Volume: 5 Issue: VII Month of publication: July 2017

DOI:

Implementation of Banking Support System Using Naive Bayes Algorithm

Shivam Agrawal[1,] Prof. Christy Jackson[2]
[1]*School of Computer Science and Engineering Vellore Institute of Technology Chennai, Tamil Nadu*
[2]*School of Computer Science and Engineering Vellore Institute of Technology Chennai, Tamil Nadu*

Abstract: Earlier the bank had a system called Banking Support Centre(BSC). The BSC system is also called as Standard Bancs Link(SBL). The objective of this project is to allow building a low maintenance web based system that is practical and easy to use, as the clients requirements keep changing. BSS should be designed in such way that, its all core functionalities to be working as per the clients requirements. This project has been designed to support all popular browsers (IE 11, Firefox, Chrome) and supports screen sizes from 5 inch onwards. The project provides a protocol for designing building blocks called Plugins. The Plugins could then be put together as a Graph (represented as an XML) to build a specific workflow.

Keywords: Banking Support System(BSS), SBL, OneSearch, Notifications, Error Partial.

I. INTRODUCTION

Earlier the bank had a system called Banking Support Centre(BSC). The BSC system is also called as Standard Bancs Link(SBL). SBL is used in business support centers and back office operations for the day to day operations. SBL framework is getting obsolete as it is not supported by browsers beyond IE11 as it uses old technologies such as, HTML and JavaScript. So, the bank is upgrading the system, by re-building the system using MVC Architecture.

Banking Support System(BSS) is a website created for business support centers and back office operations for the day to day operations. The website includes various operations like Rescheduling loans, Debt review, Customer Enquiry and few other banking operations related to retail banking. A customer will approach a consultant in the bank who can use BSS to perform banking operations, as per the customers requirements.

The objective of this project is to allow building a low maintenance web based system that is practical and easy to use, as the clients requirements keep changing. Banking Support System(BSS) is designed in such way that, its all core functionalities are working as per the clients requirements. This project has been designed to support all popular browsers (IE 11, Firefox, Chrome) and supports screen sizes from 5 inch onwards. The project provides a protocol for designing building blocks called Plugins.

A Plugin is a basic building block of a workflow. A plugin expects some data (in the form of a Model object(s)) as input and returns one or more Model object(s) as output.

The Input to a plugin could be an outcome of another plugin. Model dictionary and Global Data Dictionary is used by the navigation utility to store and forward the data to various plugins during the execution of a workflow. The Plugins could then be put together as a Graph (represented as an XML) to build a specific workflow.

A. OneSearch Framework

One Search framework, is an Advanced Search Box used to retrieve the information based on the number of parameters given, such as CIF number (1 Parameter) or Address and CIF number (2 Parameters) and so on.

B. Notification Framework

Notification Framework is a notification bar, which is meant to display notifications from the framework or from external applications. The notifications could be just a message OR could contain a link to trigger another workflow. It is a REST (REpresentational State Transfer) based service known as RESTFul Web Service where data travels over HTTP Protocol.

C. ErrorPartial Framework

The Error Partial is meant to display error message partially on the screen based on the Transaction Number. For example, error messages, such as 404 HTTP error or Server error, Error can also be based on login, log off or while triggering a workflow or performing a task.

II. LITERATURE SURVEY

On the clients side, they wanted to simplify the entire process banking operations. They wanted one system which would cater to all their needs. They created a system called Banking support Center (BSC) or StandardBANCSLink (SBL). It follows an n-tier architecture (Figure:2.1). It has been developed using ASP .Net, HTML and JavaScript. This system is in turn connected to various other systems like BANCS, XPression, Documentum and VMAX. The BANCS is used for the transactions and The VMAX System is used only when a loan reschedule needs to take place. Xpression and Documentum are used to generate reports and take printouts etc.

