Document and Passage Retrieval Based on Hidden Markov Models

Elke Mittendorf
mittendorf@inf.ethz.ch

Peter Schäuble
schauble@inf.ethz.ch

Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich (Switzerland)

Abstract

Introduced is a new approach to Information Retrieval developed on the basis of Hidden Markov Models (HMMs). HMMs are shown to provide a mathematically sound framework for retrieving documents—documents with predefined boundaries and also entities of information that are of arbitrary lengths and formats (passage retrieval). Our retrieval model is shown to encompass promising capabilities: First, the position of occurrences of indexing features can be used for indexing. Positional information is essential, for instance, when considering phrases, negation, and the proximity of features. Second, from training collections we can derive automatically optimal weights for arbitrary features. Third, a query dependent structure can be determined for every document by segmenting the documents into passages that are either relevant or irrelevant to the query. The theoretical analysis of our retrieval model is complemented by the results of preliminary experiments.

1 Introduction

We introduce a new approach to Information Retrieval, i.e. document retrieval and passage retrieval. Documents are considered as being produced by stochastic processes. A first stochastic process generates text fragments that are relevant to a certain query. A second stochastic process generates text fragments independent of any particular query. The generation of text fragments by the two stochastic processes is modeled by means of two Hidden Markov Models (HMMs). Whether one of these two HMMs generates a text fragment with a high probability or with a low probability depends on the distribution of the query features within the text fragment.

In the case of document retrieval, the documents are assigned scores that depend on the ratio of the probability that the document was generated by the first stochastic process and the probability that the document was generated by the second stochastic process. As usual, the documents are presented to the user in decreasing order of their scores. In the case of passage retrieval, the score of a passage depends on the probability that the passage itself was generated by the first stochastic process and the text fragments before and after the passage are generated by the second stochastic process.

There are three problems, that are considered difficult in Information Retrieval. First, it is not well known how complex features (e.g. phrases, proximity data, negations, cooccurrence and cocitation data etc.) should be used for indexing. Second, we lack a general weighting scheme for arbitrary indexing features and for arbitrary document collections. Third, the optimal segmentation of a long document into segments that are either relevant or irrelevant to a query is another open problem. Our approach encompasses promising capabilities to solve these three problems at least partially. First, information about positions of features can be conserved, because in our approach a document is considered as being produced by a stochastic process. Conventional retrieval models do not take into account the positions where the indexing features occur. Second, the problem of feature weighting is partially solved by an effective and efficient algorithm (Baum–Welch) which determines automatically the stochastic parameters of the model. These stochastic parameters depend on the domain but not on the queries. Third, the retrieval method proffers itself as a method well suited for retrieving passages of arbitrary lengths and formats. Furthermore our approach is based on a theoretically sound framework of probabilistic retrieval. A theoretical analysis and some preliminary experiments show that our approach to passage retrieval works well.

The approach described in this paper is a second attempt to employ HMMs (which play an important role in speech recognition at the present time). A first attempt by Schäuble and Glavitsch [21] had several drawbacks. The new approach presented in this paper is similar to a recently proposed method for spotting keywords in a speech recording [18]. It would seem that in the future, HMM matching will also become popular in fields other than speech recognition [7], [4], [9], [10]. HMM matching may supersede dynamic programming techniques as used for instance in [25] because in contrast to the dynamic

programming approach, with its intuitive and empirical characteristics, HMMs are mathematically more sound and they encompass dynamic programming as a special case [1].

In what follows we summarize *related work* on passage retrieval. Methods for retrieving passages of predefined lengths (e.g. sentences, paragraphs, sections, pages, 30-word segments, etc.) are described in: [8], [13], [14], and [22]. As already mentioned, methods for retrieving passages of arbitrary lengths and formats are preferable due to their greater effectiveness.

O'Connor [15] enhances his original method by "compound passages" which consist of several adjacent sentences. A rather complicated system of rules specifies whether a single sentence or a compound passage should be retrieved, e.g. if every sentence of the compound passage matches the query in a specific way and if the sentences are connected by phrases like "however" or "on the other hand." O'Connor's approach is domain independent whereas the knowledge based approach of Hahn [3] heavily depends on the domain of discourse. The approaches by O'Connor and Hahn lack a procedure to optimize automatically the characteristics of their retrieval model.

Salton, Allan, and Buckley [19] describe a hierarchical approach. They first retrieve entire documents by means of a global/local text comparison where it may be difficult to find necessary thresholds for this first step. In subsequent steps, the retrieved units are divided into subunits, where a subunit is retrieved iff its score is higher than the score of the unit to which it belongs. Another yet interesting method for retrieving passages of arbitrary lengths and formats is described in [5] where the similarities of consecutive text blocks are computed, graphed, and smoothed. The local minima of the resulting graph are assumed to be the boundaries of the passages. In contrast to our approach, the passage retrieval methods mentioned above lack a theoretical framework, e.g. a formal justification within the framework of probabilistic retrieval.

