CHAPTER 5

CONCEPT BASED QUERY EXPANSION

5.1 INTRODUCTION

Language Modelling (LM) is used in many IR studies producing good experimental results (comparable to best IR systems), and also a solid theoretical setting. Classical LM approaches are not dependent between indexing units, which are unigrams or bigrams. A word can be related to other words. An example is synonymy relationship as such intra-term relationships should be integrated into LM.

Present information retrieval systems like web search engines consist of a standard interface of a single input box which accepts keywords. User submitted key words are matched against a collection index to locate documents with those keywords, sorted out by various methods. When a user query has multiple topic-specific keywords accurately describing needed information, the system will return good matches; but as user queries are short with inherently ambiguous natural language, this retrieval model is prone to errors/omissions.

A critical language issue for effective retrieval is term mismatch problem: indexers and users do not use similar words usually. This is a vocabulary problem (Furnas et al 1987), compounded by synonymy and polysemy. Synonymy with word inflections (plural forms), fail to retrieve relevant documents, with a recall decrease (the system’s ability to retrieve
relevant documents). Polysemy which are different words with same/similar meanings can lead to erroneous or irrelevant document retrieval, thereby implying a precision decrease (system’s ability to retrieve relevant documents). To offset this issue many approaches were proposed including relevance feedback, search results clustering, interactive query refinement and word sense disambiguation. A successful technique is expanding original query with other words which capture best user intent or produce a useful query, i.e., a query that will retrieve relevant documents. Automatic query expansion (AQE) has an IR history suggested by Maron and Kuhns (1960) which investigated many seminal techniques that were later improved and extended in ways like vector feedback (Rocchio 1971; Ide 1971), term-term clustering (Lesk 1969; Harper and Rijsbergen 1978), and comparative analysis of term distributions (Doszkocs 1978; Porter 1982). Many small scale experiments resulted in inconclusive results about retrieval effectiveness of techniques, with gain in recall compensated by corresponding precision loss.

Query expansion methods were long studied - with debatable success on many occasions. This study presents a probabilistic query expansion model based on similarity thesaurus constructed automatically. The latter reflects domain knowledge about specific collections from which it is constructed. Two important issues with query expansion are addressed here: selection and weighting of additional search terms. Compared to earlier methods, queries are expanded by addition of terms similar to query concept, instead of selecting terms similar to query terms.

5.2 QUERY EXPANSION

As data volume dramatically increased while searcher supplied query terms remained very low, it led to a revamp of research on Query Expansion. Web search is a case in point. According to Hitwise
average query length was of 2.30 words in 2009, same as that reported ten years earlier in (Lau and Horvitz 1999). While there was a slight increase in long queries numbers (of five or more words), most prevalent queries were still those of one, two, or three words. Thus, the vocabulary problem is still more serious as query terms paucity reduces chances of handling synonymy while data heterogeneity and size make polysemy effects more severe.

The need/scope of Automatic Query Expansion (AQE) increased. Recently, many AQE techniques were presented with varied approaches leveraging many data sources and using sophisticated methods to find query term correlated new features. Now, there are firm theoretical foundations and better understanding of AQE utility and limitations; e.g., critical parameters affecting method performance, what type of queries AQE is useful for, etc. Simultaneously, basic techniques are increasingly used along with other mechanisms to increase effectiveness, including method combination, active selection of information sources, and method application’s discriminative policies. These scientific advances were corroborated by positive experimental findings in laboratory settings. In fact, AQE regained popularity due to evaluation results from Text REtrieval Conference series (TREC,http://trec.nist.gov/), where most participants who used this technique reported noticeable retrieval performance improvement.

AQE is now a promising technique to improve document ranking retrieval effectiveness and that it is being adopted in commercial applications, mainly in desktop and intranet searches. For instance, Google Enterprise, My SQL and Lucene provide users with AQE facility capable of being turned on/off. It is yet to be regularly used in major operational web IR systems like search engines.
There are many reasons for AQE’s limited uptake in web search. First, fast response times needed by web search applications prevent use of computationally expensive AQE techniques. Second, current AQE techniques though optimized to perform well on average, are unstable and cause degradation of search service in some queries. Also AQE emphasis on improving recall (as opposed to guaranteeing high precision) is less important, as there are relevant documents with many users looking only at first and last page for results. Third, AQE acceptance is an issue, due to limited usability and IR system transparency when implementing AQE: the user may be confused when system retrieves documents without original query terms. But such features are less important in many IR applications (search by experts in specialized domains), where a straightforward AQE application has no major contraindications.

