Invited paper presented to the MF/IR workshop, SIGIR 2003, Toronto, Canada

The unified model revisited

Stephen Robertson

Microsoft Research
7 JJ Thomson Avenue
Cambridge CB3 0FB, UK
(and City University, London)
ser@microsoft.com

Abstract

This paper reconsiders the unified probabilistic model of information retrieval, proposed by the author with M.E. Maron and W.S. Cooper in 1982, as a reconciliation of the Maron & Kuhns and Robertson & Sparck Jones models. Some basic concepts of the unified model, such as documents, user needs, and terms as properties of these, are discussed and reformulated in the light of later work. The issue of the event space underlying the model is also re-assessed. An event space consisting of a Cartesian product of four random variables is proposed: two observed, the texts of the document and query, and two hidden, the models assumed to underly the texts. Relevance is seen as a derived random variable within this space. The product space should not, however, be flattened: its structure is important and must be retained.

In this paper I will revisit some work done over 20 years ago, with M.E. (Bill) Maron and W.S. (Bill) Cooper (Robertson, Maron and Cooper, 1982; Robertson, Maron and Cooper, 1983). I will also consider some more recent developments. The two central, related issues are: (1) how we construe the basic objects of the IR space, their properties and relationships; and (2) how we interpret the notion of event space in this context, for the purpose of statistical models and experiments.

1 What was the problem?

When I visited Berkeley in the spring of 1981, I discovered that Bill Maron and I had independently been thinking about (and getting nowhere with) the same problem. This was the possible relationship between Bill's work (with J.L. Kuhns) from some 20 years previously, on a probabilistic model of indexing (Maron and Kuhns, 1960), and my more recent work (with Karen Sparck Jones), on a probabilistic model of searching (Robertson and Sparck Jones, 1976). Both models appeared to address the same question, how to assess the probability that a particular document will be judged as relevant to a particular user request by that user. The problem was that the two models seemed to address this question in different and apparently incompatible ways.

We eventually reached a formulation of the problem, as follows. In both models we are concerned with the *properties* of documents or queries (for example the terms they contain). However, the two models have very different views of where these properties belong and how they might be used. In Maron and Kuhns' model (referred to as Model 1), the human indexer is supposed to have a specific document in front of him/her, and to be imagining the kinds of users who might find this document useful. He or she has no knowledge of individual users or their queries, but knows something about the kinds of users the retrieval service might expect to have as clients, and how they ask questions. Thus the terms are assumed to characterise the users and their requests – this association is regarded as a fixed point, which the indexer cannot affect or change. Then the indexer's function is to index (represent) a document in ways that will match well those queries put by users who will find this document relevant.

By contrast, the Robertson/Sparck Jones model (Model 2) starts from the searcher end. Documents are assumed to be already indexed, and the searcher-end task is to represent queries in ways that will match well those documents that the user will find relevant. Here there is no direct view of individual documents – only of documents characterised by their predefined properties.

The difference becomes more evident when we consider the possibility of relevance feedback. In Model 1 we might use known relevance judgements to modify the indexing of specific documents, so as to improve their representations for future users. In Model 2, we may use relevance judgements to modify the query, for future searches for the same user information need. Neither model is capable of dealing with the other form of feedback.

Each model relies on an assumed fixed point to optimise something else which it takes as variable. The obvious next step to consider is to assume no fixed points, but to treat both as variable and to optimise both. But at least in the naïve version of this idea, losing both fixed points means that everything is lost – ”properties” which do not characterise anything cannot be said to be properties, and must be regarded as undefined.

2 Objects and properties

2.1 An example

I would like to demonstrate some of the above, and some aspects of possible solutions, with an example. This is not exactly IR, at least not document retrieval, but in this example the symmetry of the situation is clearer than in the document/user need case.

We consider the process of initiating employment – employers seeking people to fill their vacancies and job seekers trying to find suitable vacancies. In terms of systems or services, we can approach from either end: there exist in the world both databases of vacancies (details provided by employers, searched by job seekers) and databases of job seekers (details provided by themselves, searched by employers). In the first case, we may assume, vacancies are described in terms of their own characteristics or properties, and the aspiring job seeker has to formulate his/her query in those terms. In the second, presumably, the job seeker provides a CV, and the employer has to formulate a query in *those* terms. We may represent the situation graphically as in Figure 1.

