Inferring Query Models by Computing Information Flow

P.D. Bruza and D. Song
Distributed Systems Technology Centre
University of Queensland, 4072
Australia
{bruza, dsong}@dstc.edu.au

ABSTRACT

The language modelling approach to information retrieval can also be used to compute query models. A query model can be envisaged as an expansion of an initial query. The more prominent query models in the literature have a probabilistic basis, that is, for each term *w* in the vocabulary, the probability of *w*, given the query *Q*,, is computed. This paper introduces an alternative, non-probabilistic approach to query modelling whereby the strength of information flow is computed between the query *Q* and the term *w*. Information flow is a reflection of how strongly *w* is *informationally contained* within the query *Q*. In other words, the basis of the query model generation is information inference. The information flow model is based on Hyperspace Analogue to Language (HAL) vector representations, which reflects the lexical co-occurrence information of terms. Research from cognitive science has demonstrated the cognitive compatibility of HAL representations with human processing, and therefore HAL vectors would thus seem to be a potentially useful basis for inferring query expansion terms. Query models computed from TREC queries by HAL-based information flow are compared experimentally with two probabilistic query language models. Experimental results are provided showing the HAL-based information flow model be superior to query models computed via Markov chains, and seems to be as effective as a probabilistically motivated relevance model.

Main topics: Retrieval language models; Retrieval, Query expansion and fusion; Information Retrieval Theory

1. INTRODUCTION

Since the Cranfield experiments in document retrieval during the sixties, it has become well known that user queries to an information retrieval system are typically imprecise descriptions of the given information need. This phenomenon has been particularly emphasized with respect to queries on the web. Web queries average between two and three terms in length. Such short queries are, almost certainly, poor descriptions of the associated information need.

Various query expansion techniques have been developed in order to improve the initial query from the user. The goal of automatic query expansion is to automatically expand the user's initial query *Q* with terms related to the query terms in *Q* yielding a query Q'. The expanded query Q' is then used to return documents to the user. Various models and techniques have been proposed for determining the expansion terms.

Thesaurus-based techniques use a thesaurus, or ontology, as the source of expansion terms, for example, WordNet [22]. Global collection expansion techniques involve analyzing a collection of documents and computing term associations within the collection, for example, on the basis of term co-occurrence [1, 7, 9, 15]. An initial query is expanded by those terms strongly associated with query terms. Local collection expansion techniques generally follow a two-stage process, sometimes referred to as pseudo-relevance feedback. The initial query *Q* is issued to retrieve a ranked list of documents. The top *N* of these documents constitutes the local collection. Terms are extracted from these documents and used to expand *Q*, for example, local context analysis identifies terms within the neighbourhood of query terms present in documents of the local collection [19].

Other techniques involve implicit relevance feedback assuming all documents in the local collection to be relevant [3], or selecting terms from the local collection using various methods, for example, from simple term frequency to a probabilistic basis such as Robertson's Selection Value [16]. In both global and local techniques, the initial query can sometimes be massively expanded into a query of hundreds of terms [3].

The language modelling approach to information retrieval has allowed query expansion to be re-considered as a language modelling problem. More specifically, a query language model comprises estimating the probability $P(t \mid Q)$ of every term *t* in the vocabulary in the light of the initial query $Q = (q_{1,} \ldots, q_m)$. Intuitively, those terms *t* with probabilities above a threshold can be considered more useful candidate terms for expanding the initial query *Q*. There have been a number of promising approaches proposed for estimating query language models.

Lavrenko and Croft recently estimate the query language model in terms of a Relevance Model [12]. The query *Q* is considered to be a random sample from the unknown relevance model *R. R* can be envisaged as an unknown process from which words can be sampled, so if query terms $q_1, \ldots, q_m$ have been sampled, what is the probability of term *t* will be sampled next. Essentially this probability can be expressed in term of the probability of co-occurrence between *t* and *Q*, which is estimated by sampling the query terms from *t* via a number of unigram distributions M_i. The top 50 ranked documents retrieved by the query Q are used to serve as these distributions. They stated that "from a traditional IR perspective, our method is a massive query expansion technique".