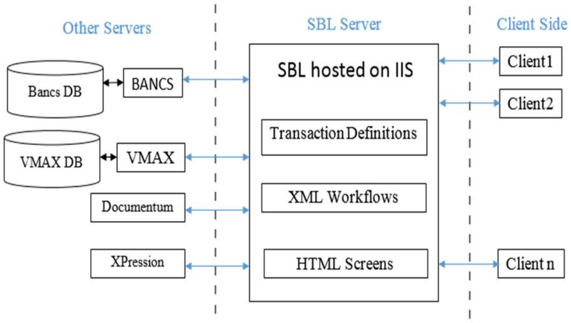

Figure. 1. System Architecture

Bancs Simulator has been designed to interact with the Bancs. User can send values and transmit it through the Bancs simulator , which processes the same and the byte stream hits the Bancs to get the proper output. It sends the same form of input what SBL forms and transfer to Bancs.

The main advantage of using the simulator is to form the input data mapped with the XML and check the output result sent by Bancs without consuming the resource of SBL or any other E-channels. Bancs Simulator is designed based on Model-View-Controller(MVC) architecture. It was designed in this pattern so as to bring about better and easier management and to allow code reusability.

ModelViewController (MVC) is a software architectural pattern for implementing user interfaces on computers. It divides a given software application into three interconnected parts, so as to separate internal representations of information from the ways that information is presented to or accepted from the user.

Microsoft SQL Server is a relational database management system developed by Microsoft. As database server, it is a software product with the main function of storing and re- trieving data requested by other software applications that can operate either on the same computer or on another computer in a network (including the Internet) Microsoft has at least a dozen different editions of Microsoft SQL Server, for different audiences and workloads ranging from small single-computer applications to major Internet-facing applications with many concurrent users.

HyperText Markup Language, HTML generally short, is the standard markup language used to create web pages. With CSS, and JavaScript, HTML is a cornerstone technology used to create Web pages, as well as to create user interfaces for web and mobile applications. Web browsers can read HTML files and make them visible or audible web pages.

A workflow consists of an orchestrated and repeatable pattern of activity enabled by a systematic organization of resources in processes that transform materials, provide ser- vices or process information. It can be represented as a sequence of operations, said the work of one person or a group, organization, personnel, or one or more simple or complex mechanisms.

From the perspective of more abstract level or higher, workflow can be considered as a view or representation of actual work. The flow being described can refer to a document, product or service that is transferred from one stage to another.

SBL framework is a component based framework. It is divided into several building blocks which could be joined together in the form of a graph. A work flow is an XML rep- resentation of the graph FNS has created a custom application (WorkFlow Designer) to visually create, update the workflows.

The WorkFlow Designer allows the developer to use the custom built-in components and add new components as well. For one to add a new component, the coding would be done in C# and once it is built the .dll file will be added in the bin folder of the SBL server, and then theyd be able to use it like a built-in component. Once the workflow has been created, it will be exported as an XML.

III. SYSTEM DESIGN

To solve the complex problem, the system is divided into smaller parts called modules. Each modules can be devloped either individually or in parallel, each module interlinked with eachother i.e., the output of one module is the input for the other module. It comprises of four modules namely Screens (One Search, Notification, Error Partial), Workflow, BSS Framework, Transactions

A. Screens

The screens developed for this project are in HTML. The various screen which are developed are OneSearch, Notifi- cation and ErrorPartial. The screen validations are done in Javascript & AngularJS and the database/3rd party system interactions happen on the server. .Net is used on the server side to achieve the same. All the screens developed needs to comply with certain standards and guidlines laid down by the client. The flow of the screen is not determined by the screen itself, but by the workflow.

B. Workflow

SBL framework is a component based framework. It is divided into several building blocks which could be joined together in the form of a graph. A work flow is an XML rep- resentation of the graph. FNS has created a custom application (Workflow designer) to visually create, update the workflows. Once the workflow has been created, it could be exported as an XML.

A workflow is nothing but a graphical representation of the flow of the system. It is very similar to the concept of flow charts. It also has a start and stop node. It can also handle multiple switch statements and if conditions. Each element is either a component or a connector. Other than the available components, we could create more components. The components are created in C# and then after the build of the C class, the .dll file is placed in the XML workflow folder of the SBL system.