We can easily imagine that the set of possible properties of vacancies is quite distinct from the set of possible properties of job seekers – they are naturally described in different ways. Just for example: a job seeker would have an attribute ”age” which a vacancy would not have (or if it did, it would have quite a different meaning, and no match would be expected). Similarly, a vacancy relates to a particular locational in the organisational structure of the employing organisation – the job seeker would not normally have such a characteristic.

2.2 The original unified model approach

We can see the same situation, perhaps less obviously, in the context of documents and queries (it becomes more obvious if we consider not just topic-descriptive words but metadata as well). One insight which was needed for unification was that we should treat these two separately. Documents have properties (including the words in them); user needs have properties (including the words in which they are expressed); and retrieval involves some matching between these two respective sets of properties. This I believe to be a powerful insight, quite outside the probabilistic modelling problem: it allows for a number of things that do not quite fit into a simple view where both documents and queries are assumed to be described with the same set of terms. For example:

1. One side may have properties which simply do not occur on the other side (e.g. a document may have citations);

2. Matching of terms-as-document-properties against terms-as-user-need-properties may be asymmetric (e.g. a general term used by a user to describe his/her need may be matched with a specific document term, but not vice versa);

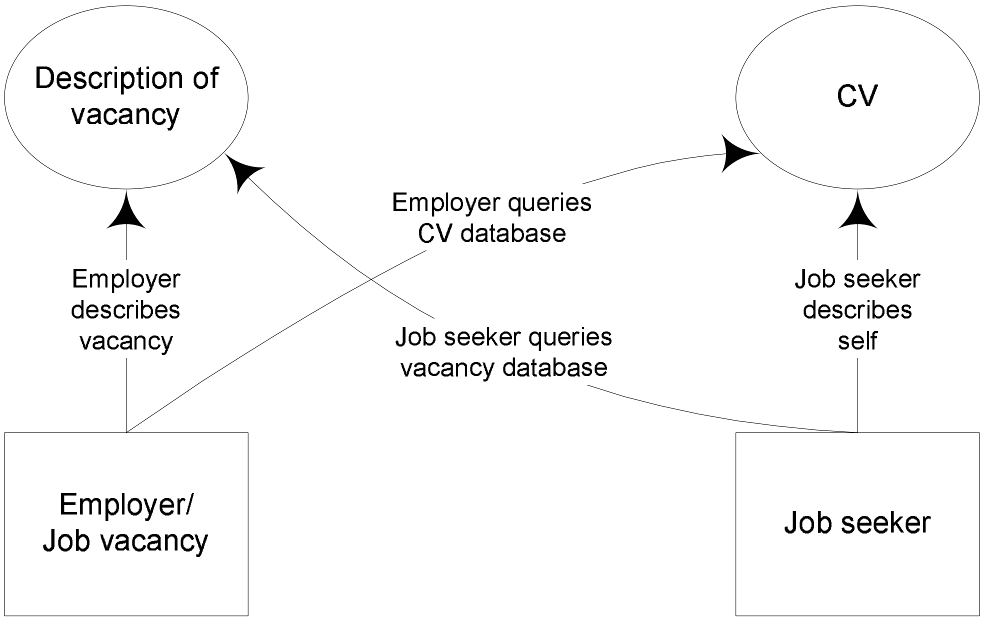

Figure 1: Employment – vacancies and job seekers

3. Documents may be indexed using some kind of category codes (for example Dewey Decimal classification codes), while queries might be expressed in natural language.

This view may be pursued somewhat further in a particular way, as follows (this was explored in (Robertson et al., 1983)). An initial information retrieval process, in the absence of relevance information, involves describing both documents and queries in terms of their own natural properties, and a using a standard matching between these two sets of properties (as in Figure 2). However, once relevance information is available on either side, a document about which we have relevance information may be additionally described in terms of the properties of queries to which it is relevant, and vice versa. Then we have to reconcile all this information in the matching process. We saw this in terms of Figure 3. More particularly, we saw Models 1 and 2 as each bypassing one set of properties and describing one entity directly in terms of the other set of properties.

2.3 A modified view

Even if we accept and keep the original insight that document and user need properties should in principle be regarded as different things, the step to Figure 3 is arguable. It might be taken to represent a view of expansion based on relevance feedback (query expansion in one case, document indexing expansion in the other case), where we maintain the distinction between the original properties and those added by expansion. However, it does not seem to accommodate another form of feedback, which is simply reweighting of existing terms. Nor is it necessarily the best way to think of expansion.

