CHAPTER 6

MOBILE AGENTS BASED INFORMATION RETRIEVAL

6.1 INTRODUCTION

As volume of content and content types on internet increases in size it is clear that online publishing needs web content management/ distribution. This study aims to demonstrate information retrieval technology’s mobile agent use. It demonstrates workability of this technology combination in e-learning Application.

The Internet is a main source of multimedia content for entertainment, education, and business. Effective or efficient method of data retrieval from Internet is needed and content is an important media in this situation. Humans have amassed enormous amounts of information over time and material is scattered around the world. It is clear that an optimal path to create useful information sources is distribution of the task of digitizing historical and heritage material in analogue formats including books, manuscripts, music scores, maps, photographs, videos, analogue tapes and phonograph records. To achieve this, libraries, museums, and archives globally, large or small, need policies, guidance, and tools to digitize collections to ensure availability economically.

The attraction of this technology for application in distributed digital libraries is its assumption that each system (Content Store) is different. Mobile Agents provide users with “what, where, and how” to access
information from systems. Such methods to find out how to access varied
archives/collections, each with various database languages and services are
both useful and essential as archives/content globally begin to convert
collections into digital formats. Despite potential and industry activity, mobile
agents were unused in digital library/content domain. Open-source digital
library management system ‘Fedora’ is an exception promoting distributed
digital library architecture through operable access to digital objects/
communication across the net based on Web services Agents.

Industry interest resulted in a mobile agents, Web services and
related standards development. Though young, adopters are using this
technology proving its potential.

6.2 RELATED WORKS

Huang et al (2009) stated a hierarchical mobile agent framework for
the high speed and convenience of the Internet in online applications. It was
proposed for handling key management and access control problems between
mobile agent and host. In advanced networking research, issues on mobile
agents have always been of popular interest. Effectively utilizing resources
over the Internet greatly enhances the efficiency of an organization and
economizes computational overhead. Mobile agents were challenged by
execution barriers from security problems. Providing a crypto-system with
workable secure-control methods is essential for access activities in the
Internet. The procedures of key generation and operation are very simple;
users with greater accessibility can directly access the decryption key of the
subordinate members, but the latter was not allowed to access encryption key
of the former. It economizes the exhaust of storage space. It raises the security
of key management, and controls access to distributed environment in non-
specific network. It successfully secures the accessing relationship between
the mobile agent and the host while economizing the exhaust of storage space.
The achievement facilitates efficient operation of mobile agents, and provides a secure execution environment for mobile computing.

Gao et al (2011) stated an agent migration approach with a multi-agent system to avoid problems like network overload, limited resources of mobile devices, mobile users not always online during retrieval of data automatically from one or multiple remote biological data sources. To address the above mentioned problems, an agent migration approach with a multi-agent system was integrated to overcome the high latency or limited bandwidth problem by moving their computations to the required resources or services. It also explains the system architecture, the migration strategy, as well as the security authentication of agent migration. The agent migration approach can also be applied to retrieving non-data web resources, for example, sending mobile agents to some bioinformatics web servers and retrieving analysis results to mobile devices. Since user-friendly and publicly accessible web-servers represent the future direction or developing practically more useful models, simulated methods, or predictors.

Fortino and Russo(2012) created an agent called Event-driven Lightweight Distilled State Charts-based Agents (ELDA) Meth, a novel agent-oriented methodology supported by a CASE tool for the simulation-based prototyping of Internet-based distributed agents systems (DAS) that enables rapid prototyping based on visual programming, automatic code generation and dynamic validation, was presented. The distinctive characteristics of ELDA Meth are: effective agent model for distributed computing systems, simulation-based agent-oriented methodology for design validation before implementation and deployment, integration with other methodologies to exploit their well-defined method fragments, and CASE tool support for supporting all development phases from modeling to simulation and implementation. ELDA Meth used both stand-alone for the modeling and
evaluation of DAS and coupled with other agent-oriented methodologies for enhancing them with simulation-based validation. In particular, it is based on the ELDA agent model, and provides key programming abstractions very suitable for highly dynamic distributed computing and on a CASE tool-driven iterative process fully supporting the modeling, simulation, and implementation phases of DAS. ELDA Meth has been applied to prototype several kinds of DAS such as mobile e-Marketplaces, content delivery infrastructures, and information retrieval systems. A case study in the information retrieval domain which has demonstrated the suitability and great effectiveness of ELDA Meth for the rapid prototyping of Internet-based DAS are explained.

