www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 10 October, 2014 Page No.8782-8787

AI Web-Based Agent for Banks

Nikhila Kamat[1], Michelle D'cruz[2], Carmelita Dabre[3]

[1]Xavier Institute of Engineering, Mahim Causeway,
Opp. Raheja Hospital, Mahim, Mumbai - 400 016, India
nikhila300@gmail.com

[2]Xavier Institute of Engineering, Mahim Causeway,
Opp. Raheja Hospital, Mahim, Mumbai - 400 016, India
michelledcruz17@gmail.com

[3]Xavier Institute of Engineering, Mahim Causeway,
Opp. Raheja Hospital, Mahim, Mumbai - 400 016, India
carmelitadabre0@gmail.com

Abstract: ***Purpose of our application is to be used by banks for web-clients. Banks are always looking for more powerful and user-friendly applications to upgrade their existing websites. So, our hypothesis is that AI would perform on all platforms using artificial algorithms to analyze and understand user's queries relating to the bank's loan, account, policy etc. This shows that AI would perform as per design and expectation. No specific format would be required for the client to ask questions. For instance, the question "what is the interest on home loan?" can be framed as "Let me know about interest on home loan". It has integrated text-to-speech concept & graphical representation of a person speaking out answers. It also consists of ATM finder and Branch locator systems for other bank related help***.

Keywords: AI, banks, user-friendliness, Tree – Miner.

1. Introduction

1.1 Title and Interpretation

Let us understand what the title "AI WEB-BASED AGENT FOR BANKS" interprets –

AI - Our system will have two artificial algorithms TREE-MINER and PATTERN-MATCHER. It will use its in-built artificial intelligence to analyze client's queries and answer them intelligibly.

WEB-BASED - It is the application that will be developed for web-users and web-developers. It is a software package that will be accessed through the web browser. The software and database reside on a central server rather than being installed on the desktop system and is accessed over a network.

AGENT - An intelligent agent is a system that perceives its environment and takes actions which maximize its chances of success. The simplest intelligent agent is a program that solves specific problems and is called as logical agent. It will portray interaction and connection between the agent and the client.

FOR BANKS - It is the system that will be designed for the benefit of the bank. The bank will use it to automate operations.

1.2 Motive

AI known as Artificial Intelligence is a key component similar to the functioning of brain. After all, banking related questions will have to be dealt with consistency of brand image and its perception. Plus, it needs to be designed to interact with structured and unstructured data, and make use of it in a way that supports the experience. This gets exponentially more difficult as data increases. So AI should get 'smarter' by virtue of more data. Our AI system will try to understand what client wants to convey and immediately provide suitable answers to their queries related to bank. For example, if we ask "What is the time?" it will intelligently answer in this way – "Sorry, do you want the present time or the bank timing? If you want the bank timing, then it is from 9am to 5pm."

Aims:

1. Provide answers for decisions, repetitive tasks.
2. Holding of huge amounts of information.
3. Minimizing employee training costs.
4. Centralizing the decision making process.
5. Making things more efficient by reducing time to solve problems.
6. Combining various human expert intelligences w.r.t bank.
7. Reducing the number of human errors.

Objectives:

1. **Efficiency and Reliability:**
   System requires no upgrades since all new features are implemented on the server and automatically delivered to the clients. Maintaining the entire secured database on the server will be more efficient compared to storing all the customer data on the spreadsheet or physically in the record books.
2. **Cost-Effective Maintenance:**
   A compatible web browser application typically requires hardly any disk space on the client's system [6]. So there is little disk maintenance cost.
3. **Cross-Platform Compatibility:**
   Web applications integrate easily into other server-side web procedures. It provides cross-platform compatibility in most cases (i.e., Windows, Mac, Linux, etc.) because they operate within a web browser window.
4. **User-Friendliness:**
   System design and flow will have user-friendliness approach.
5. **Easy Accessibility:**
   Records and other information can be easily accessed and stored.

2. Existing System

We all know that retail banking is getting dissolved in bigger multi-national banks due to the increasing use of online banking [4].

