A Context-Based Information Agent for Supporting Intelligent Distance Learning Environments

Mohammed A. Razek, Claude Frasson, Marc Kaltenbach
Département d'informatique et de recherche opérationnelle Université de Montréal
C.P. 6128, Succ. Centre-ville Montréal, Québec Canada H3C 3J7
{abdelram frasson kaltenba} @iro.umontreal.ca

Abstract
The large amount of information now available on the Web can play a prominent role in building a cooperative intelligent distance learning environment. We propose a system to provide learners with useful information in a group discussion. Finding the right information at the right moment is quite a difficult task, especially when the learner's interests are continually updated during the discussion. This paper presents a context-based information agent that can observe conversations among a community of learners on the Web, interpret the learners' inputs, and then assess the current context of the session. The agent must be able to adapt its behavior autonomously to the changing context, build a new query to get updated information from the Web, and originate the search task. Then, it can filter the results, organizing, and presenting information useful to the learners in their current activities. We claim that specifying the context of a search better can significantly improve search results. An important task, therefore, is to assess the context. For this, we have developed dominant meaning space. That is a new set based measure to evaluate the closeness between queries and documents. Our experiments show that the proposed method greatly improves retrieval effectiveness, in terms of average overall accuracy as well as that in the top twenty documents. This work is the core component of a new pedagogical agent to help people learn tasks defined within greater Web-based tutoring systems.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval model; [**Document and Text Processing**]: Document Capture-document analysis.

General Terms
Algorithms, Measurement, Performance, Experimentation.

Keywords
Intelligent distance learning environment, Search, Context, Dominant meanings space.

1. Introduction
Current Web-Based Tutoring Systems (WBTS), designed to provide distance learning environments, are still poor at adapting to the specific of learners'. We need very flexible knowledge bases in many kinds of resources (alternative explanations, examples of at various levels of complexity, exercises, etc.) which can be used opportunistically by WBTS' controlling agent. Considering the huge amount of information available on-line and of rapidly evolving knowledge, a finite knowledge base will never be sufficient. Few WBTS account for the benefit of information that can be taken directly from the Web during a learning session [17]. Finding the right information at the right time is a very time-consuming task, however, particularly because Web search systems present thousands of results, most of which are inappropriate.

We have developed the Confidence Intelligent Tutoring System (CITS) [14] to provide a Cooperative Intelligent Distance Learning Environment (CIDLE) for a community of learners to improve on-line discussions. The CITS employed a machine learning technique predicts preferred learning styles [15]. Therefore, it infers their behavior, and adapts the presentation in accordance with their particular learning styles. To be adaptive and dynamic, the CITS searches the Web and brings related documents to every concept.

This paper addresses the problem of how to enable CITS to search the Web and find sufficient information during a learning session. Most useful would be an autonomous agent that can observe conversation, interpret the inputs from learners, and then specify the current context. The agent would adapt its behavior accordingly, build a new query, and originate the search task. As a result, it would filter, organize, and present information that is immediately useful to the learners. We claim that making the context more specifying can significantly improve search results. The agent specifies the context based on the learner context-of-interest [2] and domain knowledge and also helps establish what we call the "dominant meaning" of a query. Achieving this goal requires an accurate representation of the context of the domain knowledge being taught.

Using context-of-interest [2] and the context surrounding a user-selected phrase [6], researchers have already studied several aspects of the use of learner's information interests. For good primary retrieval, we need to specify queries that are better related to the search context. In fact, we need to find a way of constructing this context and then using it to expand the query. We used the benefit of the domain knowledge in CITS.

The idea is to represent domain knowledge as a hierarchy of concepts [3]. In our proposed approach, each concept consists of some dominant meanings, and each dominant meaning links with a text fragment that defines it. We claim that the more any query consists of dominant meanings, the more closely it is related to its search context. For parsing documents coming from the Web, we design a dominant meaning space method. This measures

WWW 2003, May 20-24, 2003, Budapest, Hungary.

semantic space between the original query and the collection of retrieved documents based on existing sets of dominant meanings. For example, suppose that two users discus some concept in a course on data structure. Based on the dominant meanings of the query, a search for the term "Array-Based" must return stack-related information if carried out from a conversation related to stack or return queue-related information and the conversation is about queue concept.

In this work, we describe the Context-based Information Agent (CBIA), one that observes discussions during a cooperative learning session and extracts its main concept but also searches the Web to find related documents and presents them to learners. CBIA has been fully implemented and integrated into CITS.