Latiri et al (2003) designed an approach for query expansion based on fuzzy association rules between sets of terms. These fuzzy association rules were derived from the corresponding fuzzy closed term sets, which are from the target corpus. Discovered fuzzy contextual inter-term correlations reflect faithfully, the specified terms importance degrees in a user-defined query. To validate this approach, one method was applied to a corpus of the OFIL collection. Information retrieval (IR) focuses on the process of determining and assessing the adequacy between a user-query and a collection of documents, yielding a subset of relevant documents. The query expansion aims to reduce an eventual query/document mismatch by expanding the query using "correlated" terms. An approach based on the use of association rules was proposed to detect such correlations, in order to improve retrieval effectiveness by reducing such mismatch. By considering the term-document relation as a fuzzy binary relation, a fuzzy conceptual approach was proposed to extract fuzzy association rules. An experimental study, on real textual collections, has been conformed intuitive hypothesis that the synergy between association rules and query expansion was fruitful. Results of the study
showed a significant improvement in the performances of the information retrieval system, both in terms of recall and precision.

Cui et al (2002) suggested Query expansion as an effective way to resolve the short query and word mismatching problems. The specific characteristics of web searching can be done by web query logs which records the availability of large amount of user interaction information. The gap between the query space and the document space are narrowed. For new queries, high-quality expansion terms can be selected from the document space on the basis of these probabilistic correlations. This method was tested on a data set that is similar to the real web environment. A series of experiments showed that the log-based method can achieve substantial performance improvements, not only over the baseline method without expansion, but also with respect to the local context analysis. New method was proposed to extract probabilistic correlations between query terms and document terms by analyzing query logs. These correlations were then used to select high-quality expansion terms for new queries. The experimental results show that the log-based probabilistic query expansion method can greatly improve the search performance and has several advantages over other existing methods.

6.3 MOBILE AGENTS

Mobile Agents are independent smart programs moving through networks, seeking and interacting with available/compatible services on user’s behalf. Mobility is an agent’s orthogonal property, as not all agents are mobile. An agent can sit and communicate with surroundings through conventional means like various remote procedures calling and messaging.
6.3.1 Stationary Agents

A stationary agent executes on systems alone, where it begins execution. When information not on a system, is required it needs to interact with agents on other systems. A communication mechanism like RPC is used.

6.3.2 Mobile Agent

A mobile Agent is not tied to a system where it starts execution as it can migrate through a network. Transport ability enables mobile Agents to enter a system having an object with which the agent wants to interact. It takes advantage of being in the source host/network as object.

6.3.3 Characteristics of Mobile Agents

Following are some of the characteristics of Mobile Agents

(i) Mobile Agents reduce the network load:

In distributed systems, infraction between peer systems relies on communication protocol involving multiple infractions to compete a specific task. Mobile Agents enable packaging a conversation to dispatch it to a destination host, where the infraction takes place locally. When large data volumes scored at a remote server, data should be processed there. Data is located rather than transferred over the Network.

(ii) Mobile Agents Overcome network latency:

Mobile Agents are dispatched from a central controller to act locally and to directly execute controller directions.
(iii) Mobile Agents encapsulate protocols.

The code which is incoming/outgoing data encapsulates data itself. When a Mobile Agent migrates it carries data and code interpreting data during communication.

(iv) Mobile Agents executes asynchronously and atonality

The task to carry out has encapsulations with the agent, which is then dispatched through the network. The connection need not be continuous as it can reconnect at a later time to collect agent.

(v) Mobile Agent Adapt dynamically:

Mobile Agents can sense their execution environment reacting automatically to changes. Multiple Mobile Agents can distribute themselves among network hosts to maintain optimal configuration.

(vi) Mobile Agents are Naturally Heterogeneous:

As Mobile Agents transport layer independently being dependent only an execution environment, it ensures optimal conditions for seamless system integration.