The paper is organized as follows. In Section 2, a short introduction into the theory of HMMs is given. In Section 3, a document retrieval method is introduced which is based on HMMs. In addition, it is shown under which assumptions, this retrieval method is consistent with the probability ranking principle. In Section 4, the document retrieval method is generalized to passage retrieval. In Section 5, the results of our experiments are presented. In Section 6, some conclusions are drawn.

2 Hidden Markov Model Theory

In this section, we present a short introduction into the HMM theory. Readers who are familiar with HMMs may skip this section while other readers who are interested in a thorough presentation of the theory are referred to [6] or to [11].

The structure $\lambda = \langle S, V, a, b \rangle$ represents a discrete *Hidden Markov Model* (or HMM for short) iff the following conditions are satisfied.

1. $S = \{s_0, \ldots, s_{n-1}\}$ is a finite set of *states* where s_0 always denotes the so-called *initial state.*
2. $V = \{v_0, \ldots, v_{m-1}\}$ is a finite set of *output symbols.*
3. $a : S \times S \rightarrow [0,1], (s_i, s_j) \mapsto a_{i,j}$ is a function determining the *transition probabilities* $a_{i,j}$ such that $\sum_{j=0}^{n-1} a_{i,j} = 1$ for all $i = 0, \ldots, n-1$.
4. $b : S \times V \rightarrow [0,1], (s_j, v_k) \mapsto b_{j,k}$ is a function determining the *output probabilities* $b_{j,k}$ such that $\sum_{k=0}^{m-1} b_{j,k} = 1$ for all $j = 0, \ldots, n-1$.

A HMM defines a stochastic process that generates output symbols step by step. Every step consists of a state transition from a state s_i to a state s_j and of the generation of an output symbol v_k upon arriving at state s_j.

The transition probability $a_{i,j}$ is the probability that s_j is the next state, given that s_i is the current state. The output probability $b_{j,k}$ is the probability that v_k is generated upon arriving at s_j. In this way, the so-called *Markov assumptions* are satisfied, i.e. these probabilities do not depend on the previously visited states and on the previously generated output symbols.

Every HMM λ is assigned a probability space $\langle \Omega, P_\lambda \rangle$. The sample space Ω consists of pairs (σ, ν) where the function $\sigma : \mathsf{N}^+ \rightarrow S, t \mapsto \sigma_t$ determines an infinite sequence $\sigma_1, \sigma_2, \ldots$ of states and the function $\nu : \mathsf{N}^+ \rightarrow V, t \mapsto \nu_t$ determines an infinite sequence $\nu_1, \nu_2, \ldots$ of output symbols. Let $x = (s_0, x_1, \ldots, x_T)$ be a finite sequence of states that starts with the initial state s_0 and let $y = (y_1, \ldots, y_T)$ be a finite sequence of output symbols. We define the following three events.

$$A(x) := \{(\sigma, \nu) \,|\, x = (s_0, \sigma_1, \ldots, \sigma_T)\} \quad (1)$$

$$B(y) := \{(\sigma, \nu) \,|\, y = (\nu_1, \ldots, \nu_T)\} \quad (2)$$

$$E(x, y) := A(x) \cap B(y) \quad (3)$$

The set $A(x)$ denotes the event where at the outset the state sequence x is traversed while generating any sequence of output symbols. The set $B(y)$ denotes the event where at the outset the sequence of output

symbols y is generated by traversing any state sequence. Finally, the set $E(x,y)$ denotes the event where at the outset y is generated by traversing the state sequence x.

When defining

$$P_\lambda(E(x,y)) \quad := \quad \prod_{t=1}^{T} c(x_{t-1}, x_t, y_t) \tag{4}$$

where

$$c(s_i, s_j, v_k) := a_{i,j} b_{j,k}, \tag{5}$$

it can be shown that $\langle \Omega, P_\lambda \rangle$ is a probability space. For simplicity we will subsequently write $P_\lambda(x,y)$, $P_\lambda(y)$, and $P_\lambda(x)$ instead of $P_\lambda(E(x,y))$, $P_\lambda(B(y))$, and $P_\lambda(A(x))$ respectively.

When using HMMs, there are three problems of interest:

1. Given a HMM $\lambda = \langle S, V, a, b \rangle$ and a sequence of output symbols y, find the probability $P_\lambda(y)$ that y was generated along *any* state sequence.
2. Given a HMM $\lambda = \langle S, V, a, b \rangle$ and a sequence of output symbols y, find a state sequence x along which y is generated with the highest probability $P_\lambda(x,y)$.
3. Given a set Y consisting of sequences y, find a HMM which generates these sequences y with high probabilities.

The first problem is solved by means of the *forward algorithm* that computes the probability

$$P_\lambda(y) \quad = \quad \sum_x P_\lambda(x,y) \tag{6}$$

with a time complexity $O(T*(n+\langle\#\text{positive } a_{i,j}\text{'s}\rangle))$ even though the sum in (6) consists of n^T summands. The second problem is solved by means of the *Viterbi algorithm* that computes

$$\text{argmax}\{P_\lambda(x,y) \,|\, x\} \tag{7}$$

with a time complexity $O(T * (n + \langle\#\text{positive } a_{i,j}\text{'s}\rangle))$ even though a most probable sequence out of n^T different sequences has to be found. The last problem is solved by the elegant *Baum-Welch algorithm* which is based on the following proposition.