In another approach to query language modelling, Lafferty and Zhai generate query language models using Markov chains [10]. Given a query model θ_q and a word *w*, the probability of expanding θ_q using w can be calculated as a language model $P(w|\theta_q)$, which is estimated using the Markov chain method according to the prior probability of *w* and the translation model for generating the query model from *w*. The Markov chain starts from the initial word *w* and the alternates between words and documents. For a given word, a document is selected according to its document language model. For the selected document, a word is then selected according to its posterior probability. The Markov chain lasts until a query term is selected or a limit of steps is reached.

Even though probabilistic approaches to query language modelling are promising, there are other points of departure. There is a growing body of research from cognitive science in which corpus-based representations of terms and concepts are being computed which correlate with human processing [4, 11, 13, 14]. One such model is Hyperspace Analogue to Language (HAL) [4, 13]. HAL represents words as vectors in a high dimensional space based on lexical co-occurrence. HAL is significant because the term associations computed by this model correlate with human judgements in word association tasks. In other words, HAL representations would seem to be a promising basis on which to compute term associations for query expansion.

We have recently proposed an information flow model based on HAL vectors [18]. The goal of this model is to produce information-based inferences which correlate with human inferences with regard to information. In essence, the HAL-based information flow model computes the degree to which term *j* is informationally contained/conveyed/carried by the terms $i_1, \ldots, i_m$. The theoretical basis of the information inference is drawn from Barwise and Seligman's account of information flow [2]. From a philosophical point of view, the model is in accord with the views of Gärdenfors [6] and Newby [14], who advocate a semiotic-cognitive stance with regard to information representation and processing (rather than a probabilistic one). If the terms $i_1, \ldots, i_m$ are query terms, then those terms *j* flowing informationally from these query terms can be considered as candidate query expansion terms. So, instead of a probabilistic foundation for the query language model via $P(t \mid Q)$, we propose a query language model based on the degree of information flow between the query *Q* and a vocabulary term *t*.

The goal of this paper is to use the HAL-based information flow model to compute a query language model and evaluate its effectiveness by comparing its performance with prominent probabilistic query language models. At a broader level, we are aiming to gain an initial picture of how information inference fares in relation to traditional probabilistic inference.

2. COMPUTING HAL- BASED INFORMATION FLOW

2.1 HAL- Hyperspace analogue to Language

A human encountering a new concept derives its meaning via an accumulation of experience of the contexts in which the concept appears. This opens the door to "learn" the meaning of a concept through how a concept appears within the context of other concepts. Following this idea, Burgess and Lund developed a representational model of semantic memory called Hyperspace Analogue to Language (HAL), which automatically constructs a high dimensional semantic space from a corpus of text [4,13]. The space comprises high dimensional vector representations for each term in the vocabulary. Given an *n*-word vocabulary, the HAL space is a *n* x *n* matrix constructed by moving a window of length *l* over the corpus by one word increment ignoring punctuation, sentence and paragraph boundaries. All words within the window are considered as co-occurring with each other with strengths inversely proportional to the distance between them. After traversing the corpus, an accumulated co-occurrence matrix for all the words in a target vocabulary is produced. Note that the word pair in HAL is direction sensitive, i.e. the co-occurrence information for words preceding every word and co-occurrence information for words following it are recorded separately by its row and column vectors. For the purposes of computing information flow, we have not found it useful to preserve the order information, so a term is represented by the addition of its row and column vectors in the HAL matrix.

The quality of HAL vectors is influenced by the window size; the longer the window, the higher the chance of representing spurious associations between terms. Burgess and Lund used a size of ten in their studies [4]. In the experiments reported below a window size of eight was used to construct the HAL matrix because in our previous work, this value tended to produce more precise representations of terms [18]. In addition, it is sometimes useful to identify the so called *quality properties* of a HAL-vector.

Intuitively, the quality properties of a concept or term *c* are those terms which often appear in the same context as *c*. Quality properties are identified as those dimensions in the HAL vector for *c* which are above a certain threshold (e.g., above the average weight within that vector). HAL vectors are normalized to unit length before information flow computation. For example, part of the normalized HAL vector for "*superconductors"* computed from a corpus of Associated Press news feeds is as follows:

superconductors = < U.S.:0.11 american:0.07 basic:0.11 bulk:0.13 called:0.15 capacity:0.08 carry:0.15 ceramic:0.11 commercial:0.15 consortium:0.18 cooled:0.06 current:0.10 develop:0.12 dover:0.06 electricity:0.18 energy:0.07 field:0.06 goal:0.06 high:0.34 higher:0.06 improved:0.06 japan:0.14 loss:0.13 low:0.06 make:0.07 materials:0.25 new:0.24 require:0.09 research:0.12 researching:0.13 resistance:0.13 retain:0.06 scientists:0.11 semiconductors:0.10 states:0.11 switzerland:0.06 technology:0.06 temperature:0.48 theory:0.06 united:0.10 university:0.06>