(vii) Mobile Agents are robust and fault Tolerant.

Unfavorable situation/events build robust and fault tolerant distribution system easily.
6.4 MOBILE AGENT APPLICATION

Some of the mobile agent applications are:

(i) Electronic commerce
(ii) Personal Assistance
(iii) Secure Brokering
(iv) Distribution IR:

IR is an example of a mobile Agent application where instead of moving large data amounts to search engine to create search indexes, it dispatches agents to remote information sources, where it creates search indexes locally to be shipped back to origin later. Mobile Agents perform extended searches not constrained by hours when creator’s computer is operational.

(v) Telecommunication Network Services:

(vi) Work flow application and groupware

(vii) Monitory and Notification

(viii) Information dissemination

(ix) Parallel Processing.

6.4.1 Multiple Agents executes a task in parallel: (Remote File Update)

An Agent updates files by replacing one specified word occurrences. The philosophy operating an agent is as follows:
It is beneficial for files larger than a specific size (saves network Bandwidth) to perform a distributed files update as it saves file downloading/uploading and distributes update load to multiple servers.

6.4.2 Controlling an Agent

The example discussed has a parent agent that lets its child roam and collect information while on a short leash. Agent’s travel itinerary remains with the origin, with parent agent while child agent travels from host to host. The parent agent is stationary, keeping the itinerary of the traveling agent. Itinerary is in destination list. The traveling agent dispatches via proxy for the list’s each destination element.

6.4.3 Agent is parallel execution

The Agent delegates tasks to multiple Agents. If an agent searches hosts for information it does either sequentially or parallelly by creating 10 agents (workers) each of whom visits one host.

The parallel execution collaboration scheme is simple. The Agent creates a worker for each destination that he visits. The worker agents are dispatched to respective destinations. Each worker implements task executed by message handler. When it receives a message on completion of task, the worker returns result as a reply to parent agent accommodates incoming results.

6.5 METHODOLOGY

The Mobile agent is a technology needing a “killer application”. Though, motions of transportable computer programs or mobile code are around for some time, deployment of technology remains highly limited.
Experimental system described here provides concept proof attempting to bring technology deployment closer to reality. Every system has a LMS; a different reformulated query is sent to various systems through mobile agents during retrieval.

Mobile agent’s appeal as a distributed computing paradigm, compared to conventional client-server paradigm, is not questioned and is documented. Its deployment not being widespread is due to many reasons, chief of which is that “there is no application that can be implemented only with this paradigm”, and so there is no big reason for the paradigm to displace established paradigms like a client-server. One justification to use mobile agent is that paradigm is appropriate to compute on networks with wireless links, as with proper implementation, mobile agents
(i) Allow efficient/economical communication channels use which may have low bandwidth, high latency and be error-prone.

(ii) Enable use of portable, low-cost, personal communications devices to perform complex tasks even when device is not linked to the network.

Another attractive paradigm property is that it allows an application to be really distributed, as tasks in an application, embodied in a mobile agent, are worked out on participating systems in a decentralized process.

Previous chapters reveal that query reformation is effective with use of 15 additional words. Mobile agent concept generates 75 additional words with 5 queries being used by a mobile agent which retrieve query based documents. Retrieved documents median are used as actual retrieved documents.

Experiments are conducted with MovieLens dataset and Reuters dataset and also with college Learning Management System (LMS).

6.5.1 Collection Selections and Ranking

The Mobile Agents play important role in distributed information retrieval for collection selection and ranking problems. The central brokering agent selects only Top N limited number of relevant collections for applying the search according to the given query. The Mobile agents have been already trained with nominal language modelling technique and then they have been migrated to all the document collection in the distributed network and then apply the collection selection and ranking algorithms for the best ranking of
relevant documents. The probabilities of a term $t_i$ for the given collections $C_j$ have been estimated for each query term using the following formula.

$$P(t_i|C_j) = 0.4 + 0.6 * T*I$$

$$T = \frac{df}{df+50+150+nw+aw}$$

$$I = \frac{\log(|C|-0.5)}{\log(|C|-1.0)}$$

where $df$ - number of documents in the Collection $C_j$

$cf$ - number of collections containing the term $t_i$

$|C|$ - total number of collections

$nw$ - no of words in the Collections

$aw$ - average no. of words of collection being ranked

The result merging is the next step in this distributed information retrieval to combine the result for each query term $q$ and collection $C_j$ to form a belief that a given collection will satisfy the query.

$$W_{i,Q} = \sum_{q \in Q} \frac{P(q|C_j)}{|Q|}$$

The results have been analyzed for both the precision and recall for each of the Top N Collections.