C. BSS Framework

The bank has a new system called BSS. BSS is a website created for business support centres and back office operations for the day to day operations and it

is an up-gradation of Banking Support Center(BSC) which is being still used by the bank. BSS Consists of multiple features which are newely added to the system namely Search, Notification, BusinessToken, Error Partial and many more. BSS follows MVC (Model View Controller) Architecture. It is a software architectural pattern for implementing user interfaces on computers.

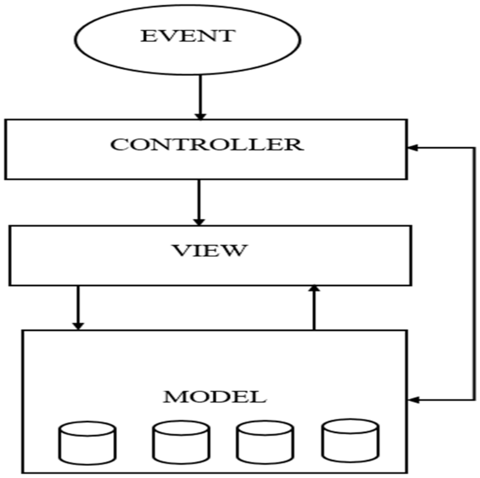

Figure. 2. Architecture of MVC

MVC architecture has been used by the BSS, the advan- tages of this approach is, It gives logical bifurcation of our business functionality and models. MVC uses Front Controller approach. That approach means a common controller for all pages processes the request. Very clean separation of concerns. View and Controller are neatly separated. The proposed system would look like Figure 3

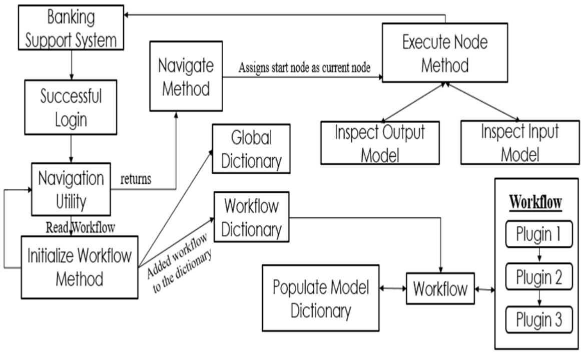

Figure. 3. System Architecture for BSS

D. Transactions

The front end connects to Bancs Host. It sends a byte string containing the input taken from the screen. BANCS sends a byte String Back to the frontend. Once the response is sent the connection is closed. Each message cannot be greater than 984 bytes. A Send or Receive message to Bancs is called as a Transaction. Each Transaction has a Number associated with it. Message sent is referred as TX Transaction. Data Received via bancs is called as an RX Tran.

IV. PROPOSED SYSTEM

A. OneSearch Module

1) *One Search Autocomplete:* The AutoComplete widget provides suggestions depending on the typed text. It also allows multiple value entries. The

suggestions shown by the widget can come from a local array or from a remote data service. The autocomplete is designed using
a) Kendo UI Autocomplete.

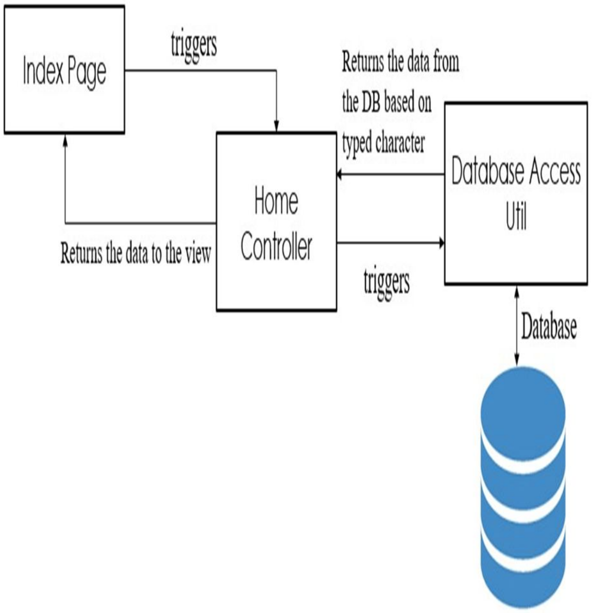
Index Page
triggers
Returns the data from the DB based on typed character
Database Access Util
Home Controller
Returns the data to the view
triggers
Database