Proposition: Let $\lambda = \langle S, V, a, b \rangle$ and $\overline{\lambda} = \langle S, V, \overline{a}, \overline{b} \rangle$ be two discrete HMMs that are based on the same set of states and on the same set of output symbols. In addition, let Y be a set of training sequences where T_y denotes the length of $y = (y_1, \ldots, y_{T_y})$ for every $y \in Y$. Assume that the transition probabilities a and $\overline{a}$ and the output probabilities b and $\overline{b}$ are related by the following reestimation formulas

$$\overline{a}_{i,j} \quad = \quad \frac{\sum_{y \in Y} \sum_{t=1}^{T_y} \gamma_{i,j}[y](t)}{\sum_{y \in Y} \sum_{j=0}^{n-1} \sum_{t=1}^{T_y} \gamma_{i,j}[y](t)}, \quad \overline{b}_{j,k} = \frac{\sum_{y \in Y} \sum_{i=0}^{n-1} \sum_{t: y_t = v_k} \gamma_{i,j}[y](t)}{\sum_{y \in Y} \sum_{i=0}^{n-1} \sum_{t=1}^{T_y} \gamma_{i,j}[y](t)} \tag{8}$$

where $\gamma_{i,j}[y](t)$ is the probability that the process generates y along any state sequence such that the states s_i and s_j are visited at time $t-1$ and at time t respectively.

$$\gamma_{i,j}[y](t) \quad := \quad \alpha_i[y](t-1) c(s_i, s_j, y_t) \beta_j[y](t)$$

$$\alpha_j[y](t) \quad := \quad \begin{cases} 0 \text{ if } j \neq 0 \text{ and } t = 0 \\ 1 \text{ if } j = 0 \text{ and } t = 0 \\ \sum_i \alpha_i[y](t-1) c(s_i, s_j, y_t) \text{ if } 0 < t \end{cases}$$

$$\beta_i[y](t) \quad := \quad \begin{cases} 1 \text{ if } t = T_y \\ \sum_j c(s_i, s_j, y_{t+1}) \beta_j[y](t+1) \text{ if } t < T_y \end{cases}$$

If the reestimation formulas yield the same probabilities (i.e. $\overline{a} \equiv a$ and $\overline{b} \equiv b$), then

$$\sum_{y \in Y} P_\lambda(y) \tag{9}$$

is a *local maximum* with respect to a and b. This local maximum is the probability that λ generates a sequence y of the training set Y. Otherwise (i.e. if $\exists i,j : a_{i,j} \neq \overline{a}_{i,j}$ or $\exists j,k : b_{j,k} \neq \overline{b}_{j,k}$), the strong inequality

$$\sum_{y \in Y} P_\lambda(y) \quad < \quad \sum_{y \in Y} P_{\overline{\lambda}}(y) \tag{10}$$

holds. *End of proposition.*

This proposition leads to the Baum-Welch algorithm to find a HMM which generates the sequences y of a given set Y with high probabilities. First, an appropriate initial HMM is chosen. Second, the transition probabilities and the output probabilities are reestimated by using the reestimation formulas and the training set Y. According to the proposition, the new HMM generates the training sequences with a higher probability than the original HMM does (unless we are already in a local maximum). The second step can be repeated until an appropriate convergence criterion is met.

3 Document Retrieval Based on Hidden Markov Models

In this section we introduce a document retrieval method based on HMMs. In the next section we show that this document retrieval method lends itself well to passage retrieval. At the outset of this section, we describe how documents are converted into sequences of output symbols. We then come to the key idea of our approach where we use HMMs generating such sequences of output symbols. Finally, we conclude this section by showing that our document retrieval method based on HMMs encompasses a conventional retrieval method that is based on a well known weighting scheme and on a simple normalization scheme.

When using HMMs, the documents have to be considered as sequences of certain output symbols. If we would define the set of output symbols V to be the set of all terms Φ in a document collection, the cardinality of this set makes any estimation of parameters unfeasible and the whole model would also be very sensitive to changes in the document collection. Pattern recognition solves this problem by extracting significant features and by comparing them to the query. This idea is not new in information retrieval; for instance, it is used to estimate parameters of probabilistic retrieval models. In our case we need a mapping $\vec{v} : \Phi \times Q \rightarrow \mathsf{R}^k$,

$$\vec{v}(\varphi, q) \quad := \quad \begin{pmatrix} sim_0(\varphi, q) \\ \vdots \\ sim_{k-1}(\varphi, q) \end{pmatrix}. \tag{11}$$

such that a vector $\vec{v}(\varphi, q)$ describes the similarity between a token φ and a query q in a way that preserves information about relevance. That means if φ_1 and φ_2 both seem equally relevant to the query they should be mapped to vectors very close to each other with respect to an appropriate distance measure.

