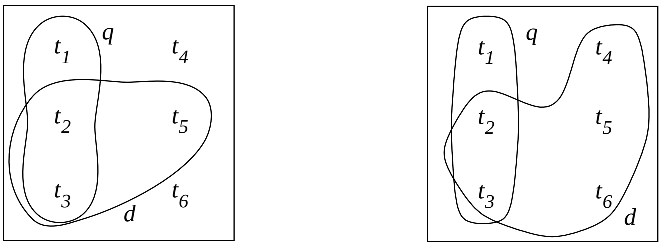
Language Models and Uncertain Inference in Information Retrieval

Norbert Fuhr
University of Dortmund, Germany

1 Introduction: IR as inference

In the logical view on IR systems, retrieval is interpreted as implication [Rijsbergen 86]: Let d denote a document (represented as logical formula) and q a query, then retrieval deals with the task of finding those documents which imply the query, i.e. for which the formula $d \rightarrow q$ is true. Due to the intrinsic uncertainty and vagueness of IR, we have to switch to uncertain inference. Using a probabilistic approach, the probability $P(d \rightarrow q)$ that the implication goes through should be computed as the conditional probability $P(q|d)$, as pointed out by Rijsbergen.



$P(d \rightarrow q) = 2/3$
$P(q \rightarrow d) = 2/3$

$P(d \rightarrow q) = 2/3$
$P(q \rightarrow d) = 2/5$

Figure 1: $P(d \rightarrow q)$ vs. $P(q \rightarrow d)$

As a simple example, assume that we have a probability space where terms represent disjoint events, as shown in figure 1. If all terms are assumed to be equiprobable, the left-hand example in this figure gives us a probability $P(d \rightarrow q) = P(q|d) = 2/3$. Later, Nie has shown [Nie 89] that in some cases, it may be reasonable to consider also the implication $P(q \rightarrow d)$, which should be defined according to Rijsbergen as $P(d|q)$. Whereas the first implication measures the exhaustivity of a document wrt. a query, the latter can be used as a measure of specifity (see the example in figure 1).

In the following, we will show how language models relate to the concept of IR as uncertain inference. For this purpose, we use a specialization of the uncertain inference approach, namely the probabilistic concept space (PCS) model developed by Wong and Yao [Wong & Yao 95], and we show how some of the language models proposed so far fit into this model.

2 The probabilistic concept space model

From a logical point of view, text retrieval models represent documents as sets of (weighted) propositions. In order to set up a basic framework for these models, Wong and Yao assume a concept space U consisting of a set of elementary, disjoint concepts c_i.

Any proposition p is a set of concepts, i.e. a subset of the concept space ($p \subseteq U$). Boolean combinations of propositions can be expressed as set operations on this concept space.

In order to support probabilistic inference, a probability function $P(.)$ over U is defined, i.e.

$$\sum_{c_i \in U} P(c_i) = 1$$

Now queries and documents are treated as propositions as well, Considering the probability function, we have

$$\begin{aligned}
P(d) &= \sum_{c_i \in d} P(c_i) \\
P(q) &= \sum_{c_i \in q} P(c_i) \\
P(q \cap d) &= \sum_{c_i \in q \cap d} P(c_i) \\
P(d \rightarrow q) &= P(q|d) = \frac{P(q \cap d)}{P(d)} \\
P(q \rightarrow d) &= P(d|q) = \frac{P(q \cap d)}{P(q)}
\end{aligned}$$

The last two formulas show that the two implication directions only differ by the normalization factor. We will exploit this fact below when we compare the Ponte/Croft language model with Hiemstra's.

Whereas text retrieval is based on terms, the PCS model uses concepts as elementary propositions; thus, we have to define the relationship between terms and concepts. A straightforward approach identifies each term with a concept; following this idea, e.g. the vector space model can be explained in terms of the PCS model. Alternatively, one can assume that terms are overlapping; this approach forms the basis for explaining Boolean and fuzzy retrieval as well as thwe probabilistic binary independence indexing and retrieval models. Since language models assume probabilistic independence of terms, we have to follow the idea of terms as overlapping in concept space for these models, too. Thus, we assume that there are terms t_i, t_j with $t_i \cap t_j \neq \emptyset$. For simplicity, let us also assume that the terms form a complete cover of the concept space U.