Figure. 4. One Search Autocomplete

When the users enter a text on the Autocomplete search box, the index page triggers the Home Controller. Home Con- troller then in-turn triggers DataAccessLibrary which consists of a function called Update Search Events. Update Search Events is a function which retrieves the data from the database based on the text typed and returns the results to the Home Controller. Home Controller returns the data to the search box (i.e., index page), which performs the autocomplete function- ality on the textbox.

2) Display One Search Results

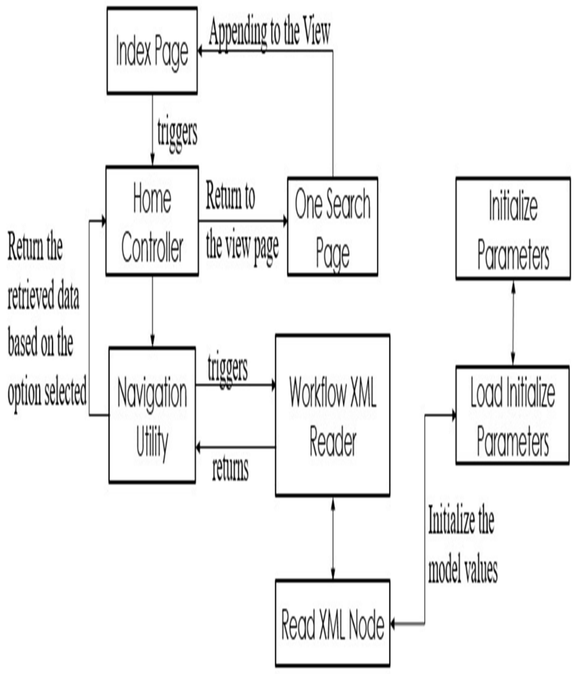

Figure. 5. Display One Search Results

When the user has selected an option from the autocomplete widget or if the user has entered the keyword **Search:** following with CIF Number and respected option and clicks on Submit button. On click of submit button the Home Controller Triggers Navigation Utility (i.e., DisplaySearchResults -> GetInitializeParameterFromWorkflow) and returns the results back to the Home Controller. The respected WorkflowName or the CommandName with the CIF number and Option are displayed dynamically. Clicking on the CommandName will trigger the following workflow and if the values were given then it is inherited in the fields.

GetInitializeParameterFromWorkflow takes WorkflowPath and EntryNode as input parameters. GetInitializeParameterFromWorkflow triggers ReadXmlNode with WorkflowPath as an input parameter and returns the results back to the function (i.e., GetInitializeParameterFromWorkflow). A loop istriggered in the GetInitializeParameterFromWorkflow checking the values based on the list created in LoadInitializeParameter (i.e., InitParameters), the WorkflowName and the WorkflowEntryNode is assigned. The assigned values are added to the Dictionary called InitializeParametersDictionary.

The ReadXmlNode reads the Xml with WorkflowPath as an input parameter. ReadXmlNode selects nodes based on the plugins which consists of Initialize Parameter. The selected nodes are checked if the nodes are not empty and if the nodes have child nodes. If the condition is satisfied, ReadXmlNode initializes the model values to LoadInitializeParameter and is triggered with list values as input parameters.

LoadInitializeParameter is used to assign the model (i.e., InitializeParameter) with the child node values which was retrieved from the ReadXmlNode. The assigned values are added to the list, which is created using InitializeParameter (i.e., Workflow Model) called InitParameters. Once the InitPa- rameters contains the model values, LoadInitializeParameter returns to ReadXmlNode. ReadXmlNode returns to GetInitial- izeParameterFromWorkflow with WorkflowName and Work- flowEntryNode.

B. Notification Module

The notification bar is meant to display notifications from the framework or from external applications. The notifications could be just a message OR could contain a link to trig- ger another workflow. It is a REST(REpresentational State Transfer) based service known as RESTFul Web Service where data travels over HTTP Protocol.