2.1 Drawbacks of Existing System

2.1.1 Bigger Multi-National Banks

Some websites of banks like HDFC, ICICI, and Citi have a facility of FAQ dynamic page containing different forms linked to each other. Since the facility provided is clickable (radio buttons and drop down lists), it takes a lot of time for the client to get desired and precise information.

2.1.2 Smaller Banks

Adds upto three times training cost and employees' salary cost 24x7, 365 days operation which is not affordable in smaller banks. Also, there is no user-friendliness.

Instead, incorporation of AI Help Page on bank's website will give precise information to the clients by asking questions on one-to-one basis.

3. Proposed System

Our proposed system "AI WEB BASED FOR BANKS" gives one-to-one interaction between the clients and the agent. It provides user-friendliness and it is dynamic. The results obtained are precise. It doesn't take a lot of time for the client to get desired and precise information, compared to existing system. Banks can use & access our application from the server. Its simultaneous use by many clients is possible.

3.1 Modules

3.1.1 Client Registration/Login

There is a client registration/login page. It is only for valid and genuine clients. If logged in validly, then the client can proceed further, else not. Once the client has registered, he can login to access the Bank Website and ask queries to the AI agent.

3.1.2 Branch-Locator and ATM-Finder

The system also has these features incorporated in it. It is to make the clients convenient to find the nearest branch/ATM and find its location. Clients can get information about various branch locations by searching station or city name; and areas by matching area name.

3.1.3 Artificial Intelligence Help Page

AI will talk as a real banking operative. This agent provides all the necessary information asked by the client.

3.1.4 Effective GUI

There will be a graphical representation of the real person speaking out answers to the queries asked by the client.

3.1.5 Text-to-Speech

Once the query is asked by the client, the agent speaks out the answer. Also this feature provides the text of the answers spoken out, in case the client wants to copy it and save it in some document for future use.

3.2 Analysis

SWOT Analysis: SWOT is an acronym of Strengths, Weaknesses, Opportunities and Threats and as it suggests that it is not purely a method used for controlling areas of planning and risk, but it is also used to highlight areas of the project that could be maximized to the benefit of the whole project or individual areas where some competitive advantage may be gained [8].

Table 1: SWOT Analysis

Strengths	What are the business benefits of completing the project? Will the project require new technology?
Weakness	Is the proposed schedule realistic? What are the drawbacks of project?
Opportunities	Do competitors have any weaknesses? What are the latest industry trends?
Threats	Is there well-established competition already in the marketplace? Has new technology been fully tested?

Feasibility: It focuses on organizational objectives, current technology, budget, integration with other systems [3]. Its types are as under:

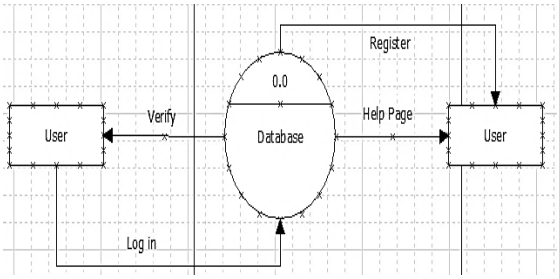
1. **Functioning:** Our project is operationally feasible because the time and personnel requirements are satisfied and processing methods are accepted by clients.
2. **Practical:** It is an excellent tool for trouble-shooting and long-term planning and lays emphasis on the project management and the overall coordination and technicality.
3. **Monetary:** It determines the positive economic benefits to the organization that the proposed system will provide.

Viability: It is the disciplined & documented process of thinking through an idea from its "Logical beginning to its Logical End!" Types of viability are:

1. **Managerial:** It provides information and guidelines to improve the overall project administration and project management plan [7].
2. **Schedule:** It is necessary to determine whether the specific deadlines are mandatory or desirable.
3. **Legal:** It is a measure of how well a solution can be implemented within existing legal/contractual obligations.
4. **Resource:** It involves questions such as how much time is available to build the new system, type and amount of resources required, dependencies and developmental procedures [2].

