

Multi-level Association Graphs – A New Graph-Based Model for Information Retrieval

Hans Friedrich Witschel

NLP department

University of Leipzig

04009 Leipzig, P.O. Box 100920

witschel@informatik.uni-leipzig.de

Abstract

This paper introduces *multi-level association graphs* (MLAGs), a new graph-based framework for information retrieval (IR). The goal of that framework is twofold: First, it is meant to be a meta model of IR, i.e. it subsumes various IR models under one common representation. Second, it allows to model different forms of search, such as feedback, associative retrieval and browsing at the same time. It is shown how the new integrated model gives insights and stimulates new ideas for IR algorithms. One of these new ideas is presented and evaluated, yielding promising experimental results.

1 Introduction

Developing formal models for information retrieval has a long history. A model of information retrieval “predicts and explains what a user will find relevant given the user query” (Hiemstra, 2001). Most IR models are firmly grounded in mathematics and thus provide a formalisation of ideas that facilitates discussion and makes sure that the ideas can be implemented. More specifically, most IR models provide a so-called *retrieval function* $f(q, d)$, which returns – for given representations of a document d and of a user information need q – a so-called retrieval status value by which documents can be ranked according to their presumed relevance w.r.t. to the query q .

In order to understand the commonalities and differences among IR models, this paper introduces the

notion of *meta modeling*. Since the word “meta model” is perhaps not standard terminology in IR, it should be explained what is meant by it: a meta model is a model or framework that subsumes other IR models, such that they are derived by specifying certain parameters of the meta model.

In terms of IR theory, such a framework conveys what is common to all IR models by subsuming them. At the same time, the differences between models are highlighted in a conceptually simple way by the different values of parameters that have to be set in order to arrive at this subsumption. It will be shown that a graph-based representation of IR data is very well suited to this problem.

IR models concentrate on the *matching* process, i.e. on measuring the degree of overlap between a query q and a document representation d . On the other hand, there are the problems of finding suitable representations for documents (*indexing*) and for users’ information needs (*query formulation*). Since users are often not able to adequately state their information need, some interactive and associative procedures have been developed by IR researchers that help to overcome this problem:

- *Associative retrieval*, i.e. retrieving information which is associated to objects known or suspected to be relevant to the user – e.g. query terms or documents that have been retrieved already.
- *Feedback*, another method for boosting recall, either relies on relevance information given by the user (relevance feedback) or assumes

top-ranked documents to be relevant (pseudo feedback) and learns better query formulations from this information.

- *Browsing*, i.e. exploring a document collection interactively by following links between objects such as documents, terms or concepts.

Again, it will be shown that – using a graph-based representation – these forms of search can be subsumed easily.

2 Related work

2.1 Meta modeling

In the literal sense of the definition above, there is a rather limited number of meta models for IR, the most important of which will be described here very shortly.

Most research about how to subsume various IR models in a common framework has been done in the context of Bayesian networks and *probabilistic inference* (Turtle and Croft, 1990). In this approach, models are subsumed by specifying certain probability distributions. In (Wong and Yao, 1995), the authors elaborately show how all major IR models known at that time can be subsumed using probabilistic inference. Language modeling, which was *not* known then was later added to the list by (Metzler and Croft, 2004).

Another graph-based meta modeling approach uses the paradigm of *spreading activation* (SA) as a simple unifying framework. Given semantic knowledge in the form of a (directed) graph, the idea of spreading activation is that a measure of relevance – w.r.t. a current focus of attention – is spread over the graph’s edges in the form of *activation energy*, yielding for each vertex in the graph a degree of relatedness with that focus (cf. (Anderson and Pirolli, 1984)). It is easy to see how this relates to IR: using a graph that contains vertices for both terms and documents and appropriate links between the two, we can interpret a query as a focus of attention and spread that over the network in order to rank documents by their degree of relatedness to that focus.