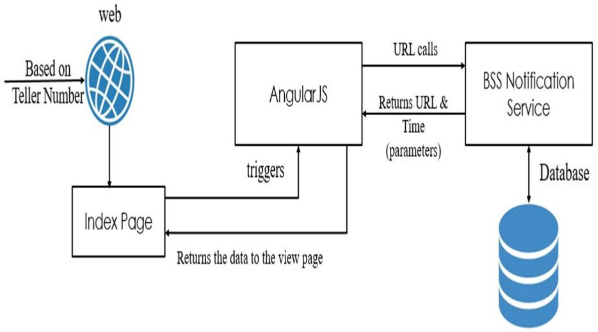

Figure. 6. Notification Framework

A web service is a generic term for a software function that is hosted at a network addressable location. Services are designed using SOA(Service Oriented Architecture). SOA is a style of software design where services are provided to the other components by application components, through a communication protocol over a network. The basic principles of Service Oriented Architecture are independent of vendors, products and technologies. The service is Asynchronous in nature. The service is called every 10 seconds and the service URL and TIME comes from the WorkflowXmlSettings file.

C. ErrorPartial

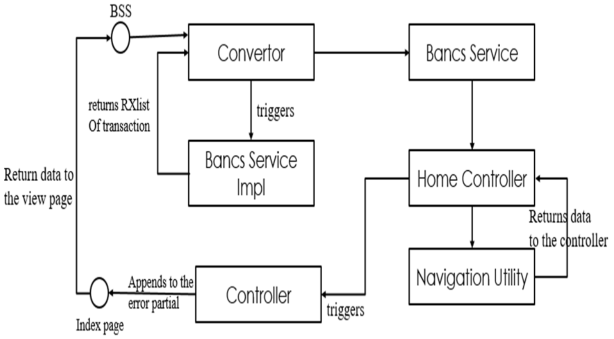
BSS
Convertor
Bancs Service
returns RXlist Of transaction
triggers
Bancs Service Impl
Return data to the view page
Home Controller
Returns data to the controller
Navigation Utility
Appends to the error partial
Controller
triggers
Index page

figure. 7. Error Partial Framework

The Error Partial is meant to display error message partially on the screen based on the Transaction Number. For example error messages such as 404 HTTP error or Server error, Error can also be based on login, log off or while triggering an workflow or performing a task. The error messages will be displayed based on the transaction number, For example transaction number for Login is 009001, if the user enters the wrong password the login transaction will not be fired and the error message will be displayed, which will be retrieved from the respected controller(i.e., Login Controller) .

The Transaction Numbers are fixed for every action per- formed, such as Login[009001], Log Off [009003] and so on. The Error Partial plays an major role in the Banking Support System(BSS) i.e., if the system has a got failure in performing any component an error message will be displayed stating why the action is not performed.

D. Screens

The screens were designed logically, but the flow of the screens is decided by the WorkFlow. The screens by itself have no actual functionality. The main screens developed for this project are broadly classified into the following

1) *One Search Autocomplete Screen:* The One Search Autocomplete is desined using KendoUIAuto- complete. It retrieves various details like workflows, workflows associated with 1 parameter, workflows associated with 2 parameters and so on, based on the command name and the results of the autocomplete can be viewed as a drop-down list. Kendo UI framework is a comprehensive set of commer- cial user interface widgets, dependent on jQuery and writ- ten in JavaScript, to be used in the development of web- sites/applications that run on the web platform. In short, the Kendo UI toolkit of widgets provides UI solutions for

showing and acquiring data by way of common user interface widget patterns (for example, data grids, calendar selection, autocomplete and so on).

2) *One Search View Page Screen:* This screen helps to renders all the workflow(s) or a particular workflow based on the command name on the screen and triggering the respected workflow on clicking the workflow name.
3) *Notification Menu View:* The Menu is designed using HTML and CSS. The calls which are made to the Services is using AngularJS. AngularJS acts as a Middle ware. AngularJS is a very powerful JavaScript Framework. It is used in Single Page Application (SPA) projects. It can be added to an HTML page with a <script> tag. AngularJS extends HTML attributes with Directives, and binds data to HTML with Expressions.
4) *Error Partial Screen:* received. The RXTransaction 009003 renders a new screen 009003 which acts as the Log Off Page. This screen allows the user to exit the Banking Support System (BSS) application with an 'OK' message.