We might instead think of always maintaining a matching process like Figure 2, but modifying both descriptions in the light of feedback. Thus identifying a good expansion term for this query (from relevant documents) does not mean building a completely different kind of user need description from the original user-chosen terms, but rather adding to the original. This can be seen as valid even if the new ”term” did not

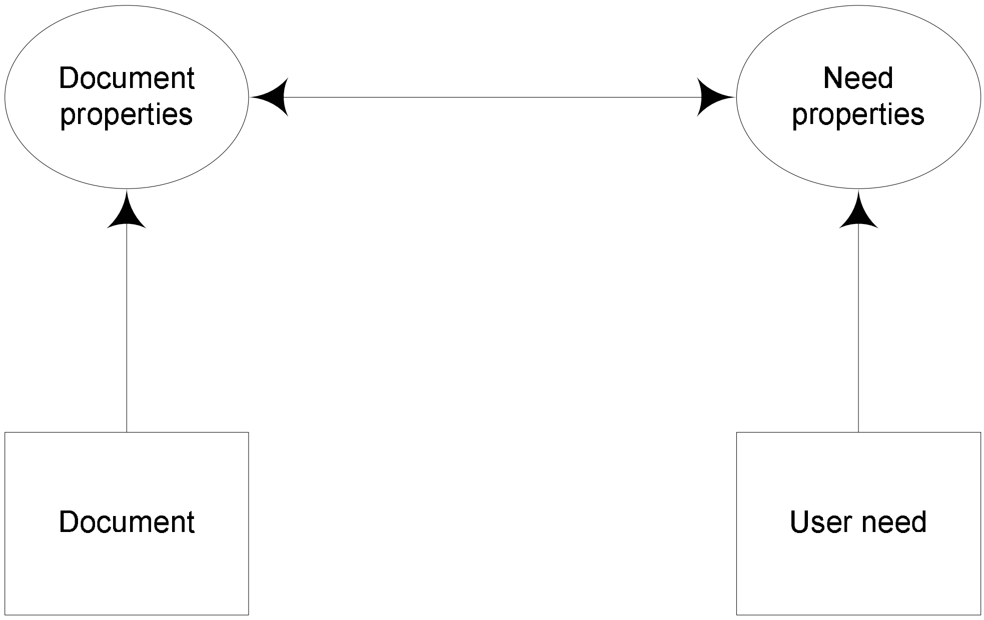

Figure 2: Before relevance information: Model 0

even exist in the original description language of the user. For example, we may discover that a category code is a good expansion term for a query, even if the user was unaware of category codes when they formulated the query. One way to think of this process is as in Figure 3, but we can also think of it in terms of Figure 2 combined with description modification.

Thus we might represent the situation, including feedback, as in Figure 4. Recent work by Bodoff (Bodoff and Robertson, 2003) is formulated in these terms.

I believe this again to be a useful insight, and indeed to be a better way of describing the kinds of relevance feedback algorithms that we have been using (at least for query modification) for some years. It does however introduce some problems which were not apparent in the Figure 3 view. One issue concerns the role played by the absence of a property in a description. (This is in any case quite a complex issue.) If we were to assume that the vocabulary from which the user chooses terms to describe their need is identical with that from which the author chose terms for the document, then we might make certain assumptions about the absence of a term from either description (in one view, absence of a term is a property with the same status as presence of a term). However, if there are elements of the document description of which the user has no knowledge, and which therefore simply do not occur in the user's vocabulary, their absence from the query may need to be interpreted differently.

3 Event spaces

The problem of understanding and choosing appropriate event spaces in the IR context was explored in a more recent paper (Robertson, 2002), as well as in the first Unified Model paper (Robertson et al., 1982). Here I will attempt to pull together and further develop some of this discussion.

