ADAPTIVE INTELLIGENT PERSONAL ASSISTIVE APPLICATION

ENROLMENT NO.	-	13103410
STUDENT NAME	-	SIDDHARTH AGARWAL
ENROLMENT NO.	-	13103414
STUDENT NAME	-	AKANKSHA SINGH
NAME OF SUPERVISOR	-	MR. PRASHANT KAUSHIK

DECEMBER 2016

SUBMITTED IN PARTIAL FULFILMENT OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING & INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I)

TABLE OF CONTENTS

(II)

DECLARATION

We hereby declare that this submission is my/our own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Jaypee Institute of Information Technology, Noida, Sector 62

Signature: ___________ _____________

Date: 21st December, 2016

Name: Siddharth Agarwal

Enroll. No: 13103410

Name: Akanksha Singh

Enroll. No: 13103414

(III)

CERTIFICATE

This is to certify that the work titled **“Adaptive Intelligent Personal Assistive Application”** submitted by **“Siddharth Agarwal”** and **“Akanksha Singh”** in partial fulfillment for the award of degree of Bachelor of Technology of Jaypee Institute of Information Technology University, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor: ________________

Name of Supervisor: Mr. Prashant Kaushik

Designation: Assistant Professor

Date: 21st December, 2016

(IV)

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our mentor – Mr. Prashant Kaushik who gave us the golden opportunity to work on the state of the art topic of Machine Learning and Artificial Intelligence, and helped us in doing all our research work and guided us all along. Without his guidance and persistent help, this project would have not been possible.

Also, in performing our assignment, we had to take the help and guideline of some other respected persons, who deserve our greatest gratitude. Many people, especially our friends and family, have made valuable comment suggestions on this proposal which gave us an inspiration to improve our assignment. We would also like to expand our deepest gratitude to all those who have directly and indirectly guided us in writing this assignment.

Signature of the Student: ________________

Name of Student: Siddharth Agarwal

Enrollment Number: 13103410

Signature of the Student: ________________

Name of Student: Akanksha Singh

Enrollment Number: 13103414

(V)

SUMMARY

An intelligent personal assistant (or simply IPA) is a software agent that can perform tasks or services for an individual. These tasks or services are based on user input, location awareness, and the ability to access information from a variety of online sources. Examples of such an agent are Apple's Siri, Google's Google Now (and later Google Assistant). Amazon Alexa, Microsoft's Cortana. Examples of tasks that may be performed by a smart personal agent-type of Intelligent Automated Assistant include schedule management (e.g., sending an alert to a dinner date that a user is running late due to traffic conditions, update schedules for both parties, and change the restaurant reservation time) and personal health management. Intelligent personal assistant technology are enabled by the combination of mobile devices, application programming interfaces (APIs), and the proliferation of mobile apps.

Similar to this, our project aims to create such an Adaptive Intelligent Personal Assistive Application whose purpose is to help its users perform basic tasks like – Schedule E – Mail, Spam Filtering according to User Labels etc., somewhat similar JARVIS or SIRI. However, this is initially expected to run on textual commands (and not voice commands) and be able to classify the given text command into objects and actions that will be performed on those objects. This can be accomplished by Text Classification – using written command and classify them into objects and actions. Hence, the first step is feature extraction, feature selection, applying appropriate learning algorithm to train the neural network.

Once we are able to train the neural network using examples, outside information and other data sets, it is used to classify the test data set. The two major classes needed for our application are – Object and Action. Also, action parameters are required to be defined that specify the weight of action and degree of importance and priority of the action.

(VI)

LIST OF FIGURES

(VII)

LIST OF TABLES

INTRODUCTION

1.1 – GENERAL INTRODUCTION

Artificial intelligence (AI) is intelligence exhibited by machines. In computer science, an ideal "intelligent" machine is a flexible rational agent that perceives its environment and takes actions that maximize its chance of success at some goal. As machines become increasingly capable, facilities once thought to require intelligence are removed from the definition.

The central problems (or goals) of AI research include reasoning, knowledge, planning, learning, natural language processing (communication), perception and the ability to move and manipulate objects. Approaches include statistical methods, computational intelligence, soft computing (e.g. machine learning), and traditional symbolic AI. The field was founded on the claim that human intelligence "can be so precisely described that a machine can be made to simulate it." This raises philosophical arguments about the nature of the mind and the ethics of creating artificial beings endowed with human-like intelligence, issues which have been explored by myth, fiction and philosophy since antiquity.

An intelligent personal assistant is a software agent that can perform tasks or services for an individual. These tasks or services are based on user input, location awareness, and the ability to access information from a variety of online sources (such as weather or traffic conditions, news, stock prices, user schedules, retail prices, etc.). Examples of such an agent are Apple's Siri, Google's Google Now (and later Google Assistant). Amazon Alexa, Microsoft's Cortana, IBM's Watson (computer), Facebook's M (app) and One Voice Technologies (IVAN).

Examples of tasks that may be performed by a smart personal agent-type of Intelligent Automated Assistant include schedule management (e.g., sending an alert to a dinner date that a user is running late due to traffic conditions, update schedules for both parties, and change the restaurant reservation time) and personal health management (e.g., monitoring caloric intake, heart rate and exercise regimen, then making recommendations for healthy choices).

Intelligent personal assistant technology are enabled by the combination of mobile devices, application programming interfaces (APIs), and the proliferation of mobile apps. However, intelligent automated assistants are designed to perform specific, one-time tasks specified by user voice instructions, while smart personal agents perform ongoing tasks (e.g., schedule management) autonomously.

1.2 – CURRENT OPEN PROBLEMS IN MACHINE LEARNING AND TEXT CLASSIFICATION

Open Problems in Machine Learning

There are many examples of machine learning problems. Much of it focuses on classification problems in which the goal is to categorize objects into a fixed set of categories. Examples:

- Optical Character Recognition: categorize images of handwritten characters by the letters represented
- Face Detection: find faces in images (or indicate if a face is present)
- Text Classification In Spam Filtering: identify email messages as spam or non-spam
- Topic Spotting: categorize news articles (say) as to whether they are about politics, sports, entertainment, etc.
- Spoken Language Understanding: within the context of a limited domain, determine the meaning of something uttered by a speaker to the extent that it can be classified into one of a fixed set of categories
- Medical Diagnosis: diagnose a patient as a sufferer or non-sufferer of some disease

Apart from classification problems, there are other important learning problems. In classification, we want to categorize objects into fixed categories. In regression, on the other hand, we are trying to predict a real value. A richer learning scenario is one in which the goal is actually to behave intelligently, or to make intelligent decisions. For instance, a robot needs to learn to navigate through its environment without colliding with anything.

Open Problems in Text Classification

Intermediate Form – Intermediate forms with varying degrees of complexity are suitable for different classification purposes. For a fine-grain domain-specific knowledge discovery task, it is necessary to perform semantic analysis to derive a sufficiently rich representation to capture the relationship between the objects or concepts described in the documents. However, semantic analysis methods are computationally expensive and often operate in the order of a few words per second. It remains a challenge to see how semantic analysis can be made much more efficient and scalable for very large text corpora.

Multilingual Text Refining - Whereas data classification is largely language independent, text classification involves a significant language component. It is essential to develop text refining algorithms that process multilingual text documents and produce language-independent intermediate forms. While most text classification tools focus on processing English documents, classification from documents in other languages allows access to previously untapped information and offers a new host of opportunities.

Domain Knowledge Integration - Domain knowledge, not catered for by any current text classification tools, could play an important role in text classification. Specifically, domain knowledge can be used as early as in the text refining stage. It is interesting to explore how one can take advantage of domain information to improve parsing efficiency and derive a more compact intermediate form. Domain knowledge could also play a part in knowledge distillation. In a classification or predictive modelling task, domain knowledge helps to improve learning/classification efficiency as well as the quality of the learned model (or mined knowledge). It is also interesting to explore how a user's knowledge can be used to initialize a knowledge structure and make the discovered knowledge more interpretable.

Personalized Autonomous Classification - Current text classification products and applications are still tools designed for trained knowledge specialists. Future text classification tools, as part of the knowledge management systems, should be readily usable by technical users as well as management executives. There have been some efforts in developing systems that interpret natural language queries and automatically perform the appropriate classification operations. Text classification tools could also appear in the form of intelligent personal assistants. Under the agent paradigm, a personal miner would learn a user's profile, conduct text classification operations automatically, and forward information without requiring an explicit request from the user.

Classifying Unstructured Text - Some types of text documents like scientific research papers are usually written strictly in a pre-specified format, which makes it easier to classify them, because of positional information of attributes. However, most text documents are written in an unstructured manner, so classification has to be done on the basis of attributes such as presence or absence of keywords and their frequency of occurrence.

Handling Large Number of Attributes: Feature Selection Using Statistical and Semantic Pre-Processing Techniques - Features useful in text classification are simple words from the language vocabulary, user-specified or extracted keywords, multi-words or metadata. In text classification literature, the steps involved in feature reduction are mainly applying pre-processing such as stop-word removal, stemming etc. Text documents generally use words from a large vocabulary, but all words occurring in a document are not useful for classification.

1.3 - PROBLEM STATEMENT

An intelligent personal assistant is a software agent that can perform tasks or services for an individual. These tasks or services are based on user input, location awareness, and the ability to access information from a variety of online sources. Examples of such an agent are Apple's Siri, Google's Google Now (and later Google Assistant). Amazon Alexa, Microsoft's Cortana. Examples of tasks that may be performed by a smart personal agent-type of Intelligent Automated Assistant include schedule management (e.g., sending an alert to a dinner date that a user is running late due to traffic conditions, update schedules for both parties, and change the restaurant reservation time) and personal health management. Intelligent personal assistant technology are enabled by the combination of mobile devices, application programming interfaces (APIs), and the proliferation of mobile apps. However, intelligent automated assistants are designed to perform specific, one-time tasks specified by user voice instructions, while smart personal agents perform ongoing tasks (e.g., schedule management) autonomously.

