Ambiguous requests: implications for retrieval tests, systems and theories

Karen Spärck-Jones[1], Stephen E. Robertson[2], Mark Sanderson[3]

[1]Computer Laboratory, University of Cambridge, UK
[2]Microsoft Research Cambridge and City University, UK
[3]Department of Information Studies, University of Sheffield, UK

ser@microsoft.com, m.sanderson@shef.ac.uk

In early 2006, as a result of a series of conversations between Steve Robertson, Mark Sanderson and Karen Spärck-Jones, Karen circulated a note summing up our discussions, which were on the topic of ambiguous requests. At the core of our discussion was the question: is too much information retrieval research focussed on search tasks where the query unambiguously defines the user's need? Karen took great interest in this topic and examined it from many angles. There was input from the two of us, but as can be seen from the writing style, the text is principly and delightfully Karen's.
Steve Robertson, Mark Sanderson

Abstract

Retrieval system experimentation has assumed that user requests represent a single information need. The problem is identifying and meeting this need. Search engine experience demonstrates that this assumption is far from holding in the real world. Responding appropriately to this fact raises new issues for research on retrieval system theory, design, and evaluation.

1 The problem

Retrieval requests can be, and short requests normally are, ambiguous. The ambiguity may be of word *sense*, or of reference *aspect*. The request "house" may mean 'building', 'home', or 'firm', and the request "house prices" may refer to actual prices or economic factors. There is also the issue of request *type* e.g. topic vs home-page seeking (Broder 2002), etc; this paper addresses ambiguity for topic requests only: this is quite enough of a problem to start with.

The presumption hitherto in retrieval testing, system design, and theory formulation has been that one is dealing with a single underlying information *need*, which may be more or less adequately (fully, etc) expressed. Thus a system works to the extent that its output meets this need as judged by the user with the need. An effective system, as thus defined, can be built by working upwards from the grounding

provided by the Probability Ranking Principle. Since the core issue considered here is relating ambiguous requests to the PRP, it is repeated here: *If documents retrieved in response to a request are ordered by decreasing probability of relevance to the user's imputed need on the data available, then the system's effectiveness is the best to be gotten for the data* (Robertson 1977[1]). The single need presumption has been reinforced by a long experimental history of test data with multi-word requests, from the Cranfield sentences through the initial TREC topics to later tests with the shortest topic components, their titles, which average round four terms (Voorhees and Harman 2005).

But what to do if you are stuck with the reality, manifest in search engine logs, that requests average just over two terms: two terms rather than one may reduce the range of possible senses and aspects, but not typically to just one sense/aspect; and single term requests are normally ambiguous. From the system's point of view, a submitted request is thus a *bundle* of separate, even if related or overlapping, requests; and there is no way of determining, from the face of the request or (as is usually the case) immediately and directly from the user, which the actual request is.[2]

There is plenty of empirical evidence that visible submitted single requests are actually bundles for the following two reasons. First that requests, especially short ones, are linguistically ambiguous, not just in the classical sense of words with multiple senses present in a dictionary, but also ambigous across place names, person names, acronyms, etc. Second because humans (users) are squashy in their relevance judgements, innevitably influenced by the context of their search. Thus as system builders we have to find a way of hedging our bets in choosing what to output, and in particular with a modern ranking system, how to rank output items. This in turn clearly calls for some corresponding consideration, elaboration, or revision, of any underlying theory that is couched in terms of the user's need. Can one, for example, speak of multiple Probability Ranking Principles (PRPs), and a requirement to develop some smooth way of integrating these to deliver a single output ranking for a retrieval run? At the same time there are significant implications for retrieval system testing. Thus it appears to be necessary to unbundle a submitted request into a set of requests, and to consider every document to be assessed, separately, for its relevance to each deemed corresponding need.