This example demonstrates how a word is represented as a weighted vector whose dimensions comprise other words. The weights represent the strength of association between "superconductors" and other words seen in the context of window: the higher the weight of a word, the more it has lexically co-occurred with "superconductors" in the same context(s).

In summary, a concept[1] c_i is a vector representation: $c_i = \left\langle w_{c_i p_1}, w_{c_i p_2}, \ldots w_{c_i p_n} \right\rangle$ where $p_1, p_2, \ldots, p_n$ are called dimensions of c_i, *n* is the dimensionality of the HAL space, and $w_{c_i p_i}$ denotes the weight of p_i in vector of c_i. A dimension is termed a property if its weight is greater than zero. A property p_i of a concept c_i is a termed quality property iff $w_{c_i p_i} > \partial$, where ∂ is a non-zero threshold value. Let $QP(c)$ denote the set of quality properties of concept *c*.

Combining concepts

Concept combination is important in IR, as combinations of words in a query topic may represent a single underlying concept, for example, *space program*. An important intuition in concept combination is that one concept can dominate the other. For example, the term "space" can be considered to dominate the term "program" because it carries more of the information in the phrase. Given two concepts $c_1 = \left\langle w_{c_1 p_1}, w_{c_1 p_2}, \ldots w_{c_1 p_n} \right\rangle$ and $c_2 = \left\langle w_{c_2 p_1}, w_{c_2 p_2}, \ldots w_{c_2 p_n} \right\rangle$, the resulting combined concept is denoted $c_1 \oplus c_2$. The following concept combination heuristic is essentially a restricted form of vector addition whereby quality properties shared by both concepts are emphasized, the weights of the properties in the dominant concept are re-scaled higher, and the resulting vector from the combination heuristic is normalized to smooth out variations due to differing number of contexts the respective concepts appear in.

Step 1: Re-weight c_1 and c_2 in order to assign higher weights to the properties in c_1.

$$w_{c_1 p_i} = \ell_1 + \frac{\ell_1 * w_{c_1 p_i}}{\underset{k}{Max}(w_{c_1 p_k})} \quad \text{and} \quad w_{c_2 p_i} = \ell_2 + \frac{\ell_2 * w_{c_2 p_i}}{\underset{k}{Max}(w_{c_2 p_k})}$$

$$\ell_1, \ \ell_2 \in (0.0, \ 1.0) \text{ and } \ell_1 > \ell_2$$

For example, if $\ell_1 = 0.5$ and ℓ_2=0.4, then property weights of c_1 are transferred to interval [0.5, 1.0] and property weights of c_2 are transferred to interval [0.4, 0.8], thus scaling the dimensions of the dominant concept higher.

[1] The term "concept" is used somewhat loosely; it can be envisaged as "term" in the traditional IR sense

Step 2: Strengthen the weights of properties appearing in both c_1 and c_2 via a multiplier α; the resultant highly weighted dimensions constitute significant properties in the resultant combination.

$$\forall(p_i \in QP(c_1) \wedge p_i \in QP(c_2)) \quad | \quad w_{c_1 p_i} = \alpha * w_{c_1 p_i},$$
$$w_{c_2 p_i} = \alpha * w_{c_2 p_i} \text{ , where } \alpha > 1.0$$

Step 3: Compute property weights in the composition $c_1 \oplus c_2$:

$$w_{(c_1 \oplus c_2) p_i} = w_{c_1 p_i} + w_{c_2 p_i}, 1 \le i \le n$$

Step 4: Normalize the vector $c_1 \oplus c_2$. The resultant vector can then be considered as a new concept, which, in turn, can be composed to other concepts by applying the same heuristic.

