Language models and probability of relevance

Stephen Robertson

Microsoft Research
Cambridge, UK
and City University
London, U.K.
ser@microsoft.com

Djoerd Hiemstra

Microsoft Research
Cambridge, UK
and University of Twente
Enschede, The Netherlands
hiemstra@cs.utwente.nl

1 A formulation of the Language Model for IR

The basic formula used in several of the papers which take a language modelling approach to IR can be written as follows:

$$P(D, T_1, T_2, \ldots, T_n) = P(D) \prod_{i=1}^{n} ((1 - \lambda) P(T_i) + \lambda P(T_i|D)) \tag{1}$$

D is a document, $\{T_i\}$ are query terms. The whole represents the probability that the query could have been generated from the language model representing the document (here simplified to the $P(T_i|D)$ values), but with a smoothing parameter λ, which allows terms some chance of coming from a general language model (the $P(T_i)$ values). The conception is that the user has a document in mind, and that s/he generates the query from this document; the equation then represents the probability that the document that the user had in mind was in fact *this* one.

Hiemstra [1] gives the same equation a slightly different justification. The basic assumption is the same (the user is assumed to have a specific document in mind and to generate the query on the basis of this document), but instead of smoothing, the user is assumed to assign a binary importance value to each term position in the query. An important term-position is filled with a term from the document; a non-important one is filled with a general language term. If we define $\lambda_i = P(\text{term position } i \text{ is important})$, then we get

$$P(D, T_1, T_2, \ldots, T_n) = P(D) \prod_{i=1}^{n} ((1 - \lambda_i) P(T_i) + \lambda_i P(T_i|D)) \tag{2}$$

which reduces to the former equation on the assumption that the probability of importance is the same for all positions.

Actually, Hiemstra's full (extended) model is somewhat more complex. As well as the processes of assigning importance and choosing terms, there is a translation stage where a term may be translated into another related term. This is useful for various situations, such as structured queries and cross-lingual retrieval; it allows explicitly for the possibility that a meaningful query term does not appear in a specific relevant document, which the basic model does not.

2 The source document concept

The idea that a query arises from a specific document might make sense if we assume that there is exactly one relevant document in the collection, which the user knows (or guesses) something about. In that case, the system may or may not rank this document first, but as soon as the user reaches the document in question, the search must be finished. Indeed, if the model is self-consistent, the probabilities assigned to all documents in the collection should sum to one, and should change as the user makes judgements. Furthermore, the only form of relevance feedback that now makes sense is negative – in the case of a positive judgement all remaining probabilities become zero.

It is possible to be a little more vague about the origin of the query, by keeping the formal model as defined above, but suggesting informally that the user has some kind of ideal relevant document in mind, and that s/he will judge a document relevant if it is sufficiently close (similar) to this ideal. This is the line taken, for example, by Ponte [2]. The problem with this concept is that it immediately compromises one of the advantages of the LM approach. One feature which the LM approach might be said to share with the probabilistic model approach to IR is that it is quite explicit about the measure of association or similarity between the document and the query. Both models avoid the general, loose notion of similarity that characterises many other approaches, in favour of an explicitly defined measure that comes out of the assumptions of the model. This is so in the LM approach if it is assumed that there is a specific precursor document: the measure of similarity is now the probability that this particular document is the one. But the vaguer concept relies on some undefined (in the model) notion of similarity between a specific document and the conceived ideal document.

3 Hiemstra's relevance feedback model

Hiemstra [1] proposes a version of his model which allows for multiple relevant documents and relevance feedback. In this version, each relevant document is assumed to have generated the query independently. Then, for feedback, each known example of a relevant document can be used to provide evidence about the importance λ_i of each query term. This evidence is accumulated across the examples, and the result is an estimate of λ_i, potentially different for every term i, which may be used in a new search. (Hiemstra does not address the question of query expansion.)

A problem with this model is that the notion of each query term position having some general (average) importance independent of the document is not really compatible with the assumed generation process. In the generation model, if a term does not occur in a particular document, it *must* have been unimportant when the query was generated from that document. So we would have to conclude that as the same query is generated from different documents, the same query term position is sometimes unimportant and sometimes important – but it nevertheless results in the same actual word each time.

This problem would not occur in the same form under Hiemstra's extended model (which has a translation stage as well as a choice stage) – one might then assume that importance has been pre-assigned to term positions, independently of the generating document, and that a term which seems to be important but which does not occur explicitly in this particular document is the result of a translation step (of which we might be uncertain). But the idea of the same query arising by independent generation processes is still fairly far-fetched.

4 Ponte's relevance feedback model

Ponte [2], by contrast, addresses query expansion but not reweighting. Terms are selected from known relevant documents by some selection value or offer weight, inspired in part by previous work on term selection and in part by the parameters available from a language model for each of these example documents. The combination as a term selection value, although based on the set of example documents, does not appear to relate to any language model of this set as a whole.

These additional terms are added to the original query, and the result is treated as a new query (in the usual relevance feedback fashion, applied to some new documents). In other words, for each individual document, we ask the question "What is the probability that this document generated the query?" – even though we know that the new part of the query was generated from some *set* of documents, not including the document in question, and the user has already told us, once for each member of the set, that the old part of the query was generated from this individual document.

Both Hiemstra's and Ponte's approaches may be good heuristic solutions to the problem of incorporating relevance feedback into the language model approach. Neither of them really addresses the question of the relationship between the language model and relevance feedback – in particular, in what way a language model may be trained on the basis of relevance feedback data.