Given the conventional 'unitary request' approach to retrieval, the challenges that ambiguous requests present for system design, theory and testing are not trivial. In particular, research hitherto has rested, at all levels, on a simple dichotomy, namely relevant vs non-relevant: relevant to this request (i.e. need) or non-relevant to this request (need). Using multiple relevance grades imposes a more refined grading along this single line, but is not a way of recognising bundles. However once we accept there can be multiple sense/aspect variations of the given request, we are faced with the possibility that a document may be relevant to one interpretation and not to another.

[1]This is a paraphrase of the original PRP. The original paper actually also discusses some cases which cause problems for the PRP, which are examples of the kind of ambiguity that is the subject of the present paper. However the examples are not often described in subsequent literature and in the rest of this paper we assume the PRP to have been interpreted in this straightforward manner.

[2]Thus there could in principle be members of the request bundle that were actually just alternative surface linguistic expressions, or *representations*, of the same underlying need. This may seem unlikely, since the notion of expressive variants is normally associated with visibly different but synonymous words, rather than invisibly different word meanings. But given that needs can genuinely be broad and vague, it remains a logical possibility. However since the major issue is the potential existence of multiple needs underlying a single surface request form, i.e. that the request has several *interpretations*, the rational assumption to make, once we abandon the single need presumption, is that there is a one-to-one correspondence between the members of the need set and the request bundle. Thus what we are doing is working with the notional one-to-one surrogates for needs, i.e. the logically separate requests represented by the various sense/aspect possibilities for the single visible submitted request.

2 Problem analysis

Suppose that we have a request R with two interpretations, i.e. R is really a bundle $\{R\} = \{r1, r2\}$. Suppose, further, that we know what these possible interpretations are, that is we have suitable knowledge of language and the world. For simplicity for now we assume that R consists of a single word "r". The $r1$ and $r2$ represent different senses/aspects of "r". It is then reasonable to suppose, further, that as part of the business of assessing returned documents – and this content interpretation is the essence of assessment anyway – we can correlate each individual document specifically, and separately, with $r1$ and $r2$. That is we can say whether each document is relevant or not to the $r1$ need, and similarly, independently, for the $r2$ need.

However even if we have this complete information about what R could mean, and can assess each returned document accordingly, we do not know which of $r1$ and $r2$ the user has in mind. Of course if we did know, we would select whichever of $r1$ or $r2$ applied and seek, using our best system design and resources, seek to deliver an appropriately ranked output for this interpretation of R, i.e. rely on a single PRP. But as we do not know whether $r1$ or $r2$ applies (and even if they have something semantic in common they are not identical), it seems we have to find some way to force two distinct, i.e. conflicting, PRPs into a single output ranking corset.

In reality of course, our knowledge of the possible interpretations of R is limited (our systems are not artificial intelligences), and the same applies to our (the system's) ability to assess relevance. Thus we already know that the system cannot assess relevance in the way that humans assess relevance, even if we suppose there is no problem of request interpretation. The system has to deem relevance or, in e.g. probabilistic frameworks, estimate it.

But this does not imply that the situation with ambiguous requests is quite the same as the familiar one. In the familiar situation the assumption is that the request is the surface form of a single underlying need, though this need may be broad and vague. Whether the need changes over time, including through interaction with the system, is a separate matter. Whether a need can be well expressed is also strictly independent of whether it is well expressed by the submitted request. The important point is that in practice the assumption hitherto has been that the system is dealing, at any specific time, with a single need and the problem is only that this need may be more or less adequately/fully/precisely expressed. Request ambiguity has been viewed as a matter of expressive failure, not of need multiplicity.

Thus even where alternative request representations are developed as variant search queries, as for instance in InQuery (Croft 2000), these are still seen as attempting to capture the one underlying need better, rather than as reflecting different needs. The different representation forms may be deemed 'better' or 'worse' through some representation-type weighting scheme, but the essential model is a combinatorial one, of bringing all the different ways of expressing the request together, the better to determine the likely relevance, and hence output rank position, for each matching document (Croft 2000, Chapter 1). There is never any question of recognising and trying to deal explicitly with request ambiguity as a function of need multiplicity, and hence of teasing different request interpretations apart and dealing with them separately. The nearest such a system gets to this is in an inverted and indirect way, since the results of the ranking process de facto impute whatever interpretation of the request is embodied in each ranked document back onto the request. Depending on the query term makeup of the matching documents, the process may therefore implicitly allow for different underlying needs rather than different expressions of the same need.