This paper is organized as follows. Section 2 gives a brief introduction to previous work. Section 3 briefly describes the overall structure of CITS, and presents the various characteristics of our Context-Based Information Agent. Section 4 discusses the role of the Context-Based Information Agent and describes our probabilistic dominant meanings method and weighting model. Section 5 presents the results of experiments conducted to test our methods. And section 6 concludes the paper.

2. Related Work

A great deal of work has been done on building systems that use context to find information. IntelliZap [6] is based on the client-server paradigm: a client application running on a user's computer extracts the context around the text highlighted by the user. Server-based algorithms analyze the context, selecting the most important words, and then prepare a set of augmented queries for subsequent search. But a problem can occur when the content surrounding the marked query is not enough to specify any of the query's contexts.

SearchPad [1] is an agent that explains the context as a set of previous information requests produced by the user. This system works collaboratively with result documents, and maintains relations between queries and links considered useful. SearchPad maintains these relations as its search context. The mean idea at [18] is to expand an original query in the context of top-ranked pages retrieved during the first step. To extract the context, Xu et al. [18] proposed a technique called local context analysis. On the one hand, this technique selects expansion features based on their co-occurrence with the query terms. On the other hand, it selects them from the documents that have been retrieved most often. Afterward, it ranks concepts by their co-occurrence with the query terms in the top ranked documents and uses the highest ranked concepts for expansion.

The remembrance agent [16] continually observes what is being written into an Emacs editor window. Afterward, the remembrance agent is continually searches the user's personal text files and other remembrance agents to find documents.

The main difference between these systems and CBIA is that the letter analyzes the context of dominant meanings in text typed by learners. Based on the knowledge base of its domain knowledge, the CBIA can build a new Web searcher to find adequate results.

In the area of using automatic query expansion to enhance search results, Following [4, 5], Hang [8] proposes other methods dependent on the analysis of learner logs. Hang's methods create probabilities of correlations among terms in queries and retrieved documents mined from learner logs. These methods have an obvious difficulty: if the clicked documents have words unrelated to the search context, then the documents retrieved using the expanded query will be inadequate. For example, suppose that the query is about Java Programming Language, and the expanded query contains words such as swing, API, applet, and so on. If a clicked document includes a high probability of correlation for the word "swing", it will be signed as relevant. This document is irrelevant, in fact, because the main topic of the document is about the swing function and its properties in Java rather than the Java programming language.

Although adding new words to queries enhances their performance [11], most attempts at automatically expanding queries have failed to improve retrieval effectiveness [12]. However, Qiu [13] suggests that this is because two problems in were not solved: the selection of suitable terms and the weighting of selected additional search terms. Furthermore, we see the need for a suitable statistical method (threshold) of specifying relevant documents. If a document is assumed relevant but in fact is not, then most of the words added to the query and extracted from that document will be unrelated to the query's context. The quality of the documents retrieved is likely to be low. But, if a document really is relevant, then most of the words added to the query will indeed be related to the search context. Consequently, the retrieval quality of final results is likely to be high. Thus, good primary retrieval is very important.

To overcome this problem, we suggest a measure called *dominant meaning space* which focuses on the main topic of the query and its dominant meanings. In this sense, a query is supposed to look for its dominant meanings in a document [7] rather than for keywords. We represent this meaning in the form of a meaning vector. The closeness of a query and a document is indicated by a value of the dominant meaning probability between them. The next subsection sheds light on the CITS system and its functions.

3. CITS and Distance Learning Challenges

CITS is a web-based tutoring system for computer supported intelligent distance learning environment. The main purpose of this tool is to help in building an intelligent learning environment on-line. The architecture of CITS [14] is based on five types of agent: the cognitive agent, based on a machine learning technique, discerns various types of knowledge from learners and predicts their learning style [15]; the behavior agent is responsible for studying the behavior of learners in order to validate their styles as predicted by the cognitive agent; the guide agent finds learners with similar interests and introduces them to each other; the Context-Based Information agent deals with domain knowledge and searches the Web for extra information; and the confidence agent establishes the conditions of a successful conversation between learners. The knowledge base built by this system consists of two types of knowledge: (i) knowledge about the learner and (ii) domain knowledge.