This project aims to create such an Adaptive Intelligent Personal Assistive Application whose purpose is to help its users perform basic tasks like – Schedule E – Mail, Spam Filtering according to User Labels etc., somewhat similar JARVIS or SIRI. However, this is initially expected to run on textual commands that is - be able to classify the given text command into objects and actions that will be performed on those objects. This can be accomplished by Text Classification – using written command and classify them into objects and actions. Hence, the first step is feature extraction, feature selection, applying appropriate learning algorithm to train the neural network. Once we are able to train the neural network using examples, outside information and other data sets, it is used to classify the test data set. The two major classes needed for our application are – Object and Action. Also, action parameters are required to be defined that specify the weight of action and degree of importance and priority of the action.

1.4 - OVERVIEW OF PROPOSED SOLUTION APPROACH AND ITS NOVELTY / BENEFITS

The Classification Problem can be stated as a training data set consisting of records. Each record is identified by a unique record id, and consist of fields corresponding to the attributes. An attribute with a continuous domain is called a continuous attribute. An attribute with a finite domain of discrete values is called a categorical attribute. One of the categorical attribute is the classifying attribute or class and the value in its domain are called class labels. Classification is the process of discovering a model for the class in terms of the remaining attributes. The objective is to use the training data set to build a model of the class label based on the other attributes such that the model can be used to classify new data not from the training data set attributes. The objective is to use the training data set to build a model of the class label based on the other attributes such that the model can be used to classify new data not from the training data set.

An Artificial Neural Network, often just named a neural network, is a mathematical model inspired by biological neural networks. A neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system changing its structure during a learning phase. Neural networks are used for modelling complex relationships between inputs and outputs or to find patterns in data. In supervised learning, we are given a set of example pairs (x,y), x є X, y є Y and the aim is to find a function f in the allowed class of functions that matches the examples. In other words, we wish to infer the mapping implied by the data. The cost function is related to the mismatch between our mapping and the data and it implicitly contains prior knowledge about the problem domain. Tasks that fall within the paradigm of supervised learning are pattern recognition (also known as classification) and regression (also known as function approximation). The supervised learning paradigm is also applicable to sequential data (e.g., for speech and gesture recognition).

Tasks of Neural Network:

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:

- Function approximation, or regression analysis, including time series prediction and modelling.

- Classification, including pattern and Sequence.
- Recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind source separation and compression.

Advantages of Neural Network:

i. Artificial neural networks make no assumptions about the nature of the distribution of the data and are not therefore, biased in their analysis. Instead of making assumptions about the underlying population, neural networks with at least one middle layer use the data to develop an internal representation of the relationship between the variables.

ii. Since time-series data are dynamic in nature, it is necessary to have non-linear tools in order to discern relationships among time-series data. Neural networks are best at discovering nonlinear relationships.

iii. Neural networks perform well with missing or incomplete data. Whereas traditional regression analysis is not adaptive, typically processing all older data together with new data, neural networks adapt their weights as new input data becomes available.

BACKGROUND STUDY

2.1 – LITERATURE SURVEY

2.1.1 – Summary Of Research Papers

Paper – 1:

Title	A Brief Review of Machine Learning and its Application
Authors	WANG Hua, MA Cuiqin, ZHOU Lijuan
Year Of Publication	2009
Publication Details	International Conference on Information Engineering and Computer Science
Summary	Machine learning studies how to use computers to simulate human learning activities, to obtain new knowledge, identify existing knowledge, and improve the performance and achievement. Learning involves processing the outside information to knowledge and putting it into the repository. The implementation process uses the knowledge of repository to complete a certain task, and to feed back the information obtained in learning, and guide further study. Types of learning techniques include – Rote learning - A memory based method to store the new knowledge and call for it when necessary. Knowledge accessing is in a stable and direct way. Inductive learning method summarizes general knowledge from sufficient specific examples, and distils a general law of things. Analogy Learning describes the similarity between objects concisely and carries out learning by comparing similar things, mapping their concept, to get new. Verified new knowledge is put

	into repository while unverified is put in as referenced knowledge. Explained Learning is based on the interpretation and analyses of the current instances, reduce a cause and effect explanation tree and use it to answer new instances. Learning Based on Neural Network involves the topology structure of network, right values and work rules of network. Combining of the two can form the main characters of a network. Knowledge discovery of repository is a process to identify effective, novel, potential, useful and understanding model from large amounts of data. Major steps include - Data selection, Data pre-processing, data integrity and consistency, processing of noisy data, Data transformation and Data classification. Applications of machine learning technology: in the field of marketing and financial forecasts.
Web Link	http://ieeexplore.ieee.org/iel5/5362513/5362514/05362936.pdf

Paper – 2:

Title	Automatic Text Classification: A Technical Review
Authors	Mita K. Dalal, Mukesh A. Zaveri
Year Of Publication	2011
Publication Details	International Journal of Computer Applications (0975 – 8887) Volume 28– No.2, August 2011
Summary	Automatic Text Classification is a semi-supervised machine learning task automatically assigning a given document to a set of pre-defined categories based on its textual content and extracted features. It involves assigning a text document to a set of pre-defined classes, Generic strategy includes - document pre-processing, feature extraction / selection, model selection, Training and testing the classifier. Data pre-processing reduces the size of the input text documents by sentence boundary determination, stop-word elimination and stemming. Feature extraction identifies important words in a text document using TF-IDF (term frequency-inverse document frequency), LSI (latent semantic indexing), and multi-word. Then, the text document is represented as a document vector, and an appropriate machine learning algorithm is used to train the text classifier. Automatic Text Classification: Issues - Classification of unstructured text – Unstructured text documents have to be classified on the basis of attributes and their frequency of occurrence. Handling large number of attributes- Features useful in text classification are extracted keywords, multi-words or metadata. But all such words are not useful for classification. Selection of appropriate machine learning technique - Naïve Bayesian gives a probabilistic classification of a text document.

	Its implementation is straightforward and learning time is less, however, its performance is not good for categories defined with very few attributes/ features. SVM is found to be very effective for 2-class classification problems but it is difficult to extend to multi-class classification. Decision tree can be used as a text classifier when there are relatively fewer number of attributes but difficult to manage for large number of attributes.
Web Link	http://www.ijcaonline.org/archives/volume28/number2/3358-4633

Paper – 3:

Title	Recent Trends in Text Classification Techniques
Authors	Nidhi, Vishal Gupta
Year Of Publication	2011
Publication Details	International Journal of Computer Applications (0975 – 8887) Volume 35– No.6, December 2011
Summary	Text Classification assigns a text document to one of a set of predefined classes. Its tasks can be divided into two sorts: supervised classification and unsupervised classification. The BoW (Bag Of Words) method is used to form a vector representing a document, with one component in the vector for each word in the document. The various algorithms discussed are – K- Nearest Neighbour Algorithm: Simple to implement, robust to noisy data but large amount of time needed to do the computations and hence impossible to implement for huge samples. Bayesian Classification: Highly sensitive to feature selection, fast and easy to implement but manageable only for low dimensions, quality is degraded if feature words are interrelated. Support Vector Machine: Highly accurate, less susceptible to overfitting and complexity is independent of the feature space dimension. Complex in implementation and scalability problem. Association based classification: High classification accuracy, strong flexibility at handling textual data , huge set of mined rules , challenging to store, retrieve, prune, and sort a large number of rules efficiently for classification, overfitting problem.

	Term Graph Model: Gives improved accuracy, computational complexity of the graph representation for text classification is the main disadvantage. Decision Tree Induction: Capability to learn disjunctive expressions, handle both numerical and categorical data, based on heuristic algorithms. Neural Network: Nonlinear models, flexible in modelling real world complex relationships, able to estimate the posterior probabilities, with increase in the number of input and hidden nodes, the parameters needed for neural network also increases this result in overfitting of the data. On comparison, SVM performs well, due to their ability to learn independent of the dimensionality of the feature space.
Web Link	http://research.ijcaonline.org/volume35/number6/pxc3976125.pdf

Paper – 4:

Title	A Review Paper On Algorithms Used For Text Classification
Authors	Bhumika, Sukhjit Singh Sehra, Anand Nayyar
Year Of Publication	2013
Publication Details	International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Summary	The need of automatically retrieval of useful knowledge from the huge amount of textual data in order to assist the human analysis is fully apparent. An overview of syntactic and semantic matters, domain ontology, and tokenization concern is given. The different steps of text classification process include: Document collection, pre-processing, indexing, feature selection, classification, performance evaluations. Tasks of text classification algorithms: text categorization, text clustering, concept classification, information retrieval, information extraction. Classification is the process of discovering a model for the class in terms of the remaining attributes and to use the training data to build a model of class label based on the other attributes such as to classify new data. Classification algorithms include: Classification using Decision Trees: (a) Sequential decision tree based classification - Each of the internal node has a decision associated with it and each of the leaves has a class label attached to it. (b) Parallel formulation of decision tree based classification - Synchronous Tree Construction Approach, Partitioned Tree Construction Approach, Hybrid Parallel Formulation.