But this is a very low-key recognition of ambiguity and one which does not tackle the request ambi-

guity problem head on. The issue is what to do if, as when a very short request is presented, there is every reason to recognise that this can stand for several needs, and these needs may be sufficiently distinct to make it desirable to address this multiplicity explicitly, rather than pretend it does not exist. [This might, indeed, be a *generally* better strategy, i.e. one that ought to be built in from the start rather than at worst ignored or at best hopefully finessed, as is currently the case. However it might be unnecessary, or at least unproductive, work as requests got longer and hence less ambiguous.]

The root question, then, is what kind of leverage can be got to identify the members of the request bundle, and how to handle the resulting system matching process and output.

3 A solution

The tool for dealing with the system design issues is apparently already available and has indeed been deployed as the basis for standard current systems: it consists of distributional data about word occurrences and cooccurrences, and such weighting factors as tf, idf and so forth.

For example, we can envisage some such strategy as follows, taking single-term requests as a conveniently simple case. We construct lexical entries for the words (or stems, or multi-word units, but for simplicity we refer here simply to words), that consist of those other words that cooccur significantly in the file with each. These word *mate* sets will probably, in thoroughly time-honoured fashion, consist of a mix of sense and aspect indicators, but might supply good enough characterisations of (the) different meanings of the initial request word and hence be taken as indicators of the different needs underlying the single surface request term. Thus we can establish a mate set $m1, m2, ...$ for each of $r1, r2, ...$ (including the request word "r" in each).

Then if we take each mate set for the word "r" in R as defining the members $r1$, $r2$ .. in the bundle $\{R\}$, we can compare the word list for each document with the mate set for each member of the bundle, and assign each matching document to the appropriate member of the bundle. This is a query expansion process: each interpretation $r1$, $r2, ...$ of the original request R is represented by its mate set. Query expansion has it own problems (e.g. topic reinforcement versus dilution), and an alternative strategy could be to use the mate sets solely to allocate documents that match the single original request term "r" to different needs. However we can continue with the first strategy here, though it is only an illustratation of how different underlying needs might be handled, as more conventional in spirit. Developing this strategy to deal with requests containing more than one word does not need anything distinctively new: just some way of generating a derived 'lexical entry' for a whole request from the entries for its constituent words and then proceeeding as before.

Now suppose we have done all of this. The question that naturally arises is 'Why should we merge the outputs for the different requests $r1$, $r2, ...$ to give a single output ranking?' There seems to be no *logical* reason for doing this, given the assumption that we have been able to establish a request bundle and also do this reasonably accurately, and given the further standard retrieval assumption that we have been able to characterise and assign documents respectably for each separate request $r1$, $r2, ...$ (as now represented by their mate sets), so the respective rankings for $r1$, $r2, ...$ has some legitimacy. For the rest of this Section we assume separate ranks will result from each request.

Of course the practicalities of such a strategy as that outlined may be far from trivial, but that is not the issue here. The important point is whether there is a problem with the PRP. Why should there be, as a *theoretical* problem? The PRP is very general and simply says that ranking for a request will be

correlated with relevance to its underlying need. It says nothing about how the request is expressed, other than that however it is expressed constitutes evidence for the need.

The strategy just considered is not, however, saying that we replace the user‘s request by a set of alternative expressions, with the implication that some evidence combination method is required that constitutes a principled interpretation of the PRP. The strategy just described deals with ambiguous requests in a more radical way by replacing a single need by inferred multiple needs which each have their own request expression in the form of the mate sets. Applying the PRP to each of these separately is then quite straightforward and there is no theoretical problem. That is, under this view, the putative theoretical problem of how to apply the PRP when requests are ambiguous just goes away. We have separate needs, their requests, and their corresponding PRPs.