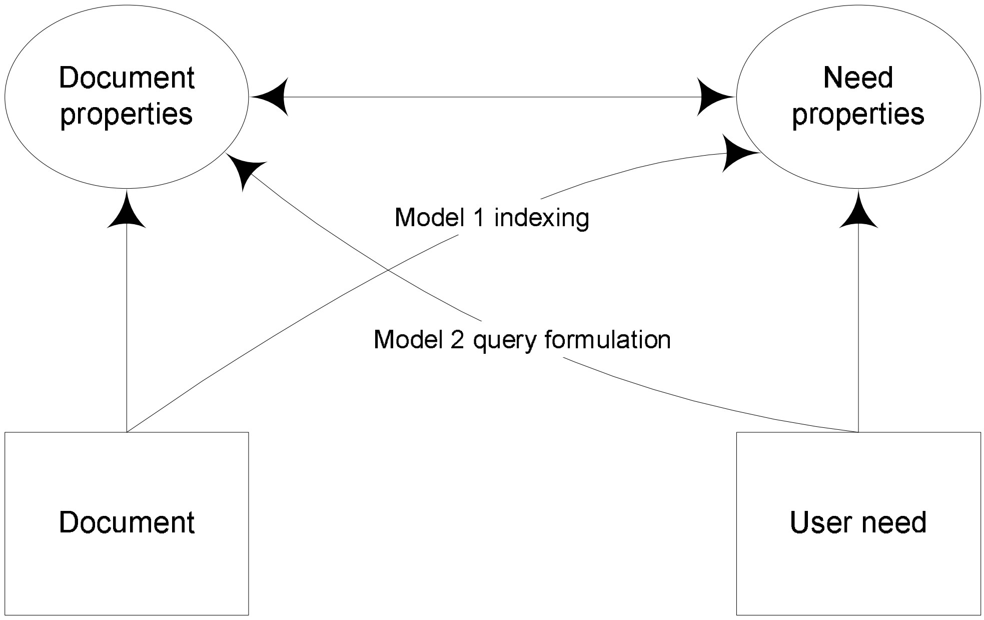

Figure 3: After relevance information: Model 3 = Model 0 + Model 1 + Model 2

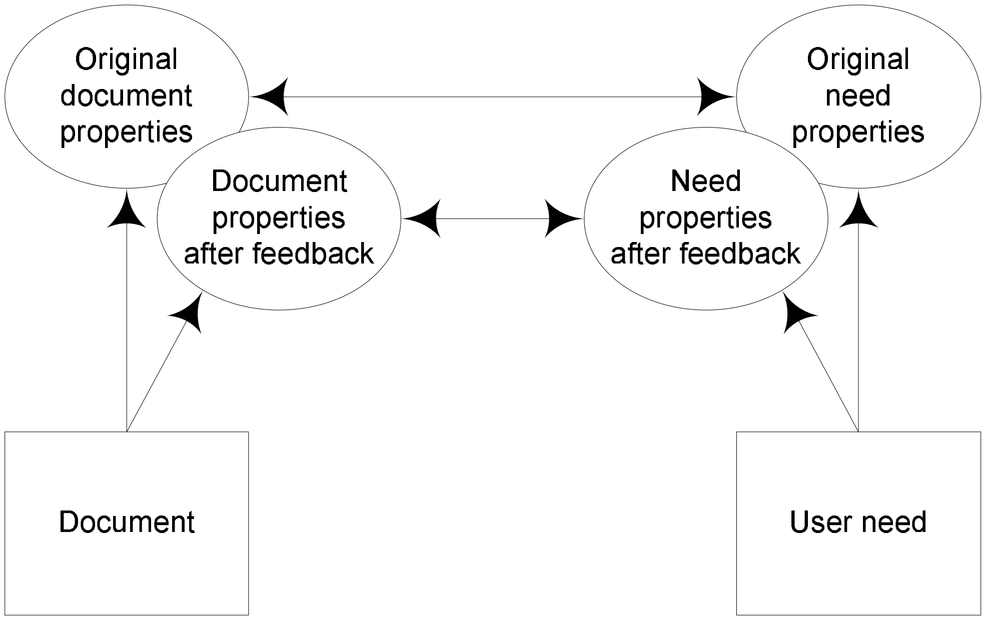

Figure 4: After relevance information: New unified model

3.1 Event spaces and random variables

When we define an event space, we start with a random event, that is an event whose outcome is not determinate. The event space is the set of possible outcomes, and we look for a probability measure on this space. A random variable is a *deterministic* function of the outcome – it acquires its 'random' status by virtue of the initial event being random.

Even before we worry about the probability measure, we see some difficulties of interpretation. Suppose for example that we consider the quintessential random event, a coin toss. If we are concerned only with the 'heads or tails' aspect of coin tossing, then we can define the outcome of the random event as H or T. The immediate random variable, call it R_1, is simply the identity function – if H, H, or if T, T. (In principle we can of course construct or derive further random variables from this one, by means of further deterministic transformations of these values, although in this case there is not very much scope.)