In order to deploy the information flow model in an experimental setting, the queries have to analysed for concept combinations. In particular, the question of which concept dominates which other concept(s) needs to be resolved. As there seems to be no reliable theory to determine dominance, a heuristic approach is taken in which dominance is determined by multiplying the query term frequency (*qtf*) by the inverse document frequency (*idf*) value of the query term. More specifically, query terms can re ranked according to *qtf*idf*. Assume such a ranking of query terms: $q_1, \ldots, q_{m.}$ (m > 1). Terms q_1 and q_2 can be combined using the concept combination heuristic described above resulting in the combined concept $q_1 \oplus q_2$, whereby q_1 dominates q_2 (as it is higher in the ranking). For this combined concept, its degree of dominance is the average of the respective *qtf*idf* scores of q_1 and q_2. The process recurses down the ranking resulting in the composed query "concept" $((..(q_1 \oplus q_2) \oplus q_3) \oplus \ldots) \oplus q_m)$. This denotes a single vector from which query models can be derived. If there is a single query term (m =1), it's corresponding normalized HAL vector is used for query model derivation.

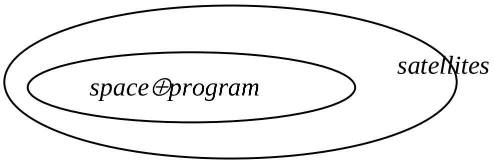
As it is important to weight query terms highly, the weights of query terms which appeared in the initial query were boosted in the resulting query model by adding 1.0 to their score. Due to the way HAL vectors are constructed, it is possible that an initial query term will not be represented in the resulting query model. In such cases, the query term was added with a weight of 1.0. Pilot experiments show that the boosting heuristic performs better than the use of only query models without boosting query terms.

2.2 Computing Information Flow

Barwise & Seligman have proposed an account of information flow that provides a theoretical basis for establishing informational inferences between concepts [2]. For example,

space, program |- *satellites*

illustrates that the concept "satellites" is carried informationally by the combination of the concepts "space" and "program". Said otherwise, "satellites" *flows* informationally from "space" and "program". Such information flows are determined by an underlying information state space. A HAL vector can be considered to represent the information "state" of a particular concept (or combination of concepts) with respect to a given corpus of text. The degree of information flow between "satellites" and the combination of "space " and "program" is directly related to the degree of inclusion between the respective information states represented by HAL vectors. Total inclusion leads to maximum information flow and can be visualised as follows:



Inclusion is a relation ⊆ over the concept space. For example, the above diagram is denoted by *space⊕program⊆ satellites.*

Definition 1 (HAL-based information flow)

$i_{1,} \ldots, i_n |- j$ iff degree($\oplus c_i \subseteq c_j$) > λ

where c_i denotes the conceptual representation of token *i*, and λ is a threshold value. (For ease of exposition, $\oplus c_i$ will be referred to as c_i because combinations of concepts are also concepts).

Note that information flow shows truly inferential character, i.e., concept *j* is not necessarily a dimension of the $\oplus c_i$. The degree of inclusion is computed in terms of the ratio of intersecting quality properties of c_i and c_j to the number of quality properties in the source c_i :

$$\text{degree}(c_i \subseteq c_j) = \frac{\sum_{p_l \in (QP(c_i) \wedge QP(c_j))} w_{c_i p_l}}{\sum_{p_k \in QP(c_i)} w_{c_i p_k}}$$

In terms of the experiments reported below, the set of quality properties $QP_i(c_i)$ in the source HAL vector c_i is defined to be all dimensions with non-zero weight (i.e., $\partial > 0$). The set of quality properties $QP_j(c_j)$ in the target HAL vector c_j is defined to be all dimensions greater than the average dimensional weight within c_j. These definitions for determining the quality properties in the source concept c_i and target concept c_j were determined via pilot studies in information flow computation.

2.3 Deriving query models via information flow

Given the query $Q = (q_{1,} \ldots, q_m)$, a query model can be derived from Q in the following way:

- Compute $\text{degree}(\oplus c_i \subseteq c_t)$ for every term *t* in the vocabulary, where $\oplus c_i$ represents the conceptual combination of the HAL vectors of the individual query terms $q_i, 1 \le i \le m$ and c_t represents the HAL vector for term *t*.
- The query model $Q' = \langle t_1 : f_1, \ldots, t_k : f_k \rangle$ comprises the top *k* information flows

Observe that the weight f_i associated with the term t_i in the query model is not probabilistically motivated, but denotes the degree to which we can infer t_i from *Q* in terms of underlying HAL space.