There is still, within this framework, an evaluation difficulty. This is that the request bundle/need set the system identifies is not necessarily the bundle/set the user might allow for, when invited to think as it were self-consciously and explicitly in terms of test evaluation protocols. There has to be some way of connecting the set of system output rankings with this user's set of request interpretations, i.e. of needs, and clearly if there is no good pairwise distribution, evaluation is going to founder. But this does not imply the strategy described, or any other with the same aims, is at fault. If there is no user-supplied information to differentiate the members of the request bundle, the system has to do whatever it can. The requirement is rather to develop an appropriate evaluation strategy, where multiple judgements are sought for the same request. There is some existing work in this area: historically, TREC's interactive track considered the multiple *aspects* of topics (Robertson, 2000); more recently in the definition questions in TREC‘s QA task, distinct *nuggets* of information were identified and systems that maximised their retrieval of the nuggets were scored higher (Voorhees 2005). However, in both cases, there was an assumption that the query had one core interpretation: answers, no matter how many or diverse, were bounded by the single interpretation. What is needed is to go beyond the maxim of “one interpretation per query”.

Thus is may be possible to have the users actively (i.e. imaginatively, thinking outside their own box) construct request bundles, i.e. need sets for their original requests, say by providing paraphrase sentences, phrases or term lists to distinguish bundle members, and then to link their bundles to those identified by the systems being assessed, e.g. by lexical overlap. Assessment would then be per bundle. The effort required for the user to assign each retrieved document directly to its appropriate need, on their judgement, i.e. to evaluate the entire retrieved set against all the needs, would be crushing. There are manifold details to address, for example what happens when the system has two component requests with their separate output rankings assigned to a single user request interpretation. But these are strictly evaluation methodology issues, not ones for the view of how system and theory deal with ambiguous requests.

There is also, perhaps, a slight slipperiness in the treatment of the PRP in the strategy just described. In the simple interpretation of the PRP, the user expresses their need as their *request*, and (although queries are not mentioned in the PRP) it is assumed the request becomes the search *query*, for which matching delivers the ranked output. The request is the same as the query which is the same as the user-side evidence to which the PRP refers. In practice, the initial request is processed, so the query that represents is may be quite different. The application of the PRP from the point of view of implementation takes the query as the user-side evidence, not the original request. When the transformation consists of no more than losing stop words, or query parsing, or request term frequency counting, this seems reasonable; and in fact other operations, like query term weighting that exploits collection data, do not

subvert this user-side view, though they appear to: they are just manipulating collection evidence in a mechanically convenient way.

The presumption in this account is that such request-to-query transformations are within the system. In practice, a user's uttered request may be manually preprocessed before query submission (as in the classical library service environment). The particular form of liberties taken with an original ambiguous request under the proposed strategy might thus be deemed compatible with the usual rather open view of the pre-query stage of retrieval. It's just pushing this a bit further by allowing the user to be confused about what their need is, not just how they should express it.

4 Problem ramifications

Suppose, however, that it is an ineluctable fact of real user behaviour that they do not want to be presented with multiple separate output rankings, and also (as seems equally the case in practice) that they do not want to be bothered by being asked to select/confirm input request interpretations, i.e. to identify $r1$, or $r2$, etc as indicators of separate needs. This is perfectly reasonable, because from their point of view they have only a single need, and believe they have expressed it adequately, so what they want is the ranking for that need. Indeed they are likely to be especially irritated if the system's idea about possible needs is all wrong. Certainly adopting the tactic of presenting separate rankings independently, say as pages of ten items with the most likely list page first followed by the others, is not an attractive option since it is very high risk. The user may well get nothing relevant to their actual need for one or more whole pages. It is a much higher risk than the current actual approach which, by assuming a single need, ignores the possibility of alternative request interpretations but may in fact, because matching with short requests is undiscriminating, get at least one response per need per page.