To elucidate the scenario supported by CITS, suppose that learner A needs to discuss a specific concept, say queue, in a course on data structure. To achieve this, CITS tries to find a second learner, B, with similar interests but more knowledge. Suppose also that A prefers to begin with written concept, lots of explanation, and does not like drawing. B, on the other hand, prefers to begin with

drawn figures, lots of examples, and does not like written concepts. The problem comes when B explains something to A. He discusses it from his own view, which depends on less written concepts. Consequently, A will find it hard to understand. In this situation, CITS based on a machine learning technique would predict their learning styles and thus adapt the presentation to suit both A and B.

Following the same example, CITS needs very flexible knowledge bases of data structure in many kinds of resources (alternative explanations, examples of varying levels of difficulty, exercises, etc...) which can be used opportunistically by its controlling agent. How we can build this knowledge base and deal with it? If the built-in domain knowledge is inadequate, how can the system retrieve more information from the Web? And how can control this information (i.e. acquiring, restoring, extracting)? In this paper, we try to answer these questions.

Current Web search engines allow users to enter queries. Users receive ranked documents. We suggest an approach that modifies the construction of the search context by using the dominant meanings of the query as further inputs. As shown in Figure 1, when two learners open a learning session using CITS about a specific concept, say queue (Figure 1.1), the CBIA deals with the session as follows: on the one hand, it observes discussions during the session and captures some words. With these words along with their dominant meanings, the CBIA constructs a query about the context of the main concept. On the other hand, it uses the constructed query to search the Web for related documents and presents them to both learners. When it receives the results, the CBIA parses them and posts new recommended results to its user interface (Figure 1.2). The CBIA supplies learners with search results list. That list shows the documents that rank highest, and allows the learner to retrieve its full content by clicking on it (Figure 1.3).

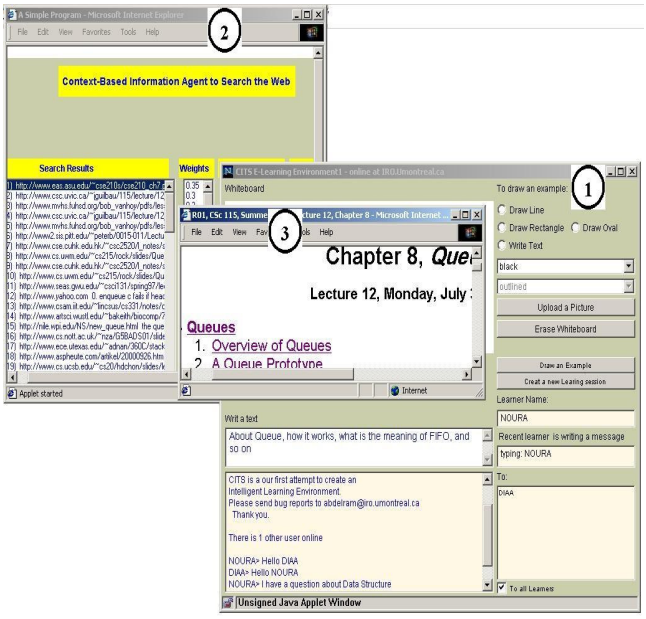

Figure 1 CITS User Interface

A diagrammatic summary of CBIA is shown in Figure 2. Here is a summary of the functions of the CBIA user interface. It allows learners to interact with their profiles for adding, deleting, and modifying the context-of-interest. It provides the learners with several documents related to the current learning session's concept. And it allows them to browse on the Web.

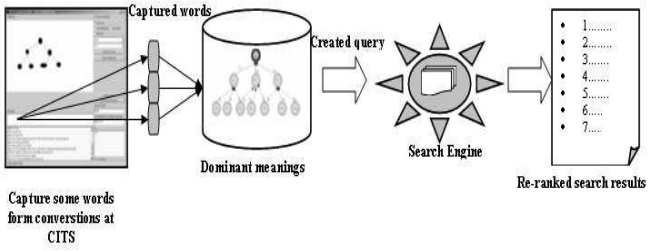

Figure 1 Information and Processing Flow of the Context-Based Information Agent.

For effective primary retrieval, we need to specify queries that are closely related to the search context. In fact, we must define a way by which to construct this context and then use it for expanding the query. The framework of CBIA consists of three components:

- Extract the main concept of the current learning session.
- Construct a query related to the context of session.
- Re-ranking the results coming from the Web.

4. The Role of CBIA

To extract the main concept of a current learning session, three challenges must be met: how to construct dominant meanings for each word, how the system decides which intended meaning to choose, and how it selects words that must be added to the original query. The following subsections explain these in more details.

