Document Understanding using LSTM (Long Short Term Memory) Neural Networks

By
Muhammad Kamran
NUST201463910MSEECS60014F

Supervisor
Dr. Faisal Shafait
Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Technology (MS IT)

In
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(November 2016)

Approval

It is certified that the contents and form of the thesis entitled "**Document Understanding using LSTM (Long Short Term Memory) Neural Networks**" submitted by **Muhammad Kamran** have been found satisfactory for the requirement of the degree.

Advisor: Dr.Faisal Shafait

Signature:……..

Date:……...

Committee Member 1: Dr Asad Anwar Butt

Signature:…….

Date:…….

Committee Member 2: Dr Anis ur Rehman

Signature:…….

Date:…….

Committee Member 3: Dr Ahmad Salman

Signature:.…….

Date:.…….

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: Muhammad Kamran

Signature:……..

Acknowledgements

I take this opportunity to express my profound gratitude and deep regards to my supervisor Dr Faisal Shafait and my GEC members Dr Asad Anwar Butt, Dr Anis ur Rehman & Dr Ahmed Salman for helping me throughout the last year to successful accomplishment of my thesis. Their blessing, help and guidance enabled me to complete objective of my thesis.

I would also like to thank TUKL-NUST lab members including Tayyaba Naz, Omer Asif, Ahsan Jalal, Muhammad Zubair and Mohsin Ghaffar who motivated and helped to bring the best out of what I could achieve. A special thanks to Junaid and Sarwat two final year undergraduate students for being there whenever I got stuck during the thesis work.

Table of Contents

List of Figures

Abstract

To endure through the throat cutting contention, business industry and researchers are compelled to take intelligent and proficient decisions. This gives rise to the requirement of an automated solution powered by cutting edge technology that can extract the information in the blink of an eye and concoct unprocessed scanned image data, emanating in resourceful and perspicacious information at one's disposal. The competency can be consummated through evaluating the scanned document in a certain manner. Nonetheless, to extract bibliographic data from the scanned document image, there is a necessity of understanding of complex semantics of structured textual and visual data, which is precisely what majority of the existing automated solutions curtail. Therefore, our ambition was to develop the Document Understanding system that will be able to analyze the document scanned image data considering the current business and research requirements of an organization. Powered by the highly acclaimed Long Short Term Memory (LSTM) neural networks and complemented by its rich and user friendly interface on top, our system is a dynamic bibliographic extracting solution. Our solution will save your precious time.

This thesis presents a document understanding system for scanned images of medical journals articles documents. Using Long Short Term Memory(LSTM) neural network, the bibliographic data that includes titles, authors, affiliation and abstract is extracted conveniently. Results show phenomenal agreement with theoretical predictions and significant improvements over previous efforts in this domain. The work presented here has abstruse and sagacious implications for future studies of logical analysis of layouts of documents and may one day help solve many such existing problems.

Chapter 1:

INTRODUCTION AND MOTIVATION

Over the past few years many techniques have been designed to remit the challenges appearing in the document understanding domain. Document Understanding system mutate meaningful information into a formal representation. Deduction of meaning of communication from written text and its representation is the reason of inclination to this topic. We will use Long Short Term Memory (LSTM) for training a model that can detect a relationship between written text and its representation.

1.1 Logical Layout Analysis

Analysis of the logical layout of documents, not only enables an automatic conversion into a semantically marked-up electronic representation but also reveals options for developing higher-level functionality.

Plain-text exploration process in huge databases of scientific journals usually takes a lot of time. This procedure can be accelerated by the addition of some metadata. For instance, describing chunks of the texts that embody title or abstract, which helps in targeting the research on the respective fields. This extraction of information from the images of the document is mostly manually done. With the advent of time and growth of paper databases, automatic extraction of meta-data became a very relevant obstacle. This information consists of both textual and logical objects. The textual components include event dates, names, or ID numbers. The logical objects may either be extracted from the structure of the document or from information, which is included from existing databases.

