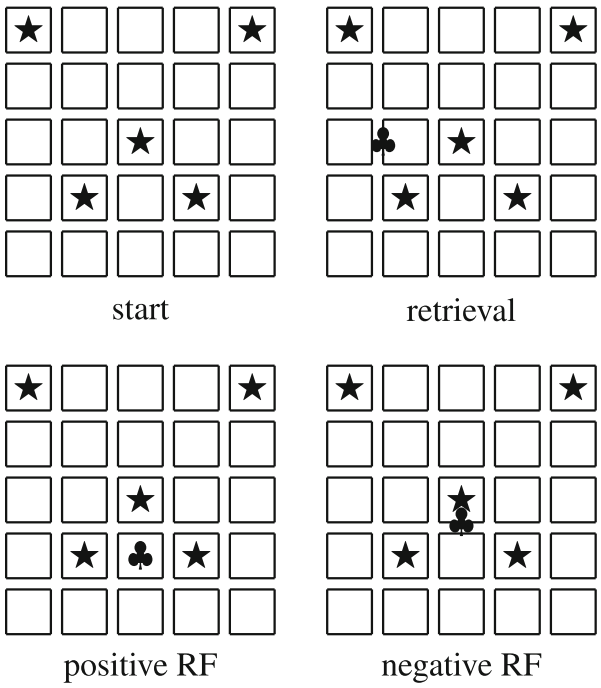

Fig. 1.17 Positive and negative Relevance Feedback

1.4 Probabilistic Models

The probabilistic models of IR have in common the presence of a universe of elementary events where an event is a single occurrence of a process or phenomenon; it is called "elementary" since it cannot be decomposed into simpler occurrences. In the literature of probability theory, the notion of elementary event is distinguished from that of event; the latter is a subset of elementary events. The notion of event is crucial because a probability measure that maps an event to the unit range $[0, 1]$ is applied to obtain the degree of belief that an elementary event belongs to the event.

The role played by the probabilistic models has become important since the Boolean model has been difficult to apply in IR tasks; the researcher has to cope with the lack of ranking, and the end user has to face null output and output overload. The VSM succeeded in improving the user's experience since it provides some ranking, but it leaves the problem of finding the coefficients of the linear combinations open.

The probabilistic models provide an answer to this problem yet at a principled level, that is, it explains how to provide the weights of coordination level used when Boolean logic is applied to IR and also provides an explanation of why the TFIDF of the VSM has been so effective. However, its impact is not only at a principled level—at present, the probabilistic models are also well accepted at the industrial

level. In this section, we survey two important probabilistic models for IR: the relevance model and the language models.

1.4.1 Relevance Model

We call this model "relevance model" because it explicitly represents relevance. The most known implementation, which is illustrated in this section, is also known as "BM25" or "Okapi" or "Robertson-Sparck Jones" model. According to this model, the document collection is viewed as a universe of elementary events. The terms or in general the content descriptors resulting from indexing the collection determine document subsets and therefore implement the events of the probability space. The key notion of this model is that relevance is a document set A, and therefore, it is modeled as an event.

As these events are subsets of a probability space, a probability measure can be applied, and each event (i.e., terms, relevance and their subsets, and logical combinations) is assigned a probability; see also Fig. 1.18 which depicts the event "relevance" as the subset A and the event "a term occurs" as the subset B. To each event, a probability measure P assigns a real number in the unit range in the same way the coordination level assigns a weight to the document and the term.

An algebraic illustration of the relevance model is provided in the following. Suppose a universe of events (i.e., event space) Ω can be split into A and $\bar{A}$ such that

$$A \cup \bar{A} = \Omega \qquad A \cap \bar{A} = \emptyset$$

Similarly, the space is split according to the terms resulting from indexing the collection as follows:

$$B_i \cup \bar{B}_i = \Omega \qquad B_i \cap \bar{B}_i = \emptyset \qquad i = 1, \ldots$$

Note that the same procedure can be applied for Boolean combinations of the terms.