5.2.1 Applications of AQE

The different applications of AQE are

5.2.1.1 Question answering

Question answering (QA) goal is providing concise responses (instead of full documents) to some natural language questions like “What are the different forms of worship in India?” Similar to document ranking, QA faces a fundamental mismatch problem between question and answer vocabularies.

5.2.1.2 Multimedia information retrieval

With digital media and libraries increasing, multimedia documents (e.g., speech, image, and video) search is widely used. Generally, multimedia IR systems use text-based search over media metadata like annotations,
captions, and surrounding html/xml descriptions. When metadata is absent, IR relies on some multimedia form of content analysis, combined with AQE techniques.

5.2.1.3 Information filtering

Information filtering (IF) monitors a stream of documents, selecting those relevant to a user. Documents arrive continuously and user’s information need evolves over time. Examples of filtering application domains are blogs, e-mail, e-commerce, and electronic news.

5.2.1.4 Cross-language information retrieval

Cross-Language Information Retrieval (CLIR) retrieves documents written in a language other than the user’s query language.

5.2.1.5 Other applications of AQE

Other AQE applications include

(i) Text categorization ((Zelikovitz and Hirsh 2000), (Hidalgo et al 2005)),

(ii) Search of hidden web content not indexed by standard search engines (Graupmann et al 2005),

(iii) Mobile devices query completion (Kamvar and Baluja 2007),

(iv) e-commerce (Chen, et al., 2004),

(v) Mobile search(Church and Smyth 2007),

(vi) Training corpora acquisition (Huang, et al., 2005; Perugini and Ramakrishnan 2006),

(vii) Expert finding (Macdonald and Ounis 2007),
(viii) Federated search (Shokouhi et al 2009),
(ix) Paid search advertising (Wang et al 2009; Broder et al 2009), and
(x) Slot-based document retrieval (Suryanto et al 2007)

5.2.2 AQE Process

AQE is broken into 4 steps as shown in Figure 5.1: data source preprocessing; candidate expansion features generation and ranking; expansion features selection and query reformulation.

![Figure 5.1 Main steps of automatic query expansion](image)

Figure 5.1 Main steps of automatic query expansion

Data source preprocessing transforms raw data source used to expand user query into a format more effectively processed by subsequent steps. It consists of a phase of intermediate features extraction, followed by appropriate data structures construction for easy access to and their manipulation. Data source preprocessing is independent of a particular user query to be expanded but specific to data source type and expansion method under consideration. Many query expansion techniques are based on information contained in top-ranked items retrieved in response to document
collection’s original user query. To compute initial retrieval run, the collection should be indexed and then run query against it.

In second AQE stage, system generates and ranks candidate expansion features. The reason for its importance is that query expansion methods choose only a limited proportion of candidate expansion features to add to query.

Original query is the input to this stage and the transformed data source; output is an expansion features set, usually with associated scores. Original query is preprocessed to remove common words and/or extract important terms for expansion (importance being approximated e.g., by their inverse document frequency).

After candidate features ranking, top elements are chosen for query expansion. Selection is on individual basis, without considering mutual dependencies between expansion features. This is a simple assumption though there are experimental results that suggest that independence assumption is justified (Lin and Murray 2005). Usually limited features are selected for expansion, partly as resulting query can be processed rapidly and partly because retrieval effectiveness of a small set of good terms is not less successful than combining candidate expansion terms, due to noise reduction; (Salton and Buckley 1990; Harman 1992).