A very general introduction of spreading activation as a meta model is given in the early work by (Preece, 1981) All later models are hence special

cases of Preece’s work, including the multi-level association graphs introduced in section 3. Preece’s model subsumes the Boolean retrieval model, coordination level matching and vector space processing.

Finally, an interesting meta model is described by (van Rijsbergen, 2004) who uses a Hilbert space as an information space and connects the geometry of that space to probability and logics. In particular, he manages to give the familiar dot product between query and document vector a probabilistic interpretation.

2.2 Graph-based models for associative retrieval and browsing

The spreading activation paradigm is also often used for associative retrieval. The idea is to reach vertices in the graph that are not necessarily directly linked to query nodes, but are reachable from query nodes via a large number of short paths along highly weighted edges.

Besides (Preece, 1981), much more work on SA was done, a good survey of which can be found in (Crestani, 1997). A renewed interest in SA was later triggered with the advent of the WWW where hyperlinks form a directed graph. In particular, variants of the PageRank (Brin and Page, 1998) algorithm that bias a random searcher towards some starting nodes (e.g. an initial result set of documents) bear close resemblance to SA (Richardson and Domingos, 2002; White and Smyth, 2003).

Turning to browsing, we can distinguish three types of browsing w.r.t. to the vertices of the graph: index term browsing, which supports the user in formulating his query by picking related terms (Doyle, 1961; Beaulieu, 1997), document browsing which serves to expand result sets by allowing access to similar documents or by supporting web browsing (Smucker and Allan, 2006; Olston and Chi, 2003) and combined approaches where both index terms and documents are used simultaneously for browsing.

In this last category, many different possibilities arise for designing interfaces. A common guiding principle of many graph-based browsing approaches is that of interactive spreading activation (Oddy, 1977; Croft and Thompson, 1987). Another approach, which is very closely related to MLAGs, is a multi-level hypertext (MLHT), as proposed in

(Agosti and Crestani, 1993) – a data structure consisting of three levels, for documents, index terms and concepts. Each level contains objects and links among them. There are also connections between objects of two adjacent levels. An MLHT is meant to be used for interactive query formulation, browsing and search, although (Agosti and Crestani, 1993) give no precise specification of the processing procedures.

2.3 Contribution of this work

Compared to Preece’s work, the MLAG framework makes two sorts of modifications in order to reach the goals formulated in the introduction: in order to subsume more IR models, the flexibility and power of Preece’s model is increased by adding real-valued edge weights. On the other hand, a clearer distinction is made between local and global information through the explicit introduction of “level graphs”.

With the introduction of *levels*, the MLAG data structure becomes very closely related to the MLHT paradigm of (Agosti and Crestani, 1993), MLAGs, however, generalise MLHTs by allowing arbitrary types of levels, not only the three types proposed in (Agosti and Crestani, 1993). Additionally, links in MLAGs are weighted and the spreading activation processing defined in the next section makes extensive use of these weights.

All in all, the new model combines the data structure of multi-level hypertexts (Agosti and Crestani, 1993) with the processing paradigm of spreading activation as proposed by Preece (Preece, 1981), refining both with an adequate edge weighting. The framework is an attempt to be *as general as necessary* for subsuming all models and allowing for different forms of search, while at the same time being *as specific as possible* about the things that are really common to all IR models.

3 The MLAG model

3.1 Data structure

Formally, the basis of a multi-level association graph (MLAG) is a union of n level graphs L_1, \dots, L_n . Each of these n directed graphs $L_i = G(VL_i, EL_i, WL_i)$ consists of a set of vertices VL_i , a set $EL_i \subseteq VL_i \times VL_i$ of edges and a function $WL_i : EL_i \rightarrow \mathbb{R}$ returning edge weights.