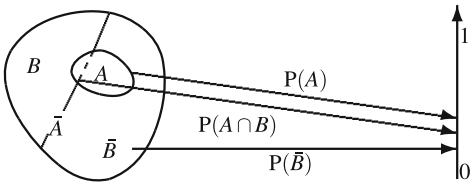

Fig. 1.18 Relevance model

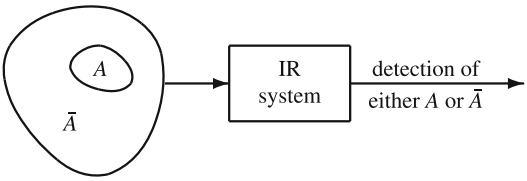

Fig. 1.19 Retrieval as a detection problem

1.4.1.1 Detection and Risk

The main problem with the relevance model is that A is not only unknown, it also changes for each information need. It follows that an IR system based on such a model has to detect whether a document is relevant although context continuously evolves and nothing is known if not for some content descriptors. As a system has to decide whether a document is relevant on the basis of a "signal" given by content descriptors, retrieval is viewed as a detection problem, as depicted in Fig. 1.19.

The dependency of A on the user's information need and ultimately on the context affecting the need is the reason why retrieval is affected by uncertainty, and therefore, the detection performed by the system is inevitably a statistical detection.

When performing such a detection, the system often (always) makes two errors: one error is to retrieve nonrelevant documents and the other error is to miss relevant documents. When making these errors, two costs arise: false alarm is the cost of retrieving (i.e., detecting) documents that are not relevant and loss of recall is the cost of not retrieving documents that are relevant. The detection costs are often encoded as follows:

- $c(A, \bar{A})$ is the cost of retrieving a document because it is decided that it is relevant (first A) when it actually is not relevant (second A); this is the cost of false alarm.
- $c(\bar{A}, A)$ is the cost of not retrieving a document because it is decided that it is not relevant when it actually is relevant; this is the cost of loss of recall.

It is defined $c(A, A)$ and $c(\bar{A}, \bar{A})$ although these costs are very often set to zero.

Although perfect retrieval, retrieval of all and only relevant documents, is impossible to obtain, optimal retrieval, retrieval of the largest number of relevant documents provided the maximum number of nonrelevant documents, can be obtained. To this end, risk is introduced. When the probability measure of the events and the costs is available, the risk of a detection can be computed for each event and can be defined as follows:

$$R(A|B) = c(A, A)\mathrm{P}(A \mid B) + c(A, \bar{A})\mathrm{P}(\bar{A} \mid B)$$

$$R(\bar{A}|B) = c(\bar{A}, A)\mathrm{P}(A \mid B) + c(\bar{A}, \bar{A})\mathrm{P}(\bar{A} \mid B)$$

An IR system decides to retrieve the documents in B when

$$R(A \mid B) < R(\bar{A} \mid B)$$

and this happens if and only if

$$\mathrm{P}(A \mid B) > \frac{c(A,\bar{A}) - c(\bar{A},\bar{A})}{c(\bar{A},A) + c(A,\bar{A}) - c(A,A) - c(\bar{A},\bar{A})}$$

The latter equation shows that the costs are like knobs operating on a device that emits documents. When the cost of loss of recall that can be accepted by the system or the end user increases, the number of retrieved documents increases since the threshold decreases; this can be explained by the fact that when $c(\bar{A},A)$ increases, the miss of relevant documents is less tolerated, and therefore, the system accepts to retrieve further documents. How the costs of loss of recall and false alarm explain the threshold is depicted in Fig. 1.20 when the costs of correct decision are null.

1.4.1.2 Probability Ranking Principle

The effectiveness of the relevance model rests on the PRP:

> 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.