The image set $\vec{v}(\Phi, Q) \subseteq \mathsf{R}^k$ may contain infinitely many elements. These elements are grouped into disjoint clusters by means of an appropriate vector quantization technique [12]. Every cluster is uniquely represented by a centroid and the (finite) set of centroids (or their indexes) serves as an output alphabet. In this way, every pair (q, d) consisting of a query q and of a document d is assigned a sequence $Y(q, d) = y$ where

$$Y : Q \times D \rightarrow V^*, (q, d) \mapsto y \tag{12}$$

denotes the corresponding function and every element y_t of the sequence

$$Y(q, d) = y = (y_1, \ldots, y_T) \tag{13}$$

denotes an output symbol v_k representing a centroid which approximates a vector $\vec{v}(\varphi, q)$

We now describe the *key idea* of our approach. At the same time we disclose the relationship between our retrieval method based on HMMs and a probabilistic retrieval method based on the probability space $\underline{Q} \times \underline{D}$ containing simple events $(\underline{q}, \underline{d})$ as described in [2]. Such a simple event $(\underline{q}, \underline{d})$ consists of an individual usage (or an interpretation) of the query q and of an individual usage (or an interpretation) of the document d. Depending on the interpretation, d may be considered as relevant to q (i.e. $(\underline{q}, \underline{d}) \in R$) or d may be considered as irrelevant to q (i.e. $(\underline{q}, \underline{d}) \in \overline{R}$).

We assume a person π sitting in a room and generating sequences y in the following way. In a first step, π selects an information need expressed by a query q. In a second step, π writes a document he or she considers as relevant to q. In probabilistic retrieval, these two steps correspond to a random selection of a simple event $(\underline{q}, \underline{d}) \in R$. In a third step, π applies the function Y and he or she submits the sequence $y := Y(\underline{q}, \underline{d})$ to the outer world who does not know to which query and to which document the sequence y corresponds. Within the framework of probabilistic retrieval,

$$P(y \,|\, R) \quad = \sum_{(q,d):Y(q,d)=y} P(q, d \,|\, R) \tag{14}$$

is the probability that y was generated by π.

Similarly, we assume another person π' sitting in a different room and generating sequences y in the following way. In first step π' selects an information need expressed by a query q. In a second step, π' writes a document independent of whether he or she considers d as relevant to q or not. In probabilistic retrieval, these two steps correspond to a random selection of a simple event $(\underline{q}, \underline{d}) \in \underline{Q} \times \underline{D}$. In a third step, π' applies the function Y and he or she submits the sequence $y := Y(\underline{q}, \underline{d})$ to the outer world who does not know to which query and to which document the sequence y corresponds. Within the framework of probabilistic retrieval,

$$P(y) \quad = \sum_{(q,d):Y(q,d)=y} P(q, d) \tag{15}$$

is the probability that y was generated by π'.

The sequences y produced by the person π are considered as being generated by a stochastic process which we approximate by means of a HMM λ such that

$$P_\lambda(y) \quad \approx \quad P(y \,|\, R). \tag{16}$$

Similarly, the sequences y produced by the person π' are considered as being generated by a stochastic process which we approximate by means of a HMM λ' such that

$$P_{\lambda'}(y) \approx P(y). \tag{17}$$

The HMM λ is called the *passage model* and the HMM λ' is called the *background model* for reasons that will become clear in the next section. Using the HMMs λ and λ' we define a retrieval method by

$$RSV(q,d) := \log\left(\sqrt[T]{\frac{P_\lambda(Y(q,d))}{P_{\lambda'}(Y(q,d))}}\right). \tag{18}$$

This definition can be justified if we make three assumptions. First, the approximations (16) and (17) are assumed to be exact. Second, we assume that all queries and all documents are mapped to sequences $Y(q,d)$ of the *same* positive length T. This assumption implies that we can restrict ourselves to the scoring function

$$RSV'(q,d) := \log\left(\frac{P_\lambda(Y(q,d))}{P_{\lambda'}(Y(q,d))}\right). \tag{19}$$

Third, we assume that Y is injective such that $P(Y(q,d)\,|\,R) = P(q,d\,|\,R)$ and $P(Y(q,d)) = P(q,d)$. This assumption facilitates the simplification

$$RSV'(q,d) = \log\left(\frac{P(q,d\,|\,R)}{P(q,d)}\right) = \log\left(\frac{P(q,d,R)}{P(q,d)P(R)}\right) = \log\left(P(R\,|\,q,d)\right) - \log\left(P(R)\right)$$

which shows that under the three assumptions made above, RSV and $P(R\,|\,q,d)$ induce the same weak ordering of the documents because both the Tth root and the logarithm log are strictly monotonic on the open interval $(0,\infty)$. Thus, our retrieval method based on HMMs is consistent with the probability ranking principle by Robertson [17]. For the case where the image $Y(Q,D)$ consists of sequences of *different lengths*, the Tth root in (18) provides a normalization such that short sequences do not have preference over long sequences. The same normalization is successfully used in keyword spotting systems [18], [24]. This completes the presentation of a document retrieval method based on HMMs and its relationship to probabilistic retrieval.