The last AQE step is query reformulation; how to describe expanded query to be submitted to the IR system, usually amounting to assigning a weight to each feature describing expanded query – termed query reweighting. One of the commonly used query reweighting technique is modeled after Rocchio’s formula for relevance feedback (Rocchio 1971) and later improvements (Salton and Buckley 1990), adapted to an AQE setting.
5.2.3 Methods of AQE

Automatic query expansion/modification based on term data co-occurrence was studied for nearly thirty years. Various methods in literature are classified in the following four groups:

(i) **Simple co-occurrence data use**: Similarities between terms are calculated based on association hypothesis and used to classify terms through a similarity threshold value (Lesk 1969; Minkar et al 1972). Thus, an index terms set is subdivided into similar terms classes. A query is expanded by adding all classe’s terms containing query terms. The idea of term classification into classes and treating members of same class as equivalent was too naive an approach to be good (Sparck-Jones 1991).

(ii) **Document classification use**: Documents are classified through a document classification algorithm. Infrequent terms in a document class are taken to be similar and clustered in same term class (thesaurus class) (Crouch 1990). Indexing of documents and queries is enhanced by replacing a term by a thesaurus class or adding a thesaurus class to index data. Retrieval effectiveness depends on parameters which are hard to determine (Crouch and Yong 1992). Further, commercial databases have millions of documents all of which are highly dynamic. Document number is larger than terms number in the database. Hence, document classification is more expensive and should be undertaken more often than simple term classification.

(iii) **Syntactic context use**: Term relations are generated based on linguistic knowledge and co-occurrence statistics (Grefenstette
1992; Ruge 1992) using grammar and a dictionary to extract a list of terms for each term t. This contains all terms that modify t. Similarities between terms are calculated using modifiers from the list. Then a query is expanded by adding terms similar to any query term producing only slightly better results than using original queries (Grefenstette 1992).

(iv) Relevance information use: Relevance information constructs a global information structure, a pseudo thesaurus (Salton and Buckley 1980) or a minimum spanning tree (Smeaton and Rijsbergen 1983). A query is expanded by this global information structure. This method’s retrieval effectiveness depends on user relevance information. Moreover, experiments failed to yield consistent improvement in performance. But, direct use of relevance information by extracting terms from relevant documents was effective in interactive information retrieval (Harman 1992; Sal1990). This approach does not help queries without relevance information. Semi-automatic query expansion was also studied (Hancock-Beaulieu 1992). In contrast to fully automated methods, user is involved in selecting additional search terms in a semi-automatic expansion process. A list of candidate terms is computed by one of the above methods and presented to the user who finally decides. Semi-automatic query expansion experiments did not result in major improvement in retrieval effectiveness (Ekmeckioglu et al 1992).

Among many approaches, automatic query expansion through using plain co-occurrence data is the simplest. Compared to present approaches, a
similarity thesaurus (Schäuble and Knaus 1992) is the basis of query expansion. First the similarity thesaurus construction is shown and then a query expansion model is presented to overcome drawbacks in using plain co-occurrence data.

5.3 DOCUMENT RANKING WITH AQE

IR systems and search engines mostly rely on computing terms importance in the query and in documents to determine answers. Similarity sim(q, d) between query q and document d can be expressed as

\[ sim(q, d) = \sum_{t \in q \cap d} w_{t, q} w_{t, d} \]  

(5.1)

where \( w_{t, q} \) is weight of term t in query q and \( w_{t, d} \) is weight of term in document d, respectively, according to system’s weighting function. A term’s weight is proportional to term frequency and inversely proportional to frequency and documents length having the term. This formulation accounts for several ranking models that are directly or indirectly traced it, including probabilistic relevance model (Robertson, et al., 1998), vector space model (Salton and McGill 1983), statistical language modelling (Zhai and Lafferty, 2001), and deviation from randomness (Amati et al 2001).

The formula of ranking scheme is modified to accommodate query expansion, abstracting from specific underlying weighting model. The AQE input consists of original query q and a data source which computes and weights expansion terms. AQE output is a query q’ formed by an expanded terms set with associated weights w’. New weighted query terms compute similarity between query q’ and document d:

\[ sim(q', d) = \sum_{t \in q' \cap d} w'_{t, q'} w_{t, d} \]  

(5.2)
Typical data sources to generate new terms is a collection being searched itself and simplest way to weight query expansion terms is by using weighting function of the ranking system. If complex features than single terms are used for query expansion (phrases), underlying ranking must handle such features.