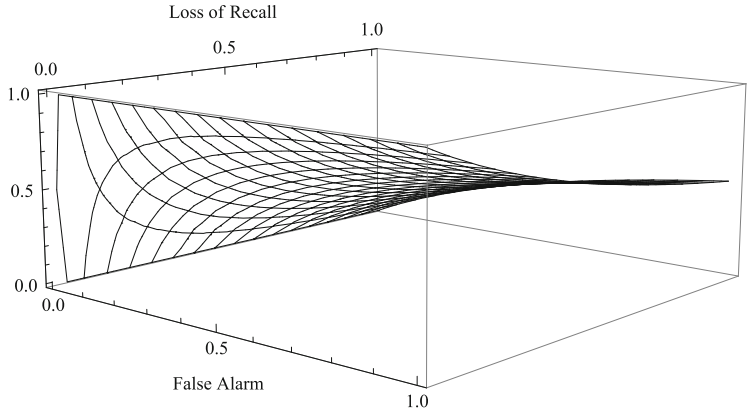

Fig. 1.20 Probability of relevance threshold as function of loss of recall and false alarm

The PRP was first introduced by Maron and Kuhns (1960), and a discussion is reported by Robertson (1977). The reason why the PRP is significant in IR is the link between the principled approach to ranking and the effectiveness measures. The risks defined within the PRP are actually probabilistic definitions of recall and fallout. Thanks to this probabilistic view, the maximization of the probability of relevance determines the maximization of recall at a given maximum tolerated fallout. In this way, the PRP states that the retrieval effectiveness is optimized when recall is maximum for each fixed cost of fallout. In practice, the principle tells to determine the Bs such that the fallout is less than a given threshold and to choose the B that maximizes the recall among the previously determined Bs.

1.4.1.3 Model Estimation

Corresponding to the events of the probability space, two main random variables can be defined:

$$X_j(\omega) = 1 \qquad t_j \text{ occurs in } \omega$$
$$X_A(\omega) = 1 \qquad \omega \text{ is relevant}$$

For example, suppose the following four documents ω_1 = "apple banana", ω_2 = "apple banana cherry", ω_3 = "banana apple", and ω_4 = "banana" are indexed and three terms are extracted from the documents. We have that

$$X_{\text{apple}}(\omega_1) = 1 \qquad X_{\text{banana}}(\omega_1) = 1 \qquad X_{\text{cherry}}(\omega_1) = 0$$
$$X_{\text{apple}}(\omega_2) = 1 \qquad X_{\text{banana}}(\omega_2) = 1 \qquad X_{\text{cherry}}(\omega_2) = 1$$
$$X_{\text{apple}}(\omega_3) = 1 \qquad X_{\text{banana}}(\omega_3) = 1 \qquad X_{\text{cherry}}(\omega_3) = 0$$
$$X_{\text{apple}}(\omega_4) = 0 \qquad X_{\text{banana}}(\omega_4) = 1 \qquad X_{\text{cherry}}(\omega_4) = 0$$

In general, when k terms are extracted, the multidimensional random variable

$$X = (X_1, \ldots, X_k)$$

can be defined, thus yielding up to 2^k possible outcomes. Let B be a subset of the set of elementary events mapped to a given outcome of X; for example, when $X = (0, 1, 0)$, we have that $B = \{\omega_2, \omega_4\}$. These subsets can be restricted to the subset A of relevant documents, thus obtaining the outcomes of X conditioned to relevance. When the elementary events ω are assigned a probability measure $\mathrm{P}(\omega)$, the following probabilities can be computed:

$$\mathrm{P}(B \mid A) \qquad \mathrm{P}(B \mid \bar{A})$$

The actual use of the relevance model requires the estimation of $P(B|A)$ and $P(B|\bar{A})$. The documents in B are described by k properties which are in turn described by a random variable X. The simplest approach is binary:

$$X_j(\omega) = 1 \quad \text{term } j \text{ occurs in } \omega \quad \omega \in B$$