We conclude this section by showing that our document retrieval method based on HMMs encompasses a conventional retrieval method that is based on a well known weighting scheme and on a simple normalization scheme. For we assume that both λ and λ' are based on constant Markov chains. Thus, we have $S = \{s_0\}$, $S' = \{s'_0\}$, $a_{0,0} = 1$ and $a'_{0,0} = 1$. The similarity function $\vec{v} : \Phi \times Q \rightarrow \mathsf{R}$ is chosen to be a mapping into the set of one dimensional vectors (i.e. into the real numbers).

$$\vec{v}(\varphi,q) := ff(\varphi,q)\cdot idf^2(\varphi), \tag{20}$$

where ff denotes the features frequency and idf denotes the inverse document frequency as explained in [20]. If the feature frequencies in all queries are assumed to be bounded, and if the collection is as small as e.g. MEDLARS, the image set $\vec{v}(\Phi,Q)$ is small enough to serve as an output set for λ and λ'. We therefore define $V := \vec{v}(\Phi\times Q) = \{v_0,\ldots,v_{m-1}\}$. The output probabilities of the background model λ'

$$b'_{0,k} := u(v_k) \tag{21}$$

are initialized by an arbitrary probability function $u : V \rightarrow [0,1]$, e.g. such that $u(v_k)$ is the relative frequency of the symbol v_k in some training collection. The output probabilities of the passage model λ are initialized as follows.

$$b_{0,k} := \frac{u(v_k)\exp(\kappa v_k)}{c(\kappa)} \tag{22}$$

where

$$c(\kappa) := \sum_{i=0}^{m-1} u(v_k)\exp(\kappa v_k). \tag{23}$$

Let us now consider the representation of a document d consisting of T tokens, represented relative to a query as in (13). By definition we have for $\vec{v}(\varphi,q) = y_t$ that $y_t = ff(\varphi,q)\cdot idf^2(\varphi)$, $i := 1,\ldots,T-1$. We then can show that

$$RSV(q,d) = \frac{\kappa\sum_{\varphi\in\Phi} ff(\varphi,d)ff(\varphi,q)idf^2(\varphi)}{T} - \log c(\kappa). \tag{24}$$

Thus, this retrieval model yields the same list of ranked documents as vector space retrieval with the scalar product of $ff * idf$-vectors and with a normalization by the number of tokens.

For larger collections the set $\vec{v}(\Phi,Q)$ must be compressed to a codebook with a reasonable size. That leads to a small quantization error in (24) which has marginal effects on the ranking order.

4 Passage Retrieval Based on Hidden Markov Models

In this section, we present a simple passage retrieval method based on HMMs. We use the passage model λ and the background model λ' to build a larger HMM called document model μ. We describe how the

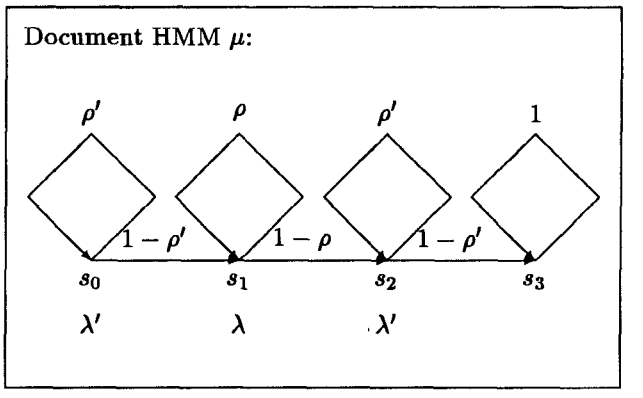


Figure 1. Structure of the document HMM μ, i.e. a simple HMM for passage retrieval.

Viterbi algorithm is used to assign scores to passages of a document. Finally, we show how this passage retrieval method is related to vector space retrieval.

The main idea is that the passage model λ and the background model λ' are concatenated to obtain a document model μ which generates relevant text fragments while traversing λ, and which generates arbitrary text fragments while traversing λ'.

We started with a simple structure for μ as shown in Figure 1. In state s_3, only end-of-document markers are generated. The probability that an end-of-document marker is generated in another state equals to zero.

Suppose we are given a query q and a long document, say a book d. The query and the document are mapped by (12) to a sequence of output symbols $Y(q,d) =: y$. Every path x through the document model μ with positive probability $P(x,y)$ has the form

$$x = (s_0, \ldots, s_0, s_1, \ldots, s_1, s_2, \ldots, s_2, s_3).$$

The paths of this form correspond bijectively to the passages of the book in an obvious way.

We use the Viterbi algorithm to trace for a path through the document model μ, i.e. for a path that has produced y with maximal probability. We call such a path a *best path*. The passage that corresponds to a best path is assumed to be a best (i.e. a most probable) passage with respect to the query.

