is computed once off-line at indexing time by iteratively calculating the page ranks of pages at time $t + 1$ from the page ranks calculated in the iteration at time t until they do not change significantly anymore. Once the page rank of every page is calculated it can be used during querying. One possible way to use page rank during querying is as follows: Select the documents that contain all query terms and rank those documents by their page rank. In practice, web search engines like Google use many more factors in their ranking than just page rank alone.

[1] see http://dmoz.org and http://wikipedia.org

7 Putting things together

There is no such thing as a dominating model or theory of information retrieval, unlike the situation in for instance the area of databases where the relational model is *the* dominating database model. In information retrieval, some models work for some applications, whereas others work for other applications. For instance, the probabilistic retrieval model of Section 4 might be a good choice if examples of relevant and non-relevant documents are available; language models in Section 5 are helpful in situations that require models of language similarity; and the page rank model of Section 6 is often used in situations that need modeling of more of less static relations between documents. Despite their differences, these three models all use probability theory. This brings up the questions: Could we combine the models in one coherent model? and, What would such a model look like?

Let's consider the scenario of searching scientific papers as for instance done by Citeseer, Google Scholar or Scopus[2], that is, given a text query, for instance "theory of information retrieval", the system should retrieve the most important research papers in the field. To find the most important research papers on the theory of information retrieval, we need a model that fulfills the following requirements: First and foremost, an important research paper should mention the query terms "theory", "information" and "retrieval" more often than we would expect in random texts. Second, the paper should be cited a lot, preferably by papers that are cited a lot themselves. Third, the paper should fulfill a number of simple criteria and intuitions that we have about good research papers, such as 1) it should be written recently; 2) it should be written in an ISI-rated journal; 3) it should contain examples; 4) it should contain real program code, etc.

1. The paper should mention the query terms

To fulfill the first requirement, i.e., to find papers that use similar language as our query, a language modeling approach would be appropriate. So, we assign the result of Equation 4 to every document. Note however that Equation 4 defines the probability of a query given a document, but obviously, the system should rank by the probability of the documents given the query. These two probabilities are related by Bayes' rule as follows.

$$P(D|T_1, T_2, \cdots, T_n) \;=\; \frac{P(T_1, T_2, \cdots, T_n|D)P(D)}{P(T_1, T_2, \cdots, T_n)} \tag{7}$$

The left-hand side of Equation 7 cannot be used directly because the independence assumption presented above assumes terms are independent given the document. So, in order to compute the probability of the document D given the query, we need to multiply Equation 4 by $P(D)$ and divide it by $P(T_1, \cdots, T_n)$. Again, as stated in the previous paragraph, the probabilities themselves are

[2]http://citeseerx.ist.psu.edu, http://scholar.google.com, and http://scopus.com

of no interest, only the ranking of the document by the probabilities is. And since $P(T_1, \cdots, T_n)$ does not depend on the document, ranking the documents by the numerator of the right-hand side of Equation 7 will rank them by the probability given the query. This shows the importance of $P(D)$, the marginal probability, or prior probability of the document, i.e., it is the probability that the document is relevant if we do not know the query (yet). But how to define $P(D)$ properly?

2. The paper should be cited a lot

In fact $P(D)$ defines a static ranking function as described earlier for the page rank model of Section 6. If one research paper cites another, the cited paper is endorsed by the citing paper. Interestingly, Brin and Page (1998) were inspired by citation analysis when developing page rank, and they call the hyperlink graph a *citation graph* in their paper. So, the page rank model would be an excellent model to fulfill requirement 2, and since the page rank score is really a document prior $P(D)$ it can be easily combined with the language modeling score as shown by Equation 7. So, in terms of the language modeling approach, static rankings are simply document priors, i.e., the a-priori probability of the document being relevant, that should be combined with the probability of terms given the document. Document priors can be easily combined with standard language modeling probabilities and are as such powerful means to improve the effectiveness in web search (Kraaij et al. 2002).

3. Intuitions about good research papers

Our third requirement lists a number of intuitions about the properties of a good research paper. Let's assume that these properties are easily detected for a paper. So, for every document we know if it is: 1) written recently (for instance after 2004), 2) published in an ISI-rated journal, 3) containing examples, 4) containing real code. However, are these 4 properties all equally important? If not, how important is each property? Could it be that some properties are not important at all? It might not surprise the reader at this point that the probabilistic retrieval model of Section 4 is able to answer these questions. As said, the probabilistic retrieval model needs examples of relevant and non-relevant documents, in this case examples of important research papers and unimportant research papers. Suppose we ask a group of users to use a basic version of our system for some time, and to rate the research papers found for each query as *important* or *not important*. Using this data, we can estimate for each document its probability of relevance given its 4 properties. This probability can be combined with the page rank model by using it in Equation 6 to replace $^1/_{\#\text{pages}}$. The resulting random surfer of this model follows citations with some probability λ, but selects a "random" page by the probability of relevance given the document properties with probability $1 - \lambda$.

Discussion

Let's again explain our model's implications by the analogy of monkeys. Suppose we ask the monkeys from the previous sections to follow random citations in research papers. Instead of starting from a completely random paper they are more likely to start from papers that have a high probability of relevance given their properties, and each time they are allowed to follow a citation, they might also (with probability $1 - \lambda$) return to the important papers. If, after letting them surf around for a while, the number of monkeys on each paper does not change significantly anymore, we ask them to stop following citations. Instead they pick three random words from the paper in which they ended up, one word at a time.

The process is defined in such a way that every research paper has a very small probability of having a monkey ending up there that also picked the words "theory", "information", and "retrieval". The document with the highest probability of this event is most likely to be an important paper on the theory of information retrieval.

8 Conclusion and further reading

This paper describes three information retrieval models in a tutorial style in order to explain the consequences of modeling assumptions. Once the reader is aware of the consequences of modeling assumptions, he or she will be able to choose a model of information retrieval that is adequate in new situations. Although each model is well-suited for certain applications and not so useful for others, their modeling assumptions do not necessarily contradict each other, and unified modeling approaches are certainly possible.

A much more elaborate version of this paper that covers eight models instead of only three can be found in the book by Goker and Davies (2009). The book focuses on current trends and achievements in information retrieval. It provides a basis for understanding recent developments in the field and outlines directions for information search technologies in the near future and beyond. The book contains exercises, making it a good candidate for information retrieval courses in both undergraduate and graduate programs.

References

Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems 30*(1-7), 107–117.

Goker, A., and J. Davies (2009). *Information Retrieval: Searching in the 21st Century*. John Wiley and Sons, Ltd., ISBN-13: 978-0470027622, November 2009.

Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *Proceedings of the Second European Conference*

on Research and Advanced Technology for Digital Libraries (ECDL), pp. 569–584.

Kraaij, W., T. Westerveld, and D. Hiemstra (2002). The importance of prior probabilities for entry page search. In *Proceedings of the 25th ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, pp. 27–34.

Miller, D., T. Leek, and R. Schwartz (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval (SIGIR'99)*, pp. 214–221.

Ponte, J. and W. Croft (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st ACM Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 275–281.

Robertson, S. and K. Spärck-Jones (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science 27*, 129–146.