	Classification using Neural Network: Multi-layer perceptions and Back Propagation algorithm. Artificial neural networks make no assumptions about the nature of the distribution of the data and are not therefore, biased in their analysis. Neural networks perform well with missing or incomplete data. Clustering algorithm: Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. Types of clustering algorithms - Hierarchical methods and Partitioning Methods.
Web Link	http://www.ijaiem.org/Volume2Issue3/IJAIEM-2013-03-13-025.pdf

Paper – 5:

Title	Text Classification in Data Mining
Authors	Anuradha Purohit, Deepika Atre, Payal Jaswani, Priyanshi Asawara
Year Of Publication	2015
Publication Details	International Journal of Scientific and Research Publications, Volume 5, Issue 6, June 2015
Summary	Text classification is the process of classifying documents into predefined categories based on content which retrieve texts in response to a user query. Extracting keywords is done using Porter Stemmer and Tokenizer, word set is formed using Association Rule and Apriori algorithm. The Probability of the word set is calculated using Naive Bayes classifier. Merged Porter Stemmer, Naive Bayes and Association rule are more efficient overall. Association rule classification finds association or correlation relationships among a large set of data items. Naïve Bayes classifier uses the maximum a posteriori estimation for learning a classifier. After pre-processing, association rule classification is applied to the set of data where each frequent word set is considered as a single transaction. Porter stemmer algorithm is used to remove unnecessary words. The association rule is used to derive feature sets from pre-classified text documents. The concept of Naive Bayes Classifier is then used on derived features sets to calculate the probability of derived word sets. Data mining refers to extracting or "mining" knowledge from large amounts of data. Mining means assigning a document or object to one or more classes and is done based on attributes, behaviour or subjects. Association rule classification discovers

	relationships among items in a transactional database. Support, Confidence, Strong Association rules are then formed. Data set are divided into training set and testing set and then proposed algorithm is used to perform the experiments. After repeated experiments over a variety of data sets keeping all other parameter constant, accuracy of 75% was achieved.
Web Link	http://www.ijsrp.org/research-paper-0615/ijsrp-p4262.pdf

Paper – 6:

Title	Text Categorization with Support Vector Machines: Learning with Many Relevant Features
Authors	Thorsten Joachims
Year Of Publication	1998
Publication Details	10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings
Summary	Explores the use of SVM for learning the text classifiers from examples. It analyses particular properties of learning with text data and identifies why SVMs are appropriate for this task. Goal of text categorization is the classification of the documents into a fixed number of predefined categories. Since categories may overlap, each category is treated as separate binary classification problem. First step is to transform documents leading to the attribute representation of text with each word corresponding to a feature. SVMs are based on Structural Risk Minimization principle from computational learning theory, finding a hypothesis for which we can guarantee the lowest true error. SVMs find a hypothesis, with the error on the training set and the complexity measured by VC-Dimensions, which minimizes this upper bound on the true error by effectively and efficiently controlling the VC-Dimension. SVMs are universal learners, and can be used to learn polynomial classifiers, radical basic function networks, and three- layered sigmoid neural nets. SVMs measure the complexity of hypotheses based in the margin with which they separate the data not the number of features i.e. we can generalize in presence of many features. The experimental results show that the SVM consistently achieve good performance on text categorization tasks, outperforming

	existing methods substantially and significantly. With their ability to generalize well in high dimensional feature spaces, SVMs eliminate the need for feature selection, making the application of text categorization considerably easier. Further SVMs perform good avoiding catastrophic failure and do not require any parameter tuning, hence making SVM a very promising and easy to use method for learning text classifier from examples.
Web Link	http://www.cs.cornell.edu/people/tj/publications/joachims_98a.pdf

Paper – 7:

Title	A Comparison of Event Models for Naive Bayes Text Classification
Authors	Andrew McCallum, Kamal Nigam
Year Of Publication	1998
Publication Details	Association For Advancement Of Artificial Intelligence - Workshop On Learning For Text Classification
Summary	Multi-variate Bernoulli model is a Bayesian Network with no dependencies between words and binary word features. Multinomial model is a uni-gram language model with integer word counts. Multi-variate Bernoulli performs well with small vocabulary sizes but multinomial performs better at larger vocabulary sizes. Multi – Variate Bernoulli Event Model specifies that a document is represented by a vector of binary attributes indicating which words occur and do not occur in the document. The document is considered as the "event," and the absence or presence of words are the attributes. Multinomial Event Model specifies that a document is represented by the set of word occurrences from the document. The individual word occurrences are considered as "events" and the document is the collection of word events. This approach assumes that the text data was generated by a parametric model, and uses training data to calculate Bayes-optimal estimates of the model parameters, classifies new test documents using Bayes' rule and calculate the posterior probability that a class would have generated the test document in question. In the multi-variate Bernoulli event model, a document is a binary vector over the space of words and its probability is given by the product of the probability of the attribute values over all word

	attributes. The multinomial model captures word frequency information in documents. The document is an ordered sequence of word events, drawn from the same vocabulary. The lengths of documents are independent of class. In the experiments performed over the data set the multivariate Bernoulli event model reaches a maximum of 41% accuracy with only 200 words. The multi-variate Bernoulli shows its best results at a smaller vocabulary than the multinomial, and that the multinomial has best performance at a larger vocabulary size. Multinomial has the highest accuracy of 74% at 20000 words, and multi-variate Bernoulli is best with 46% accuracy at 1000 words.
Web Link	http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf

Paper – 8:

Title	Recurrent Convolutional Neural Networks for Text Classification
Authors	Siwei Lai, Liheng Xu, Kang Liu, Jun Zhao
Year Of Publication	2015
Publication Details	AAAI'15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence
Summary	A key problem in text classification is feature representation since traditional methods ignore the contextual information and cannot capture the semantics of the words. Multiple Neural Network techniques have been proposed and compared. Recursive Neural Network captures the semantics of a sentence via a tree structure. Its performance depends on the performance of the textual tree construction that has a time complexity of O (n^2), where n is the length of the text. Thus, RecursiveNN is unsuitable for modelling long sentences or documents. Recurrent Neural Network analyses a text word by word and stores the semantics of all the previous text in a fixed-sized hidden layer all in a time complexity of O (n). Its advantage is the ability to better capture the contextual information. However, the RecurrentNN is a biased model, where later words are more dominant than earlier words. Convolutional Neural Network (CNN), an unbiased model can fairly determine discriminative phrases in a text with a max-pooling layer. It better captures the semantic of texts in a time complexity of O (n). However, it is difficult to determine the window size: small window sizes may result in the loss of some critical information, whereas large windows result in an enormous parameter space. Recurrent Convolutional Neural Network (RCNN) applies a bi-directional recurrent structure, to capture the contextual

	information to the greatest extent possible. Second, it employs a max-pooling layer, to capture the key component in time complexity of O (n). On comparison of these algorithms to the widely used traditional methods (BoW, LR), the neural network approaches outperform the traditional methods. Also RCNN outperforms the CNN in all cases because CNNs use a fixed window of words whereas RCNNs use the recurrent structure to capture a wide range of contextual information.
Web Link	https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/download/9745/9552

Paper – 9:

Title	An Analysis of Single-Layer Networks in Unsupervised Feature Learning
Authors	Adam Coates, Honglak Lee, Andrew Y. Ng
Year Of Publication	2011
Publication Details	JMLR Workshop and Conference Proceedings Volume 15, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics
Summary	Unsupervised feature learning involves training of the network without any given examples or training data set and find the output based on the inputs being provided, to teach the network. Several simple factors, such as the number of hidden nodes in the model, may be more important to achieving high performance than the learning algorithm or the depth of the model. Several feature learning algorithms (sparse auto-encoders, sparse RBMs, K-means clustering, and Gaussian mixtures) are applied to CIFAR, NORB, and STL datasets using only single - layer networks. Results show that large numbers of hidden nodes and dense feature extraction are critical to achieving high performance. Best performance is based on K-means clustering, which is extremely fast, has no hyper parameters to tune beyond the model structure itself, and is very easy to implement. Despite the simple system, accuracy beyond all previously published results on the CIFAR-10 and NORB datasets was achieved (79.6% and 97.2% respectively). A major drawback of many feature learning systems is their complexity and expense. At a high-level: Extract random patches from unlabelled training images. Apply a pre-processing stage to the patches. Learn a feature-mapping using an unsupervised learning algorithm. Extract features from equally spaced sub-patches covering the input image. Pool features together over

	regions of the input image to reduce the number of feature values. Train a linear classifier to predict the labels given the feature vectors. Choices of unsupervised learning algorithms: Sparse auto-encoder, Sparse restricted Boltzmann machine, K-means clustering, Gaussian mixtures. It is shown more generally that smaller stride and larger numbers of features yield monotonically improving performance, which suggests that while more complex algorithms may have greater representational power, simple but fast algorithms can be highly competitive.
Web Link	http://www.jmlr.org/proceedings/papers/v15/coates11a.html

2.1.2 - INTEGRATED SUMMARY OF LITERATURE STUDIED

As a summation of all the research papers studied, a lot many things were understood like what is machine learning along with its types, and some of the problems it solves. Further we learned about data mining and text classification steps – parsing, tokenization, stemming, categorization etc. The following was integrated from the above literature studied.

What is Machine Learning?

Machine learning studies computer algorithms for learning to do stuff. It involves learning to complete a task, or to make accurate predictions, or to behave intelligently. The learning that is being done is always based on some sort of observations or data, such as examples, direct experience, or instruction. The emphasis of machine learning is on automatic methods.

The primary goal of machine learning research is to develop general purpose algorithms of practical value. Such algorithms should be efficient. In the context of learning, the amount of data that is required by the learning algorithm is an important factor. Learning algorithms should also be as general purpose as possible. Types of learning techniques include – Rote learning, Inductive Learning Method, Analogy Learning, Explained Learning, Learning Based on Neural Network and Knowledge discovery. All these techniques can be performed Supervised (guided by predefined set of inputs and outputs), or Unsupervised (guided by set of Examples).