However, if we are interested in any other aspects of the coin toss, we have to push back the definition of 'outcome' to an earlier stage in the process. For example, we may toss the coin and let it fall on the ground, and observe not just which face is on top but also the north-south orientation of the face. Then the outcome of the random event might be taken to be a coin in a certain position on the ground, and we have two random variables, namely $R_1 =$'read the face as H or T' and $R_2 =$'read the orientation of the face as the compas bearing of the top of the face, in degrees'. Provided that the face has an identifiable top, R_2 is a perfectly good random variable. So is R_1 – in fact it is identical to the R_1 defined previously, except that we have moved part of its definition from the random event to the random variable.

Alternatively, in this case, we could regard the outcome as sets of pairs of R_1 and R_2 values. This would be reasonable provided that we are confident in advance that we are not interested in any other aspect of the 'real' outcome, namely the position on the ground, and provided also that we do not need to appeal to any other aspect to explain a relationship between R_1 and R_2.

At this point it is appropriate to discuss the probability measure. The simplest form of measure is one which assigns a probability to each possible outcome, indeed the simplest of all would assign equal probability to every distinct outcome. Then probabilities for sets of outcomes, and therefore for random variable values or sets of values, can be derived simply. This form of probability measure seems obviously applicable to the first coin-tossing example where the outcome must be H or T, but would appear not to be applicable to the second, where the outcome is a position on the ground. However, it is not a necessary feature of a probability measure that it assigns a probability to each individual outcome – the requirement is that it should assign probabilities to some set of sets of outcomes (the measurable sets) satisfying certain conditions. In fact if the random variables of interest are all known, then we may want to define the measurable sets in terms of the random variables. In the second coin-tossing example, this is equivalent to assigning a probability distribution to the pairs of values (R_1, R_2).

However, if we are exploring possible ways of understanding a particular event space, we should avoid fixing on particular random variables and using them in this way, since the exploration may involve considering others. To assign a probability distribution *only* at the level of sets defined by a particular collection of random variables is to restrict our exploration to this particular collection and to random variables derivable from them. Any random variable which requires access to the original event is now outside our scope.

3.2 The uses of random variables

It may be argued that it is indeed the function of random variables to define classes of events. It is at least arguable that a collection of single, individual events, each one different from all the others in unknown ways, is not a good subject for statistical study. In some sense we look to random variables (which define classes of similar events: a class of events is those which have the same value of this random variable) to give us the sort of structure we require for statistical study. When we cannot see all the variables we feel we need, then we tend to postulate other, hidden variables, whose values might be (statistically speaking) inferred from what we can observe. But in order to tell us anything, we must have fewer random variable values (or values of combinations of random variables) than there are elements in the event space. Otherwise no generalisation is possible.[1]

In fact, even in the second coin-tossing example, our definition of outcome already lumps together many distinct events, even if not enough to make it easy to define a probability distribution. In defining the outcome as the final position of the coin on the ground, it is clear that we allow (in principle) any number of individual tosses to result in the coin ending up in exactly the same position on the ground (the fact that this is unlikely in practice simply reflects the number of coin-tossing events we expect to occur). To treat coin-tosses as truly individual events, though not impossible, is a little hard to imagine. It is particularly hard to get a handle on all the features (random variables) one might possibly identify which may characterise the individual event. However, it is clear that in this context, position on the ground would be just one of many random variables, and H/T and north-south orientation just two more which may be derived from position.

But an alternative to treating the basic events (coin tosses) as completely individual, as indicated above, is to hypothesise one or more hidden variables on these events, which encapsulate all that we need to know about their 'individuality'. In terms of probability measures and distributions, if we treat the outcome of a coin-toss as a pair of values of two random variables (H/T and north-south orientation), then the minimum required for a probabilistic description is a joint probability distribution over these two variables. If we now add a hidden variable, then the minimum is now a joint probability distribution over these three variables. Provided that we can hypothesise some reasonable form for this new probability distribution over three variables, we do not necessarily need to think any further about the space of individual events.

This is a fair argument. However, there are circumstances in which we do indeed need to preserve (or at least worry about) individuality. This is not evident at all from the coin-tossing examples (coin-tossing is of course used for various purposes for exactly this reason); however, the next section will expand on this point and IR will provide an example.

3.3 Structured event spaces

In my paper in the MF/IR workshop in 2002 (Robertson, 2002), I considered the problem of event spaces that have some internal structure. I considered one example of a parent-child relationship and another of a cross-product space – the latter being applicable to documents and queries in IR. In both cases a simple interpretation of the event space (as a set of unrelated events, which is the basic assumption in much statistical work) leads to problems with the notion of individuality.