Suppose B can be mapped to X. It follows that

$$P(B|A) = P(X = x|X_A = 1) \qquad P(B|\bar{A}) = P(X = x|X_A = 0)$$

When $X = x$ is $X_1 = x_1, \ldots, X_k = x_k$,

$$P(X = x|X_A = 1) = P(X_1 = x_1, \ldots, X_k = x_k|X_A = 1)$$

Since there are 2^k possible outcomes, the number of estimations is exponential, and its estimation is in practice infeasible although k might be small; this problem is known as the curse of dimensionality and may be addressed by assuming conditional stochastic independence between the X_js defined as follows[2]:

$$P(X_1 = x_1, \ldots, X_k = x_k|X_A = 1) = \prod_{j=1}^{k} P(X_j = x_j|X_A = 1)$$

and

$$P(X_1 = x_1, \ldots, X_k = x_k|X_A = 0) = \prod_{j=1}^{k} P(X_j = x_j|X_A = 0)$$

Suppose

$$p_j = P(X_j = 1 \mid X_A = 1) \qquad q_j = P(X_j = 1 \mid X_A = 0)$$

It follows that

$$P(X = x \mid X_A = 1) = \prod_{j=1}^{k} p_j^{x_j}(1 - p_j)^{1-x_j}$$

$$P(X = x \mid X_A = 0) = \prod_{j=1}^{k} q_j^{x_j}(1 - q_j)^{1-x_j}$$

[2]Cooper (1995) showed that this assumption can be weakened.

The application to IR gives the likelihood ratio of the Binary Independence Retrieval (BIR) model where the likelihood ratio is

$$L(x) = \frac{\mathrm{P}(X = x \mid X_A = 1)}{\mathrm{P}(X = x \mid X_A = 0)} = \frac{\prod_{j=1}^{k} p_j^{x_j}(1 - p_j)^{1-x_j}}{\prod_{j=1}^{k} q_j^{x_j}(1 - q_j)^{1-x_j}}$$

and the log-likelihood ratio is

$$\ell(x) = \log L(x)$$

that is,

$$\ell(x) = \sum_{j=1}^{k} x_j w_j + \sum_{j=1}^{k} \log \frac{1 - p_j}{1 - q_j}$$

where $w_j = \log \frac{p_j(1-q_j)}{q_j(1-p_j)}$ is called Term Relevance Weight (TRW).

In the relevance model, the query is directly not modeled, whereas relevance is modeled since it is represented as a subset of documents. However, the BIR model requires query modeling due to efficiency reasons since the calculation of the log-likelihood would require k additions. To reduce the computational cost, which might be large when k is large, it is supposed that a query is given as input so that the summation is limited to the TRWs of the query terms.

The estimation of the TRWs is based on the maximum livelihood estimators (MLEs) of the p_js and q_js. Provided a training subset of documents, the following Maximum Likelihood Estimators (MLEs) are used:

$$\hat{p}_j = \frac{r_j + \frac{1}{2}}{R + 1} \qquad \hat{q}_j = \frac{n_j - r_j + \frac{1}{2}}{N - R + 1}$$

where R is the number of relevant documents in the training set, $r_j \leq R$ is the number of relevant documents indexed by term j, N is the number of documents, and n_j is the number of documents indexed by term j; the constants are commonly utilized to smooth the estimators.

1.4.1.4 Best Match N. 25

Robertson and Walker (1994) proposed a variation of the TRW which became one of the most effective weighting schemes. Best Match N. 25 (BM25) basically multiplies the TRW by a saturation component, thus obtaining the following weight:

$$w_{ij} = \mathrm{TRW}_j \mathrm{SATURATION}_{ij}$$

where the first component is the TRW also known as Robertson and Sparck Jones (1976)'s weighting scheme defined as

$$\text{TRW}_j = \log \frac{r_j + 0.5}{R - r_j + 0.5} - \log \frac{n_j - r_j + 0.5}{N - n_j - R + r_j + 0.5}$$

The TRW is multiplied by a saturation component

$$\text{SATURATION}_{ij} = \frac{(k_1 + 1) f_{ij} (k_3 + 1) g_j}{(k + f_{ij})(k_3 + g_j)}$$

of term j in document i. For each document, the saturation component is a monotonically increasing function of the frequency, f_{ij}, of j in i. The shape of this function is tuned by a number of parameters and variables; $k = k_1((1 - b) + b\frac{l_i}{l})$, l is the average document length, l_i is the length of document i, b is a parameter (usually 0.75), k_1 and k_3 are parameters (usually, 1.2 and something between 7 and 1000, respectively), and g_j is the frequency of term j in the query.