The following algorithm yields a list of disjoint passages that are ranked in an appropriate order. We will call this algorithm *Viterbi ranking*. The top ranked passage is a passage that corresponds to a best path. We then cut off the top ranked passage, we consider the book without this passage, and again, we trace for a best path. This is repeated until there are as many passages on the list as needed by the user.

Problems might occur with the Viterbi ranking, since a retrieved "passage" may contain a gap due to a cut off in a previous step. This problem did not occur in our experiments.

We now analyze this passage retrieval method in detail. Again we investigate a relationship with vector space retrieval. Consider this particular document HMM μ (Figure 1), the similarity function defined in (20), and the output probabilities defined in (21) and (22), where the document frequencies are obtained from an appropriate training collection. Let p be a passage consisting of T_p tokens and let x_p be the corresponding path. We can show that

$$\log(P(x_p, Y(q,d))) = \kappa \sum_{\varphi \in \Phi} ff(\varphi,p) ff(\varphi,q) idf^2(\varphi) - T_p \log\left(c(\kappa)\frac{\rho}{\rho'}\right) + \log G, \tag{25}$$

where G is a value that is constant for all possible paths x_p. The probability $P(x_p, Y(q,d))$ can be regarded as the score for a passage p. After each step of the Viterbi ranking, the passage with the highest score (25) is appended to the list.

Equation (25) discloses the following properties. First we note that the score only depends on

1. the scalar product of the $ff * idf$-vectors, representing the passage p and the query q,
2. the parameter $c(\kappa)$ times the ratio ρ/ρ'.

For every token φ in the passage, the score is incremented by $\kappa \cdot ff(\varphi,q) \cdot idf^2(\varphi)$ and it is decremented by $\log(c(\kappa)\rho/\rho')$. If $\kappa ff(\varphi,q) idf^2(\varphi) > \log(c(\kappa)\rho/\rho')$ the token ϕ causes an increase of the score; otherwise, it causes a decrease of the score. By changing κ and the ratio ρ/ρ', the expected length of the retrieved passages can be manipulated. The larger $c(\kappa)\rho/\rho'$, the shorter is the average length of the retrieved passages. We also can see that for this particular passage retrieval method a retrieved passage always starts and ends with a token that occurs in the query. To avoid this we are deliberating on other HMM structures as well as on other similarity functions, i.e. other than the function (20).

We have presented simple HMMs for passage retrieval and we have shown that in this case the Viterbi ranking provides meaningful scores. Based on the experience we get from our experiments, we hopefully will be able to design more sophisticated HMMs.

5 Experiments

In this section, we describe three experiments. The first two experiments are about passage retrieval based on HMMs with different output probability distributions. The third experiment is about retrieving passages of fixed formats (sentences) which is used as a reference method.

Evaluating passage retrieval is problematic, since appropriate evaluation methods do not exist. It is not clear how to consider the percentage of the retrieved passage that is relevant and the percentage of the relevant passage that is covered by the retrieved passage. Moreover, we lack relevance judgements containing detailed information about relevant passages.

To get an idea of the effectiveness of our passage retrieval method, the following framework for experiments has been designed. Standard document collections are considered as one large document without document boundaries. The concatenation of all documents of a particular collection is called a book d. The set Φ is the set of terms, consisting of word classes obtained by Porter's reduction algorithm [16] after removing the stop words listed in [23, pp. 18–21]. The similarity function is defined by (20). The models λ and λ' are concatenated as shown in Figure 1.

Using the Viterbi ranking, the documents d_j of the book d are ranked as follows. A document d_j is ranked next if the currently best passage starts in d_j. The whole document d_j is then cut off from the book d. We decided to rank only this document, even if the retrieved passage covers more than one document of the collection. This can be justified by the fact that a user usually starts reading a retrieved passage from the beginning.

Below we describe three different experiments. Two experiments are based on HMMs and one experiment serves as a reference method.

Experiment 1: (HMM0) For testing the initialization analyzed in Section 4, λ is defined by (22) and λ' by (21). Thus in the model μ we have $b_{0,k} = b_{2,k} = u(v_k)$ and $b_{1,k} = u(v_k)\exp(\kappa v_k)/c(\kappa)$, k=0,..., m-1. The parameter κ is set to 0.2. No training is performed.

Experiment 2: (HMM1) The output probabilities of λ and λ' are optimized by means of the reestimation formula (8). The training data for λ consists of $\{(q,d) \mid d \text{ relevant to } q\}$ and the training data for λ' consists of $\{(q,d) \mid d \in D,\ q \in Q\}$.

For all experiments the transition probabilities ρ and ρ' are set to be equal, $\rho = \rho' = 0.9999$.

A comparison of vector space retrieval and our method of passage retrieval would not be fair, since vector space retrieval uses the information about the document boundaries and passage retrieval does not. For that reason, we let the vector space retrieval perform a primitive kind of passage retrieval in the following way: all documents d_j of a collection are segmented into sentences, since this is the smallest reasonable unit for a passage. After stopwording and stemming, a conventional vector space retrieval with cosine measure and $ff*idf$-weights is performed on the sentences. From the list of ranked sentences a document ordering is built in the following way: the top ranked document is the one which contains the top ranked sentence. Then we append the document to the list in which the sentence occurs that is ranked next, but only if this document has not already been ranked.