### 6.6 Recommendation Systems

Recommendation system is an important e-learning tool, as more institutions implement their features into their learning management system. Recommendation system is planned to overcome the huge data quantities in a
dataset. e-commerce is an initial application that used recommender system to increase sales and offers a wide variety of items through the net. e-commerce customers are analogous to learner (student) in e-learning systems. Customer/learner gets recommendations for smart-search.

A recommender system is an information filtering technology designed to determine items most likely to be of the customer (learners) tastes. Recommender system type varies based on affinity between customer and item which is identified with matched pairs. There are two frequently used recommendation system; one being collaborative filtering where a system analyses historical interaction alone, and the other is content based filtering system based on profile attributes.

This study combines hybrid recommendation systems for real world problems. Obtaining recommendation is a critical component in an intelligent decision making system, due to the challenge of personalizing advertising efforts.

It analyses usage data across users to find and match user-item pairs. It collects user feedback as item ratings and exploits similarities in rating behavior amongst many users to determine how to recommend an item.

There are two applications in collaborative filtering

(i) Neighborhood based approaches

(ii) Model based approaches

In neighborhood approach, a user’s subset is chosen based on rating similarities. In term such weighted ratings produce predictions for learners.
Algorithm 6:

Step 1: Assign a weight to all users based on similarity with active user.

Step 2: Select k top users having highest similarity with active user (Neighborhood).

Step 3: Compute a prediction from a weighted combination of selected neighbors rating.

The $W_{a,u}$ is the similarity weight measure between user ‘u’ and active user ‘a’. The Pearson correlation Co-efficient between ratings of two users is commonly used to find out the weight.

$$W_{a,u} = \frac{\sum_{i \in I}(r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I}(r_{a,i} - \bar{r}_a)^2 \sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}}$$  \hspace{1cm} (6.3)

$I$ – is set of all items

$r_{u,i}$ – Rating of user ‘u’ for item ‘i’

$r_u$ – Average rating of user ‘u’.

To produce prediction of recommendation of an item ‘i’ for user ‘a’ is generally computed as weighted average of deviations from neighbor’s mean.

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in k}(r_{u,i} - \bar{r}_u) * W_{a,u}}{W_{u \in k} * W_{a,u}}$$  \hspace{1cm} (6.4)

$P_{a,i}$ - Prediction for the active user ‘a’ for item ‘i’

$W_{a,u}$ - Similarity between user ‘u’ with active user ‘a’

$k$ – Neighborhood set of most similar users.
6.7 EXPERIMENTAL SETUP AND RESULTS

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 and recall 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 6.1 and Table 6.2 respectively. Figure 6.2 and 6.3 show the same.

Table 6.1 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>0.7509</td>
<td>0.8026</td>
<td>0.7838</td>
<td>0.8804</td>
</tr>
<tr>
<td>0.1</td>
<td>0.6877</td>
<td>0.7593</td>
<td>0.7396</td>
<td>0.8304</td>
</tr>
<tr>
<td>0.2</td>
<td>0.6672</td>
<td>0.6853</td>
<td>0.6978</td>
<td>0.7525</td>
</tr>
<tr>
<td>0.3</td>
<td>0.6659</td>
<td>0.6781</td>
<td>0.6700</td>
<td>0.7436</td>
</tr>
<tr>
<td>0.4</td>
<td>0.6121</td>
<td>0.6514</td>
<td>0.6327</td>
<td>0.7138</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5993</td>
<td>0.6394</td>
<td>0.6281</td>
<td>0.7007</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5525</td>
<td>0.6436</td>
<td>0.6172</td>
<td>0.7053</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5207</td>
<td>0.6097</td>
<td>0.5990</td>
<td>0.6682</td>
</tr>
<tr>
<td>0.8</td>
<td>0.4828</td>
<td>0.5809</td>
<td>0.5710</td>
<td>0.6358</td>
</tr>
<tr>
<td>0.9</td>
<td>0.4730</td>
<td>0.5400</td>
<td>0.5082</td>
<td>0.5918</td>
</tr>
<tr>
<td>1</td>
<td>0.2687</td>
<td>0.4595</td>
<td>0.4283</td>
<td>0.3816</td>
</tr>
</tbody>
</table>
Figure 6.2 Precision values for various techniques for MovieLens dataset