3. EXPERIMENTS

3.1 Experimental set-up

Two experiments to be reported here use the AP89 collection (disk 1) for TREC[2] topics 1 – 50, and the AP 88&89 collection (disks 1 and 2) using TREC topics 101-150 and 151-200. Only the titles of the topics were used as queries. We attempted to set up the experiment to allow comparison against the Markov chain and Relevance Models mentioned in the introduction. Table 1 summarizes the collection and query characteristics:

	Experiment 1	**Experiment 2**
Query set	Topics 1-50 (titles only)	Topics 101-150 and 151-200 (titles Only)
Average Query Length	3.24	3.8 and 4.5
Collection	AP89	AP88 & 89
Number of Documents	84, 678	164, 597
Size of Vocabulary	137, 728	249, 453

Table 1: Test collections and queries

HAL spaces were constructed from both collections using a window size of 8 words (*l* = 8). Stemming was not performed during HAL space construction.

The following query models were evaluated for their effectiveness:

HAL-based Information Flow Model (IM): This model was chosen to investigate whether information flow analysis contributes positively to query model derivation. IM is a global collection based model for query expansion. The top 85 information flows were used in the query model (*k*=85). This value produced best performance during a series of pilot studies.

Information Flow Model with pseudo-relevance feedback (IM w/pseudo): Pseudo-relevance feedback has consistently generated improved effectiveness. This model was implemented by constructing a high dimensional context (HAL) space by using the top fifty documents in response to a query, and thereafter deriving a query model deriving from this local collection. The fifty documents were retrieved by the baseline model. The top 60 information flows were used in the query model (*k*=60). This value produced the best performance during a series of pilot studies.

For each query topic, the query terms are combined using our concept combination heuristic into a single query vector ($\ell_1 = 0.5, l_2 = 0.3, \alpha = 2.0$). This vector is then used to derive a query model.

In the baseline model, documents are indexed using the document term frequency and inverse collection frequency components of Okapi BM25 formula [16] (parameters (k_1=1.2, k_2=0.0, *b*=0.75). Query vectors are produced using query term frequency with query length normalization [20], which is defined similarly to the BM25's document term frequency with parameter k3=1000.

[2] TREC stands for the Text Retrieval Conference series run by NIST. See trec.nist.gov

The matching function employed between document and query vectors was dot product as advocated by Lafferty and Zhai [10, 20].

Note that in the baseline, Markov Chain, and Relevance models terms were stemmed, whereas in the information flow models with and without pseudo feedback, terms were not stemmed as pilot studies revealed that information flow models perform slightly better without stemming.

3.2 Results: Experiment 1

This experiment evaluated the effectiveness of IM on the AP89 collection with TREC query topics 1-50. This experiment allows a direct comparison of the HAL-based information flow model with Lafferty & Zhai's Markov chain based query model both with and without pseudo-relevance feedback. These results are detailed in Table 2 and Figure 1. Note that these results do not include topic 47 which was omitted in Lafferty and Zhai's experiments [10].

3.3 Results: Experiment 2

In the second experiment, the information flow model is investigated in the context of the larger AP 88&89 collection using TREC topics 101-150,151-200. This experiment allows a performance comparison between the information flow model and Lavrenko and Croft's Relevance model [12]. The results are shown in Table 3, Figures 2 and 3.

3.4 Discussion

The first observation is the low baseline performance (average precision 0.185. .221, 0.296) for the three query topic sets. This is due to only the titles being used. The average precision scores using the corresponding TDN (title, description, narrative) queries are (0.269, 0.329, 0.343).

The results of experiment 1 suggest that the HAL-based information flow model outperforms the Markov chain query language model. It is notable that the information flow model without feedback outperforms the Markov chain model with feedback. (Note the baseline performance on both collections is poor due to only titles being used as queries.