Experiment 3: (sent) Vector space retrieval of sentences with standard cosine measure.

Each experiment is performed on two disjoint subsets of the test collection to have the possibility to separate a test set and a training set.

Setting A: The tests are performed on odd numbered queries. For experiment 2 the even numbered queries and their relevance assessments are used to train the HMMs.

Setting B: The tests are performed only on even numbered queries. Training is performed with odd numbered queries.

We show the recall-precision graphs determined by the experiments on the MEDLARS test collection. Figure 2 shows the recall-precision graphs for test setting A and Figure 3 shows the recall-precision graphs for test setting B.

From experiment 2 one can see that the Baum–Welch algorithm is able to assign appropriate weights to extracted similarity features. Another observation is that the results for experiment 2 did not depend on the initializations of the HMMs before training. Experiments with different initializations have shown the same results. That shows that the reestimation formula (8) is not trapped by inappropriate local maxima.

Furthermore, we can see that HMM based retrieval is clearly more effective in adjusting the right size of a passage than a rigid presegmenting into sentences. In other experiments, in which passages were

Figure 2. Effectiveness of HMM based passage retrieval on the MEDLARS test collection using odd numbered queries, (setting A).

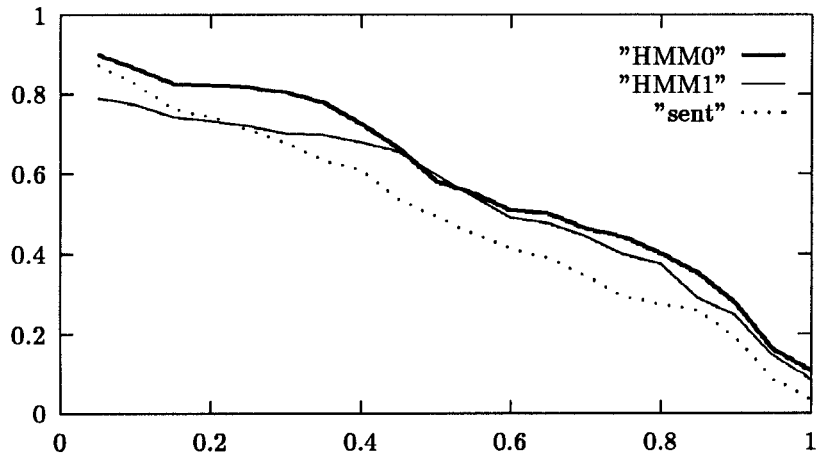
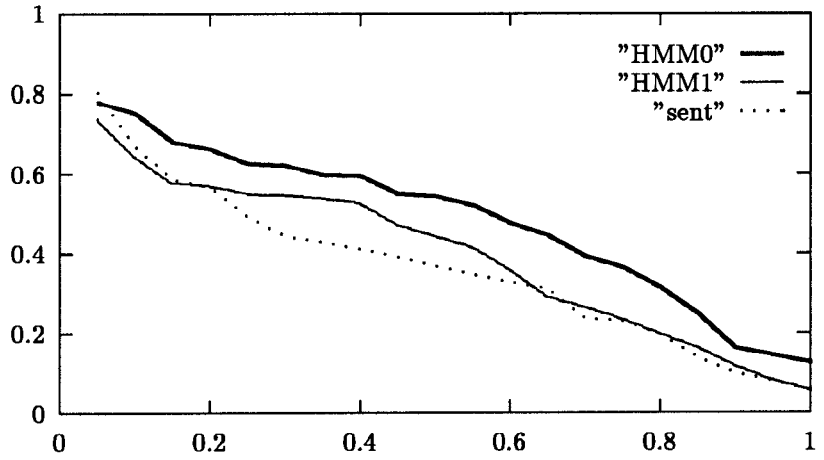


Figure 3. Effectiveness of HMM based passage retrieval on the MEDLARS test collection using even numbered queries, (setting B).



presegmented to contain a fixed number of tokens, the retrieval effectiveness was at best as good as that in experiment 3, but usually worse.

Due to the lack of appropriate relevance judgements and evaluation measures, our evaluation framework does not measure, how much information relevant to the query is covered by and contained in a retrieved passage. However, it was interesting to observe that very often—when consecutive documents are relevant to a query—the most relevant passage consisted exactly of those adjacent documents. For instance, in the MEDLARS test collection, the consecutive documents 164–172 are relevant to query 1. Interestingly, our passage retrieval method identified these consecutive documents as a single passage and it assigned this passage the highest score.

6 Conclusions and Outlook

The analysis presented in Section 4 as well as the experiments presented in Section 5 show that our approach to passage retrieval based on HMMs is feasible. In addition, we have shown that HMMs provide a mathematically sound framework for passage retrieval which can be explained in terms of probabilistic retrieval.