5.4 PROPOSED CONCEPT BASED QUERY EXPANSION

A similarity thesaurus (Schäuble and Knaus 1992) is a matrix consisting of term-term similarities. Compared to a co-occurrence matrix, similarity thesaurus is based on how collection terms are “indexed” by documents. It is shown that a similarity thesaurus is built automatically using arbitrary retrieval with documents roles and terms interchanged. Terms play role of retrievable items and documents constitute their "indexing features.”

With this, a term $t_i$ is represented by a vector in document vector space (DVS) defined by all documents of the collection. The $d_{ik}$'s signify feature weights of the indexing features of documented with respect to the term $t_i$ and $n$ being the collection’s features (documents) number. The normalized tf.idf weighting scheme (Salton and Buckley 1988) is adopted and defines feature weights $d_{ik}$ by feature frequency (ff), inverse item frequency (iif), and maximum feature frequency (maxff) as follows.

$$d_{ik} = \frac{(0.5 + 0.5 \frac{ff(d_k, t_i)}{\max ff(t_i)})iif(d_k)}{\sqrt{\sum_{j=1}^{n}((0.5 + 0.5 \frac{ff(d_k, t_j)}{\max ff(t_j)})iif(d_j))^2}}$$

(5.3)

where $ff(d_k, t_i)$ is the within-item frequency of feature $d_k$ in term $t_i$.

$$iif(d_k) = \log\left(\frac{m}{|d_k|}\right)$$ is the inverse item frequency of feature $d_k$. 

m is the number of items in the collection and $|d_k|$ is number of different items indexed by feature $d_k$.

$|d_k|$ is the number of terms appearing in document $d_k$. $\text{maxf}(t_i)$ is maximum within-item frequency of features in item $t_i$.

Feature frequency $ff(d_k, t_i)$ specifies the occurrences number of $d_k$ in $t_i$. It is analogous to term frequency $\text{tf}(t_i, d_k)$ when documents are indexed by terms. Definition of inverse item frequency reveals that a short document has an important role than a longer one. When two terms co-occur in a long document, the probability that two terms are similar is lesser than if they co-occurred in a short document. It is then derived that

$$|t_i| = \sqrt{\sum_{k=1}^{n} d_{ik}^2} = 1$$  \hspace{1cm} (5.4)

Thus, $t_i$ is a unit vector representing term in document vector space DVS. These definitions define similarity between two terms $t_i$ and $t_j$ through using a similarity measure like a simple scalar vector product:

$$SIM(t_i, t_j) = t_i^T t_j = \sum_{k=1}^{n} d_{ik} d_{jk}$$  \hspace{1cm} (5.5)

Similarity thesaurus is built through determining similarities of all term pairs $(t_i, t_j)$, resulting in a symmetric matrix whose values are in the following range:

$$0 \leq SIM(t_i, t_j) \leq 1$$

A query $q$ is represented by a vector $q = (q_1, q_2, ..., q_m)^T$ in term vector space (TVS) defined by all collection terms. Here, the $q_i$'s are weights
of the search terms \( t_i \) contained in query \( q \); \( m \) is total number of collection terms.

Probability that term \( t \) is similar to query concept \( q \) is \( P(S|q,t) \).

Bayes' theorem is applied to estimate probability:

\[
P(S|q,t) = P(S|t) \frac{P(q|S,t)}{P(q|t)} - \frac{P(S|t)}{P(q|t)} P(q|S,t)
\]  

(5.6)

It is assumed that terms distribution in all queries which has a similar term is independent:

\[
P(S|q,t) = \frac{P(S|t)}{P(q|t)} \prod_{i=1}^{m} P(q_i|S,t)
\]

\[
= \frac{P(S|t)}{P(q|t)} \prod_{i=1}^{m} \frac{P(S|q_i,t)}{P(S|t)} P(q_i|t)
\]

\[
= \frac{1}{P(q|t).P(S|t)} \prod_{i=1}^{m} P(S|q_i,t)P(q_i|t)
\]  

(5.7)

Another assumption is that similarity between a term and query concept depends on terms in the query and no other terms. Hence,

\[
P(S|q,t) = \frac{1}{P(q|t).P(S|t)} \prod_{i=1}^{m} P(S|t_i,t)P(t_i|t)
\]  

(5.8)

\( P(S|t_i,t) \) is the probability that query term \( t_i \) is similar to term \( t \). \( P(t_i|t) \) is probability that query term \( t_i \) represents query \( q \). \( P(q|t) \) is the probability that query \( q \) will be submitted to IR system. \( P(S|t) \) is probability that term \( t \) is similar to an arbitrary query.