In order to connect the levels, there are $n - 1$ connecting bipartite graphs (or inverted lists) $I_{1,2}, \dots, I_{n-1,n}$ where each inverted list $I_{j,j+1}$ consists of vertices $VI_{j,j+1} = VL_j \cup VL_{j+1}$, edges $EI_{j,j+1} \subseteq (VL_j \times VL_{j+1}) \cup (VL_{j+1} \times VL_j)$ and weights $WI_{j,j+1} : EI_{j,j+1} \rightarrow \mathbb{R}$. Figure 1 depicts a simple example multi-level association graph with two levels L_d and L_t for documents and terms.

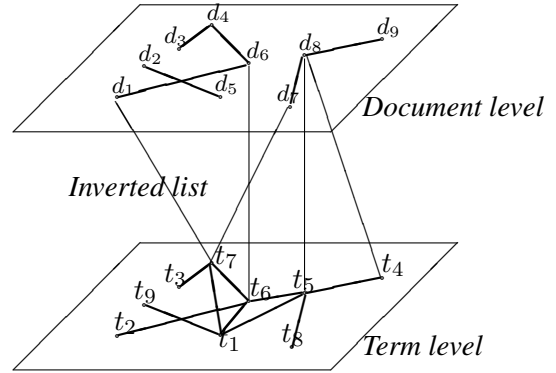


Figure 1: A simple example MLAG

Assuming that the vertices on a given level correspond to objects of the same type and vertices in different levels to objects of different types, this data structure has the following general interpretation: Each level represents associations between objects of a given type, e.g. term-term or document-document similarities. The inverted lists, on the other hand, represent associations between different types of objects, e.g. occurrences of terms in documents.

3.2 Examples

3.2.1 Standard retrieval

The simplest version of a multi-level association graph consists of just two levels – a term level L_t and a document level L_d . This is the variant depicted in figure 1.

The graph I_{td} that connects L_t and L_d is an inverted list in the traditional sense of the word, i.e. a term is connected to all documents that it occurs in and the weight $WI(t, d)$ of an edge (t, d) connecting term t and document d conveys the degree to which t is representative of d ’s content, or to which d is about t .

The level graphs L_t and L_d can be computed in various ways. As for documents, a straight-forward way would be to calculate document similarities, e.g. based on the number of terms shared by two documents. However, other forms of edges are possible, such as hyperlinks or citations – if available. Term associations, on the other hand, can be computed using co-occurrence information. An alternative would be to use relations from manually created thesauri or ontologies.

3.2.2 More levels

In order to (partly) take document structure into account, it can be useful to introduce a level for document parts (e.g. headlines and/or passages) in between the term and the document level. This can be combined with text summarisation methods (cf. e.g. (Brandow et al., 1995)) in order to give higher weights to more important passages in the inverted list connecting passages to documents.

In distributed or peer-to-peer environments, databases or peers may be modeled in a separate layer above the document level, inverted lists indicating where documents are held. Additionally, a peer’s neighbours in an overlay network may be modeled by directed edges in the peer level graph. More extensions are possible and the flexibility of the MLAG framework allows for the insertion of arbitrary layers.

3.3 Processing paradigm

The operating mode of an MLAG is based on the spreading activation principle. However, the spread of activation between nodes of two *different* levels is not iterated. Rather, it is carefully controlled, yet allowing non-linear modifications at some points.

In order to model spreading activation in an MLAG, we introduce an *activation function* $A_i : VL_i \rightarrow \mathbb{R}$ which returns the so-called *activation energies* of vertices on a given level L_i . The default value of the activation function is $A_i(v) = 0$ for all vertices $v \in VL_i$.

In the following, it is assumed that the MLAG processing is invoked by *activating* a set of vertices A on a given level L_i of the MLAG by modifying the activation function of that level so that $A_i(v) = w_v$ for each $v \in A$.

A common example of such activation is a query

being issued by a user. The initial activation is the result of the query formulation process, which selects some vertices $v \in A$ and weights them according to their presumed importance w_v . This weight is then the initial activation energy of the vertex.