We described a rather simple way to use HMMs, since the passage model and the background model are based on constant Markov chains. Despite this simple approach, the results of the experiments are absolutely encouraging. Even more importantly, this simple approach provides a better understanding of both document retrieval based on HMMs and passage retrieval based on HMMs, because both types of retrieval could be interpreted in terms of conventional vector space retrieval.

Having shown the feasibility of HMM based retrieval and having understood the relationship between HMM based retrieval, probabilistic retrieval, and vector space retrieval, we look forward to using more sophisticated HMMs that model non-trivial structures of the running text (stochastic grammars). We hope that the Baum–Welch algorithm is able to estimate appropriate weights for the various syntactic structures that can be identified by state-of-the-art natural language processing devices.

References

1. J. S. Bridle. Stochastic Models and Template Matching: some Important Relationships Between Two Apparently Different Techniques for Automatic Speech Recognition. In *Institute of Acoustics, Autumn Conference*, pages 452a–452h, 1984.
2. N. Fuhr. Probabilistic Models in Information Retrieval. *The Computer Journal*, 35(3):243–255, 1992.
3. U. Hahn. Topic Parsing: Accounting for Macro Structures in Full-Text Analysis. *Information Processing & Management*, 26(1):135–170, 1990.
4. Y. He, N.Y. Chen, and A. Kundu. Handwritten Word Recognition Using HMM with Adaptive Length Viterbi Algorithm. In *International Conference on Acoustics, Speech, and Signal Processing*, pages III–153–III–156, 1992.
5. M. A. Hearst and C. Plaunt. Subtopic Structuring for Full-Length Access. In *ACM SIGIR Conference on R&D in Information Retrieval*, pages 59–68, 1993.
6. X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition.* Edinburgh University Press, Edinburgh, 1990.
7. J. J. Hull. A Hidden Markov Model for Language Syntax in Text Recognition. In *11th IAPR International Conference on Pattern Recognition. Vol.II*, pages 124–127, 1992.
8. R. Kuhlen and M. Hess. Passagen-Retrieval–auch eine Möglichkeit der automatischen Verknüpfung in Hypertexten. In *Information Retrieval'93*, pages 100–115, Konstanz, Deutschland, October, 1993. Hochschulverband für Informationswissenschaft (HI).
9. J. Kupiec. Hidden Markov Estimation for Unrestricted Stochastic Context-Free Grammars. In *International Conference on Acoustics, Speech, and Signal Processing*, pages I–177–180, 1992.
10. J. Kupiec. Robust Part-of-Speech Tagging Using a Hidden Markov Model. *Computer Speech and Language*, 6:225–242, 1992.
11. L. W. Levinson, L. R. Rabiner, and M. M. Sondhi. An Introduction to the Application of the Theory of Probabilistic Functions of Markov Process to Automatic Speech Recognition. *The Bell System Technical Journal*, 62(4):1035–1075, 1983.
12. Y. Linde, A. Buzo, and R. M. Gray. An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communication*, 28(1):84–95, 1980.
13. A. Moffat, R. Sacks-Davis, R. Wilkinson, and J. Zobel. Retrieval of Partial Documents. In *TREC-2 Proceedings*, 1993.
14. J. O'Connor. Retrieval of Answer-Sentences and Answer-Figures from Papers by Text Searching. *Information Processing & Management*, 11(5/7):155–164, 1975.
15. J. O'Connor. Answer Passage Retrieval by Text Searching. *Journal of the ASIS*, 31(4):227–239, 1980.
16. M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 14(3):130–137, 1980.
17. S. E. Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

18. R. C. Rose and D. B. Paul. A Hidden Markov Model Based Keyword Recognition System. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 129–132, 1990.

19. G. Salton, J. Allan, and C. Buckley. Approaches to Passage Retrieval in Full Text Information Systems. In *ACM SIGIR Conference on R&D in Information Retrieval*, pages 49–58, 1993.

20. G. Salton and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

21. P. Schäuble and U. Glavitsch. A Probabilistic Hypermedia Retrieval Model Based on Hidden Markov Models. Workshop Intelligent Access to Information Systems, Darmstadt, Nov. 1990.

22. C. Stanfill and D. Waltz. Statistical Methods, Artificial Intelligence, and Information Retrieval. In P. S. Jacobs, editor, *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, pages 215–226. Lawrence Erlbaum Associates, 1992.

23. C. J. van Rijsbergen. *Information Retrieval.* Butterworths, London, second edition, 1979.

24. L. D. Wilcox and M. A. Bush. HMM-Based Wordspotting for Voice Editing and Indexing. In *European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 25–28, 1991.

25. P. Willet and A. M. Robertson. Searching for Historical Word-Forms in a Database of 17th-Century English Text Using Spelling-Correction Methods. In N. Belkin, P. Ingwersen, and A. M. Pejtersen, editors, *ACM SIGIR Conference on R&D in Information Retrieval*, pages 256–265, 1992.