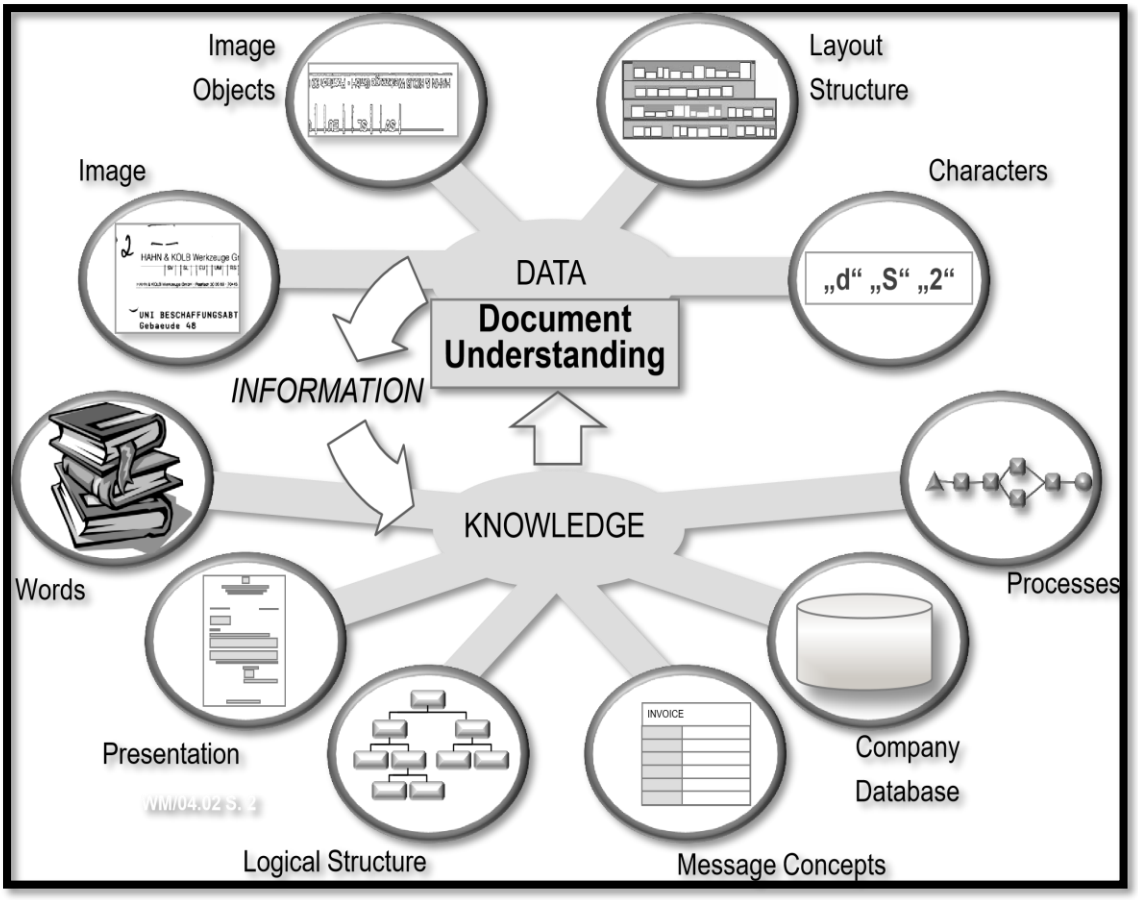


Fig 1.1: Document Understanding System [3].

For achieving this, document understanding classification gives a system to automatically convert essential information from a raster image into a formal portrayal. Therefore, it is a pattern of analysis in which the interpretation of communication is

concluded from the amalgamation of the written text whereas the logical labelling is the automated extraction of bibliographic data from the first page i-e the title page of the research journal papers which is automatic categorization of the function blocks. So we should have to add suitable and precise classification labels to blocks where an image and segmentation is inured.

Normally we have four labels on main title pages of research journal papers that include Title, Author, Abstract and Affiliation. The technique of automatically drawing

out such information establishes with scanning the article, transforming the bitmapped image into text by optical character recognition (OCR), zoning the neighbor text to set up the text zones, and then identifying the zones by the respective labels (title, author, affiliation, abstract.).