Desirability: It is a study made to decide whether or not the proposed system is "worthwhile" [9]. Its types are:

1. **Cultural:** It investigates all the environmental factors involved to successfully carry out a project. It evaluates the impact of the project on the local culture.
2. **Social:** It determines whether the proposed project will be satisfactory for the people or not and examines if the probability that project is accepted by the group of people.
3. **Safety:** It analyses if the project is capable of implementation and operated safely with minimum adverse effects [5].
4. **Market:** It takes into account the importance of the project in the selected area to determine whether it will support a specific development.



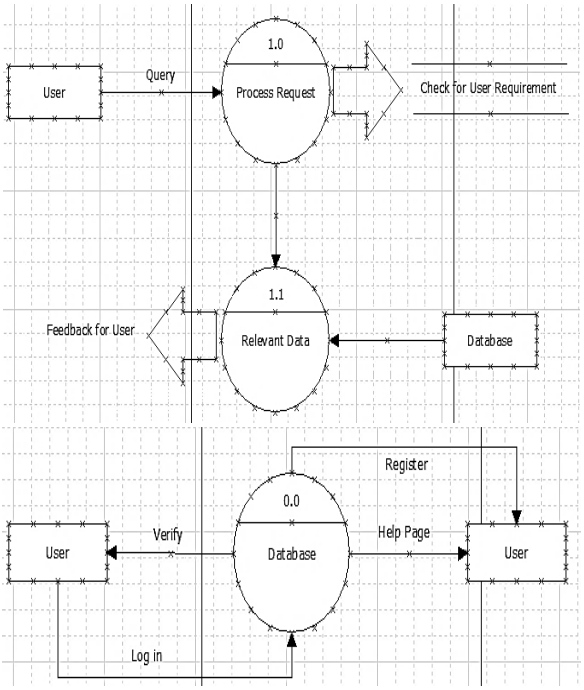


Figure 1: Data Flow Diagrams: Levels 0, 1, 2

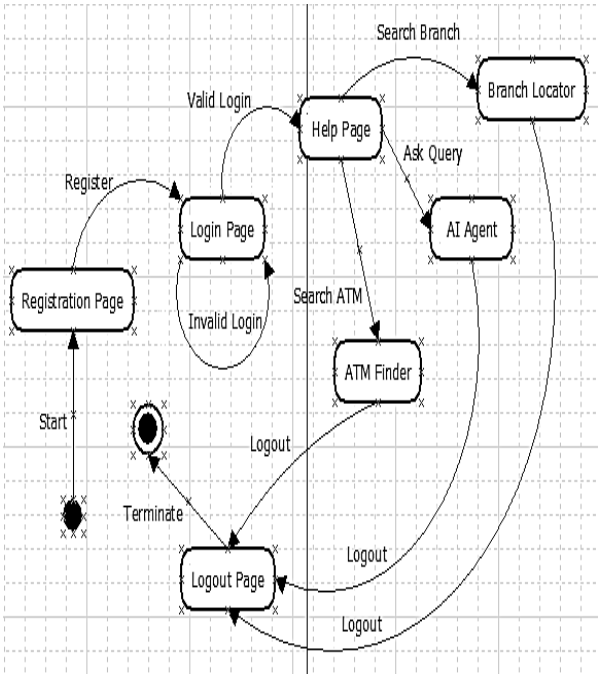


Figure 2: State-Chart Diagram

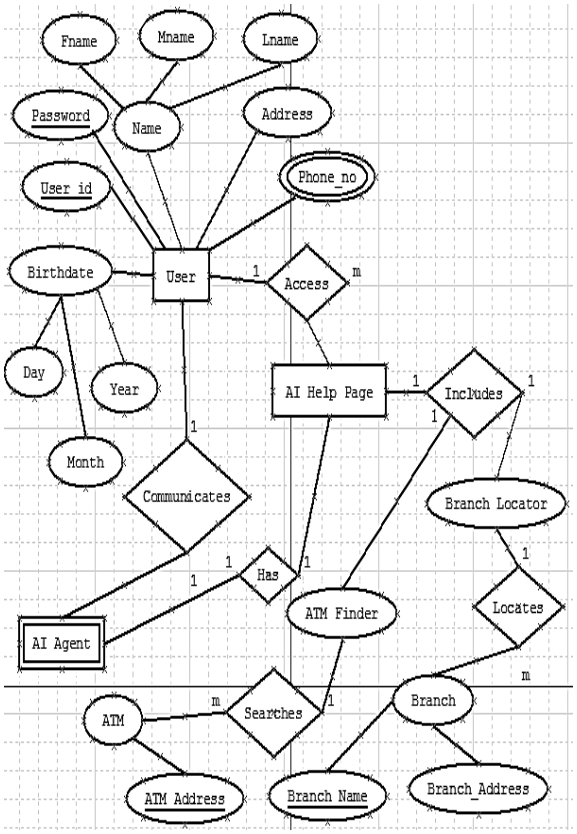


Figure 3: E-R Diagram

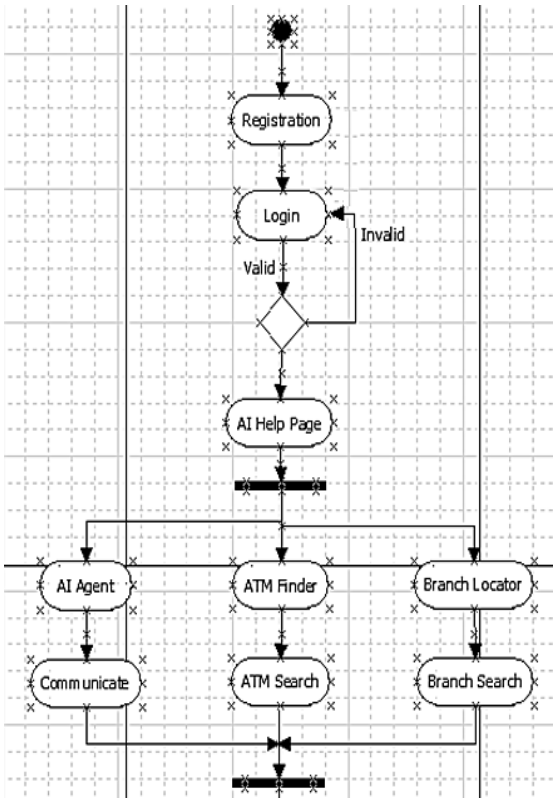


Figure 4: Activity Diagram

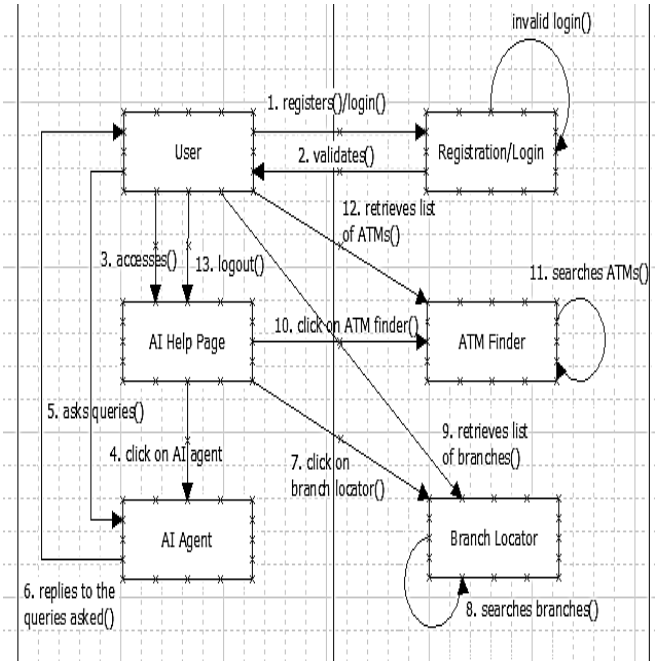


Figure 5: Collaboration Diagram

3.3 Algorithms

Tree–Miner Algorithm:

TREE-MINER(D, minsup):
F1 = {frequent 1-subtree};
F2 = {classes $[P]_1$ of frequent 2-subtrees};
for all $[P1]_1 \in E$ do enumerate-Frequent-Sub_i

ENUMERATE-FREQUENT-SUBTREES([P]):
for each element (x, i) $\in$ [P] **do**
$[P^i_x] = \acute{O}$;
for each element (y, j) $\in$ [P] **do**
R = {(x, i) x (y, j)};
L(**R**) = {L(x) x L(y)};
if for any $R \in$ **R**, *R* is frequent
then $[P^i_x] = [P^i_x] \cup \{R\}$;
Enumerate-Frequent-Subtrees($[P^i_x]$);

TREE-MINER performs depth-first search (DFS) for frequent subtrees, [1] using a novel tree representation called scope-list for fast support counting, as discussed below:

Scope-List Representation: Let X be a k-subtree of a tree T. Let x_k refer to the last node of X. We have used the notation L(X) that refers to the scope-list of X. Each element of the scope-list is a triple (t, m, s), where t is a tree id (t_{id}) in which X occurs, m is a match label of the (k-1) length prefix of X, and s is the scope of the last item x_k. Recalling prefix match label, gives the positions of nodes in T that matches the prefix. Since the given prefix can occur multiple times in a tree, X can be associated with multiple match labels as well as multiple scopes. The initial scope-lists are created for single items (i.e., labels) i that occur in a tree T. Since a single item has an empty prefix, we don't have to store the prefix match label m for

single items. FIG 3.4.1.1 shows a database of three trees, along with the horizontal format for each tree and the vertical scope lists format for each item. Consider Item 1; since it occurs at node position 0 with scope [0; 3] in tree T_0, we add (0; [0; 3]) to its scope list. Item 1 also occurs in T_1 at position n_1 with scope [1; 3], so we add (1; [1; 3]) to L(1). Finally, 1 occurs with scope [0; 7] and [4; 7] in tree T_2, so we add (2; [0; 7]) and (2; [4; 7]) to its scope-list. In a similar manner, the scope lists for other items are created.

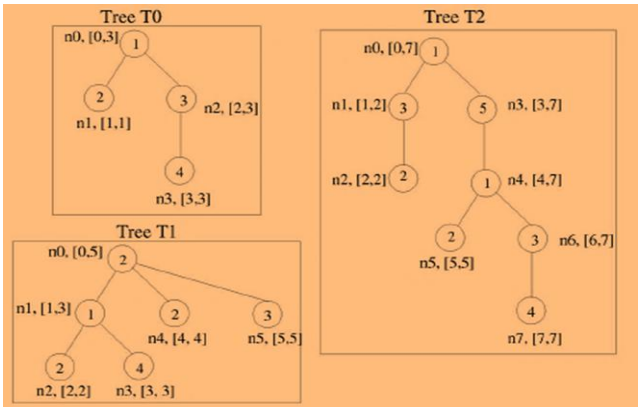


Figure 6: Database 'D' Of 3 Trees

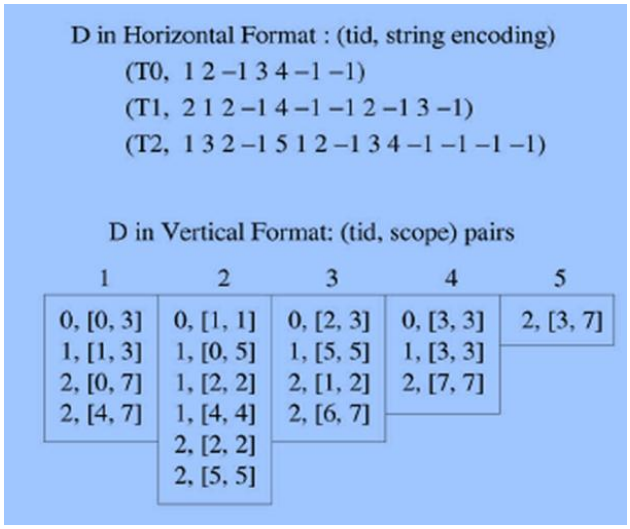


Figure 7: Format of 'D'