We may illustrate the problem with a cross-product space. Suppose we have two sets of events $\mathcal{A}$ and $\mathcal{B}$. Each set of events has its own random variables which may partition it in various ways; but suppose also that we are primarily interested in the interactions – any $A \in \mathcal{A}$ can interact with any $B \in \mathcal{B}$, and we may observe further random variables from the interaction.

3.3.1 Hidden variables

We may now choose to consider the event space consisting of all interactions (that is all pairs $(A, B) \in \mathcal{A} \times \mathcal{B}$), and classify the interaction events by means of the complete collection of random variables (those that apply to $\mathcal{A}$, those that apply to $\mathcal{B}$, and those that apply to the interaction specifically). As discussed in the workshop paper, any random variable that applies to e.g. $A \in \mathcal{A}$ can be re-interpreted as applying to pairs $(A, B) \in \mathcal{A} \times \mathcal{B}$ without difficulty, and random variables which naturally reside in the product space can obviously be used. All of this works fine, and we can treat the cross-product as an elementary sample space, *provided that* all the variables are observable and none is hidden. It works still if one or more of the variables that naturally reside in the product space is hidden. However, it does *not* work if any of the variables associated with either original space is hidden.

[1] This is a somewhat simple-minded argument, which does not translate easily to infinite event spaces or continuous random variables, nor does it deal with issues around modelling probability distributions.

Why is this? It has to do with individuality. Let us suppose that the variable X is associated with $\mathcal{A}$, so that $X(A)$ is a deterministic function of $A \in \mathcal{A}$. Then we can define X' on $\mathcal{A} \times \mathcal{B}$ by $X'(A, B) = X(A)$, which is again a deterministic function. We have the structural property that $X'(A, B_1) \equiv X'(A, B_2)$ ($B_1 \neq B_2$), even though (A, B_1) and (A, B_2) are two separate points in the product space (this is exactly the property that distinguishes X as a random variable on $\mathcal{A}$, rather than on $\mathcal{A} \times \mathcal{B}$). If $X = X'$ is an observed variable, the fact that these two values must be the same tells us nothing, since we already know what the values are. If, however, $X = X'$ is a hidden variable, the equality (identity) of these two values is potentially important to us, since it may give us valuable information for estimation purposes. But the identity cannot be expressed in terms of the simple event space of pairs.

Thus if we limit ourselves to the simple event space of pairs, we cannot hypothesise a hidden variable relating to either base set on its own.

To interpret this statement in terms of IR: we have base sets of documents and queries, and we are interested in properties of these (such as their texts) and also a property of pairs (relevance). Texts are normally visible, and relevance may or may not be. With exactly this set of random variables, we can simply treat the set of document-user need pairs as classified by these random variables as the event space, and postulate a probability distribution over this space (i.e. over these random variables). However, as soon as we wish to postulate any hidden variable on either documents or queries alone, this view of the event space breaks down.

3.3.2 Additional random events

One of the consequences of introducing models with hidden variables in them is that we often also introduce hidden random events. Consider for example a language-modelling view of documents. If each document potentially has its own language model (defined by a set of parameters), then the process of generating a document involves two stages. First, a particular model (set of parameters) has to be chosen, and then the text of the document has to be generated by choosing a sequence of words according to the chosen parameters. If we are to take a probabilistic view of the process, each of these stages must be regarded as stochastic.

Now choosing a document (let's say at random) from the set of all documents in the world involves determining the outcomes of both of these stochastic process simultaneously. That is, the author of the chosen document went through both the above stages, and we see the resulting text only. However, in order to make use of the language modelling view, we normally want to do something with the (hidden from us) original model, that is the parameters chosen by the author. Thus we need to disentangle the two stochastic processes. Introducing the language model as a hidden variable (or the set of language model parameters as a set of hidden variables) means that we have to distinguish two separate random events. Each of these two random events has its own event space. Since similar arguments apply to queries, we now have perhaps four random variables instead of two (five if we include relevance). This set of random variables may be represented as in Figure 5, a diagram which occurs in many minor variants in a number of published papers by various authors (e.g. (Fuhr and Buckley, 1991; Crestani, Lalmas, van Rijsbergen and Campbell, 1998; Lafferty and Zhai, 2001; Bodoff and Robertson, 2003)).