1.2 Neural Networks

Artificial neural networks (ANNs) were primitively matured as mathematical illustration of the information refining means of biological brains [1][4]. However, it is evident that ANNs have slight affinity to actual biological neurons, they enjoy endure acclaim as pattern classifiers. The primitive architecture of an ANN is a chain of minuscule nodes, which are coupled together by weighted relations. Particularly the elementary biological miniature, the bulge personifies neurons, and the connection weights personify the potency of the synapses amid the neurons. The structure is triggered by lending an input to some or all the nodes, which then escalated right through the structure onward to the weighted connections. The electrical liveliness of biological neurons naturally persues a streak of sharp “spikes”, and the triggering of an Artificial Neural Network node was initially designed to imitate the average firing rate of these spikes

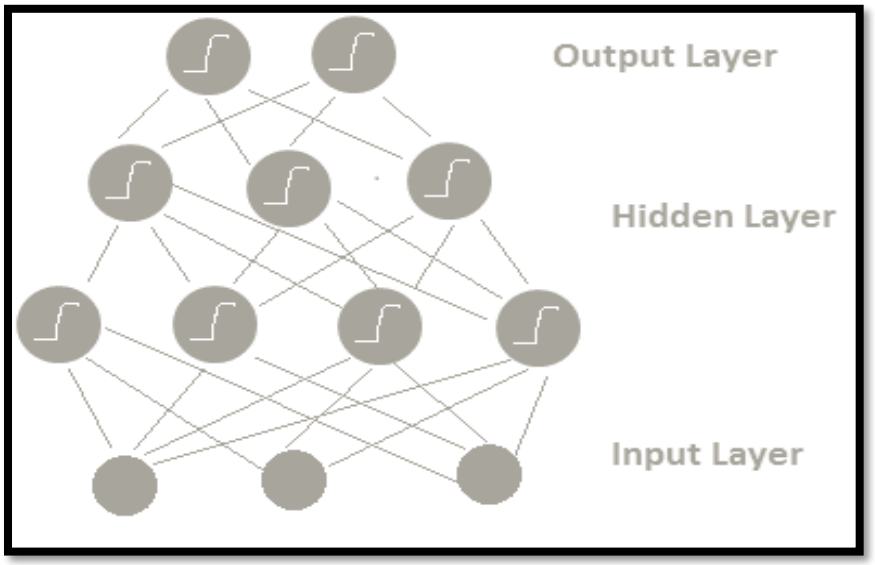


Fig 1.2: Artificial Neural Network

Recurrent networks usually follow their feedback connections to save model of up-to-date input events in the mode of activations [2] This is likely to be compelling for different applications, including speech refining, non-Markovian control, and music composition. [5] While the broadening of scope from multi-layer perceptron to recurrent neural network may seem frivolous, the significance for sequence learning are extensive. An MLP can exclusively mapped from input to output vectors, while an RNN can be postulated from the outright history of preceding inputs to each output. In fact, the corresponding aftermath to the universal approximation rationale for MLPs is that an RNN with an ample number of hidden units can approximate any quantifiable sequence-to-sequence mapping to frivolous accuracy [6]. The major point is that the recurrent connections grants a memory of preceding inputs to prevail in the network's internal state, which can then be used to put an impact on the network output.

1.2.1 Long Short Term Memory (LSTM)

Sadly, for standard recurrent neural network architectures, the magnitude of situations that can be accessed is fixed. [9] The dilemma is that the impact of a given input on the hidden layer, and accordingly on the network output deteriorates or blows up aggressively as it revolves about the network's recurrent connections. [8] This is generally called as vanishing gradient problem. Fundamentally this problem makes it pretty difficult for a recurrent neural network to learn tasks having 10 time steps or higher delays between suitable input and objective events [10] [11]

The LSTM architecture includes certain recurrently coupled subnets, familiarly called as memory blocks. These blocks can be contemplated of as a differentiable rendition of the memory wafer in a digital computer. Each block encompasses either one or more self-connected memory cells and three multiplicative units the input, output and forget gates that provide continuous analogues of write, read and reset operations for the cells. [1] The figure below gives an interpretation of a single cell LSTM memory block. An LSTM network

is established precisely like an elementary RNN, with the exception that the nonlinear units in the hidden layer are taken over by memory blocks. As a matter of fact, LSTM blocks may be infused with manageable units if imperative albeit this is normally not mandatory. Additionally, as with the RNNs, the hidden layer can be connected to any type of differentiable output layer, depending on the regression or classification task.