4.1 Constructing Dominant Meanings

We used the domain knowledge in CITS to indicate the search context. Our idea is to represent the domain knowledge as a hierarchy of concepts [3]. In our proposed approach, each concept consists of some dominant meanings, and each dominant meaning is linked with a text fragment that defines it. We claim that the more a query consists of a session's dominant meanings, the more closely a query will be related to its search context.

Suppose that our domain knowledge C, say data structure, consists of m concepts $C = \{C_k\}_{i=1}^{m}$. Each concept is represented by a set of documents $C_k = \{D_v \mid v = 1,..., r\}$, and each document is signified by a set of words $D_v = \{w_j \mid j = 1,..., n\}$. Where, a word w_j represents the word j^{th} in the document D_v that belonged to the concept C_k. At this end, we can represent each concept as follows:

$$C_k = \bigcup_{v=1}^{r} D_v = \bigcup_{j=1}^{rxn} w_j$$

Therefore, each concept consists of many words. Our goal is to reduce this number to top-T words which can represent the dominant meaning of concept C_k. To do that, we follow: Suppose that a word w_c^k symbolizes the concept C_k.

- Compute the frequency of concept C_k, which appears in document D_v:

  $F(w_c^k \mid D_v)$, where $v = 1,..., r$. (1)

- Compute the frequency of word w_j which appears in document D_v:

  $F(w_j \mid D_v)$, where $j = 1,..., n$ & $v = 1,..., r$. (2)

- Calculate a maximum value of $F(w_c^k \mid D_v)$ $\forall v$

  $$F_c = \underset{v=1,...,r}{Max} \{F(C_k \mid D_v)\},$$

- Calculate a maximum value of $F(w_j \mid D_v)$ $\forall j, v$ i.e.,

  $$F_w = \underset{v=1,...,r}{Max} \{F(w_j \mid D_v)\},\ v = 1,...,\ r$$

- Choose F_c that satisfies $0 \le F_w \le F_c$ (3)

- Finally, consider the dominant meaning probability:

$$P_{kj} = P_{kj}(w_c \mid C_k) = \frac{1}{r}\left[\sum_{v=1}^{r} \frac{F(w_j \mid D_v)}{F_c}\right], \quad (4)$$

$$j = 1,..., n \ \& \ k = 1,..., m$$

So we divide $F(w_j \mid D_v)$ by the maximum value F_c of frequency of C_k, and normalize the results by dividing by the number of documents r in collection C_k. Based on formula (3), we clearly have $0 \le P_{kj}(w_j \mid C_k) \le 1$.

For each concept C_k, we rank the terms of collection $\{P_{k1}, P_{k2},..., P_{km}\}$ in decreasing order according to formula (3). As a result, the dominant meanings of concept C_k are represented by the set of words that is corresponding to the set $\{P_{k1}, P_{k2},..., P_{kT}\}$; i.e. $C_k = \{w_1, w_2,..., w_T\}$.

4.2 Representing Dominant Meanings

The most important of this paper is to construct an efficient way to organize every C_k so that it can be stored and retrieved quickly. We define a graph that provides the fundamental operations of storing words, finding them, and removing them from it. First, we need to describe the construction of a graph to predict the best way of exploring it. To that end, we introduce a new notion called a Dominant Meaning Graph (DMG), which represents the dominant meanings of all concepts. Our DMG is different from the Conceptual Graphs (CGs) [10], which are labeled graphs in which "concept" nodes are connected by "relation" nodes. The proposed graph consists of a set of nodes, and a set of edges. Nodes represent dominant meaning words for each concept C_k. To distinguish the words in each concept we suppose that each concept is represented as follows $C_k = \{w_1^k, w_2^k,..., w_T^k\}$ $\forall k = 1,..., m$. Target word C is designed as the end node. Each edge has a nonnegative weight P_{ij}, where P_{ij} represents the dominant meaning space between words w_i and w_j according the formulas (1:3) as follows: $P_{ij} = P_{kj}(w_i \mid w_j)$.

The dynamically changing of learners' knowledge during a learning session leads to the DMG continuously changing graph's nodes and edges. In that way, it will be very hard to find a specific node. This new method gives us an advantage in exploring. The DMG is not a free tree but rather is a graph whose shape obeys the following definition:

- The root represents the target word (whole concept).
- The first children represent concepts and any node is represented by only one word.