This division introduces further complications. One question is, should I consider the sample of documents that I choose to be drawn simply from all existing documents in the world, or from all conceivable documents. Before I introduced the hidden variable, this was not a major problem: the sample of documents (= document texts) that I have was (I could assume) drawn from the collection of all existing documents (= texts), which was in turn drawn from the set of all conceivable documents (= texts). But with the hidden variable which is the language model, I now have the space of all conceivable models to worry about, and also the space of all conceivable texts that could be generated from a single model, as well as the space of all conceivable texts over all models.

Probably an appropriate model for the second stage stochastic process (generating the text from the

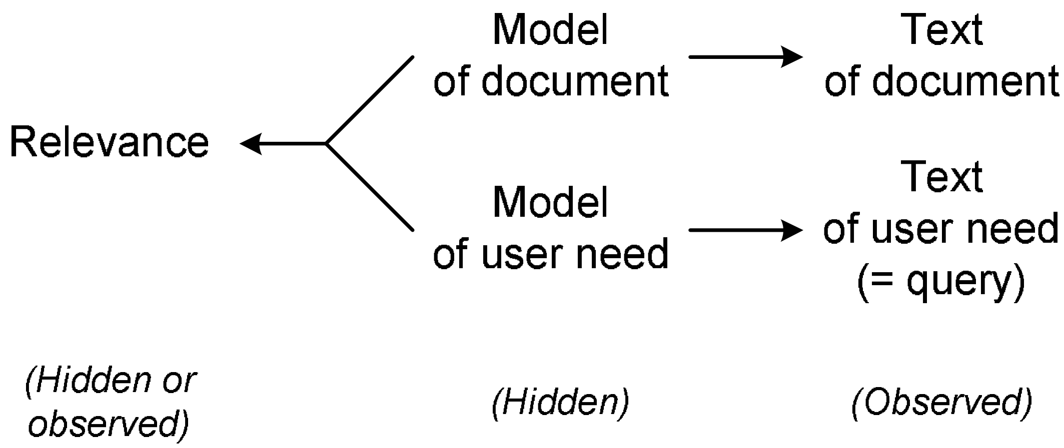

Figure 5: Random variables for document retrieval

model) would best be seen as occupying the "all conceivable" space. In other words, if T is the text and M is the model, the probability $P(T|M)$ is best modelled/estimated in the Cartesian product space of all conceivable Ts and Ms, rather than in the space of all actual documents. One reason is that in the actual documents, we are unlikely to get the same model repeated exactly. In this case, consistency would require that we model the first stage stochastic process in the same way.

4 The IR event space revisited

If we see the document retrieval situation as modelled by Figure 5, what should we take to be the event space?

There appear to be at least four elementary spaces which will play a role in the complete event space. These are: the set of all conceivable document models, the set of all conceivable document texts, the set of all conceivable user need models, and the set of all conceivable query texts. As before, one kind of event we are interested in is the event of matching a particular document with a particular query. This seems to reside in the Cartesian product space of all four of the above elementary spaces. Let us call this $M_D \times T_D \times M_U \times T_U$.

At this point we need to consider various issues. Specifically, we need to think about the fifth random variable of interest, relevance, and issues of individuality.

In the original unified model, we used the product space of individual documents with individual user needs ($D \times U$). In last year's paper, I compared this space defined by individual events with that defined by the texts only, $T_D \times T_U$. My conclusion was that we needed the former rather than the latter in order to regard relevance as a well-defined random variable in the space, because we could not assume that two users issuing the same textual query and being offered the same document would make the same relevance judgement. An alternative was to extend the latter space by including relevance as another dimension of it, $T_D \times T_U \times R$. This is the approach taken by (Lafferty and Zhai, 2003). This does not, however, allow for the hidden variables.

Reverting to $D \times U$ to allow for the hidden variables, in the context of the discussion above, may now be problematic. The problem lies in my choice of an "all conceivable" space instead of an "all existing" one. If we want to allow for all conceivable document models and all conceivable document texts, the notion of individual documents (and similarly of individual queries) seems to disappear. In effect, this dissappearance has been forced by our introduction of the hidden variables, and our associated intention to model the two processes of document (or query) generation separately.

However, relevance may not present the same difficulty in $M_D \times T_D \times M_U \times T_U$ as it did in $T_D \times T_U$. The issue is: could we reasonably assume that two people with the same user need model, when presented with the same document, would make the same relevance judgement? It might indeed be reasonable to make that assumption, on the grounds that if two people make different relevance judgements on the same document, we may take this as evidence that the two user need models were actually different (even if, for example, their initial queries were the same). This is consistent with the view that we choose to have a hidden variable (the user need model) in order to capture all the important aspects of individuality exhibited by the user need. (Whether or not specific assumptions about the form of the user need model will satisfy this criterion is of course another question.)