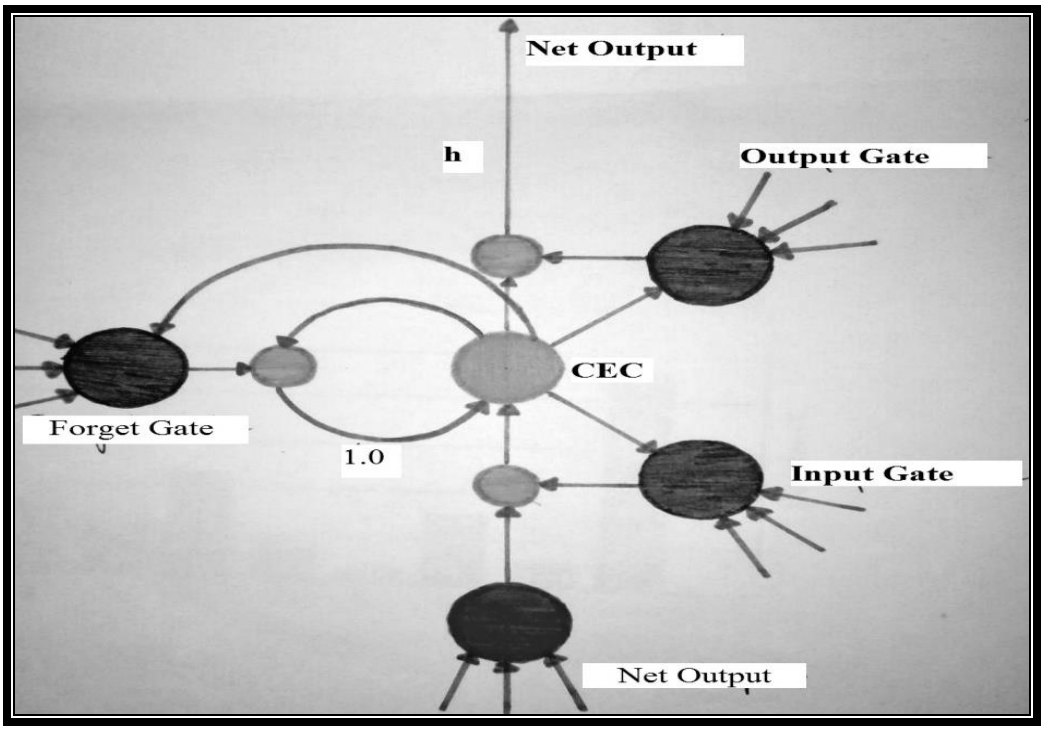


Fig 1.4: LSTM with one cell

1.3 Medical Article Record Groundtruth (MARG) Dataset

Distinguishing geometric features to layout algorithms is not easy for automated data extraction, because of the diversity of layout geometries in around 4000 journal titles recorded in MEDLINE. Ground truth data illustrative of the collection of biomedical journals ought to comprise instances of entire important layout types. [7]. The MARG site encompasses prototypes of 9 layout categories or types organized conferring to the zones of the Article Title, Author, Affiliation, and Abstract.