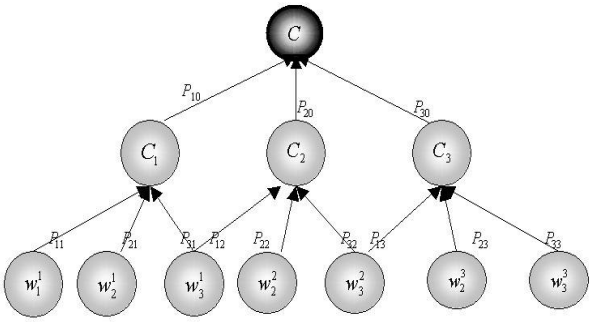

Figure 3 Dominant Meaning Graph (DMG) for representing the dominant meanings

- Every internal node has one or more children.
- All edges of children connected with their parents are stored as decreasing order from left to right. That is, the value of the left sibling's edge is greater than that of the right sibling, i.e.

  $P_{j+1,k} > P_{j,k}$, $\forall j, k \ge 0$

- The value of nodes and edges adapts dynamically by using a machine leaning approach.
- All external nodes except root are at the same level in the graph.

In the next subsection, we explain how to extract the main concept of any discussion through a learning session.

4.3 Extract Main Concept

We believe that the letter DMG can be used to extract the concept of any discussion. For example, suppose that two learners open a session about a specific concept. The CBIA extracts some words through disscusion, say $E = \{w_2^2, w_3^1, w_2^3\}$. The problem is to find which a concept probably represents these words. In other words, we need to find the best subset C_k that belongs to the whole concept C and contains most of the elements in E. It is obvious that traverse DMG is an important issue and certainly to be taken of account.

Researchers have used many techniques to traverse the graph [10]. For a large problem space, the graph's nodes must be searched in an organized way. Starting from a specific state (node) and moving to a specific goal might provide a solution. We can use Depth-First Search (DFS), Breadth First Search (BFS), and Best-First-Search to traverse graphs.

To traverse the graph, we use the hill climbing search algorithm with some modifications. It uses lists to maintain states: Search List to keep track of the current fringe. It chooses the node "C" as starting point. We mark this node to show that it has been visited. It applies a heuristic evaluation to the edges. The heuristic here is represented by the value of $P_{i,j}$, where P_{ij} represents the dominant meaning space between two words w_i and w_j.

Initially Search List consists of generated children that we intend to consider during the search. Suppose that a child is searched after it is taken out of Search List. After being opened, it can be expanded and removed. The proposed algorithm ends when a request word is extracted from Search List (success), or when we try to extract a child while it is empty (failure), or, in some cases, when we generate a goal state (success). The input of our traverse algorithm is a requested word w_r and the output would be a requested concept C_r. The pseudocode for this algorithm search is as follows:

TRAVERSEDMG (Requested word w_r)

1. ***Put*** Search List=[Starting point];
2. ***If*** Starting point = w_r ***then*** C_r = Starting point,
   Exit successfully and **return** C_r ;
3. ***while*** Search List ≠ [ ] do begin
   1. **Remove** the leftmost state from Search List, call it X;
   2. ***If*** X = w_r ***then*** C_r = paraent (X),
      Exit successfully and **return** C_r ;
   3. ***If not*** begin
      1. **Generate** children and edges of X.
      2. ***For each*** children of X
         1. Calculate the edge heuristic value $H(E_i)=P_{i,j}$;
         2. Sort children related to $H(E_i)$ as decreasing order;
         3. Add sorted children to Front of Search List;
4. If the goal has been found, announce success and return C_r .

Consider this algorithm of the directed graph in Figure 3. At each iteration, it removes the first element from the Search list. If it meets the requested word, the algorithm returns its parents (which led to the concept). If the first element is not a requested word, the algorithm generates its children and then applies heuristic evaluation $P_{i,j}$ to its adages. These states (children) are sorted in decreasing order according to the heuristic values before being inserted at the top of the search list. This brings the best state in the front of Search list. For example, if we used this algorithm to look for the main concept of the set of words $E = \{w_2^2, w_3^1, w_2^3\}$, we would get the set of corresponding concepts as $\{C_2, C_1, C_2, C_3\}$. We observe that the concept C_2 is repeated twice, therefore, it will be considered as the main concept of the desscusion. As a result, the dominant meaning vector of the concept C_2 is represented as $V(C_2, w_1^2, w_2^2, ..., w_T^2)$.