What is Text Classification? What are its Implementation techniques?

Automatic Text Classification is a semi-supervised machine learning task automatically assigning a given document to a set of pre-defined categories based on its textual content and extracted features. It involves assigning a text document to a set of pre-defined classes, Generic strategy includes - document pre-processing, feature extraction / selection, model selection, Training and testing the classifier.

Major Classification algorithms include:

K- Nearest Neighbour Algorithm: Simple to implement, robust to noisy data but large amount of time needed to do the computations and hence impossible to implement for huge samples.

Bayesian Classification: Highly sensitive to feature selection, fast and easy to implement but manageable only for low dimensions, quality is degraded if feature words are interrelated.

Support Vector Machine: Highly accurate, less susceptible to overfitting and complexity is independent of the feature space dimension. Complex in implementation and scalability problem.

Association based classification: High classification accuracy, strong flexibility at handling textual data , huge set of mined rules , challenging to store, retrieve, prune, and sort a large number of rules efficiently for classification, overfitting problem.

Term Graph Model: Gives improved accuracy, computational complexity of the graph representation for text classification is the main disadvantage

Classification using Decision Trees: Sequential decision tree based classification - Each of the internal node has a decision associated with it and each of the leaves has a class label attached to it.

Classification using Neural Network: Multi-layer perceptions and Back Propagation algorithm. Artificial neural networks make no assumptions about the nature of the distribution of the data and are not therefore, biased in their analysis. Neural networks perform well with missing or incomplete data.

Clustering algorithm: Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups.

What was found to be the best approach to perform our task?

An Artificial Neural Network, or neural network consists of an interconnected group of artificial neurons, and it processes information using a connectionist approach to computation. In most cases a neural network is an adaptive system changing its structure during a learning phase. Neural networks are used for modelling complex relationships between inputs and outputs or to find patterns in data.

Artificial Neural Network make no assumptions about the nature of the distribution of the data and are not therefore, biased in their analysis. Instead of making assumptions about the underlying population, neural networks with at least one middle layer use the data to develop an internal representation of the relationship between the variables. Neural networks are best at discovering nonlinear relationships. Neural networks perform well with missing or incomplete data. Whereas traditional regression analysis is not adaptive, typically processing all older data together with new data, neural networks adapt their weights as new input data becomes available. Neural network applies a bi-directional recurrent structure, to capture the contextual information to the greatest extent possible. Second, it employs a max-pooling layer that judges which features play key roles in text classification, to capture the key component in time complexity of O (n).

2.2 – SUMMARY OF FIELD SURVEY, EXPERIMENTAL STUDIES, NEW TOOLS

Field survey will be done as the project is still in the learning phase and will surely include the experimental phase and the testing phase with development of the project over the time. Experimental studies involves training the neural network with the training dataset and finding the results of the new test data set with the varying inputs. This whole process of training the network is most important task and needs thorough study and many experimental and testing implementations of the different libraries or frameworks available for the task of training the network. After training the network the desired feature extraction and the feature selection methods have to be applied and the text is classified under the required categories or classes. This is the testing phase of the project where the implementation will be tested against the different data sets to test the accuracy of the code and find the desired results.

Tools Used – TensorFlow

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

TensorFlow is a programming system in which you represent computations as graphs. Nodes in the graph are called *ops* (short for operations). An op takes zero or more Tensors, performs some computation, and produces zero or more Tensors. In TensorFlow terminology, a Tensor is a typed multi-dimensional array. For example, you can represent a mini-batch of images as a 4-D array of floating point numbers with dimensions [batch, height, width, channels].

A TensorFlow graph is a *description* of computations. To compute anything, a graph must be launched in a Session. A Session places the graph ops onto Devices, such as CPUs or GPUs, and provides methods to execute them. These methods return tensors produced by ops as numpy ndarray objects in Python, and as tensorflow::Tensor instances in C and C++.

TensorFlow programs use a tensor data structure to represent all data -- only tensors are passed between operations in the computation graph. You can think of a TensorFlow tensor as an n-dimensional array or list. A tensor has a static type, a rank, and a shape.

ANALYSIS, DESIGN AND MODELLING

3.1 – FUNCTIONAL AND NON FUNCTIONAL REQUIREMENTS

FUNCTIONAL REQUIREMENTS:

INPUTS:

- Documents for text classification training.
- Textual Command - that will be classified.
- Feedback on Training Set.
- Weights of Action Parameter.

OUTPUTS:

- Classified text – as object or action.
- Output could be correct or incorrect which is dependent on the training of the data set.

INTERMEDIATE STEPS:

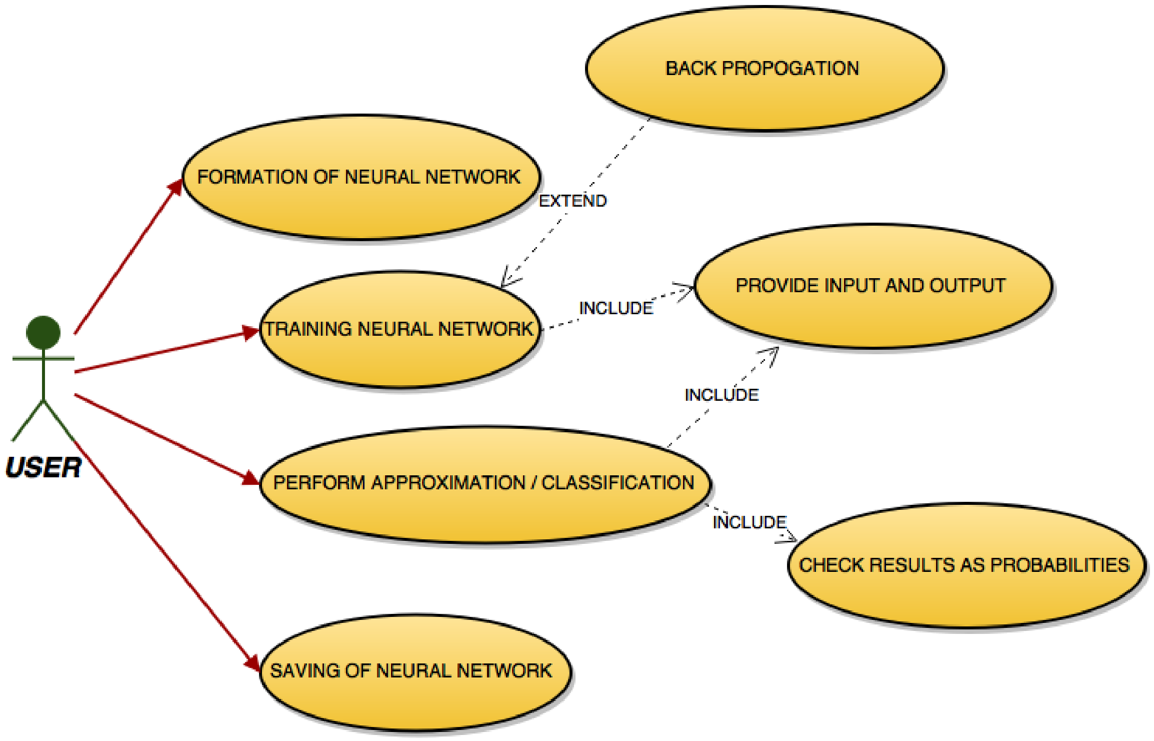
- Documents Collection - This is first step of classification process in which we are collecting the different types (format) of document like .html, .pdf, .doc, web content etc.
- Pre-Processing - The first step of pre-processing which is used to presents the text documents into clear word format. The documents prepared for next step in text classification are represented by a great amount of features.
- Tokenization: A document is treated as a string, and then partitioned into a list of tokens by Removing stop words, Stemming word - Applying the stemming algorithm that converts different word form into similar canonical form.
- Indexing - The document have to be transformed from the full text version to a document vector.
- Feature Selection - After pre-processing and indexing the important step of text classification, is feature selection to construct vector space, which improves the scalability, efficiency and accuracy of a text classifier.
- Classification - The automatic classification of documents into predefined categories has observed as an active attention, the documents can be classified by three ways, unsupervised, supervised and semi-supervised methods.