1.4.1.5 Relevance Feedback

RF in the relevance model consists of modifying the TRWs. The iterative process of RF begins with the situation in which no relevance data are available, that is, $R = 0$. It follows that at the beginning, ranking is computed by the following function:

$$g^{(0)}(z) = \sum_{j=1}^{k} z_j w_j^{(0)}$$

where

$$w_j^{(0)} = \log \frac{N - n_j + \frac{1}{2}}{n_j + \frac{1}{2}}$$

At step $t = 1, 2, \ldots$ of RF, the following function is used instead:

$$g^{(t)}(z) = \sum_{j=1}^{k} z_j w_j^{(t)}$$

where

$$w_j^{(t)} = \log \hat{p}_j^{(t)} + \log 1 - \hat{q}_j^{(t)} - \log \hat{q}_j^{(t)} - \log 1 - \hat{p}_j^{(t)}$$

$$\hat{p}_j^{(t)} = \frac{r_j^{(t)} + a^{(t)}}{R^{(t)} + b^{(t)}} \qquad \hat{q}_j^{(t)} = \frac{n_j - r_j^{(t)} + c^{(t)}}{N - R^{(t)} + d^{(t)}}$$

It is usually assumed that

$$a^{(t)} = c^{(t)} = \frac{1}{2} \qquad b^{(t)} = d^{(t)} = 1$$

but the theory of Bayesian statistics may provide additional hints.

1.4.2 Language Models

This model is named after the fact that it explicitly represents language, while relevance is not explicitly represented; in the literature, both terms "language model (LM)" and "language models (LMs)" are utilized depending on whether one is referring to the class of probabilistic retrieval models sharing the properties described in this section or to the specific probabilistic space describing how language can statistically be described. In the following, the term is meant to indicate the class of probabilistic retrieval models.

According to a LM, there is an author of a document thinking about the queries a possible end user would formulate to retrieve the document; for example, an author may write the sentences of the document in a way that they contain the answers to the users' questions. In doing that, the author writes the document using queries and variations of them, although he is assumed to have a good idea of the user's need. On the other end, there is a user assumed to have a good idea of what he is searching for.

So that documents and queries can be matched, the author and the user are assumed to use an effective language and the same language. Most importantly, it is also assumed that the documents generated by the authors are relevant to the user's information need. The LM is indeed known as a generative model since it describes how language is generated; in particular, the Language Model (LM) describes how documents and queries are generated and how a query can be viewed as the outcome of the generation fueled by a document. Figure 1.21 gives a pictorial description of

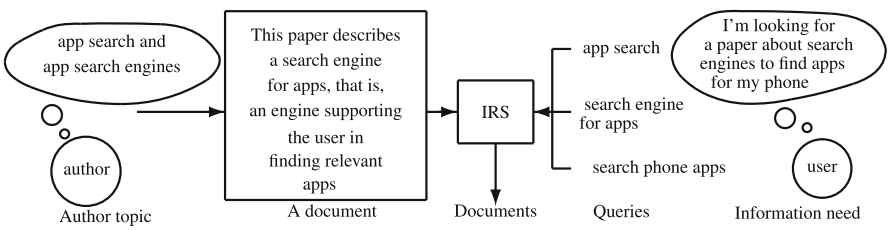

Fig. 1.21 Metaphor of the Language Model

how the LM is used in IR. On the left-hand side of the figure, an author is thinking about "app search and app search engines" and is writing a document pertinent to this topic. On the right-hand side of the figure, a user is thinking about his own information need and formulates a series of queries describing the aspects of the information need. At the center of the figure, an IR based on the LM matches the author's document and the user's queries and decides whether the document and the queries derive from the same language. The basic assumption of the LM is that a document contains information relevant to the user's information need when it is of the same language as the user's queries.