But more importantly, on a correctly rigorous view of the PRP, the submitted request is just and only the evidence available for the user, and the system has to use whatever other evidence it has to score matches and deliver its output ranking. Thus if the system has no user-side evidence that might be such as to enable it to select one need rather than others, providing a single rather than multiple output rankings is not attempting to be user convenient, but the only way to proceed in the light of the PRP. Of course if there is some user-side evidence it may not be very solid, which might suggest a strategy of offering first a complete ranking according to the preferred interpretation of the request, and then less preferred other, separate, rankings. But this is mere mechanics and is not the issue here. The issue here is what to do if there is no user-side evidence.

Thus if we start from the position that there is no direct user-side evidence of need, while there is system-side evidence, say of the kind illustrated by the mates idea considered earlier, that there may be multiple needs lurking behind the submitted request, the question is how to combine this as evidence for the value of some document with other evidence about document value so as to determine where the document is ranked in the search output. As the driver for all of this is the PRP, the evidence is all probabilistic, so the issue is the appropriate way of combining probabilities.

Suppose, for example, we have the situation shown in Figure 1a. We have the submitted request "java", where the word "java" has three senses $s1, s2, s3$, *coffee*, *island*, and *programming language* (*proglang* for short), which we will for simplicity take as standing for the three needs $n1, n2, n3$. We further suppose that in reality this request was submitted three times, at $t1, t2, t3$, in fact by three different users, $u1, u2, u3$, whose needs were respectively $n1, n2, n3$. However even if we know the users

```
a) request R = ``java'' ==> sense1 = n1 = coffee
                            sense2 = n2 = island
                            sense3 = n3 = proglang

time t1  request ``java'' u1 with n1
     t2     "    ``java'' u2 with n2
     t3     "    ``java'' u3 with n3


b)   document d1 mateset ``coffee''     prob relevance to R = .8
              d2         ``coffee''                           .5
              d3         ``proglang''  prob relevance to R = .4
              d4         ``proglang''                         .3

     prob need n1 ``coffee''   (via mateset) = .1
     prob need n2 ``island''                   .2
     prob need n3 ``proglang''                 .7

    score (prob rel * prob need)  d3 = .28
                                  d4   .27
                                  d1   .08
                                  d2   .05


c)  d1 prob need n1 ``coffee''    = .1
                 n3 ``proglang''  = .4

    score (prob rel * prob need n1) + (prob rel * prob need n3) = .4
```

Figure 1

are different, this does not provide any information about their needs in the absence of other user-side evidence. We may indeed not know the users are different. On the other hand, even if we had just a single user, and knew it, this would not necessarily imply their need was the same, each time. In other words, all we have is collection-side evidence.

This collection-side evidence could, for example, be word frequency data, perhaps for the mate data presented earlier. Thus while we have no means of knowing what senses "java" has from its frequency alone, we can get an idea of what its senses may be, and of their relative frequency, using mate sets. Thus we might find that *proglang* (i.e. the mate set for *proglang*) is more frequent across the collection than *island*, which is in turn more frequent than *coffee*. We may then project this frequency data back onto need characterisation, inferring that as *proglang* is more frequent than the other senses, it is more likely to represent the user's need (in fact at each of $t1, t2, t3$). We have of course to get numerical probabilities to represent this frequency data in some kosher way.

At the same time, we will have other probabilities, namely of relevance, computed also from fre-

quency data in the familiar way. Thus for example if we treat a mate set of words as if it were an expanded request we can get the usual kind of numbers out. We then have to combine the probability of relevance to a need with the probability of relevance of that need for each document.

For instance suppose we have the situation illustrated in Figure 1b, where mate set matches for the different senses of "java" are simplistically represented by their respective sense words, i.e. sense $s1$ of "java" is represented here by the word "coffee", which stands for the mate set for *coffee*. We can ignore any other elements of a proper scoring function here and simply multiply the two component probabilities. Thus in relation to the need $n1$ *coffee* alone, as represented by "coffee", we have document $d1$ with probability of relevance and top rank position for this need alone of $.8$, and $d2$ with probability of relevance $.5$. While for the need $n2$ with mate set "proglang" we have $d3$ and $d4$ ranked by relevance probabilities $.4$ and $.3$. However we have also computed the probabilities of $n1$ and $n3$ themselves as $.1$ and $.7$ respectively. In other words $d1$ and $d2$ have high probabilities of relevance but low probabilities of need, and $d3$ and $d4$ the reverse. Merging, as required, for a single outcome ranking using the multiplied probabilities would give the result $d3, d4, d1, d2$. Is this what we want?