In the results of Experiment 2 the comparison between the information flow model with pseudo-relevance feedback (M w/Pseudo) with the Relevance model is most pertinent as the Relevance model also uses feedback. (The top-ranked 50 documents are used as the sampling distributions). This comparison shows the HAL-based information flow model outperforms the Relevance model for both topic sets. Of notable interest is that the information flow model without feedback (IM) has similar performance to the Relevance model for query topics 101-150, and slightly inferior performance to the Relevance model for query topics 151-200. In other words, a global collection query expansion model is performing similarly to a local collection query expansion model. Experiences from the TREC conference series have consistently revealed that local collection expansion techniques outperform the global collection based techniques. In order to see if such a result was by accident, we compared the IM model (global collection model) with a pseudo-relevance feedback model based on term frequency using the top-ranked 40 documents retrieved by the baseline. The 30 most frequently occurring terms were selected for query expansion. The average precision scores for AP89-topics 1-50, AP88&89-Topics 101-150, AP88&89-topics 151-200 were 0.233, 0.294, and 0.335 respectively. The IM model's performance was (0.247, 0.265, 0.298). There is some evidence to suggest that the global collection based information flow model performs as well as, or near to, local collection based techniques.

A major disadvantage of the existing global techniques, for which it has been criticized, is its context insensitivity. For example, "program" may occur in different contexts such as "software", "postgraduate program", "space program", and so on. All these contexts exist in the same "program" vector constructed via term co-occurrence methods. We have addressed this problem in our model by introducing concept combination and information flow inference. When "program" appears in the context of "space", the related dimensions like "NASA" and "defense" will be enhanced by combining "space" and "program" using our concept combination heuristic. Those irrelevant dimensions such as "postgraduate" will be accordingly eliminated or adjusted with a lower weights. Moreover, the information flow analysis allows true inference. Terms, for example, "satellites", which are not dimensions present in the composed query vector for $space \oplus program$, can be inferred and then used for query expansion.

The IM model comprises two components: HAL representations, and information inference. In order to understand where the effectiveness of the IM is originating from, a further experiment was carried out which evaluated the effectiveness of the IM model without the inference component on the AP89 collection using topics 1-50. In other words, query topics were translated into query vectors as before, normalized, and used for retrieval. The average precision achieved was 0.197. This represents an 8% improvement over the baseline model. The IM model's average precision scored 0.247 which represents a 35% improvement over the baseline. Similarly with respect to recall the model without the inference component produced a 19% improvement (1996/3301 relevant documents retrieved) versus a 35% improvement in recall of the IM model (2269/3301). These figures suggest that the inference component contributes more to precision than the underlying HAL representation.

The performance increases of the IM model over the baseline were not as marked using query topics 101-150. As these topics are on average longer than the other topic sets, it may be the case that the translation heuristic of topics via concept combinations may not be as effective for longer queries. Further experimentation is needed to bear this out.

The improvement of the IM model with pseudo-feedback over the IM model without pseudo-feedback, parallels the general trend of local collection based query expansion techniques outperforming global collection based query expansion techniques. It is interesting that the improvements are not as marked as the improvement registered by the Markov model.

4. RELATED WORK

Schütze and Pedersen apply a singular value decomposition (SVD) algorithm to the term co-occurrence matrix, which is produced by moving a k-word sliding window, for dimensional reduction [17]. SVD is also used in the Latent Semantic Analysis (LSA) or more specifically to IR, Latent Semantic Indexing (LSI), to reduce the number of dimensions of a term-passage matrix [5, 8, 11]. It performances some reasonable inductions, i.e. some words not occurring in a passage could be inferred and the weights of some originally occurring terms are changed. Therefore, SVD and the Information Flow model seem to be complementary but address different aspects of the informational inference problem. SVD has a strong mathematical basis: Its inferential character comes from the matrix decomposition and the dimension reduction. The information flow model is based on HAL vectors and a calculation of the degree of inclusion between vectors to compute the strength of inference.

5. CONCLUSIONS

This paper compares the effectiveness of query models derived from information flow computations on vector representations of terms produced by Hyperspace Analogue to Language (a model of information representation from cognitive science) with two prominent probabilistic language models. More specifically, the information flow model outperforms a Markov chain based query language model and a relevance-based query language model. In addition, the HAL-based information flow model performs well using both local and global collection-based expansion.

The information flow approach presented here differs from the probabilistic approaches to query language models in the following ways:

- ❑ Query terms are not considered independent. HAL-based vector representations embody associations between terms, which are context sensitive. Moreover, information flow analysis involves concept combination which melds individual vector representations of query terms, thus not treating them separately.

- ❑ The degree of information flow (or informational inference) between term *w* and query *Q* is used weight the words in the query model. These weights reflect how strongly *Q* is informationally contained in *w*, rather than a conditional probability.