V. NAIVE TEXT CLASSIFICATION ALGORITHM

The first supervised learning method we introduce is the multinomial Naive Bayes or multinomial NB model, a proba- bilistic learning method. The probability of a document *d* being in class *c* is computed as
The screen is designed using HTML, CSS and JQuery. This
screens helps to render error messages on the screen, based on transaction number.

$$P(c|d) \propto P(c) \prod_{1\leq k\leq n_d} P(t_k|c)$$

Equation. 1

E. Workflow

Each screen is associated with one or more workflow. The workflow decides the flow of the screen. Once the client lands on the One Search View Page, the clients have to enter the command name with CIF Number and Option or the workflow name to be searched. The Notification View, they get various options like simple message, link message which trigger another workflow, clear all button, time duration of the notification recieved. The error partial displays an error message based on the transaction number, i.e., if an transaction or the screen could not be fired, it displays an partial message on the screen stating where the error have occured. To achieve the goal of this project we have created a workflow called Centralized Reschedule workflow. With many custom built components like Kendo Autocomplete, Display- OneSearchView, Notification Serices and ErrorPartial.

F. Transactions

1) *Login:* StandardBancsLink does not have access to the database directly. Access to the database happens using a middleware called BANCS. Transaction is a function performed on BANCS. Each transaction has a transaction number. In the workflow, when screen 009001 is rendered, a TX- Transaction 009001 is shot and a RXTransaction 032026 is received. The RXTransaction 032026 renders a new screen 032026 which acts as the Login Page. This screen allows the user to enter the Banking Support System(BSS) application. where $P(t_k|c)$ is the conditional probability of term (t_k) occur- ring in a document of class *c*.We interpret $P(t_k|c)$ as a measure of how much evidence (t_k) contributes that *c* is the correct class. $P(c)$ is the prior probability of a document occurring in class *c*. If a document's terms do not provide clear evidence for one class versus another, we choose the one that has a higher prior probability. $(t_1, t_2, \ldots, t_{n_d})$ are the tokens in *d* that are part of the vocabulary we use for classification and n_d is the number of such tokens in *d*. For example, $(t_1, t_2, \ldots, t_{n_d})$ for the one-sentence document Beijing and Taipei join the WTO might be *(Beijing, T aipei, join, WTO)*, with $n_d = 4$, if we treat the terms and and the as stop words. In text classification, our goal is to find the best class for the document. The best class in NB classification is the most likely class c_{map}: In text classification, our goal is to find the best class for the document. The best class in NB classification is the most likely class c_{map}:$c_{map} =$

$$argmax_{c\in C}\hat{P}(c|d) = argmax_{c\in C}\hat{P}(c) \prod_{1\leq k\leq n_d} \hat{P}(t_k|c)$$

Equation. 2

We write $\hat{P}$ for P because we do not know the true values of the parameters $P(c)$ and $P(t_k|c)$, but estimate them from the training set as we will see in a moment. In Equation 2, many conditional probabilities are multiplied, one for eachposition $1 \leq k \leq n_d$. This can result in a floating point
underflow. It is therefore better to perform the computation
by adding logarithms of probabilities instead of multiplying probabilities. The class with the highest log probability score is still the most probable; log(xy) = log(x) + log(y) and the logarithm function is monotonic. Hence, the maximization that is actually done in most implementations of NB is:
In the workflow, when screen 009001 is rendered, a TXTransaction 009001 is shot and a RXTransaction 009001 is received. The RXTransaction 009001 renders to same

$$c_{map} = argmax_{c \in C} [log \hat{P}(c) +$$

$$1 \leq k \leq n_d$$

Equation. 3

$log \hat{P}(t_k|c)]$
screen(Login page) and displaying an error message as Failed to Process Transaction. The RXTransaction 032026 is not fired and resulting to display an Error Message on the screen.