Figure 1c illustrates the effect where we allow a document, $d1$ here, to have a positive probability of relevance to more than one need (java programmers consume coffee). Again, does the way we have combined the need and the relevance probabilities and their effect on the final score for $d1$ reflect what we want? Further, of course, the mate set strategy could imply matches between a document and request, i.e. need, on more than one word from a mate set. This requires appropriate refinement of the scoring formula.

This is a sketch showing how one might apply the PRP in a principled way. In practice we can imagine all kinds of ad hoc strategies depending on the information available. For example we might have a means of identifying alternative senses of the request words, and of their occurrence for documents, but none of the numerical data to supply acceptable guidance about probabilities. This could lead to taking documents one each on a round-robin basis from separate rankings per need in order to obtain a final merged output ranking. This kind of thing could get extremely complicated with multi-word requests. Thus is would seem preferable to dig very earnestly into any possible source of information about need probabilities.

Equally, even if we suppose there is some user-side evidence for one need rather than another in the set as underlying the offered request, if this was extremely weak it might be preferable both in principle and in practice to try to factor this into the PRP-based scoring function covering all the inferred potential needs, rather than apply it to select a ranking tailored just to that putative need. This would then require an explicit 'need-flagging' component in the document score computation: it would not be the same kind of thing as, say, the convergence on some one need of relevance feedback. Thus the whole scoring mechanism would have to factor in, explicitly, the recognised set of needs.

Superficially, all of this may not look very different from manipulating different representations of the same need. But the underlying assumptions, and hence structure of the scoring formula, as well as the status of the specific evidence used to characterise each need, can be expected to be quite different.

5 The problem in theory

Some previous work has argued for retrieval strategies that seek diversity in the returned set, as a general approach rather than as a specific response to ambiguous requests, though the latter follows from the

former. Thus Carbonell and Goldstein (1998)'s MMR ranking method is designed to seek successively different documents. However Zhai et al. (2003) seek explicitly to allow for different interpretations of a query, treating these as subtopics, aspects of the overall topic, which should each be covered in the returned set. Both of these approaches are addressing a user's presumed interest in both novelty and relevance as they proceed down the return ranking. Seeking difference from what has already been offered may be seen as penalising redundancy. Chen and Karger (2006), on the other hand, maintain that diversity in output, reflecting different interpretations or aspects of a query, would be the indirect result of a retrieval strategy with another primary motivation, namely to seek to supply the user with some specific number k, typically low, of relevant documents: thus shooting to deliver 3 relevant documents in 10, say, would be more likely to succeed if the offered documents were different than when their content were the same. However it is not clear that the actual heuristic procedure they use, though of the generic reranking kind like the others mentioned, would have this effect. Thus while the general aim is to return enough relevant documents, the specific relationship to be satisfied by the next returned document given those already returned appears to depend on the chosen value of k, and might not automatically promote diversity. It is not clear that a retrieval strategy that depends on setting a wanted number of documents is satisfactory in itself, though if it were to promote diversity this would be an advantage from the present point of view, especially if its behaviour was stable enough for the user to be able to treat k as a rough rather than precise specification of the number of wanted documents.

More generally, if we accept that Chen and Karger's approach could in principle promote diversity, it would, like the others mentioned, be exploiting statistical data in a way which responds to topic ambiguity as embodied in the documents in the collection, rather than seeking to capture it directly through operations on the vocabulary that are then used to guide retrieval.

The conclusion reached in the previous section is that whatever else is a challenge with ambiguous requests, the existence of ambiguous requests does not require reassessment or revision of the PRP, as a grounding for a retrieval model like the classical Probabilistic Model (PM) (Spärck Jones, Walker and Robertson 2000). However it can be argued that even if there is no problem for the PRP itself, there is a problem with a model like the PM interpreting it. The PM is all about estimating relevance from the available data (e.g. term distribution data). This data is an uncertain indicator or the real relationship between document and need. One of the sources of the uncertainty in the PM is the request as expression of need, the other being the document text as expression of its content.