NON FUNCTIONAL REQUIREMENTS:

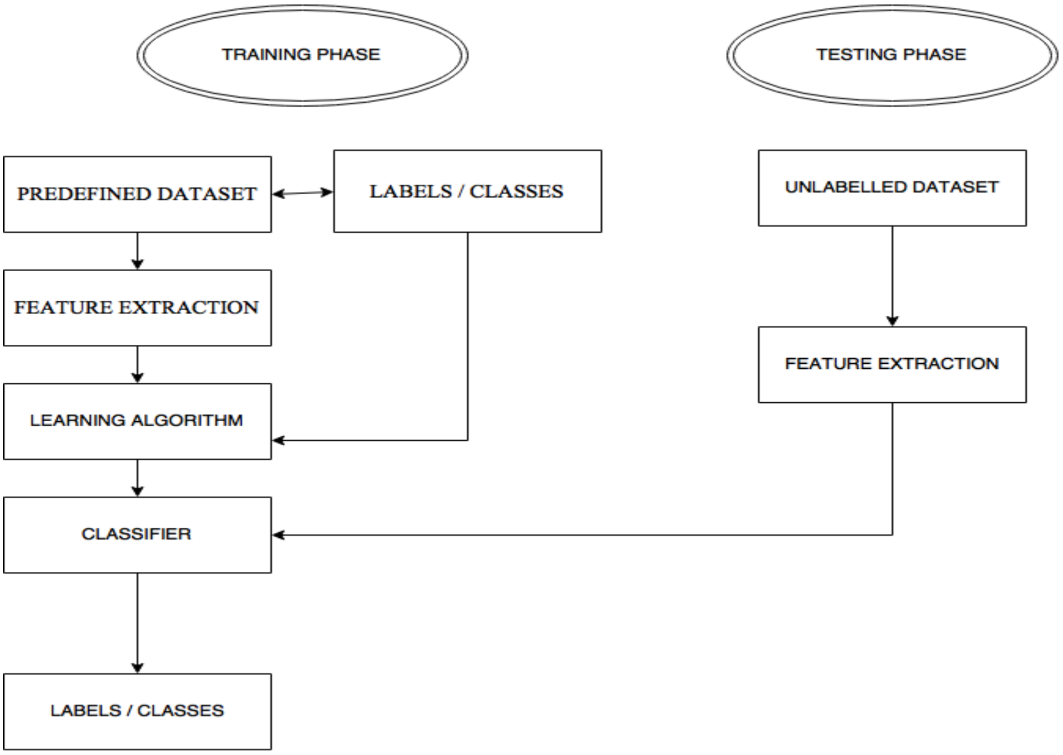
- The system is required to be efficient in memory allocation and freeing, to allow space for large document sets for training purposes.
- Fast processing for classifying into sub classes.
- Libraries involved – NLTK – Platform for building python programs to work with Human Language Data, Text Processing Libraries for classification tokenization, stemming, tagging, and semantic reasoning.
- UNIX System

3.2 – DESIGN DOCUMENTATION

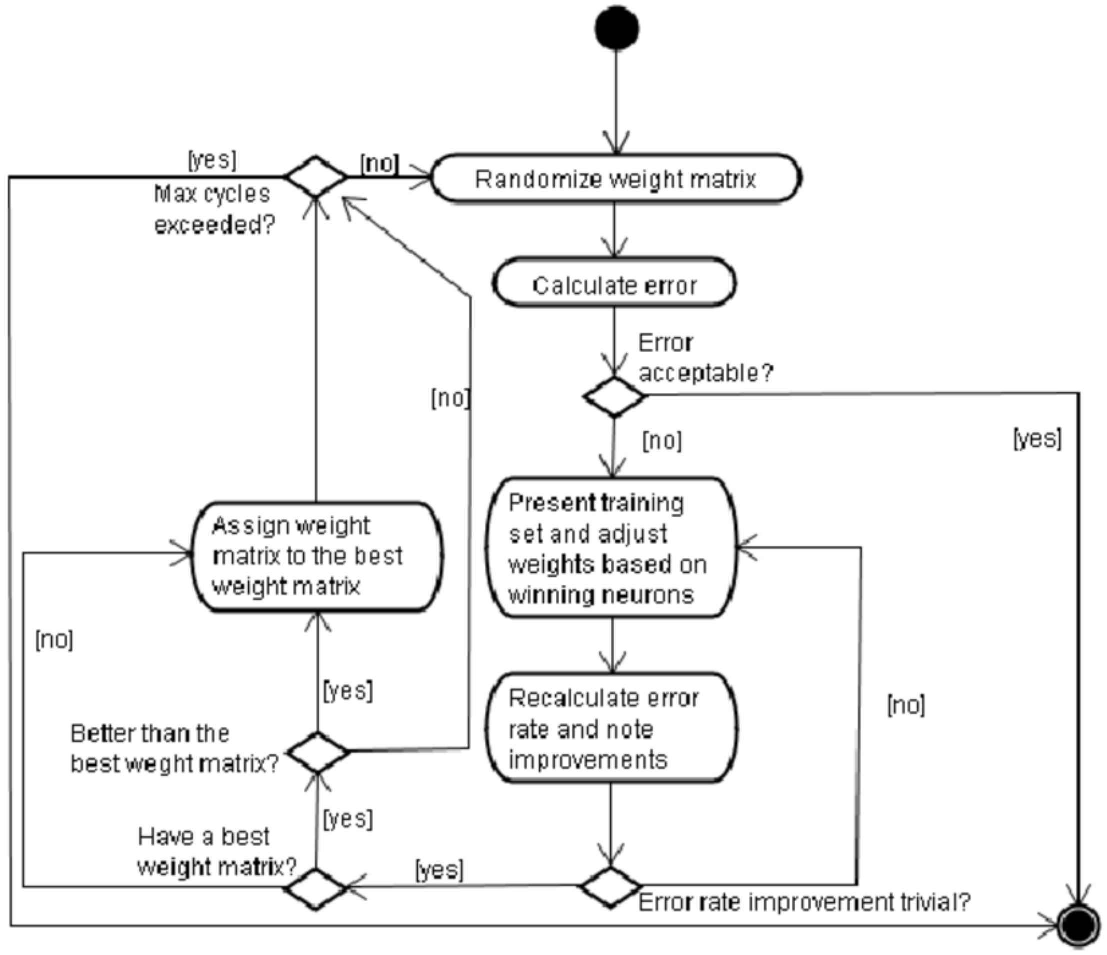
3.2.1 – USE CASE DIAGRAM



3.2.2 – CONTROL FLOW DIAGRAM



3.2.3 – ACTIVITY DIAGRAM


[yes]
Max cycles exceeded?
[no]
Randomize weight matrix
Calculate error
Error acceptable?
[no]
[yes]
[no]
Present training set and adjust weights based on winning neurons
Assign weight matrix to the best weight matrix
[no]
[yes]
Recalculate error rate and note improvements
[no]
Better than the best weght matrix?
[yes]
Have a best weight matrix?
[yes]
Error rate improvement trivial?

3.3 – ALGORITHM DESIGN

We propose a deep neural model to capture the semantics of the text. Figure below shows the network structure that we hope to follow. The input of the network is a document D, which is a sequence of words w_1, w_2….w_n. The output of the network contains class elements. We use p(k|D,θ) to denote the probability of the document being class k, where θ is the parameters in the network.

We combine a word and its context to present a word. The contexts help us to obtain a more precise word meaning. In our model, we use a recurrent structure, which is a bidirectional recurrent neural network, to capture the contexts. We define c_l (w_i) as the left context of word w_i and c_r (w_i) as the right context of word wi. Both c_l (w_i) and c_r (w_i) are dense vectors with |c| real value elements. W (l) is a matrix that transforms the hidden layer (context) into the next hidden layer. W(s_l) is a matrix that is used to combine the semantic of the current word with the next word's left context. The right-side contexts of the last word in a document share the parameters c_r (w_i). c_l (w_i) encodes the semantics of the left-side context "stroll along the South" along with all previous texts in the sentence, and c_r (w_i) encodes the semantics of the right-side context "affords an ...". Then, we define the representation of word wi, which is the concatenation of the left-side context vector c_l (w_i), the word embedding e (w_i) and the right-side context vector c_r (w_i). In this manner, using this contextual information, our model may be better able to disambiguate the meaning of the word wi compared to conventional neural models that only use a fixed window (i.e., they only use partial information about texts).

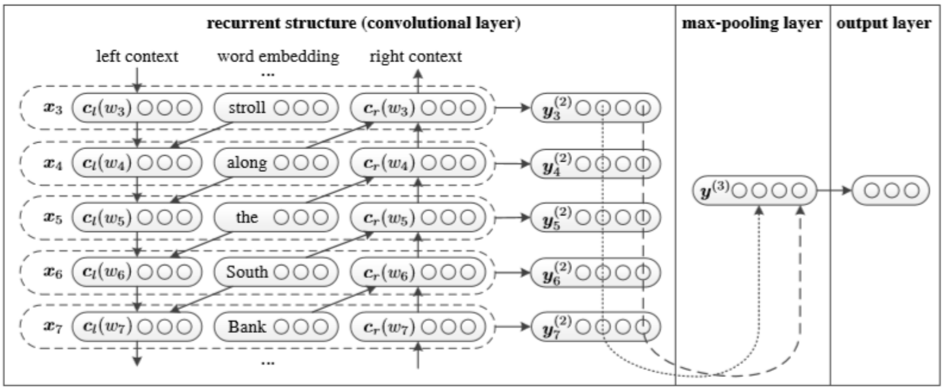


Figure 1: The structure of the recurrent convolutional neural network. This figure is a partial example of the sentence "A sunset stroll along the South Bank affords an array of stunning vantage points", and the subscript denotes the position of the corresponding word in the original sentence.