2) Log Off: StandardBancsLink does not have access to the database directly. Access to the database happens using a middle-ware called BANCS. Transaction is a function performed on BANCS. Each transaction has a transaction number.

In the workflow, when screen 009003 is rendered, a TX- Transaction 009003 is shot and a RXTransaction 009003 is
Equation 3 has a simple interpretation.Each conditional parameter log $\hat{P}(t_k|c)$ is a weight that indicates how good an indicator t_k is for c. Similarly, the prior log $\hat{P}(c)$ is a weight that indicates the relative frequency of c. More frequent classes are more likely to be the correct class than infrequent classes. The sum of log prior and term weights is then a measure of how much evidence there is for the document being in the class, and Equation 3 selects the class for which i have the most evidence.

We will initially work with this intuitive interpretation of the multinomial NB model and defer a formal derivation.

How do we estimate the parameters $\hat{P}(c)$ and $\hat{P}(t_k|c)$? We first try the maximum likelihood estimate (MLE; probtheory), which is simply the relative frequency and corresponds to the most likely value of each parameter given the training data. For the priors this estimate is:

$$\hat{P}(c) = \frac{N_c}{N}$$

Equation. 4

where N_c is the number of documents in class c and N is the total number of documents. We estimate the conditional

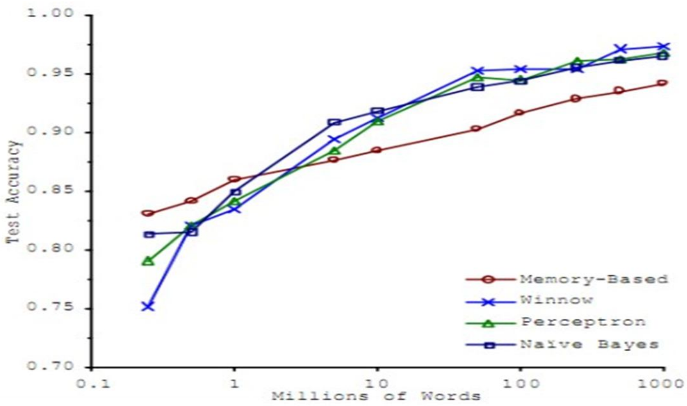

figure. 8. Training and test times for NB

probability $\hat{P}(t|c)$ as the relative frequency of term t in documents belonging to class c:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

Equation. 5

where T_{ct} is the number of occurrences of t in training docu- ments from class c, including multiple occurrences of a term in a document. We have made the positional independence assumption here, which we will discuss in more detail in the next section: T_{ct} is a count of occurrences in all positions k in the documents in the training set. Thus, we do not compute different estimates for different positions and, for example, if a word occurs twice in a document, in positions k_1 and k_2, then $\hat{P}(t_{k_1}|c) = \hat{P}(t_{k_2}|c)$. The problem with the MLE estimate is that it is zero for a term-class combination that did not occur in the training data. If the term WTO in the training data only occurred in China documents, then the MLE estimates for the other classes, for example UK, will be zero:

$$\hat{P}(WTO|UK) = 0$$

Equation. 6

Now, the one-sentence document Britain is a member of the WTO will get a conditional probability of zero for UK because we are multiplying the conditional probabilities for all terms in Equation 1. Clearly, the model should assign a high probability to the UK class because the term Britain occurs. The problem is that the zero probability for WTO cannot be "conditioned away," no matter how strong the evidence for the class UK from other features. The estimate is 0 because of sparseness : The training data are never large enough to represent the frequency of rare events adequately, for example, the frequency of WTO occurring in UK documents.