5 A general LM approach to multiple relevant documents?

What would a proper LM approach to the multiple-relevant-document case look like?

Here is one suggestion: there should be an explicit model which generates the set of relevant documents. Thus, and probably only thus, can relevant documents be seen in the LM approach to have some similarity, different from the whole collection, which would allow in principle some form of relevance feedback. But such a model would seem to have considerable difficulties.

The LM approach at present allows for $N + 1$ language models, where N is the collection size. The additional one is the general language model. The relationship between this last and the individual document models is never raised (How can a document be generated from one language model when the entire collection is generated from a different one?). This problem multiplies if we start looking at models for sets of documents.

Is there any mechanism within the LM field for considering some kind of global-and-local model structure? One possible mechanism is the one with which we are familiar: the mixture model. However, this is a curious view of the interaction between the two – two distinct low-level language models with unrelated probabilities for the same event, and a higher-level model for deciding which of these to use. It would seem to bear little relation to any real authorship process. The language that I use in writing a paper on a specialist subject is basically the same language I use when talking to my family, with only some parts modified by the context of the specific subject. It seems that what we need is a general model for some large accumulation of text, which is modified (not replaced) by a local model for some smaller part of the same text. The modification affects (for example) some parts of the vocabulary only.

If we assume that documents are multi-topic, and that relevance to a topic in the IR sense is only one aspect of such a multi-topic document, then we actually need a 3-level structure:

1. A whole-collection model…
2. …modified by a relevant-documents model…
3. …modified by an individual document model.

This last includes all the special aspects of this document that are specific to other topics, other than the topic of this particular user query. It would play something like the role of an error term in a regression or ANOVA model – "all the variance not explained by the above". It would seem to be necessary in the IR context in order to avoid the assumption that all relevant documents are the same.

The query would also be generated by 1 and 2. Thus the query, and also any known relevant documents, would be evidence about the 1+2 combination, and specifically (in comparison with the whole collection) about how 2 modifies 1. The question to be asked about each document is: Does model component 2 explain any aspect of this document? Or in other words, what evidence does this document present for a level 2 component? The task of relevance feedback would be to draw inferences (i.e. learn) about level 2.

It may be noted that in this context, the concepts of 'explanation', 'generation' and 'prediction' are more or less equivalent: saying a model *explains* a piece of text is the same as saying it *generates* it (in the language model sense), or that it *predicts* it.

5.1 Parsimonious models

A difficulty with this approach lies in comparing different models for their explanatory power concerning a piece of text. If a model can incorporate as many parameters as it wants, then it can fit any fixed set of data to any degree of accuracy. In principle, a document-specific model can explain a document completely (or rather, within the limits of the basic modelling approach, e.g. bigram), and has no need to appeal to a higher-level model of a collection of documents. The only sense in which such a higher-level model might contribute is in the sense of explaining the data more parsimoniously. Thus a collection model with only minor modifications for the individual documents would be good because it explained the same set of data as well but with less parameters.

So, for example, a unigram model of general language, modified by a small number of topic-specific unigram models, each of which had parameters for only a small number of topic-specific specialist terms, would be a very parsimonious model for a collection of documents. The modelling approach to be used would have to reward parsimony for this to work. There are indeed approaches to statistical modelling, particularly some Bayesian and maximum entropy models, which do this. But as far as we know no such model has been used in the LM domain.

5.2 The 3-level model again

Assume, then, that we have an approach to language modelling which (a) rewards parsimony, and (b) allows a more specialist model to be built parsimoniously on top of a more general one. We suppose that every document is generated by a general language model (level 1), modified by some unknown set of special models relating to the specific combination of topics referred to in this document. We will assume that they can all be rolled into a single individual-document model (level 3) which represents everything in that document that cannot be described by the general language model. However, if we have a model for a specific topic that is represented in the document, this model will take over part of the individual-document model – potentially more parsimoniously because it is applicable to all documents in which this topic is represented, and because it makes the remaining individual-document model more parsimonious.

Now the relevance hypothesis is as follows: A request (topic in the TREC sense) is generated from a specific-topic model (as usual, modifying the general language model). If and only if a document is relevant to the topic, the same model will apply to the document – that is, it will replace part of the individual-document model in explaining the document. Thus the probability of relevance of a document is the probability that this model explains part of the document – more specifically, the probability that the {1,2,3} combination is better than the {1,3} combination. Another way of putting it: there is a null hypothesis about each document, namely the {1,3} combination, and an alternative hypothesis, that the {1,2,3} combination explains it as well but more parsimoniously; we have to assign a probability to this alternative against the null hypothesis.

A document that has already been judged relevant provides us with independent evidence, additional to the query, about the level 2 part of the model.

6 Discussion

The above is clearly a sketch only, and suggests some requirements for a model.

However, one interesting observation can be made about it even at this stage. The process of developing a level 3 model for an individual document in the {1,3} case is independent of any request, and is essentially an indexing process: it asks the question "How does the language of this document differ from that of the whole collection?". It looks as if it should deal automatically with stopwords as well as with tf and idf (which one might expect a language model to do). Stopwords (or at least some of them) would be words for which the general language model is sufficient, and which therefore do not figure at all in the level 3 model of an individual document.

Acknowledgement

We would like to thank Hugo Zaragoza for comments and discussions on this paper.

References

[1] Hiemstra, D. *Using language models for information retrieval*, PhD Thesis, University of Twente, 2001.

[2] Ponte, J.M. 'Language models for relevance feedback', in *Advances in information retrieval*, Ed. W.B. Croft, Dordrecht: Kluwer, 2000, 73-95.