Once we have an initial set of activated vertices, the following general procedure is executed until some stopping criterion is met:

1. Collect activation values on current level L_i , i.e. determine $A_i(u)$ for all $u \in VL_i$.
2. (Optionally) apply a transformation to the activation energies of L_i -nodes, i.e. alter $A_i(u)$ by using a – possibly non-linear – transformation function or procedure.
3. Spread activation to the next level L_{i+1} along the links connecting the two levels:

$$A_{i+1}(v) = \sum_{(u,v) \in L_{i,i+1}} A_i(u) \cdot WI(u, v) \quad (1)$$

4. Set $A_i(u) = 0$ for all $u \in VL_i$, i.e. “forget” about the old activation energies
5. (Optionally) apply a transformation to the activation energies of L_{i+1} -nodes (see step 2).
6. Go to 1, increment i (or decrement, depending on its value and the configuration of the MLAG)

If we take a vector space view of this processing mode and if we identify level L_i with terms and level L_{i+1} with documents, we can interpret the activation energies $A_i(u)$ as a query vector and the edge weights $WI(u, v)$ of edges arriving at vertex $v \in VL_{i+1}$ as a document vector for document v .

This shows that the basic retrieval function realised by steps 1, 3 and 4 of this process is a simple dot product. We will later see that retrieval functions of most IR models can actually be written in that form, provided that the initial activation of query terms and the edge weights of $L_{i,i+1}$ are chosen correctly (section 4).

For some models, however, we additionally need the possibility to perform nonlinear transformations on result sets in order to subsume them. Steps 2 and 5 of the algorithm allow for arbitrary modifications

of the activation values based on whatever evidence may be available on the current level or globally – but *not* in the inverted list. This will later also allow to include feedback and associative retrieval techniques.

4 The MLAG as a meta model

In this section, examples will be shown that demonstrate how existing IR models of ranked retrieval¹ can be subsumed using the simple MLAG of figure 1 and the processing paradigm from the last section. This is done by specifying the following parameters of that paradigm:

1. How nodes are activated in the very first step
2. How edges of the inverted list are weighted
3. Which transformation is used in 2 and 5.

For each model, the corresponding retrieval function will be given and the parameter specification will be discussed shortly. The specification of the above parameters will be given in the form of triplets $\langle activation_{init}, edge\ weights, transform \rangle$.

4.1 Vector space model

In the case of the vector space model (Salton et al., 1975), the retrieval function to be mimicked is as follows:

$$f(q, d) = \sum_{t \in q \cap d} w_{tq} w_{td} \quad (2)$$

where w_{tq} and w_{td} are a term's weight in the query q and the current document d , respectively. This can be achieved by specifying the parameter triplet $\langle w_{tq}, w_{td}, none \rangle$. This simple representation reflects the closeness of the MLAG paradigm to the vector space model that has been hinted at above.

4.2 Probabilistic model

For the probabilistic relevance model (Robertson and Sparck-Jones, 1976), the MLAG has to realise the following retrieval function

$$f(q, d) = \sum_i d_i \log \frac{p_i(1-r_i)}{r_i(1-p_i)} \quad (3)$$

¹This excludes the Boolean model, which can, however, also be subsumed as shown in section 5.5 of (Preece, 1981)

where $d_i \in \{0, 1\}$ indicates whether term i is contained in document d , p_i is the probability that a relevant document will contain term i and r_i is the probability that an irrelevant document will contain it. This retrieval function is realised by the parameter triplet $\langle \log \frac{p_i(1-r_i)}{r_i(1-p_i)}, d_i, none \rangle$.

Now there is still the question of how the estimates of p_i and r_i are derived. This task involves the use of relevance information which can be gained via feedback, described in section 6.1.

4.3 Language models

The general language modeling retrieval function (cf. e.g. (Zhai and Lafferty, 2001)) is – admittedly – not in the linear form of equation 1. But using logarithms, products can be turned into sums without changing the ranking – the logarithm being a monotonic function (note that this is what also happened in the case of the probabilistic relevance models).