3.4 – Risk Analysis And Mitigation Plan

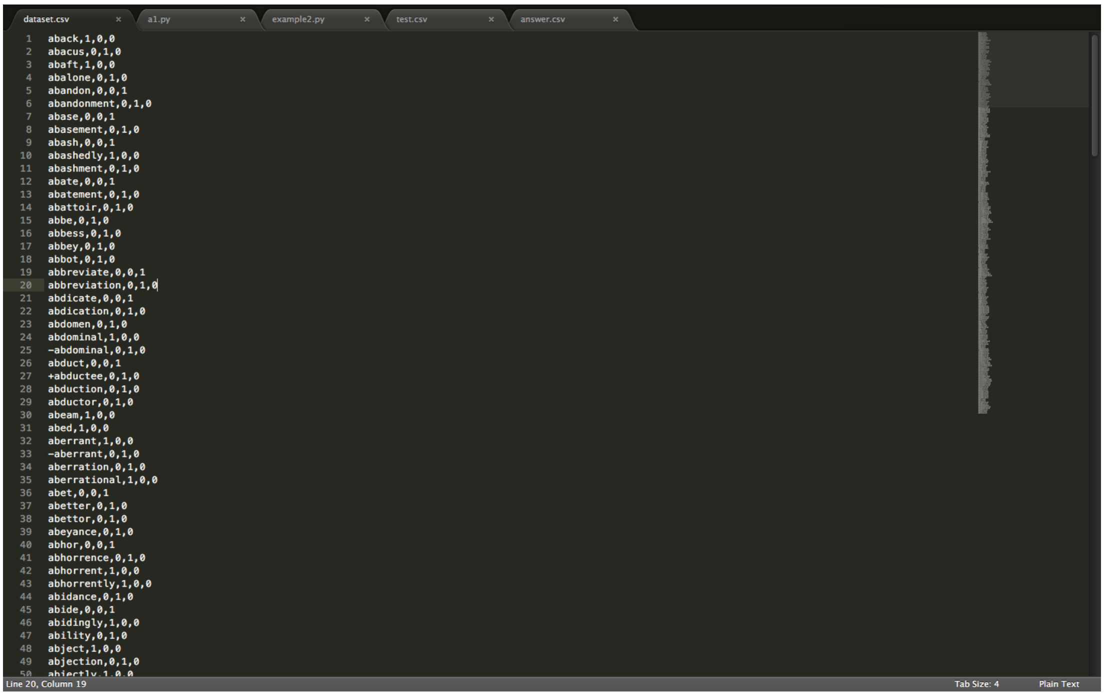
Risk ID	Description of Risk	Risk Area	Probability (P)	Impact (I)	RE(P*I)	Risk Selected for Mitigation	Mitigation on Plan if selection is 'Y'	Contingency plan, if any
1	Inability to Classify Text	Machine Learning	Low (1)	High (5)	5	'Y'	Training algorithm will be selected according to the Dataset size and Feature set size.	Keep the dataset of moderate size and the Feature sets should be extracted accordingly
2	Incorrect Text Classification	Learning Algorithm	Medium (3)	High (5)	15	'Y'	Training dataset should be specific to facilitate correct learning.	Feeding the network with more dataset.
3	Large Computational Time Complexity	Algorithm	Low (1)	Low (1)	1	'N'		

4	Segmentation Error : Code Dump	Code	Low (3)	High (5)	15	‘Y’	Debugging of the code.	
5	Library Support and Error	Library	Low (1)	Medium(3)	3	‘Y’	Re-install library compatible with the IDE and OS.	
6	System Memory	Device problem	Low (1)	High (5)	5	‘N’		

IMPLEMENTATION AND TESTING

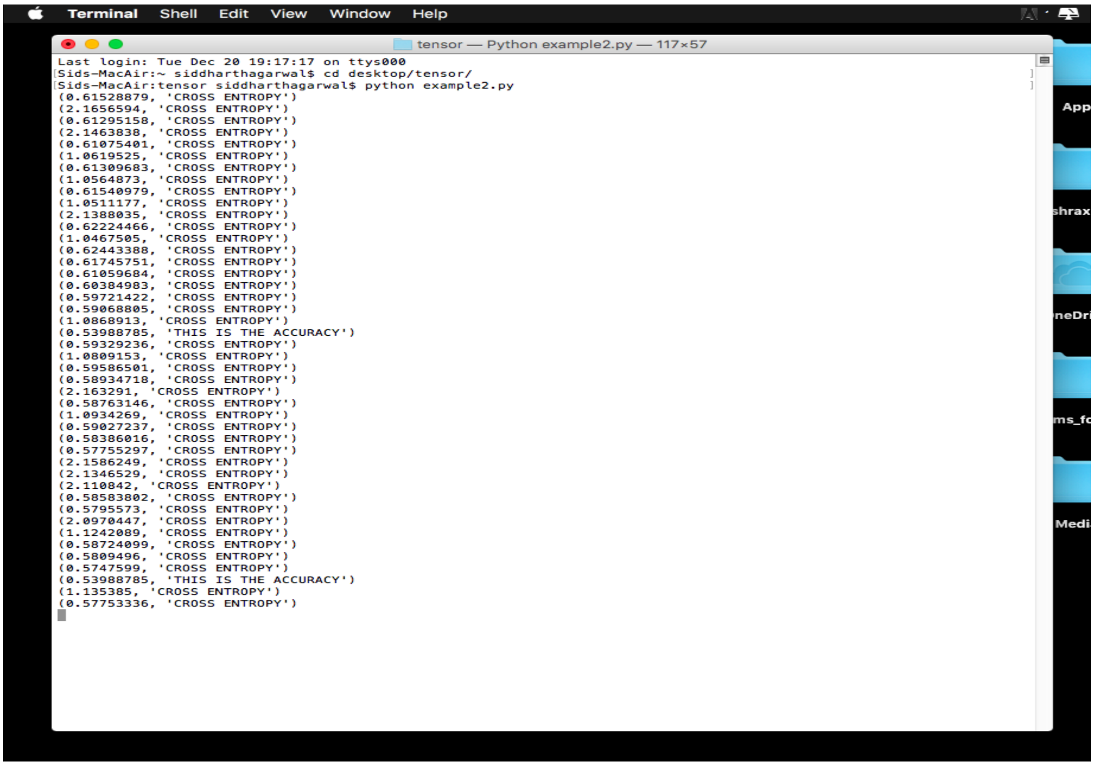
4.1 – IMPLEMENTATION DESIGN AND ISSUES

- Training Data Set



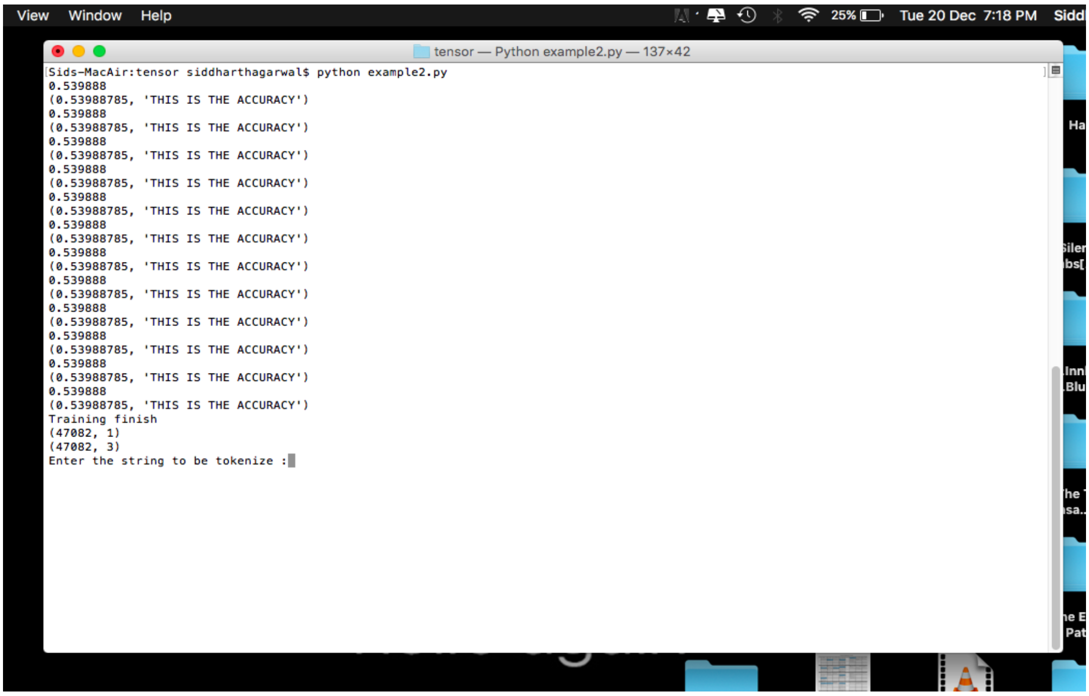
```
aback,1,0,0
abacus,0,1,0
abaft,1,0,0
abalone,0,1,0
abandon,0,0,1
abandonment,0,1,0
abase,0,0,1
abasement,0,1,0
abash,0,0,1
abashedly,1,0,0
abashment,0,1,0
abate,0,0,1
abatement,0,1,0
abattoir,0,1,0
abbe,0,1,0
abbess,0,1,0
abbey,0,1,0
abbot,0,1,0
abbreviate,0,0,1
abbreviation,0,1,0
abdicate,0,0,1
abdication,0,1,0
abdomen,0,1,0
abdominal,1,0,0
-abdominal,0,1,0
abduct,0,0,1
+abductee,0,1,0
abduction,0,1,0
abductor,0,1,0
abeam,1,0,0
abed,1,0,0
aberrant,1,0,0
-aberrant,0,1,0
aberration,0,1,0
aberrational,1,0,0
abet,0,0,1
abetter,0,1,0
abettor,0,1,0
abeyance,0,1,0
abhor,0,0,1
abhorrence,0,1,0
abhorrent,1,0,0
abhorrently,1,0,0
abidance,0,1,0
abide,0,0,1
abidingly,1,0,0
ability,0,1,0
abject,1,0,0
abjection,0,1,0
abjectly,1,0,0
```

- Training the network.