1.4.2.1 Documents and Queries as Languages

Algebraically, the LM consists of symbols, strings of symbols (i.e., n-grams), and a probability function defined on these strings. Let s be a symbol. A language is defined as a set of symbols:

$$\{s_1, \ldots, s_N\}$$

Given a language, an n-gram is a sequence of n symbols drawn from the language expressed as

$$s_{(1)} \ldots s_{(n)} \qquad n > 0$$

When $n = 1$, the n-gram is called unigram; if $n = 2$, it is called bigram; if $n = 3$, it is called trigram. A language can be viewed as an urn like the one in Fig. 1.22 from which symbols are drawn to form n-grams. Whenever a document or a query is formed, a series of symbols are drawn from the urn, and the outcome of this process is an abstract representation of the document or the query. Probabilistically, an n-gram is an experimental outcome; for example, suppose $L = \{s_1, s_2, s_3\}$ and $n = 2$. Sampling two symbols with replacement yields the following sampling space (i.e., the space of all the possible outcomes):

$$s_1s_1, s_1s_2, s_1s_3, s_2s_1, s_2s_2, s_2s_3, s_3s_1, s_3s_2, s_3s_3$$

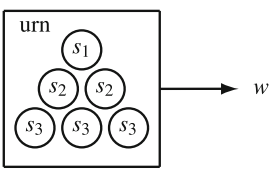

Fig. 1.22 Language as an urn

"With replacement" means that using the same symbol in an n-gram is possible; indeed, natural languages work with replacement. A sampling space contains N^n n-grams when order matters and contains $\binom{N}{n}$ n-grams when order does not matter; the space is therefore very large, thus explaining the high variety of a natural language in which the words are symbols, whereas documents and queries are n-grams.

The LM is a language provided with a probability function; for example, the language in Fig. 1.22 is an instance of the LM where $\mathrm{P}(s_j) = j/6$. In IR, a language model is often built from a document or a group of documents. Consider the following document:

upon the bench the goat lives, under the bench the goat dies

After removing stop words and stemming words, the resulting language is

$$L = \{\text{bench, goat, live, die}\}$$

Suppose $n = 2$; the LM is given by a language and by the probability function

$$\mathrm{P}(s_1, s_2) = \mathrm{P}(s_1|s_2)\mathrm{P}(s_2)$$

where

$$\{s_1, s_2\} \in \{\{\text{bench, goat}\}, \{\text{goat, live}\}, \{\text{live,bench}\}, \{\text{goat,die}\}\}$$

1.4.2.2 Query Language Model

Among the various instances of LM, the Query Language Model (QLM) is the one mostly used in IR. According to the QLM, documents are samples of a language and queries are LMs. The IR system designed according to the QLM looks for the most likely document given a query:

$$B^* = \arg_B \max \mathrm{P}(B \mid Q)$$

where Q is a query event and B is a document event. Documents are ranked by $\mathrm{P}(B \mid Q)$. However, Q is not completely known: the language is known but the probability is unknown. Therefore, the Bayes theorem is applied to obtain

$$\mathrm{P}(B \mid Q) = \frac{\mathrm{P}(Q \mid B)\mathrm{P}(B)}{\mathrm{P}(Q)}$$

thus swapping the roles played by query and document. Given a query, $\mathrm{P}(Q)$ is constant and therefore, the ranking of the documents is not affected. $\mathrm{P}(B)$ is assumed to be either uniform and then not affecting the document ranking or estimated

by external sources such as PageRank or other query-independent measures of document qualities. With regard to probability estimation, as Q is regarded as an n-gram, we have

$$\mathrm{P}(Q \mid B) = p_B(s_{(1)} \dots s_{(n)}) = p_B(s_{(1)}) p_B(s_{(2)}|s_{(1)}) \cdots p_B(s_{(n)} \mid s_{(n-1)} \cdots s_{(1)})$$