In particular, we will use the approach of comparing query and document language models by Kullback-Leibler divergence (KLD) (Lafferty and Zhai, 2001) which results in the equation

$$\begin{aligned} KLD(M_q || M_d) &= \sum_{t \in q} P(t|M_q) \log \frac{P(t|M_q)}{P(t|M_d)} \\ &\propto - \sum_{t \in q} P(t|M_q) \log P(t|M_d) \end{aligned}$$

where $P(t|M_q)$ and $P(t|M_d)$ refer to the probability that term t will be generated by the unigram language model of query q or document d , respectively. Note that we have simplified the equation by dropping a term $\sum_t P(t|M_q) \log P(t|M_q)$, which depends only on the query, not on the documents to be ranked.

Now, the triplet $\langle P(t|M_q), -\log P(t|M_d), t \rangle$ can be used to realise this retrieval function where t stands for a procedure that adds $-P(t|M_q) \log P(t|M_d)$ to the document node's activation level for terms t not occurring in d and sorts documents by *increasing* activation values afterwards.

5 Combining IR models

As can be seen from the last equation above, the language model retrieval function sums over all terms in the query. Each term – regardless of whether it

appears in the document d or not – contributes something that may be interpreted as a “penalty” for the document. The magnitude of this penalty depends on the smoothing method used (cf. (Zhai and Lafferty, 2001)). A popular smoothing method uses so-called Dirichlet priors to estimate document language models:

$$P(t|M_d) = \frac{tf + \mu p(t|C)}{\mu + |d|} \quad (4)$$

where tf is t ’s frequency in d , $p(t|C)$ is the term’s relative frequency in the whole collection and μ is a free parameter. This indicates that if a rare term is missing from a document, the penalty will be large, $P(t|M_d)$ being very small because $tf = 0$ and $p(t|C)$ small.

Conceptually, it is unproblematic to model the retrieval function by making I_{td} a complete bipartite graph, i.e. specifying a (non-zero) value for $P(t|M_d)$, even if t does not occur in d . In a practical implementation, this is not feasible, which is why we add the contribution of terms not contained in a document, i.e. $-P(t|M_q) \log P(t|M_d)$, for terms that do not occur in d .²

This transformation indicates an important difference between language modeling and all other IR models: language models *penalise* documents for the *absence* of rare (i.e. informative) terms whereas the other models *reward* them for the *presence* of these terms.

These considerations suggest a combination of both approaches: starting with an arbitrary “presence rewarding” model – e.g. the vector space model – we may integrate the “absence penalising” philosophy by subtracting from a document’s score, for each missing term, the contribution that one occurrence of that term would have earned (cf. (Witschel, 2006)).

For the vector space model, this yields the following retrieval function:

$$f(q, d) = \sum_{t \in q \cap d} w_{tq} w_{td} - \frac{\alpha}{|q|} \sum_{t \in q \setminus d} w_{td} (tf = 1) w_{tq}$$

²In order to do this, we only need to know $|d|$ and the relative frequency of t in the collection $p(t|C)$, i.e. information that is available *outside* the inverted list.

where α is a free parameter regulating the relative influence of penalties, comparable to the μ parameter of language models above.

5.1 Experimental results

Table 2 shows retrieval results for combining two weighting schemes, *BM25* (Robertson et al., 1992) and *Lnu.ltn* (Singhal et al., 1996), with penalties. Both of them belong to the family of *tf.idf* weighting schemes and can hence be regarded as representing the vector space model, although *BM25* was developed out of the probabilistic model.

Combining them with the idea of “absence penalties” works as indicated above, i.e. weights are accumulated for each document using the *tf.idf*-like retrieval functions. Then, from each score, the contributions that one occurrence of each missing term would have earned is subtracted. More precisely, what is subtracted consists of the usual *tf.idf* weight for the missing term, where $tf = 1$ is substituted in the *tf* part of the formula.

Experiments were run with queries from *TREC-7* and *TREC-8*. In order to study the effect of query length, very short queries (using only the title field of *TREC* queries), medium ones (using title and description fields) and long ones (using all fields) were used. Table 1 shows an example *TREC* query.