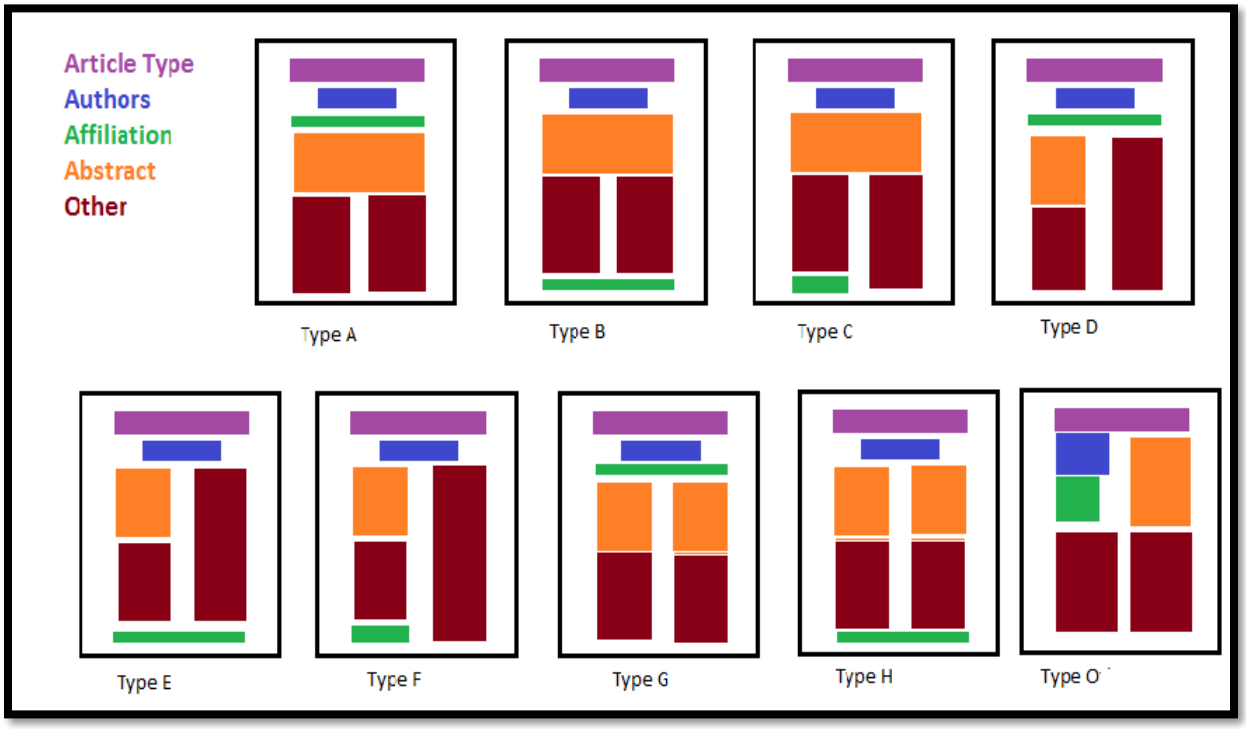


Fig 1.5: Page Layout Types

The extensive portraiture of all types is given below

• **Type A** – At the top of the page all the Title, Author, Affiliation and Abstract are placed in the assigned pattern.

• **Type B** - The top part of the page contains Title, Author and Abstract while the Affiliation is placed at the bottom part of the page.

• **Type C** - The top part of the page contains Title, Author and Abstract while the Affiliation is one columned and placed in the left column of double column text.

• **Type D** - The top part of the page contains Title, Author and Affiliation while the Abstract generally is placed in the first column. Some exceptions have the abstract lasting till a portion of the second column too.

• **Type E** - The top half part of the page contains Title, Author and Affiliation while the Abstract generally is one-columned and found above the body text of the article in majority of the times.

• **Type F**- The top half part of the page contains Title and Author. The Affiliation is located at the lowermost-left. The Abstract generally is in entire or some of the first column. Some exceptions have the abstract lasting till a portion of the second column too.

• **Type G** - The top part of the page contains Title, Author and Affiliation while the Abstract is in below two adjoining columns.

• **Type H** - The top part of the page contains Title and Author. The Affiliation is located at the bottom. The Abstract is dual-columned, and mostly placed above the body text of the article.

• **Type O** - This group, possessing all unconventional layouts confronted, containing approximately 23% of the whole journal compilation. Layouts in this group have not been grouped more as till yet.

Below is the bar graph chart that shows the exact number of journals that are included in each type.