In order to apply the PCS model, terms are mapped onto disjoint atomic concepts in the following way: We form complete conjuncts (or minterms) of all terms $t_1, \ldots, t_n$, in which each term occurs either positively or negated, i.e.

$$\begin{array}{rcl}
m_0 & = & \bar{t}_1 \cap \bar{t}_2 \cap \bar{t}_3 \cap \cdots \bar{t}_{n-1} \cap \bar{t}_n \\
m_1 & = & t_1 \cap \bar{t}_2 \cap \bar{t}_3 \cap \cdots \bar{t}_{n-1} \cap \bar{t}_n \\
m_2 & = & \bar{t}_1 \cap t_2 \cap \bar{t}_3 \cap \cdots \bar{t}_{n-1} \cap \bar{t}_n \\
m_3 & = & \bar{t}_1 \cap \bar{t}_2 \cap t_3 \cap \cdots \bar{t}_{n-1} \cap \bar{t}_n \\
\vdots & & \\
m_{2^n-2} & = & \bar{t}_1 \cap t_2 \cap t_3 \cap \cdots t_{n-1} \cap t_n \\
m_{2^n-1} & = & t_1 \cap t_2 \cap t_3 \cap \cdots t_{n-1} \cap t_n
\end{array}$$

3 Language models

After setting up the framework, now we will show how two popular language models can be interpreted in terms of the PCS model. The basic idea of language models is as follows: Queries and documents are generated by statistical language models. In order to estimate the relevance of a document with respect to a query, we ask for the likelihood that they were both created by the same language model. So we are looking for the probability of $P(d \cap q)$.

In terms of our PCS model, this probability is computed via summing up the probabilities of the disjoint concepts:

$$P(d \cap q) = \sum_m P(m \cap d) P(q \cap m) \tag{1}$$

In the most basic case, language models assume a query to be a set of terms. In the framework used here, this means that a query is a single atomic concept

$$q = m_q = t_1^{\beta_1} \cap \cdots \cap t_n^{\beta_n}.$$

Here β_i indicates whether the term t_i is present or absent in the query:

$$t_i^{\beta_i} = \left\{ \begin{array}{ll} t_i & \text{if} \quad \beta_i = 1, \\ \bar{t}_i & \text{if} \quad \beta_i = 0. \end{array} \right.$$

Thus, we get for the joint probability of d and q:

$$\begin{array}{rcl}
P(d \cap q) & = & P(m_q \cap d) P(q \cap m_q) \\
& = & P(d \cap m_q) \\
& = & P(m_q | d) P(d)
\end{array} \tag{2}$$

Next we assume independence of terms.

$$P(t_1^{\beta_1} \cap \cdots \cap t_n^{\beta_n}|d) = \prod_{i=1}^{n} P(t_i^{\beta_i}|d) \tag{3}$$

Combining this assumption with eqn 2, we get

$$P(d \cap q) = P(d) \prod_{i=1}^{n} P(t_i^{\beta_i}|d) \tag{4}$$

Now we can split the product into two parts, depending on the fact whether the term is present or absent in the query:

$$P(d \cap q) = P(d) \cdot \prod_{\beta_i=1} P(t_i|d) \prod_{\beta_i=0} P(\bar{t}_i|d) \tag{5}$$

Based on this formula, we now can explain both the Ponte/Croft model and Hiemstra's.