< top>
< num> Number: 441
< title> Lyme disease
< desc> Description:
How do you prevent and treat Lyme disease?
< narr> Narrative:
Documents that discuss current prevention and treatment techniques for Lyme disease are relevant [...]
< /top>

Table 1: A sample *TREC* query

Table 2 shows that both weighting schemes can be significantly improved by using penalties, especially for short queries, reaching and sometimes surpassing the performance of retrieval with language models. This holds even when the parameter α is not tuned and confirms that interesting insights are gained from a common representation of IR models in a graph-based environment.³

³Note that these figures were obtained without any refinements such as query expansion and are hence substantially

Weighting	TREC-7			TREC-8		
	very short	medium	long	very short	medium	long
BM25	0.1770	0.2120	0.2141	0.2268	0.2514	0.2332
+ P ($\alpha = 1$)	0.1867*	0.2194*	0.2178*	0.2380*	0.2593*	0.2335
+ P (best α)	0.1896*	0.2220*	0.2185*	0.2411*	0.2625*	0.2337
best α value	2	2	1.5	2	2	0.25
Lnu.ltn	0.1521	0.1837	0.1920	0.1984	0.2226	0.2013
+ P ($\alpha = 1$)	0.1714*	0.1972*	0.1946*	0.2176*	0.2305*	0.2040
+ P (best α)	0.1873*	0.2106*	0.1977	0.2394*	0.2396*	0.2064*
best α value	5	5	3	5	4	1.5
LM	0.1856	0.2163	0.2016	0.2505	0.2578	0.2307

Table 2: Mean average precision of BM25 and Lnu.ltn and their corresponding penalty schemes (+ P) for TREC-7 and TREC-8. Asterisks indicate statistically significant deviations (using a paired Wilcoxon test on a 95% confidence level) from each baseline, whereas the best run for each query length is marked with **bold** font. Performance of language models (LM) is given for reference, where the value of the smoothing parameter μ was set to the average document length.

6 Different forms of search with MLAGs

In order to complete the goals stated in the introduction of this paper, this section will briefly explain how feedback, associative retrieval and browsing can be modeled within the MLAG framework.

6.1 Feedback

Using the simple term-document MLAG of figure 1, feedback can be implemented by the following procedure:

1. Perform steps 1 – 4 of the basic processing.
2. Apply a transformation to the activation values of L_d -nodes, e.g. let the user pick relevant documents and set their activation to some positive constant β .
3. Perform step 3 of the basic processing with $L_i = L_d$ and $L_{i+1} = L_t$, i.e. let activation flow back to term level.
4. Forget about activation levels of documents.
5. Apply transformation on the term level L_t , e.g. apply thresholding to obtain a fixed number of expansion terms.
6. Spread activation back to the document level to obtain the final retrieval status values of documents.

lower than MAP scores achieved by systems actually participating in TREC.

In order to instantiate a particular feedback algorithm, there are three parameters to be specified:

- The transformation to be applied in step 2
- The weighting of document-term edges (if different from term-document edges) and
- The transformation applied in step 5.

Unfortunately, due to space constraints, it is out of the scope of this paper to show how different specifications lead to well-known feedback algorithms such as Rocchio (Rocchio, 1971) or the probabilistic model above.

6.2 Associative retrieval

Associative retrieval in MLAGs exploits the information encoded in level graphs: expanding queries with related terms can be realised by using the term level graph L_t of a simple MLAG (cf. figure 1) in step 2 of the basic processing, whereas the expansion of document result sets takes place in step 5 on the document level L_d . In order to exploit the relations encoded in the level graphs, one may again use spreading activation, but also simpler mechanisms. Since relations are used directly, dimensionality reduction techniques such as LSI cannot and need not be modeled.