4.4 Re-ranking Results

Our proposed probability definition should retrieval effectiveness by making some constraints on submitted queries, and retrieved documents. Following the example above, the query constructed by our proposed algorithm consists of dominant meaning vector $V(C_2, w_1^2, w_2^2, ..., w_T^2)$. In general, we suppose that the query is $V(C_h, w_1^h, w_2^h, ..., w_T^h)$ and a stream of documents coming from the Web is $\{D_s\}_{s=1}^{s=q}$. Based on the dominant meaning probability in subsection 4.1, we compute the relevance of document D_s with respect to concept C_h, as follow:

$$P(C_h \mid D_s) = \frac{1}{n}\left[\frac{F(C_h \mid D_s)}{F_c} + \sum_{j=1}^{j=T}\frac{F(w_j^h \mid D_s)}{F_c}\right], \quad (5)$$

where, $\forall j = 1,..., T \quad \forall s = 1,..., q$, and

$$F_c = \underset{s=1,...,q}{Max}\{F(C_h \mid D_s)\} > F(w_j^h \mid D_s)$$

Function $F(C_h \mid D_s)$ signifies the frequency of concept C_h which appears in document D_s. The purpose of this step is to measure the importance of each document in the stream.
Formula (5) clarifies the restrictions that must apply to documents in order for it to be relevant. These restrictions depends on following these instructions:

- Use as a threshold maximum value F_c of the frequency of the original query.
- Weigh the dominant words according to both their frequencies and the threshold.

Doing so will prevent our system from falling into the same traps as the previous system [8], in which the system signed a document as relevant. In the next section, we present the experiments and results.

5. Experiments and results

In this section we describe a series of experiments conducted to reveal practical gains from the proposed approach of dominant meanings.

5.1 Dominant meanings vs. Keywords

The goal of this experiment was to demonstrate the effectiveness of the dominant meanings approach in retrieval. It was conducted on the MED collection [9], which contains 1033 documents(MEDLINE abstracts), and 30 queries. The test collection is often used for information retrieval. Table 1 presents its main features, the number of documents. In addition, it indicates the number of queries with relevance information, the number of terms, the average number of terms per document and query, and the average number of relevant documents per query.

Table 1: Collection used for experiment.

Collection	MED
Number of Documents	1033
Number of Queries	30
Number of Terms	8663
Average Terms in Query	271
Average relevance Documents	23.2

The experiment was conducted in two stages: training and retrieval. For the former, we built a dominant meaning graph of the MED collection using the method proposed in section (4). For the comparative experiments, we computed the threshold of dominant meaning space for relating one word with its dominant meanings. We used 20% of the documents in the MED collection as a training set for fitting this threshold parameter. For the retrieval stage, documents and queries were indexed as usual. We then computed the dominant meaning space between a word in a query and its dominant meanings in a document, using formulas

(1-3). Therefore, if the dominant meaning vector of a word in the query were either greater than or equal to the threshold parameter, the document would be considered relevant; otherwise, it would be considered irrelevant. The algorithm is summed up as follows:

Training Stage:

1. Build dominant meanings graph of the MED collection.
2. Compute dominant meanings vectors for each query.

Retrieval Stage:

1. Index the collection.
2. For each query and each document:
   - Compute the average dominant meaning space between the dominant meaning vector of a word in the query and those of its dominant meaning in the document
   - **If** (the average dominant meaning space > the dominant meaning threshold)
     Then Consider the document is relevant
     else consider it irrelevant.

Table 2 shows performance improvement when the query is expanded by using dominant meanings constructed by the proposed approach. The normal evaluation measures, precision and recall, were used. Precision is the ratio of the number of relevant documents retrieved to the total number retrieved. The average precision of a query is the average of precisions calculated when a relevant document is found in the rank list. We evaluated results by applying the average precision of a set of queries for 11 points of recall. Table 2 indicates that our dominant meaning approach produced a considerable improvement 14.8% in the retrieval effectiveness.

Table 2: Improvement using Dominant meaning approach

Collection	MED
Average Precision of Original queries	0.534
Average Precision of dominant meanings queries	0.682
Improvement	14.8%

This experiment shows that significant improvement in retrieval effectiveness can be achieved by creating a dominant meaning query and using a dominant meaning graph. The latter is an appropriate model for encoding the distribution of the terms, therefore, in a document collection. And dominant meanings give more accurate estimation between words in the query and dominant meanings in the document.