Pattern –Matcher Algorithm:

PATTERN-MATCHER(D, minsup):
F1 = {frequent 1-subtree};
F2 = {classes $[P]_1$ of frequent 2-subtrees};
for (k=3; $F_{k-1} \neq Ø$; k=k+1) **do**
C_k = {classes $[P]_{k-1}$ of candidate k-subtrees};
for all trees T in D **do**
Increment count of all S <= T, S Є $[P]_{k-1}$
C_k = {classes of frequent k-subtrees};
F_k = {hash table of frequent in C_k};
Set of all frequent = $U_k F_k$;

Keyword - Pattern – Matching: Matching the prefix P of a class in a leaf against the tree T is the main step in support counting. Let X[i] denote the i^{th} node of subtree X and let X[i; . . . ; j] denote the nodes from positions i to j. We use a recursive routine to test prefix matching. At the r^{th} recursive call, we maintain the invariant that all nodes in P[0; 1; . . . ; r] have been matched by nodes in $T[i_0; i_1; . . . ; i_r]$, i.e., prefix node P[0] matches $T[i_0]$, P[1] matches $T[i_1]$, etc., and, finally, P[r] matches $T[i_r]$. Note that while nodes in P are traversed consecutively, the matching nodes in T can be far apart. If P < = T, we search for a match in T for each element (x; k) Є [P] by searching for x starting at the subtree T[i-1]. (x,k) is either a descendant or embedded sibling of P[k 1]. Either check takes complexity O(1) time. If a match is found, the support of the element (x; k) is incremented by one. For counting support (at least one occurrence in T), the count is incremented only once per tree, or else, for weighted support (all occurrences in T), we continue the recursive process until all matches have been found [1].

For instance, if the user enters a sentence "I want to know information on home loan." AI agent will pick out the keyword "home loan" and do keyword-pattern-matching to give precise answer to the user immediately and suitably.

4. Conclusion

Artificial intelligence is already very much a part of everyday life in industrialized nations. AI is helping people in every field make better use of information to work smarter, not harder. New applications of AI, which include Intelligent Agents, are providing new areas of research. AI holds the promise of overcoming the difficulties and playing a major role in the widespread promulgation of new services [9]. AI-driven customer care solutions can positively affect the bottom line, make customers happy, and free up the organization resources to higher-level objectives. AI systems have no sentient intentions to make art, but could be considered as such if a human had made them. New advances in AI are opening a door to new game genres and even new game paradigms.

References

[1] Mohammed J. Zaki, "Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, AUG 2005.

[2] L. De Raedt, and C. Helma, "Molecular Feature Mining in HIV Data," Proc. Int'l Conf. Knowledge Discovery and Data Mining, 2001.

[3] J.N. Kok and S. Nijssen, "Efficient Discovery of Frequent Unordered Trees," Proc. First Int'l Workshop Mining Graphs, Trees, and Sequences, 2003.

[4] U. Ruckert, "Frequent Free Tree Discovery in Graph Data," Special Track on Data Mining, Proc. ACM Symp. Applied Computing, 2004.

[5] Y. Chi, Y. Yang, "HybridTreeMiner: An Efficient Algorithm for Mining Frequent Rooted Trees and Free Trees

Using Canonical Forms," Proc. Int'l Conf. Scientific and Statistical Database Management, 2004.

[6] J. Han, "CloseGraph: Mining Closed Frequent Graph Patterns," ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, Aug. 2003.

[7] G. Karypis and M. Kuramochi, "An Efficient Algorithm for Discovering Frequent Subgraphs," IEEE Trans. Knowledge and Data Eng. Sept. 2004.

[8] B. Moon and Q. Li, "Indexing and Querying XML Data for Regular Path Expressions," Proc. 27th Int'l Conf. Very Large Data Bases, 2001.

9] X. Yan, "gSpan: Graph-Based Substructure Pattern Mining," Proc. IEEE Int'l Conf. Data Mining, 2002.