6.3 Browsing

Since the MLAG framework is graph-based, it is easy to grasp and to be visualised, which makes it

a suitable data structure for browsing. The level graphs can be used as a flat graphical representation of the data, which can be exploited directly for browsing. Depending on their information need, users can choose to browse either on the term level L_t or on the document level L_d and they can switch between both types of levels at any time using the inverted list I_{td} . This applies, of course, also to passage or any other type of levels if they exist.

7 Conclusions

In this paper, a new graph-based framework for information retrieval has been introduced that allows to subsume a wide range of IR models and algorithms. It has been shown how this common representation can be an inspiration and lead to new insights and algorithms that outperform the original ones. Future work will aim at finding similar forms of synergies for the different forms of search, e.g. new combinations of feedback and associative retrieval algorithms.

References

- M. Agosti and F. Crestani. 1993. A methodology for the automatic construction of a hypertext for information retrieval. In *Proceedings of SAC 1993*, pages 745–753.
- J. R. Anderson and P. L. Pirolli. 1984. Spread of activation. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 10:791–799.
- M. Beaulieu. 1997. Experiments of interfaces to support query expansion. *Journal of Documentation*, 1(53):8–19.
- R. Brandow, K. Mitze, and L. F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31(5):675–685.
- S. Brin and L. Page. 1998. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of WWW7*, pages 107–117.
- F. Crestani. 1997. Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482.
- W. B. Croft and R. H. Thompson. 1987. I3R : a new approach to the design of document retrieval systems. *Journal of the american society for information science*, 38(6):389–404.
- L. B. Doyle. 1961. Semantic Road Maps for Literature Searchers. *Journal of the ACM*, 8(4):553–578.
- D. Hiemstra. 2001. *Using language models for information retrieval*. Ph.D. thesis, University of Twente.
- J. Lafferty and C. Zhai. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR 2001*, pages 111–119.
- D. Metzler and W. B. Croft. 2004. Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5):735–750.
- R. N. Oddy. 1977. Information retrieval through man-machine dialogue. *Journal of Documentation*, 33(1):1–14.
- C. Olston and E. H. Chi. 2003. ScentTrails: Integrating browsing and searching on the Web. *ACM Transactions on Computer-Human Interaction*, 10(3):177–197.
- S. E. Preece. 1981. *A spreading activation network model for information retrieval*. Ph.D. thesis, University of Illinois at Urbana-Champaign.
- M. Richardson and P. Domingos. 2002. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *Proceedings of Advances in Neural Information Processing Systems*.
- S. E. Robertson and K. Sparck-Jones. 1976. Relevance Weighting of Search Terms. *JASIS*, 27(3):129–146.
- S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. 1992. Okapi at TREC-3. In *Proceedings of TREC*, pages 21–30.
- J.J. Rocchio. 1971. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System : Experiments in Automatic Document Processing*. Prentice Hall Inc., Englewood Cliffs, New Jersey.
- G. Salton, A. Wong, and C. S. Yang. 1975. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.
- A. Singhal, C. Buckley, and M. Mitra. 1996. Pivoted document length normalization. In *Proceedings of SIGIR 1996*, pages 21–29.
- M. D. Smucker and J. Allan. 2006. Find-similar: similarity browsing as a search tool. In *Proceedings of SIGIR 2006*, pages 461–468.
- H. Turtle and W. B. Croft. 1990. Inference networks for document retrieval. In *Proceedings of SIGIR 1990*, pages 1–24.
- C. J. van Rijsbergen. 2004. *The Geometry of Information Retrieval*. Cambridge University Press.
- Scott White and Padhraic Smyth. 2003. Algorithms for estimating relative importance in networks. In *Proceedings of KDD 2003*, pages 266–275.
- H. F. Witschel. 2006. Carrot and stick: combining information retrieval models. In *Proceedings of DocEng 2006*, page 32.
- S. K. M. Wong and Y. Y. Yao. 1995. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68.
- C. Zhai and J. Lafferty. 2001. A study of smoothing methods for language models applied to Ad Hoc information retrieval. In *Proceedings of SIGIR 2001*, pages 334–342.