The information flow model comprises two components. The first is the HAl-based representation, the second is an information inference component. The improvements in average precision appear to arise primarily from the information inference component. This suggests that further research should be directed to tune this component. In addition, investigating the inference component in tandem with alternative vector representations of terms, for example, latent semantic analysis, is an interesting avenue for further exploration.

ACKNOWLEDGEMENTS

The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of Australia.

	Baseline	IM	Markov Chain	IM w/ pseudo	Markov Chain w/pseudo
AvgPr	0.185	0.251	0.201	0.263	0.232
InitPr	0.479	0.559	0.500	0.544	0.534
Recall	1650/3261	2236/3261	1745/3261[3]	2298/3261	2019/3261

Table 2: Comparison of various query models for the AP89 collection using TREC topics 1-50 (titles), but not including topic 47

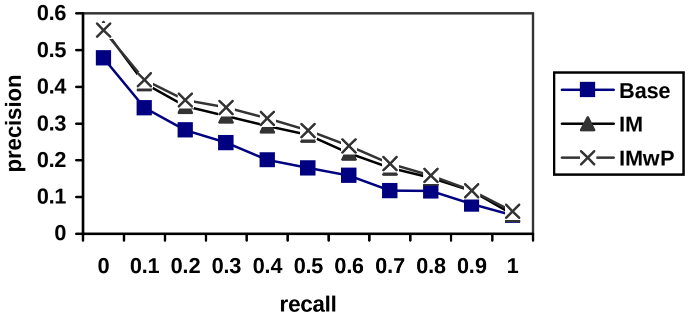


Figure 1: Precision recall curves comparing Baseline with the information flow model (with/without feedback) for the AP89 collection using TREC topics 1-50 (titles)

Topics		Baseline	IM	IM w/pseudo	Relevance Model
101-150	**AvgPr**	0.221	0.265	0.301	0.262
	InitPr	0.616	0.587	0.623	0.616
	Recall	3183/4805	3456/4805	3822/4805	3733/4805
151-200	**AvgPr**	0.296	0.298	0.344	0.318
	InitPr	0.731	0.655	0.703	0.725
	Recall	3348/4933	3125/4933	3446/4933	3222/4933

Table 3: Comparison of query models on the AP collection and topics 101-150 and 151-200 (titles)

[3] .By adding query topic 47, the AvgPr, InitPr and Recall of baseline, IM and IM w/pseudo are (0.183, 0.475, 1683/3301), (0.247, 0.554, 2269/3301) and (0.258, 0.554, 2331/3301).

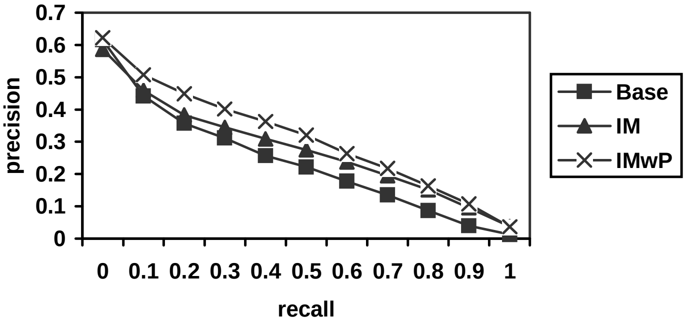


Figure 2: Precision-recall curves comparing the Baseline with Information Flow model (with/without) feedback on the AP collection, TREC topics 101-150 (titles)

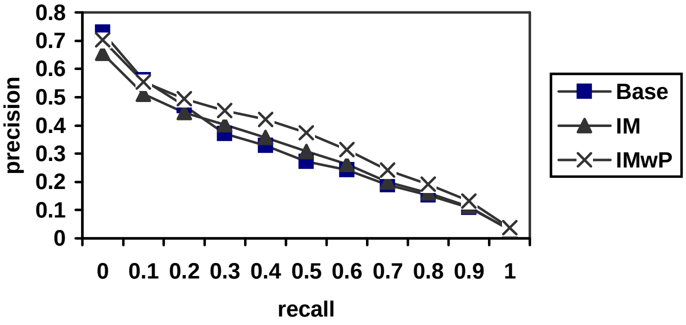


Figure 3: Precision-recall curves comparing the Baseline with Information Flow model (with/without) feedback for the AP collection, TREC topics 151-200 (titles)

REFERENCES

[1] Allan, J., Ballasteros, L., Callan, J.P., Croft, W.B., Lu, Z. (1996) Recent Experiments with INQUERY. In D.K. Harman (ed) *Proceedings of the 4th Text Retrieval Conference (TREC-4).*

[2] Barwise, J. and Seligman, J. (1997) *Information Flow: The Logic of Distributed Systems.* Cambridge Tracts in Theoretical Computer Science, 44.