From the Figure 6.2 it is seen that precision values for DNLM is higher than the precision values for all the other techniques. The precision values for DNLM is higher than the precision values for tdf.idf by 17.2% when recall is 0.01 and 25.1% when recall is 0.9. DNLM performs better than concept expansion by 9.6% when recall is 0.01 and 0.9. Similarly for Language modelling using query likelihood, DNLM performs better by 12.3 % when recall is 0.01 and 16.4% when recall is 0.09.
Table 6.2  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>0.5393</td>
<td>0.5679</td>
<td>0.5626</td>
<td>0.5879</td>
</tr>
</tbody>
</table>

Figure 6.3  Average F measure values for various techniques for MovieLens dataset

From the Figure 6.3 it is seen that average f measure values for DNLM is higher than the average f measure values for all the other techniques. The average f measure values for DNLM is higher than the average f measure values for tdf.idf by 9%, concept expansion by 3.5% and Language modelling using query likelihood by 4.4 %.
The experimental results for Reuters-21758 dataset for precision and F measure are tabulated in Table 6.3 and Table 6.4 respectively. Figure 6.4 and 6.5 show the same.

Table 6.3  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>0.8491</td>
<td>0.9185</td>
<td>0.8946</td>
<td>0.9909</td>
</tr>
<tr>
<td>0.1</td>
<td>0.7859</td>
<td>0.8697</td>
<td>0.8450</td>
<td>0.9355</td>
</tr>
<tr>
<td>0.2</td>
<td>0.7641</td>
<td>0.7896</td>
<td>0.7989</td>
<td>0.8495</td>
</tr>
<tr>
<td>0.3</td>
<td>0.7647</td>
<td>0.7804</td>
<td>0.7713</td>
<td>0.8416</td>
</tr>
<tr>
<td>0.4</td>
<td>0.7036</td>
<td>0.7504</td>
<td>0.7270</td>
<td>0.8111</td>
</tr>
<tr>
<td>0.5</td>
<td>0.6917</td>
<td>0.7395</td>
<td>0.7245</td>
<td>0.7971</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6407</td>
<td>0.7480</td>
<td>0.7155</td>
<td>0.8078</td>
</tr>
<tr>
<td>0.7</td>
<td>0.6050</td>
<td>0.7100</td>
<td>0.6977</td>
<td>0.7652</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5630</td>
<td>0.6815</td>
<td>0.6657</td>
<td>0.7309</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5548</td>
<td>0.6335</td>
<td>0.5946</td>
<td>0.6829</td>
</tr>
<tr>
<td>1</td>
<td>0.3166</td>
<td>0.5416</td>
<td>0.5035</td>
<td>0.4424</td>
</tr>
</tbody>
</table>

Figure 6.4  Precision values for various techniques for Reuters-21758 dataset
The Figure 6.4 shows the relationship between precision and recall and it is seen that precision values for DNLM is higher than the precision values for tdf.idf by 16.7% when recall is 0.01 and 23% when recall is 0.9. DNLM performs better than concept expansion by 7.8% when recall is 0.01 and 0.9. Similarly for Language modelling using query likelihood, DNLM performs better by 10.7% when recall is 0.01 and 14.8% when recall is 0.09.

Table 6.4  Average F measure values for various techniques for Reuters-21758 dataset

<table>
<thead>
<tr>
<th></th>
<th>TDF-IDF</th>
<th>Concept expansion</th>
<th>Language modeling using Query likelihood</th>
<th>DNLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>F measure</td>
<td>0.5750</td>
<td>0.6049</td>
<td>0.5982</td>
<td>0.6142</td>
</tr>
</tbody>
</table>

Figure 6.5  Average F measure values for various techniques for Reuters-21758 dataset
The average f measure values for DNLM is higher than the average f measure values for tdf.idf by 6.8%, concept expansion by 1.5 %, and Language modelling using query likelihood by 2.6%.