To eliminate zeros, we use add-one or Laplace smoothing, which simply adds one to each count:

$$\hat{P}(t|c) = \frac{T_{ct}+1}{\sum_{t' \in V}(T_{ct'}+1)} = \frac{T_{ct}+1}{(\sum_{t' \in V} T_{ct'})+B'}$$

Equation. 7

where $B = |V|$ is the number of terms in the vocabulary. Add-one smoothing can be interpreted as a uniform prior (each term occurs once for each class) that is then updated as evidence from the training data comes in. Note that this is a prior probability for the occurrence of a term as opposed to the prior probability of a class which i estimate in Equation 4 on the document level. I have now introduced all the elements we need for training and applying an NB classifier. The complete algorithm is described in Figure 8

	mode	time complexity	
	training	$\Theta(\lvert D\rvert L_{ave} + \lvert c\rvert\lvert V\rvert)$	
	testing	$\Theta(L_a + \lvert C\rvert M_a) = \Theta(\lvert C\rvert M_a)$	

Figure. 9. Graph Plotting for Naive Bayes Text Classification

VI. CONCLUSION

Banking Support System(BSS) is a website created for business support centres and back office operations for the day to day BSS includes operations like One Search which lets the user to search for a specific workflow given the keyword, Notification is a bar which is meant to display notifications from the framework or from external applications, Error Partial is meant to display error message partially on the screen based on the Transaction Number and few other banking operations related to retail banking. I have met the objective of this project, which is to create the front-end and back-end for One Search, Notification and Error Partial which helps the consultant to search specific workflow or account or CIF number with ease, get notifications based on the teller number and Error Partial displays an error message, if an workflow is not triggered or an user has entered the wrong password.

REFERENCES

[1] Vivek, Shivangi Pandey An Intelligent Information System based on Ontology (IISO) :From Febraury 2015, volume 5, Issue 2, International Journal of Advanced Research in Computer Science and Software Engineering.

[2] S. M. Patil, D. M. Jadhav Semantic Search using Ontology and RDBMS for Cricket :From International Journal of Computer Applications (0975 8887) Volume 46 No.14, May 2012.

[3] Patrick Lambrix et al, Dwebic :An Intelligent Search Engine based on Default Description Logics-1997.

[4] K. Satya Sai Prakash and S. V. Raghavan Intelligent Search Engine: Simulation to Implementation, In the proceedings of 6th International conference on Information Integration and Web-based Applications and Services (iiWAS2004), pp. 203-212, September 27 - 29, 2004, Jakarta, Indonesia, ISBN 3-85403-183-01.

[5] Isa, B. S., Boyle, J. M., Neal, A. S., and Simons, R. M. A methodology for objectively evaluating error messages. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (New York, NY, USA, 1983), CHI 83, ACM, pp. 6871

[6] Kummerfeld, S. K., and Kay, J. The neglected battle fields of syntax er- rors. In Proceedings of the Fifth Australasian Conference on Computing Education Volume 20 (Darlinghurst, Australia, Australia, 2003), ACE 03, Australian Computer Society, Inc., pp. 105111.

[7] Marceau, G., Fisler, K., and Krishnamurthi, S. Measuring the effective- ness of error messages designed for novice programmers. In Proceedings of the 42Nd ACM Technical Symposium on Computer Science Education (New York, NY, USA, 2011), SIGCSE 11, ACM, pp. 499504.

[8] Marceau, G., Fisler, K., and Krishnamurthi, S. Mind your language: On novices interactions with error messages. In Proceedings of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software (New York, NY, USA, 2011), Onward! 2011, ACM, pp. 318.

[9] TCS BaNCS Research Journal-Issue1

[10] TCS BaNCS Research Journal-Issue2

[11] Alex Ferrara, M. M. (2002).Programming .NET Web Ser-vices. OReilly Media, United States of America.

[12] Awad, E. M. (1997).Systems Analysis and Design. Tata McGrawHill Publication, New Delhi, India.[Leonard Richardson2007] Leonard Richardson, S. R. (2007).RESTful Web Ser-vices. OReilly Media, United States of America.

[13] Prasad, D. K. (2005).Data Warehouse Development Tools. Dream-tech publisher, New Delhi, India.

[14] Pressman, R. S. (1997).Software Engineering. Tata McGrawHill Publi- cation, New Delhi, India.

[15] Spaanjaars, I. (2010).Beginning ASP.NET 4: In C and VB.Wiley Publishing, Indianapolis, Indiana.