The probability of a term being similar to a query is based on factors which follow:
(i) Similarities between term and all query terms;

(ii) Weights of query terms.

The objective of this query expansion scheme is to locate suitable additional query terms having properties similar to the entire query rather than individual query terms. It was shown that such terms are found when overall similarity scheme is considered. As similarity thesaurus expresses similarity between collection terms in DVS (defined by collection documents), vector $q$ is mapped from TVS (defined by collection terms) into vector in space DVS. Thus, overall similarity between a term and query is estimated. Every query term $t_i$ is defined by unit vector $\vec{t}_i$ which in turn is defined by a documents number. $q_i$ is weight of term $t_i$ in query. The concept expressed by term $t_i$ in query has importance of $q_i \cdot t_i$ for query. It is assumed that concept expressed by the whole query depends on query terms alone. Hence, vector $q_c$ representing query concept in DVS space is virtual term vector:

$$q_c = \sum_{t_i \in q} q_i \vec{t}_i \quad \quad (5.9)$$

Similarity between a term and query $q$ is denoted by $\text{Sim}_{qt}(q,t)$. Scalar vector product is used as similarity measure:

$$\text{Sim}_{qt}(q,t) = q_c^T \vec{t}_i = \left( \sum_{t_i \in q} q_i \vec{t}_i \right)^T \vec{t}_i$$

$$= \sum_{t_i \in q} q_i \cdot (\vec{t}_i^T \vec{t}_i) \quad \quad (5.10)$$

where $(\vec{t}_i^T \vec{t}_i)$ is the similarity between two terms.
The Reuters dataset and MovieLens dataset is used for evaluating the proposed methods. The experiments are conducted as detailed in the previous chapter, with the inclusion of proposed concept query expansion method. Precision values for various techniques for MovieLens dataset and Reuters dataset is evaluated. The techniques used were tdf.idf, Language modelling using query likelihood, proposed concept expansion and proposed DNLM with the proposed feature selection method. The experimental results for MovieLens dataset for precision and F measure are tabulated in Table 5.1 and Table 5.2 respectively. Figure 5.2 and 5.3 show the same.

Table 5.1 Precision values for various techniques for MovieLens dataset

<table>
<thead>
<tr>
<th>Recall</th>
<th>TDF-IDF</th>
<th>Concept expansion</th>
<th>Language modelling using Query likelihood</th>
<th>DNLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.736205539</td>
<td>0.788517548</td>
<td>0.76768341</td>
<td>0.862264349</td>
</tr>
<tr>
<td>0.1</td>
<td>0.674231856</td>
<td>0.745941259</td>
<td>0.724450666</td>
<td>0.813317659</td>
</tr>
<tr>
<td>0.2</td>
<td>0.654106968</td>
<td>0.673295989</td>
<td>0.683480201</td>
<td>0.736996708</td>
</tr>
<tr>
<td>0.3</td>
<td>0.652870386</td>
<td>0.666211395</td>
<td>0.656261048</td>
<td>0.728296238</td>
</tr>
<tr>
<td>0.4</td>
<td>0.600086141</td>
<td>0.639949659</td>
<td>0.619683593</td>
<td>0.699085423</td>
</tr>
<tr>
<td>0.5</td>
<td>0.587566501</td>
<td>0.62820501</td>
<td>0.615208626</td>
<td>0.686306294</td>
</tr>
<tr>
<td>0.6</td>
<td>0.541628181</td>
<td>0.632310059</td>
<td>0.604541939</td>
<td>0.690827911</td>
</tr>
<tr>
<td>0.7</td>
<td>0.510531962</td>
<td>0.599050154</td>
<td>0.586735683</td>
<td>0.654412245</td>
</tr>
<tr>
<td>0.8</td>
<td>0.473339763</td>
<td>0.570691382</td>
<td>0.559262151</td>
<td>0.62271193</td>
</tr>
<tr>
<td>0.9</td>
<td>0.463725261</td>
<td>0.530560081</td>
<td>0.497772722</td>
<td>0.579615886</td>
</tr>
<tr>
<td>1</td>
<td>0.263466017</td>
<td>0.451463061</td>
<td>0.419494071</td>
<td>0.373715098</td>
</tr>
</tbody>
</table>
From the Figure 5.2 it is seen that the precision values for DNLM with the proposed concept query expansion is higher than the precision values for tdf.idf by 17% when recall is 0.01 and 25% when recall is 0.9. DNLM with the proposed concept query expansion performs better than concept expansion by 9.3% when recall is 0.01 and by 9.2% when recall is 0.9. Similarly for Language modelling using query likelihood, DNLM with the proposed concept query expansion performs better by 12.3% when recall is 0.01 and 16.4% when recall is 0.09.