Ponte and Croft [Ponte & Croft 98] consider the 'classical' implication direction $d \rightarrow q$, thus we have

$$\begin{aligned} P(d \rightarrow q) &= P(q|d) = \frac{P(d \cap q)}{P(d)} = \prod_{\beta_i=1} P(t_i|d) \prod_{\beta_i=0} P(\bar{t}_i|d) \\ &= \prod_{\beta_i=1} P(t_i|d) \prod_{\beta_i=0} (1 - P(t_i|d)) \end{aligned} \tag{6}$$

[Ponte & Croft 98].

In contrast, Hiemstra [Hiemstra 98] considers the other implication direction:

$$P(q \rightarrow d) = P(d|q) = \frac{P(d \cap q)}{P(q)} = \frac{P(d)}{P(m_q)} \prod_{\beta_i=1} P(t_i|d) \prod_{\beta_i=0} P(\bar{t}_i|d) \tag{7}$$

In addition, Hiemstra assumes that terms occuring in the document but not in the query have no effect on the relevance of a document, thus the second product of the last equation is omitted, and we arrive at the ranking formula:

$$P(q \rightarrow d) = \frac{P(d)}{P(m_q)} \prod_{\beta_i=1} P(t_i|d) \tag{8}$$

Since $P(m_q)$ is constant for a given query, its value is not needed for computing a ranking wrt. a query. So both models require the estimation of the parameters $P(t_i|d)$; Hiemstra's model also needs an estimate for $P(d)$, but this parameter can be assumed to be equal for all documents.

The parameters $P(t_i|d)$ can be derived by assuming a statistical language generation process for the document text. Thus, this parameter could be estimated as the relative within-document frequency of t_i within the document d. However, since the document is the only observation that we have for the underlying language gmodel, we have to cope with the problem of sparse data. Thus, it is assumed that there is another, more general language model which generates the whole document collection. For estimating $P(t_i|d)$, observations from the two language models are combined, i.e. by considering both the within-document frequency of t_i in d and the inverse document frequency of t_i in the collection.

4 Conclusions and open issues

Comparing the language models with the classical IR models as explained in [Wong & Yao 95], it becomes apparent that none of the classical models provides a solid theoretical solution for the document indexing problem: Given a specific document d, how should it be mapped into the concept space, i.e. how to estimate $P(t_i|d)$? This problem is solved by bthe language models, by connecting statistical models of language with probabilistic IR models. In addition, solutions for the sparse data problem are provided.

However, a closer look reveals new problems. The basic idea in the estimation of $P(t_i|d)$ is that this probability can be estimated by the relative frequency of the term t_i in the document d, and the inverse document frequency is only considered due to the sparse data problem. But this approach is inconsistent: Using the relative frequency is only appropriate in case we assume that $P(t_i \cap t_j|d) = 0$ whenever $i \neq j$. That is, in the probability estimation process, terms are disjoint events, whereas in the concept space, they are assumed to be independent.

One possible way out is the assumption that $P(t_i|d)$ is only *proportional* to the relative frequency of t_i in d. Assuming a constant factor for all query terms, the ranking would not be affected by this change.

In any case, the precise definition of a mapping from term occurrences within document texts into terms and documents in concept space still is an open issue.

References

Hiemstra, D. (1998). A Linguistically Motivated Probabilistic Model of Information Retrieval. In: Nikolaou, C.; Stephanidis, C. (eds.): *Lecture Notes In Computer Science - Research and Advanced Technology for Digital Libraries - Proceedings of the second European Conference on Research and Advanced Technology for Digital Libraries: ECDL'98*, pages 569–584. Springer Verlag.

Nie, J. (1989). An Information Retrieval Model Based on Modal Logic. *Information processing & management. 25(5)*, pages 477–491.

Ponte, J.; Croft, W. (1998). A Language Modeling Approach to Information Retrieval. In: Croft et al. (ed.): *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281. ACM, New York.

van Rijsbergen, C. J. (1986). A Non-Classical Logic for Information Retrieval. *The Computer Journal 29(6)*, pages 481–485.

Wong, S.; Yao, Y. (1995). On Modeling Information Retrieval with Probabilistic Inference. *ACM Transactions on Information Systems 13(1)*, pages 38–68.