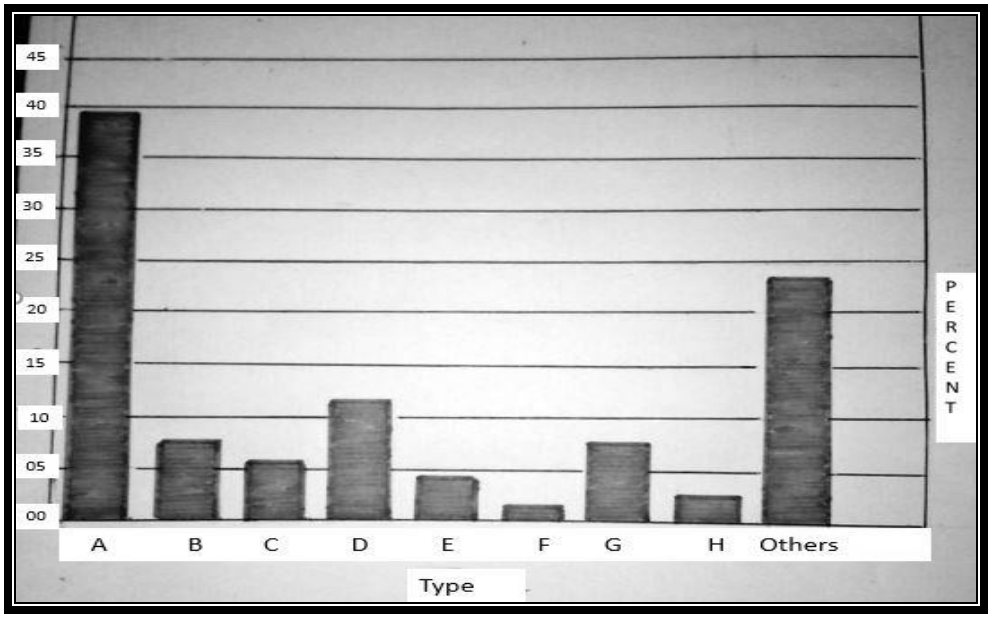


Fig 1.6: Distribution of Layout Types

1.4 Outline

In the Chapter 2, we will discuss the different alternatives presented in the literature related to Long short term memory. There is a lot of related work previously done in this domain. We will make a comparison with those substitutes available to LSTM. We will deliberate in detail the issues related to other options, which are solved by long short term memory.

In the Chapter 3, we will look at the methodology we used for our experiments. The techniques include different programming languages (Python, C++), textual feature extraction, image feature extraction and the concatenation of these features to obtain the results.

In the Chapter 4, the implementation of the techniques will be the main topic of discussion and the results obtained using those methods and techniques. Several screenshots will be attached for the authenticity of the results.

In the last Chapter, we will review the method and technique we used. We will discuss what can be done in future and what are the shortcomings of our method.

Chapter 2:

LITERATURE REVIEW

The role of a document image analysis and recognition (DIAR) system is to provide an electronic and editable version of a paper document. For example, a user wants to find quickly some interesting documents inside a corpus, based on some keywords. He could use a plain-text search while an Optical Character Recognition (OCR) would be able to extract the text from the pages. However, generally it is a waste of time for the user and the content providers, digital libraries or publishers (e.g. CiteSeer, European Library, Library of Congress, Springer, Elsevier, etc.). The raw results of an OCR appear insufficient when the user needs to focus on some structural metadata such as specific titles, list of authors, paragraphs, tables. Indeed, both users and content providers prefer working on a specific part of the document ("the advanced search") to focus their search on more meaningful zones like "Title" or "Author". For an advanced query, the amount of computations is reduced and it should return less but more interesting documents for the reader. [12] [13] [14]

Previous & Related Work

In this segment, we will provide a concise synopsis on different equivalent advents for time series processing and give references to the pioneer works. During the thesis, we will discuss LSTM, its application on our dataset and its difference from other techniques available.

Time window approaches: Any static or constant pattern matching mechanism like the feed-forward network having a consistent time window of latest inputs can be used as material sequence processing arrangement. This way of doing things has many momentous disadvantages and it is very complicated as well. Firstly, it is very troublesome to figure out the favorable time window length, if it ought to have any. Secondly, for tasks that involve long-term reliance or interdependency, a massive input window is required. A quick-fix to this problem might be to handle it with amalgamation of many different sized time windows. But we can apply this solution

only if we priory know the explicit long-term inter relation of the task to be done, which is normally almost impossible to have known. Thirdly stable time windows are not enough when we have uncertain long-term dependencies. In all the three cases, it is not advisable to use this approach.

An approach using recurrent neural network, can circumvent these issues, because RNNs do not have to need connection to the previous inputs. They can most likely imbibe to take the essence, show a Markov state and eventually by doing this they avoid the above described problems.

2.1 RNNs

Elman networks and RNNs with context units. In Elman network the information of the hidden units is added to what we call as context units, is given back to the hidden layer [15]. This arrangement is comparable to a network with a hidden layer, in which every unit is given back into each and every additional unit by means of time delayed connections with singular delay. The training of Elman networks is usually done by means of back-propagation [16]. Therefore, they do not have to go back to check for errors. The substitute methods, having context units the hidden unit is completely associated to the context units It means that they are fed directly to the context units. There may be a case in which the difference in number may occur of context units and hidden units. Almost in most the cases the training is done by BPTT or RTRL and their curtailed adaptations.