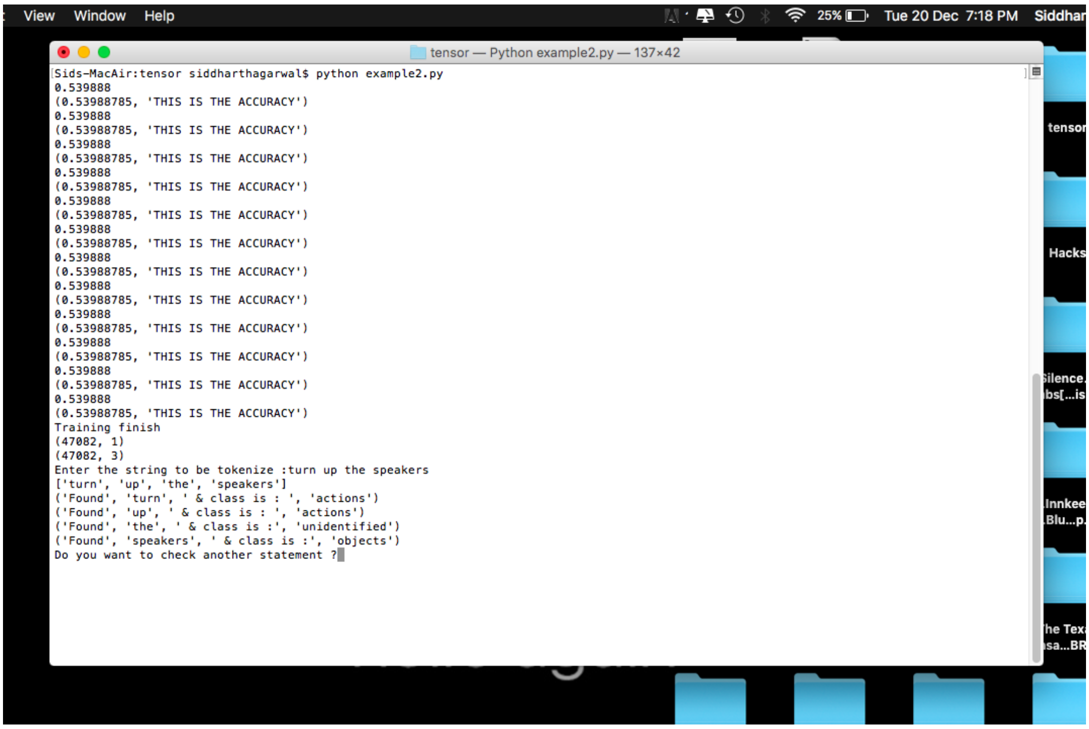
```
Last login: Tue Dec 20 19:17:17 on ttys000
Sids-MacAir:~ siddharthagarwal$ cd desktop/tensor/
Sids-MacAir:tensor siddharthagarwal$ python example2.py
(0.61528879, 'CROSS ENTROPY')
(2.1656594, 'CROSS ENTROPY')
(0.61295158, 'CROSS ENTROPY')
(2.1463838, 'CROSS ENTROPY')
(0.61075401, 'CROSS ENTROPY')
(1.0619525, 'CROSS ENTROPY')
(0.61309683, 'CROSS ENTROPY')
(1.0564873, 'CROSS ENTROPY')
(0.61540979, 'CROSS ENTROPY')
(1.0511177, 'CROSS ENTROPY')
(2.1388035, 'CROSS ENTROPY')
(0.62224466, 'CROSS ENTROPY')
(1.0467505, 'CROSS ENTROPY')
(0.62443388, 'CROSS ENTROPY')
(0.61745751, 'CROSS ENTROPY')
(0.61059684, 'CROSS ENTROPY')
(0.60384983, 'CROSS ENTROPY')
(0.59721422, 'CROSS ENTROPY')
(0.59068805, 'CROSS ENTROPY')
(1.0868913, 'CROSS ENTROPY')
(0.53988785, 'THIS IS THE ACCURACY')
(0.59329236, 'CROSS ENTROPY')
(1.0809153, 'CROSS ENTROPY')
(0.59586501, 'CROSS ENTROPY')
(0.58934718, 'CROSS ENTROPY')
(2.163291, 'CROSS ENTROPY')
(0.58763146, 'CROSS ENTROPY')
(1.0934269, 'CROSS ENTROPY')
(0.59027237, 'CROSS ENTROPY')
(0.58386016, 'CROSS ENTROPY')
(0.57755297, 'CROSS ENTROPY')
(2.1586249, 'CROSS ENTROPY')
(2.1346529, 'CROSS ENTROPY')
(2.110842, 'CROSS ENTROPY')
(0.58583802, 'CROSS ENTROPY')
(0.5795573, 'CROSS ENTROPY')
(2.0970447, 'CROSS ENTROPY')
(1.1242089, 'CROSS ENTROPY')
(0.58724099, 'CROSS ENTROPY')
(0.5809496, 'CROSS ENTROPY')
(0.5747599, 'CROSS ENTROPY')
(0.53988785, 'THIS IS THE ACCURACY')
(1.135385, 'CROSS ENTROPY')
(0.57753336, 'CROSS ENTROPY')
```

- Testing



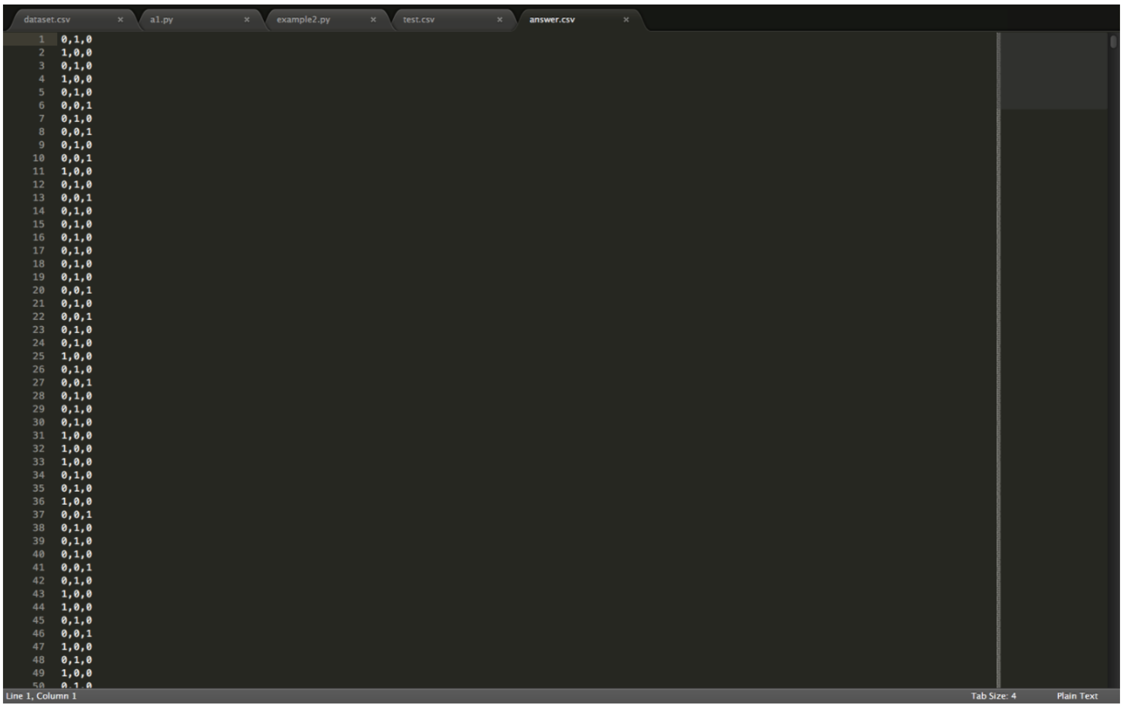
```
Sids-MacAir:tensor siddharthagarwal$ python example2.py
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
Training finish
(47082, 1)
(47082, 3)
Enter the string to be tokenize :
```

- Tokenization and classification.



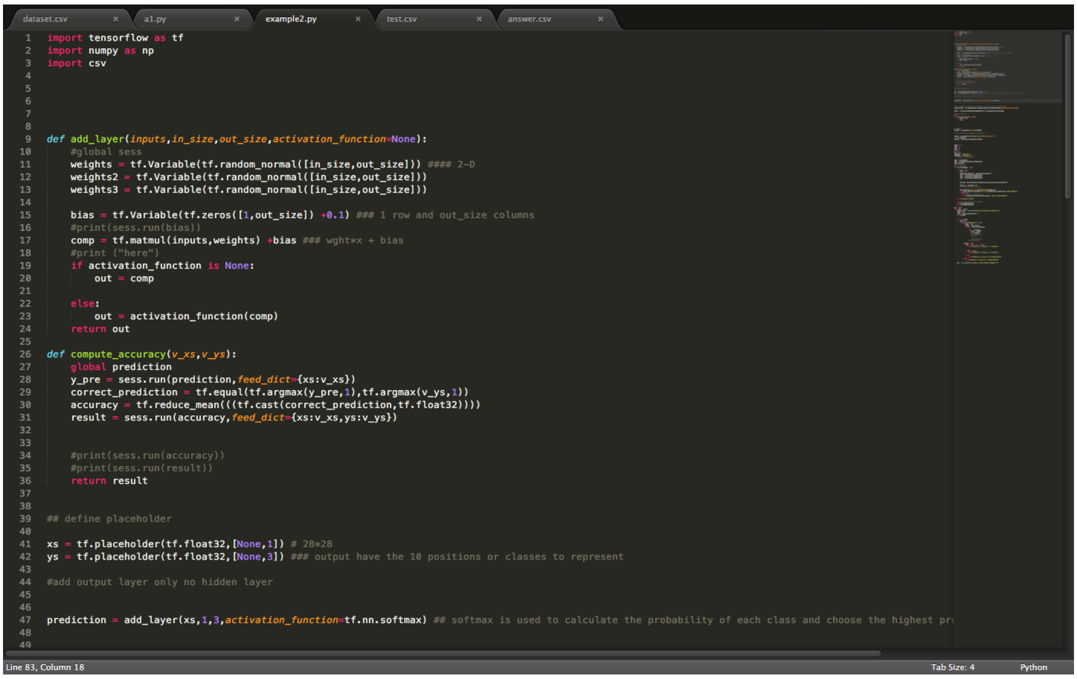
```
Sids-MacAir:tensor siddharthagarwal$ python example2.py
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
0.539888
(0.53988785, 'THIS IS THE ACCURACY')
Training finish
(47082, 1)
(47082, 3)
Enter the string to be tokenize :turn up the speakers
['turn', 'up', 'the', 'speakers']
('Found', 'turn', ' & class is : ', 'actions')
('Found', 'up', ' & class is : ', 'actions')
('Found', 'the', ' & class is :', 'unidentified')
('Found', 'speakers', ' & class is :', 'objects')
Do you want to check another statement ?
```