Thus the hidden variables of Figure 5 may allow us to get away with classifying (aggregating) events by means of random variable values, rather than always considering individual events. The event space $M_D \times T_D \times M_U \times T_U$, in which relevance is assumed to be a well-defined random variable, may be adequate for a unified model.

4.1 Event space structure again

It must be stressed, however, that the problem identified in section 3.3.1 still applies: different points in the four-way product space may share some properties. As indicated in that section, this is important because some of the variables are hidden. For example, if we know that two points come from the same user need, then they must share the same user need model. Since the user need model is hidden, this is an important piece of information. Any attempt to estimate this user need model should take this fact into account. Although we have removed the individual user need from the event space, we must preserve the mapping that determines that two points in the event space which derive from the same individual user need necessarily have the same user need model.

Thus although it appears that we should be operating in the four-way Cartesian product space, we cannot simply take the event space as this product space flattened out. Its structure remains important. In fact different parts of a probabilistic model may address different spaces. For example, if we are attempting to model the second-stage stochastic process identified above, namely the generation of text from model, in the context of documents, then we need to consider the two-way product space of document texts and document models, not the full four-way product space. Each part of a full probabilistic model for IR occupies its own event space.

5 Conclusion: an event space model for IR

The following describes an event space model that goes with the view of information retrieval represented by Figure 5.

1. Each document and query is seen as having an underlying, hidden model (referred to as the document model and the user need model respectively).

2. Document or query generation involves two stochastic processes: the choice of model and the generation of text from the model.

3. Relevance of a document to a user need is a deterministic function of the two models.

4. The full event space is a Cartesian product of four elementary spaces. These are, respectively, the space of all conceivable document models, all conceivable document texts, all conceivable user need models, all conceivable query texts.

5. This event space has important structure, and cannot be flattened out into a set of undifferentiated points. Different parts of a probabilistic model of IR fit into different subspaces of the product space.

Item 3 is a basic assumption of the full event space model: that the hidden models contain enough information about the individual document and user need to determine relevance. Item 5 is somewhat at odds with the traditional probabilistic notion of an event space.

The strength of this event space model is that it supports, in principle, any form of relevance feedback. That is, we observe document and query texts, and we also (sometimes) observe relevance. In principle, we may infer information about either or both hidden models from this combination of evidence, and therefore derive better descriptions of the models, which may be used in future retrieval activities, possibly involving the same user needs and different documents or *vice versa*. That was indeed one of the aims of the original unified model.

References

Bodoff, D. and Robertson, S. (2003), A new unified probabilistic model. Submitted to JASIST.

Crestani, F., Lalmas, M., van Rijsbergen, C. J. and Campbell, I. (1998), 'Is this document relevant?... probably: a survey of probabilistic models in information retrieval', *ACM Computing Surveys* **30**, 528–552.

Fuhr, N. and Buckley, C. (1991), 'A probabilistic learning approach for document indexing', *ACM Transactions on Information Systems* **9**, 223–248.

Lafferty, J. and Zhai, C. (2001), Document language models, query models and risk minimization for information retrieval, *in* W. B. Croft, D. J. Harper, D. H. Kraft and J. Zobel, eds, 'SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM Press, pp. 111–119.

Lafferty, J. and Zhai, C. (2003), Probabilistic relevance models based on document and query generation, *in* W. B. Croft, ed., 'Language Modelling and IR', Kluwer. To appear.

Maron, M. E. and Kuhns, J. L. (1960), 'On relevance, probabilistic indexing and information retrieval', *Journal of the ACM* **7**, 216–244.

Robertson, S. (2002), On Bayesian models and event spaces in information retrieval. Presented at SIGIR 2002 Workshop on Mathematical/Formal Methods in Information Retrieval; submitted to *Information Retrieval*.

Robertson, S. E., Maron, M. E. and Cooper, W. S. (1982), 'Probability of relevance: a unification of two competing models for information retrieval', *Information Technology - Research and Development* **1**, 1–21.

Robertson, S. E., Maron, M. E. and Cooper, W. S. (1983), The unified probabilistic model for IR, *in* G. Salton and H.-J. Schneider, eds, 'Research and development in information retrieval', Springer-Verlag, Berlin, pp. 108–117.

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