But it is possible to argue that if the request is an uncertain indicator of the underlying need, it is perfectly proper to allow the uncertainty to hold for the need itself, i.e. to take the uncertainty to project backwards onto the need itself. This follows from the fact that while meaning and language are not identical, they are symbiotically entwined, so linguistic uncertainty is also meaning uncertainty. Linguistic ambiguity implies a possible multiplicity of needs (even if not always an actual multiplicity because request ambiguities *may* be alternative expressions of the same need).

The consequence then seems to be that the data analysis and relevance estimation processes (a model like the PM is designed to support) have to be enriched to function in a sort of need unpacking and repacking way. That is, one might view the request as a mixture model of need expressions and seek to unpack them. Thus to allow for request ambiguity, i.e. for need alternatives underlying the offered request, one would look for evidence, applying the same kind of analysis in principle as outlined earlier as a system strategy for decomposing putative request bundles, to establish what requests, and hence underlying needs, are in the bundle. This analysis would have to form part of the way the abstract PM is interpreted, in the same way that term weighting notions currently interpret the PM. This approach

would leave the PRP itself as a single ranking justification, per need, but allow for multiple needs and rankings for them. Developing the PM to cover uncertainty about need, for a single given request R, should deliver a single combined output incorporating estimates of the relative status – on some data-determined basis – of the needs underlying R. This would presumably refer primarily to the existence of the need in the sense of conceptual distinctness, but might also refer to relative importance in the file. Translating such general considerations into a soundly motivated procedure for producing the actual single output ranking is, however, another matter.

The alternative would appear to be some attempt to develop what might be called a second-order PRP on top of the existing first-order one, so one might talk about ordering documents by decreasing probability of relevance to need and, where there is uncertainty about what the need is, quite separately by decreasing likelihood of need. But it is not clear what this 2PRP would really be.

6 Conclusions

In this note, the question of how search engines can best deal with ambiguous requests was discussed. The existance of such requests was established. The impact on common interpretations of the probabilistic ranking principle (PRP) was also considered and means for systems to deal with such requests were described. The relatively small selection of existing research addressing the actual building of systems capable of dealing with such requests were surveyed. As often happens in our field, however, a strong stimulus to research in this important area would come from a test collection being created that addresses such requests. Currently, no such collection exists; perhaps it is time for that situation to change.

7 References

Broder, A. 'A taxonomy of web search', ACM SIGIR Forum, 2002.

Carbonell, J. and Goldstein, J. 'The use of diversity based reranking for reordering documents and producing summaries', SIGIR, 1998, 335-336.

Chen, H. and Karger, D. 'Less is more: probabilistic models for retrieving fewer relevant documents', SIGIR, 2006, 429-436.

Croft, W.B. (Ed.), *Advances in information retrieval*, Dordrecht: Kluwer, 2000, 1-36.

Spärck Jones, K., Walker, S. and Robertson, S.E., 'A probabilistc model of information retrieval: development and comparative experiments', Parts 1 and 2, *Information Processing and Management*, 36, 2000, 779-808 and 809-840.

Voorhees, E.M. 'Question answering in TREC', in *TREC. Experiment and evaluation in information retrieval*, Ed. E.M. Voorhees and D.K. Harman, Cambridge, MA: MITP Press, 2005, 233-257.

Robertson, S.E. 'The probability ranking principle in IR', *Journal of Documentation*, 33, 1977, 130-137.

Voorhees, E.M. and Harman D.K. (Eds.) *TREC. Experiment and evaluation in information retrieval*, Cambridge, MA: MIT Press, 2005.

Zhai, C., Cohen, W.W. and Lafferty, J. 'Beyond independent relevance: methods and evaluation metrics for subtopic retrieval', SIGIR, 2003, 10-17.