In the next subsection, we present another experiment for validating the Context-Base Information Agent performance. We compared its performance with those of two major search engines: Google, and AltaVista.

5.2 Context-Based Information Agent vs. Other Search Engines

As we have just shown, dominant meaning probability performed well. The main goal of this evaluation was to assess the effectiveness of the Context-Based Information Agent in the accuracy of a search engine's results. To clarify this goal, we compared the dominant meaning probability's values of the CBIA results against those of Google[1] and AltaVista[2] . We used original queries only.

The main core of the CBIA is in using the dominant meaning vector as queries instead of the original queries. Using formulas from (1) to (3), we analyzed the concepts of the domain knowledge, say data structure, in order to extract the preliminary dominant meanings for each concept. As a result, the concept "queue", includes all related words that can be added in the dominant meaning vector, such as V (*queue* , *reer* , *front* , *fifo* , *priority*) , and so on.

The algorithm is summarized as follows:

Training Stage:

1. Build dominant meanings graph of the domain knowledge (data structure course).
2. Build original queries (concepts without dominant meanings).
3. Compute dominant meanings vectors of original queries.
4. Compute threshold for each query.

Retrieval Stage:

1. For each original query
   - Send it to the two proposed search engines
   - For each query and each document:
     i) Compute the average dominant meaning space between the dominant meaning vector of a word in the query and those of its dominant meaning in the document
     If (the average dominant meaning space > the dominant meaning threshold)
     then
     consider the document relevant
     else
     consider it irrelevant
2. Send it to the two proposed search engines via CBIA
   - For each query and each document:
   - Compute the average dominant meaning space between the dominant meaning vector of a word in the query and that of its dominant meaning in the document
     If (the average dominant meaning space > the dominant meaning threshold)
     then
     consider the document relevant
     else
     consider it irrelevant.

Since learners of tutoring systems are normally interested in the top ranked documents, this experiment will compute the average dominant meaning probability of the top 20 ranked documents. Table 4 shows the retrieval quality difference between the CBIA results and that of each standard search engines. The figures show that our dominant meaning method yields more performance improvement in both search engines: Google and AltaVista. The level of improvement changes from one search engine to the other. Our experiment shows that the degree of relevance is much higher at CBIA working with Google than with AltaVista.

[1] http:// www.google.com

[2] http:// www.altavista.com

In short, our dominant meanings method produces a significant improvement in the retrieval effectiveness of both Google and AltaVista.

Figure 3 shows that the retrieval effectiveness of the standard retrieval method with Google and AltaVista varied randomly between 0 and 0.19. This must be the reason why learners waste most of their time looking for information that might be found after many efforts to browse through the results. Meanwhile, CBIA results were consistently better than others, they recommended the top ranked documents.

Table 3 some results of Meaning probability

Query	INA &Google	Google	Improvement %	INA &AltaVista	AltaVista	Improvement %
1	0.626	0.166	36%	0.292	0.094	20%
2	0.422	0.152	27%	0.218	0.129	9%
3	0.478	0.086	39%	0.281	0.086	20%
4	0.684	0.088	60%	0.268	0.146	12%
5	0.808	0.137	67%	0.303	0.2	10%
6	0.235	0.089	15%	0.364	0.09	27%
7	0.8	0.129	67%	0.558	0.14	42%
8	0.711	0.067	64%	0.144	0.067	8%
9	0.995	0.113	88%	0.405	0.098	31%
Average			51%			20%

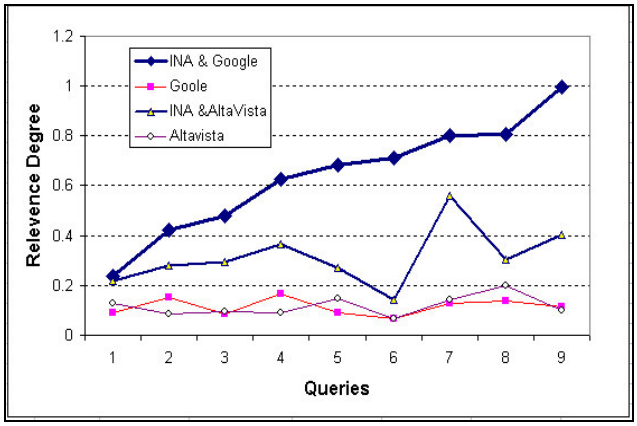

Figure 3 Improvement using CBIA with Google and AltaVista Search

6. Conclusion

In this paper, we have presented the development of a Context-Based Information agent for CITS—a Confidence Intelligent Tutoring System- to support a community of online learners in information need. The CBIA is based on a new approach to expanding queries, called a dominant meaning vector. It can supply more documents related to leaning sessions. This model is intend to develop a new probabilistic measure called a dominant meaning probability to measure the closeness between the original query concept and its dominant meaning in the documents rather than on the similarity between a query term and the terms of the documents. The experiments carried out on the two test collections show that our approach yields substantial improvements in retrieval effectiveness.