Table 5.2 Average F measure values for various techniques for MovieLens dataset

<table>
<thead>
<tr>
<th></th>
<th>TDF-IDF</th>
<th>Concept expansion</th>
<th>Language modelling using Query likelihood</th>
<th>DNLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F measure</td>
<td>0.528718</td>
<td>0.557951</td>
<td>0.551004</td>
<td>0.575810</td>
</tr>
</tbody>
</table>
Figure 5.3 Average F measure values for various techniques for MovieLens dataset

From the figure 5.3 it is seen that average f measure values for DNLM with the proposed concept query expansion is higher than the average f measure values for tdf.idf by 8.9%, concept expansion by 3.2% and Language modelling using query likelihood by 4.5%.

The experimental results for Reuters-21758 dataset for precision and F measure are tabulated in Table 5.3 and Table 5.4 respectively. Figure 5.4 and 5.5 show the same.
Table 5.3  Precision values for various techniques for Reuters-21758 dataset

<table>
<thead>
<tr>
<th>Recall</th>
<th>TDF-IDF</th>
<th>Concept expansion</th>
<th>Language modelling using Query likelihood</th>
<th>DNLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.831644106</td>
<td>0.89949308</td>
<td>0.875672961</td>
<td>0.983533761</td>
</tr>
<tr>
<td>0.1</td>
<td>0.769754052</td>
<td>0.851704269</td>
<td>0.827139846</td>
<td>0.928597104</td>
</tr>
<tr>
<td>0.2</td>
<td>0.748400666</td>
<td>0.773306475</td>
<td>0.782014571</td>
<td>0.843243592</td>
</tr>
<tr>
<td>0.3</td>
<td>0.748902623</td>
<td>0.764249838</td>
<td>0.755021124</td>
<td>0.835386254</td>
</tr>
<tr>
<td>0.4</td>
<td>0.689139901</td>
<td>0.734923375</td>
<td>0.711668277</td>
<td>0.805139854</td>
</tr>
<tr>
<td>0.5</td>
<td>0.6774131</td>
<td>0.724206713</td>
<td>0.709199407</td>
<td>0.791218501</td>
</tr>
<tr>
<td>0.6</td>
<td>0.627466661</td>
<td>0.732498213</td>
<td>0.700349971</td>
<td>0.801778771</td>
</tr>
<tr>
<td>0.7</td>
<td>0.592573475</td>
<td>0.69528761</td>
<td>0.682966143</td>
<td>0.759497059</td>
</tr>
<tr>
<td>0.8</td>
<td>0.551441906</td>
<td>0.667393946</td>
<td>0.651598754</td>
<td>0.725531569</td>
</tr>
<tr>
<td>0.9</td>
<td>0.543356262</td>
<td>0.620455421</td>
<td>0.582080004</td>
<td>0.677859969</td>
</tr>
<tr>
<td>1</td>
<td>0.310100304</td>
<td>0.530449696</td>
<td>0.492848736</td>
<td>0.439102732</td>
</tr>
</tbody>
</table>