- Testing Data Set



```
0,1,0
1,0,0
0,1,0
1,0,0
0,1,0
0,0,1
0,1,0
0,0,1
0,1,0
0,0,1
1,0,0
0,1,0
0,0,1
0,1,0
0,1,0
0,1,0
0,1,0
0,1,0
0,1,0
0,0,1
0,1,0
0,0,1
0,1,0
0,1,0
1,0,0
0,1,0
0,0,1
0,1,0
0,1,0
0,1,0
1,0,0
1,0,0
1,0,0
0,1,0
0,1,0
1,0,0
0,0,1
0,1,0
0,1,0
0,1,0
0,0,1
0,1,0
1,0,0
1,0,0
0,1,0
0,0,1
1,0,0
0,1,0
1,0,0
0,1,0
```

- CODE



```python
import tensorflow as tf
import numpy as np
import csv





def add_layer(inputs,in_size,out_size,activation_function=None):
    #global sess
    weights = tf.Variable(tf.random_normal([in_size,out_size])) #### 2-D
    weights2 = tf.Variable(tf.random_normal([in_size,out_size]))
    weights3 = tf.Variable(tf.random_normal([in_size,out_size]))

    bias = tf.Variable(tf.zeros([1,out_size]) +0.1) ### 1 row and out_size columns
    #print(sess.run(bias))
    comp = tf.matmul(inputs,weights) +bias ### wght*x + bias
    #print ("here")
    if activation_function is None:
        out = comp

    else:
        out = activation_function(comp)
    return out

def compute_accuracy(v_xs,v_ys):
    global prediction
    y_pre = sess.run(prediction,feed_dict={xs:v_xs})
    correct_prediction = tf.equal(tf.argmax(y_pre,1),tf.argmax(v_ys,1))
    accuracy = tf.reduce_mean(((tf.cast(correct_prediction,tf.float32))))
    result = sess.run(accuracy,feed_dict={xs:v_xs,ys:v_ys})


    #print(sess.run(accuracy))
    #print(sess.run(result))
    return result


## define placeholder

xs = tf.placeholder(tf.float32,[None,1]) # 28*28
ys = tf.placeholder(tf.float32,[None,3]) ### output have the 10 positions or classes to represent

#add output layer only no hidden layer


prediction = add_layer(xs,1,3,activation_function=tf.nn.softmax) ## softmax is used to calculate the probability of each class and choose the highest pr


```

4.2 – TESTING

4.2.1 - TESTING PLAN

Type of Test	Will Test be Performed?	Explanations	Software Component
Requirement Testing	Yes	As the project is based on training and testing the accuracy of the Neural Network, every aspect of the project will be tested based on different criteria.	The language used for the training and formation of the Neural Network is Python and hence this language will be used for the requirement testing.
Unit	Yes	As the training of t he Neural Network depends on the learning rate and the input dataset , so every different dataset available will act as different unit test.	For training and testing the Neural Network Tensorflow is used and the language used is Python , hence making the testing and training easy and feasible.
Integration	Yes	The Neural Network is used to calculate the and minimize the Cross Entropy, of the Classification Dataset provided ,to learn and predict the output Classes which will later be used on the Testing Dataset.	For the calculation of the Cross Entropy Logarithmic Function is used and the Activation Function used is Softmax.

Performance	Yes	After learning from the input dataset which will already be Classified , the predicted classes of the testing dataset will be checked with their original classes , hence calculating the accuracy or the performance of the learning of Neural Network.	The software component is TensorFlow library packages which includes all the predefined Activation Functions required for the calculation of the accuracy .
Stress	No	The application is of Neural Network learning from the input dataset and hence no requirement of stress testing .	
Compliance	No	As the project is still in the developing stage , hence the requirements and the kind of outputs expected are changing with the on going development.	
Security	No	Training of the Neural Network does not require any Internet connection or any other human interference , hence safe .	

Load	Yes	As the network learning is dependent upon the input dataset and different amount of datasets results in different accuracy and performance .Therefore the load testing will be performed.	The input dataset is the simple CSV(Comma Separated Values) File which will be mined using the programming language Python .
Volume	No	Volume testing is not performed due to lack of resources.	

4.2.2 – Component Decomposition And Type Of Testing Required

S.No.	List of various components(modules) that require testing.	Type of testing required.	Technique for writing test cases.
1.	Requirement Gathering	Requirement testing	Black Box Testing
2.	Input Dataset Generation	Unit testing	White Box Testing
3.	Neural Network Training	Unit Testing	White Box Testing
4.	Output Dataset Generation	Unit Testing	White Box Testing
5.	Neural Network Testing	Performance Testing	Black Box Testing
6.	OS Commands Testing	Unit Testing	White Box Testing

4.2.3 – LIST OF ALL TEST CASES

Test Case ID	Input	Expected Output	Status
1.	For the generation of the Input Dataset , a part of dictionary is used and the data is extracted according to the requirement of the Input layer of Neural Network.	Output would be the Input dataset in the format required by the Input layer of the Neural Network (i.e. Words along with their classes) .	PASS
2.	The input dataset is fed to the Neural Network to start the learning process.	Learning starts and the calculation of cross entropy begins.	PASS
3.	If the input is not in the correct form i.e. other than required by the input layer .	Error in learning.	FAIL
4.	Learning starts and the cross entropy is calculated using the logarithmic functions and the pre defined activation functions in the Tensorflow package.	Minimized output of the cross entropy at each step of learning.	PASS
5.	Generation of the Testing Dataset as required by the input layer for the prediction of the classes.	Prediction starts after the learning process is complete and the testing data is classified.	PASS

6.	Calculation of the performance or the accuracy of the generated or predicted classes are done on the given testing input of classes.	Correct classes are predicted according to the approximation or the accuracy calculated.	PASS

4.2.4 – Limitations Of The Solution

The solution proposed by our model is feed forward ,back-propagation model of the neural network and is dependent on the size and quality of the input training dataset and the testing dataset . As the quality of the training dataset improves the learning quality of the neural network also improves. The model learns and adjusts the weights and biases on the basis of the training examples and then predicts the classes for the testing dataset .

Hence , the only limitation to our model is of the learning rate and training dataset, and how well the neural network learns and adjusts the weights and biases for the testing examples.

FINDINGS AND CONCLUSION

5.1 – FINDINGS

- Data Set required for training should be apt in quantity and quality.
- Learning Rate is a trade-off between Speed and Accuracy of training. The higher the learning rate, the learning is faster however it hampers the accuracy of learning. If the learning rate is lower, the accuracy of rate is better, but learning takes a substantial amount of time.
- Weights and Biases which are adjusted by the Neural Network are dependent on the Activation function, Optimizer Function and Accuracy Function.
- Testing is heavily dependent on the quality of Testing Data.

5.2 – CONCLUSION

At the learning rate of 0.01, and with activation function as – “softmax”, the maximum accuracy achieved by our project is 53 – 54 % i.e. our neural network can classify 53% of the test cases accurately while it may give erroneous classification in certain cases.

5.3 – FUTURE WORK

Future Work that we hope to do is-

- Increase its accuracy by assigning priority to parameters and then prioritizing according to weights of these parameters.
- After correct classification, we hope to implement the commands by linking them to our operating system, so that the input command can be executed.
- The same can be extended to voice commands.

REFERENCES

- WANG Hua, MA Cuiqin, ZHOU Lijuan, A Brief Review of Machine Learning and its Application, *International Conference on Information Engineering and Computer Science*, 2009
- Mita K. Dalal, Mukesh A. Zaveri, Automatic Text Classification: *A Technical Review, International Journal of Computer Applications (0975 – 8887) Volume 28– No.2*, August 2011, 2011
- Nidhi, Vishal Gupta, Recent Trends in Text Classification Techniques, *International Journal of Computer Applications (0975 – 8887) Volume 35– No.6, December 2011*, 2011
- Bhumika, Sukhjit Singh Sehra, Anand Nayyar, A Review Paper On Algorithms Used For Text Classification, *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, 2013
- Anuradha Purohit, Deepika Atre, Payal Jaswani, Priyanshi Asawara, Text Classification in Data Mining, *International Journal of Scientific and Research Publications, Volume 5, Issue 6, June 2015*, 2015

<u>Web Links:</u>

- http://www.nltk.org/book/
- http://stackoverflow.com/questions/13788229/very-simple-text-classification-by-machine-learning
- https://en.wikipedia.org/wiki/Document_classification
- http://aitopics.org/topic/text-classification
- https://en.wikipedia.org/wiki/Artificial_neural_network
- https://en.wikipedia.org/wiki/Supervised_learning
- https://en.wikipedia.org/wiki/Unsupervised_learning
- https://en.wikipedia.org/wiki/Cluster_analysis
- https://en.wikipedia.org/wiki/Dimensionality_reduction
- http://www.nltk.org/book/ch06.html
- https://www.coursera.org/learn/neural-networks/home/welcome
- https://www.coursera.org/learn/machine-learning/home/welcome