[3] Buckley, C., Salton, S., Allan, J., Singhal, A. (1995) Automatic Query Expansion Using SMART : TREC 3. In D.K. Harman (ed) *Proceedings of the 3rd Text Retrieval Conference (TREC-3).*

[4] Burgess, C., Livesay, K. and Lund K. (1998) Explorations in Context Space: Words, Sentences, Discourse. *Discourse Processes, 25(2&3), 211-257.*

[5] Deerwester, S., Dumais, S.T., Furnas, G.W. and Landauer, T.K. (1990). Indexing by Latent Semantic Analysis. *Journal of American Society for Information Science.* 41(6): 391-407.

[6] Gärdenfors, P. (2000) *Conceptual Spaces: The Geometry of Thought.* MIT Press.

[7] Gauch, S. and Wang, J. (1997) A Corpus Analysis Approach for Automatic Query Expansion. *In Proceedings of the 6th International Conference on Information and Knowledge Management (CIKM'97),* pp. 278-284.

[8] Hofmann T. (1999) Probabilistic Latent Semantic Indexing. In Proceedings of the *22nd Annual International Conference on Research and Development in Information Retrieval (SIGIR'99*), pp. 50-57.

[9] Jing, Y. and Croft, W.B. (1994). An Association Thesaurus for Information Retrieval. In *Proceedings of the Intelligent Multimedia Information Retrieval Systems (RIAO'94),* pp. 146-160.

[10] Lafferty, J and Zhai, C. (2001) Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In Proceedings of the *24th Annual International Conference on Research and Development in Information Retrieval (SIGIR'01),* pp. 111-119.

[11] Landauer, T.K., Foltz, P.W., and Laham D. (1998) An Introduction to Latent Semantic Analysis. *Discourse Processes, 25(2&3), 259-284.*

[12] Lavrenko, V. and Croft, W.B. (2001) Relevance-Based Language Models. In Proceedings of the *24th Annual International Conference on Research and Development in Information Retrieval (SIGIR'01), pp. 120-127.*

[13] Lund, K. and Burgess C. (1996) Producing High-dimensional Semantic Spaces from Lexical Co-occurrence. *Behavior Research Methods, Instruments, & Computers, 28(2), 203-208.*

[14] Newby, G. B. (2001) Cognitive Space and Information Space. *Journal of the American Society for Information Science, 52(12), 1026-1048.*

[15] Qui, Y. and Frei, H.P. Concept Based Query Expansion. In Proceedings of the *16th Annual International Conference on Research and Development in Information Retrieval (SIGIR'93), pp. 160-169.*

[16] Robertson, S.E., Walker, S., Spark-Jones, K., Hancock-Beaulieu, M.M., and Gatford, M. (1995) OKAPI at TREC-3. In *Proceedings of the 3rd Text Retrieval Conference (TREC-3).*

[17] Schütze, H. and Pedersen, J. O. (1997) A Cooccurrence-based Thesaurus and Two Applications to Information Retrieval. *Information Processing and Management, 33(3),* pp. 307-318.

[18] Song, D. and Bruza, P.D. (2001) Discovering Information Flow Using a High Dimensional Conceptual Space. In Proceedings of the *24th Annual International Conference on Research and Development in Information Retrieval (SIGIR'01*), pp. 327-333.

[19] Xu, J. and Croft, W.B. (2000) Improving the Effectiveness of Information Retrieval with Local Context Analysis. *ACM Transactions on Information systems,* Vol. 18, No. 1, pp. 79-112.

[20] Zhai, C. (2001). Notes on the Lemur TFIDF model. Unpublished report.

[21] Voorhees, E. and Harman, D. (1998). Overview of the Sixth Text Retrieval Conference (TREC-6). In *Proceedings of the 3rd Text Retrieval Conference.*

[22] Voorhees, E. and Hou, Y. W. (1992) Vector Expansion in a Large Collection. *In Proceedings of TREC-1,* pp.343-351.