Figure 5.4  Precision values for various techniques for Reuters-21758 dataset
The Figure 5.4 shows the relationship between precision and recall for Reuters-21758 dataset and it is seen that precision values for DNLM with the proposed concept query expansion are higher when compared with the precision values for all the other techniques. The precision values for DNLM with the proposed concept query expansion is higher than the precision values for tdf.idf by 18.2% when recall is 0.01 and 31.5% when recall is 0.9. DNLM with the proposed concept query expansion performs better than concept expansion by 9.3% when recall is 0.01 and 8.7% when recall is 0.9. Similarly for Language modelling using query likelihood, DNLM with the proposed concept query expansion performs better by 12.3% when recall is 0.01 and 11.3% when recall is 0.09.

Table 5.4 Average F measure values for various techniques for Reuters-21758 dataset

<table>
<thead>
<tr>
<th>Technique Used</th>
<th>TDF-IDF</th>
<th>Concept expansion</th>
<th>Language modelling using Query likelihood</th>
<th>DNLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F measure</td>
<td>0.563152</td>
<td>0.592410</td>
<td>0.585548</td>
<td>0.609676</td>
</tr>
</tbody>
</table>

Figure 5.5 Average F measure values for various techniques for Reuters-21758 dataset
From the above pictorial representation it is seen that the proposed methodology produces higher average f measure than the existing tf.idf method, concept expansion and Language modelling using Query likelihood. The average f measure values for DNLM with the proposed concept query expansion is higher than the average f measure values for tdf.idf by 8.2%, concept expansion by 2.9%, and Language modelling using query likelihood by 4.1%.

In the second set of experiments, additional terms with query are used. Experiments are conducted for 5, 10, 15, 20, 25 additional terms. Precision values for various techniques for MovieLens dataset and Reuters dataset is evaluated. The techniques used were tdf.idf, Language modelling using query likelihood, proposed concept expansion and proposed DNLM with the proposed feature selection method. The experimental results of percentage improvement for MovieLens dataset and Reuters dataset for precision are tabulated in Table 5.5 and Table 5.6 respectively. Figure 5.7 and 5.8 show the same.

Table 5.5 Percentage Improvement in Precision for MovieLens dataset

<table>
<thead>
<tr>
<th>Additional terms</th>
<th>Percentage improvement - MovieLens</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TDF-IDF</td>
</tr>
<tr>
<td>5</td>
<td>2.6</td>
</tr>
<tr>
<td>10</td>
<td>3.08</td>
</tr>
<tr>
<td>15</td>
<td>3.26</td>
</tr>
<tr>
<td>20</td>
<td>3.14</td>
</tr>
<tr>
<td>25</td>
<td>2.97</td>
</tr>
</tbody>
</table>
Figure 5.6 Percentage Improvement in Precision for MovieLens dataset

Table 5.6 Percentage Improvement in Precision for Reuters dataset

<table>
<thead>
<tr>
<th>Additional terms</th>
<th>Percentage improvement - MovieLens</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TDF-IDF</td>
</tr>
<tr>
<td>5</td>
<td>2.8</td>
</tr>
<tr>
<td>10</td>
<td>5.275</td>
</tr>
<tr>
<td>15</td>
<td>6.48</td>
</tr>
<tr>
<td>20</td>
<td>4.68</td>
</tr>
<tr>
<td>25</td>
<td>4.2</td>
</tr>
</tbody>
</table>
5.6 CONCLUSION

In this study, a probabilistic query expansion model is presented based on an automatically constructed similarity thesaurus. A similarity thesaurus reveals domain knowledge about the particular collection from which it is constructed. The two important issues with query expansion are addressed: the selection and the weighting of additional search terms. In contrast to earlier methods, in the proposed method queries are extended by addition of terms similar to the concept of the query. Experiments are conducted for varying number of additional terms (5, 10, 15, 20, 25). Experimental results demonstrate the superiority of the proposed concept based query expansion method with respect to the precision. It is also observed that 15 additional terms achieve the maximum precision.