In the second set of experiments, investigations were conducted in the college Learning management system (LMS) -model which contained html and text based documents (about 15000 pages in 7 subjects). The experimental results for precision and F measure are tabulated in Table 6.5 and Table 6.6 respectively. Figure 6.6 and 6.7 show the same.

**Table 6.5 Precision values for various techniques for e-learning 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>0.69361</td>
<td>0.74289</td>
<td>0.72326</td>
<td>0.81237</td>
</tr>
<tr>
<td>0.1</td>
<td>0.63522</td>
<td>0.70278</td>
<td>0.68253</td>
<td>0.76626</td>
</tr>
<tr>
<td>0.2</td>
<td>0.61626</td>
<td>0.63434</td>
<td>0.64393</td>
<td>0.69435</td>
</tr>
<tr>
<td>0.3</td>
<td>0.61509</td>
<td>0.62766</td>
<td>0.61829</td>
<td>0.68615</td>
</tr>
<tr>
<td>0.4</td>
<td>0.56536</td>
<td>0.60292</td>
<td>0.58383</td>
<td>0.65863</td>
</tr>
<tr>
<td>0.5</td>
<td>0.55357</td>
<td>0.59185</td>
<td>0.57961</td>
<td>0.64659</td>
</tr>
<tr>
<td>0.6</td>
<td>0.51029</td>
<td>0.59572</td>
<td>0.56956</td>
<td>0.65085</td>
</tr>
<tr>
<td>0.7</td>
<td>0.48099</td>
<td>0.56439</td>
<td>0.55278</td>
<td>0.61654</td>
</tr>
<tr>
<td>0.8</td>
<td>0.44595</td>
<td>0.53767</td>
<td>0.52690</td>
<td>0.58668</td>
</tr>
<tr>
<td>0.9</td>
<td>0.43689</td>
<td>0.49986</td>
<td>0.46897</td>
<td>0.54608</td>
</tr>
<tr>
<td>1</td>
<td>0.24822</td>
<td>0.42534</td>
<td>0.39522</td>
<td>0.35209</td>
</tr>
</tbody>
</table>
From the Figure 6.6 it is seen that precision values for DNLM is higher than the precision values for all the other techniques for e-learning dataset. The precision values for DNLM is higher than the precision values for tdf.idf by 17.1\% when recall is 0.01 and 25\% when recall is 0.9. DNLM performs better than concept expansion by 9.3\% when recall is 0.01 and 9.2\% when recall is 0.9. Similarly for Language modelling using query likelihood, DNLM performs better by 12.3\% when recall is 0.01 and 16.4\% when recall is 0.09.

Table 6.6  Average F measure values for various techniques for e-learning 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>0.49812</td>
<td>0.52567</td>
<td>0.51912</td>
<td>0.54249</td>
</tr>
</tbody>
</table>
The average f measure values for DNLM is higher than the average f measure values for tdf.idf by 8.9%, concept expansion by 3.1 %, and Language modelling using query likelihood by 4.5%.

6.7.1 Limitations

The implementation relies on java and its network support. Hence all participating systems must host a java virtual machine and the extensive java network class library.

System security is a well-known concern for mobile agents. A remote host which allows a mobile agent to visit opens itself to the potential of security breaches. Thus, security levels are required so that the mobile agent is protected from the host, the host is protected from the agent, agents are protected from each other, and hosts are protected from each other.
6.8 CONCLUSION

The objective of this study is to demonstrate the ability of the mobile agent’s technology in information retrieval. The system demonstrates the workability of this combination of technologies in an e-learning Application. In the mobile agent concept, 75 additional words are generated and 5 queries are used by each mobile agent. The mobile agents retrieve documents based on the query. The median of the retrieved documents are used as the actual retrieved documents. Experiments are conducted with MovieLens dataset and Reuters dataset. Experiments were also conducted in the college Learning Management System (LMS). Results demonstrate the effectiveness of the mobile agent method.