1.4.2.3 Mixture and Smoothing

Due to the curse of dimensionality encountered when the relevance model was described, stochastic independence has to be assumed, thus obtaining

$$\mathrm{P}(Q|B) = p_B(s_{(1)}) \cdots p_B(s_{(n)})$$

where

$$p_B(s_{(j)}) = \frac{f(s_{(j)}, B)}{\sum_{j=1}^{n} f(s_{(j)}, B)}$$

and $f(s, B)$ is the frequency of s in B. The problem that f might be 0 is solved either by a mixture as follows:

$$\hat{p}_B(s_{(j)}) = (1 - \lambda) \frac{f(s_{(j)}, B)}{\sum_{j=1}^{n} f(s_{(j)}, B)} + \lambda \frac{f(s_{(j)}, L)}{\sum_{j=1}^{n} f(s_{(j)}, L)} \tag{1.3}$$

or by smoothing as follows:

$$\hat{p}_B(s_{(j)}) = \frac{f(s_{(j)}, B) + a}{\sum_{w \in B} f(s_{(j)}, B) + a + b} \tag{1.4}$$

Mixture and smoothing are depicted in Fig. 1.23; mixture (Fig. 1.23a) consists of repeatedly sampling from an urn chosen with a given probability, whereas smoothing (Fig. 1.23b) consists of virtually modifying the urn before sampling. With mixture, first an urn is drawn with a given probability of the urn, and then the symbols are drawn from the selected urn. After collecting a sufficiently large set of outcomes together, a histogram of the frequencies of the symbols can be drawn. With smoothing, the process is like injecting additional symbols into each urn to avoid that an urn does not contain a symbol. After injecting these additional symbols, it is possible to proceed as with a mixture.

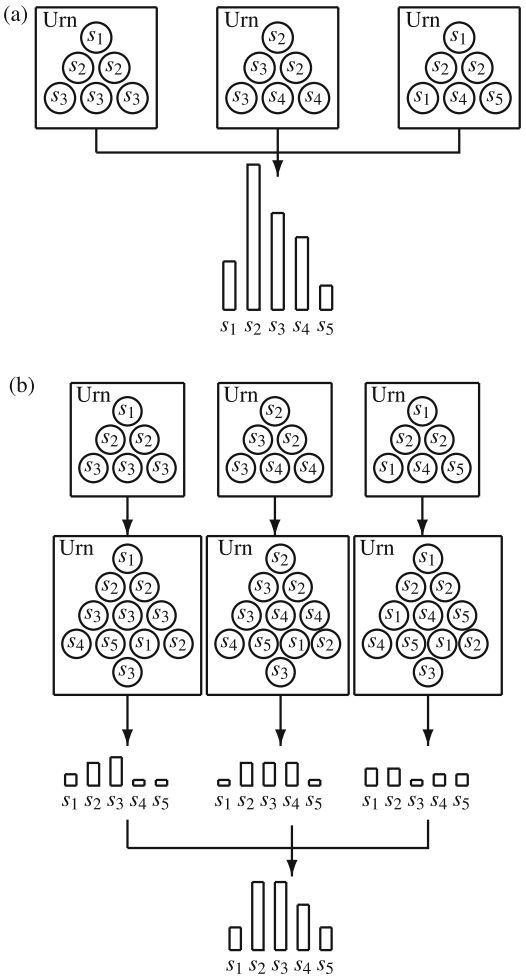

Fig. 1.23 (**a**) Mixture and (**b**) smoothing

1.5 Machine Learning

The use of Machine Learning (ML) models in IR may be explained by the difficulty in designing a single retrieval function that encompasses all the sources of evidence of a modern system (e.g., a search engine of the WWW or an "intelligent" system called to solve complex tasks). The difficulty in combining these sources is caused by the interaction between the user and system and the variety of context. ML provides an alternative approach to the problem of designing a system (i.e., a "machine") that can retrieve and rank documents in the best possible way for each

http://www.springer.com/978-3-662-48312-1

Introduction to Information Retrieval and Quantum Mechanics

Melucci, M.

2015, XVIII, 232 p., Hardcover
ISBN: 978-3-662-48312-1