References

[1] Bharat, K. SearchPad: Explicit capture of search context to support web search. In Proceedings of the 9th International World Wide Web Conference, WWW9 (Amsterdam, May). 2000.

[2] Brown P.J., and Jones G.J.F., 'Exploiting contextual change in context-aware retrieval'. Proceedings of the 17th ACM Symposium on Applied Computing (SAC 2002), Madrid, ACM Press, New York, pp. 650-656, 2002.

[3] De Bra,P., "Adaptive Educational Hypermedia on the Web". Communications of the ACM, Vol. 45, No. 5, pp. 60-61, May 2002.

[4] Efthimiadis E. and Biron P. 'UCLA-Okapi at TREC-2: Query Expansion Experiments'. Proc. of the Second Text REtrieval Conference (TREC-2). NIST Special Publication pp. 500-515, 1994.

[5] Evans D., and Efforts R., 'Design and Evaluation of the CLARIT-TREC-2 system'. *Proceedings* of the Second Text Retrieval Conference (TREC-2), NIST Special Publication, pp. 516-532, 1994.

[6] Finkelstein L. , Gabrilovich E. , Matias Y. , Rivlin E. , Solan Z. , Wolfman G. , and Ruppin E. "Placing search in context: the concept revisited". ACM Transactions on Information Systems, Vol. 20, No. 1, pp 116–131, January 2002.

[7] Gale A. W., Kenneth W. C., and Yarowsky D., 'One sense per discourse'. Proceedings of the 4th DARPA Speech and Natural Language Workshop, 1992.

[8] Hang C., Ji-Rong W., Jian-Yun N., and Wei-Ying M.' Probabilistic Query Expansion Using Query Logs'. IW3C2 Honolulu, Hawaii USA. 2002.

[9] Jing, Y. and Croft, W.B., An association thesaurus for information retrieval, *RIAO '94*, pp. 146-160,1994.

[10] Luger George F. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Addison Wesley, 2002.

[11] Mitra M., Singhal A., and Buckley C. 'Improving automatic query expansion'. W.B. Croft, A., C.J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, pp. 206--214, August 1998.

[12] Peat, H.J., and Willett, P. 'The limitations of term co-occurrence data for query expansion in document retrieval systems'. J. of the ASIS, Vol. 42, No. 5, pp. 378-83, 1991.

[13] Qiu Y., and Frei H. 'Concept based query expansion'. Proceedings of the sixteenth annual international ACM SIGIR conference on Research and Development in Information Retrieval, July 1993.

[14] Razek M., Frasson, C., and Kaltenbach M. 'A Confidence Agent: Toward More Effective Intelligent Distance Learning Environments'. Proceedings of the international Conferencew on Machine Learning and Applications (ICMLA'02), Las Vegas, USA, pp.187-193, 2002.

[15] Razek M., Frasson, C., Kaltenbach M. “Using Machine Learning approach To Support Intelligent Collaborative Multi-Agent System”, International Conference on Technology of Information and Communication in Education for engineering and industry, TICE**,** 13,14,15 November 2002 LYON, France, 2002.

[16] Rhodes, B., and Starner, T. “The Remembrance Agent: A Continuously Running Automated Information Retrieval System”, In the Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96), London, UK, pp. 487-495, April 1996.

[17] Trausan-Matu S., Maraschi D., and Cerri S. A. ‘Ontology-Centered Personalized Presentation of Knowledge Extracted from the Web’, International Conference in Intelligent Tutoring Systems, Biarritz, Lectures Notes in Computer Science, Springer Verlag, pp 259-269, June 2002.

[18] Xu, J. and Croft W. B. Improving the effectiveness of information retrieval with local context analysis. ACM Transactions on Information Systems (TOIS), Vol. 18, No. 1, 2000.