Time delay neural networks (TDNNs). Time-Delay Neural Networks (TDNNs) [17] grant entrée to previous events by means of cascaded intramural lingering lines. The hiatus they can connect is built upon the fact that which network topology they are having. Therefore, they always face the similar difficulties which pose a threat to the correctness of time window based feed-forward networks.

Nonlinear autoregressive models with exogenous inputs (NARX) networks. Nonlinear autoregressive models with exogenous inputs networks [18] acquiesce for quite

a few lucid input time-windows conceivably of singular size with contrasting material offsets. There is a very higher possibility that they are able to deal with the tasks having static long time delays but as a matter of fact it will continue to be a dilemma to suggest and choose the right size of windows. And thus, this approach seems most likely to fail if the longer duration inter dependencies are non-stationary.

Focused Back-propagation. Mozer[19] devised a technique that is able to cater the longer time lags. This technique uses time constants which dominates and put effects on activation advancements. Nonetheless, if there are longer time chasms the time constants require extrinsic small adjustments. Sun et al.'s [20] surrogate method renovates the awakening of a recurrent unit by tallying up with the previous activation and the calibrated ongoing net input. The latest net input, nonetheless, have the tendency to perplex the gathered information. This perturbation of the information also makes long term storage illogical and absurd.

Continual, Hierarchical, Incremental Learning and Development. Continual, Hierarchical, Incremental Learning and Development method also known as CHILD method was initially given by Ring [21] for taking into account the longer time lags. For every reception of incompatible or contradictory error signal by a unit in his network, he added a unit imprinting relevant connections of even higher order. Albeit Ring's method seems to be supremely fast, to cater for a time lag
associating 80 steps may not suffice for lesser units rather it may need the addition of 80 more units. When the lag durations of sequences are, unknown or may seem unseen in that case the network will not be able to make a sweeping statement or generalize.

Chunker Systems. Chunker Systems [22] perform well only in the case when the input sequence illustrates locally anticipated invariability. In that case only, this system does have the abilities to cater for the variable time lags.

Long Short Term Memory. LSTM does not go through from any of the issues that are mentioned above which different systems suffer from. It seems to be the futuristic and

best approach for recurrent networks faced with practical, pragmatic longer time delays between happening of applicable events.

2.2 RNNs versus Other Sequence Processing Approaches

Discrete Symbolic grammar learning Algorithms. SGL Algorithms [23] [24] have the tendency to quickly grasping grammatically constructed event sequences that are distinct, and noise-free. However, this algorithm fails to do fine in the presence of noise or when the inputs are sequences of real-valued values [25].

Hidden Markov Model. Hidden Markov architecture is extensively used technique for sequence processing operations. They work fine especially when the inputs are noisy or if the inputs are alternate to non-sequential temporal stretching. As they are not affected whether a work is spoken pretty fast or really slow so, that makes Hidden Markov Models in particular are very useful in speech recognition. Apart from that Hidden Markov Models are not that effective for different other assignments. [26] This is due to the fact that contrary to recurrent neural networks, HMMs are restricted to only distinct state spaces. This discrepancy makes their utilization to different time series processes inconvenient and ponderous. [27] For instance, if we have to do small counting tasks, hidden markov models requires the same number of states as the count of symbols on the most towering sequence that must be calculated. Also, there is another plus point related neural network, is when dealing with RNNs, the fundamental algorithm can be started with networks of mere 3-4 units [28]. Hence, in general recurrent neural networks are for more suitable

for the tasks that cannot be done by using hidden markov model.

Input output hidden Markov Models. The input-output hidden markov model architecture [29] blends components of mixture-of-experts and the two leading techniques, the recurrent neural network and HMMs, and is refitted by means of the expectation-maximization algorithm. Mixture of experts (ME) is one of the most popular and

interesting combining methods, which has great potential to improve performance. As far as my awareness is concerned, this model has not yet been practiced to the chores commensurate to the above described one. However, it is known to help in solving of the tasks that have longer time lags. It is basically seen to solve such tasks and maybe it will be a key in future if we see it as in a research point of view.

Genetic Programming and Program Search (GE & PIPE). Genetic Programming [30] and probabilistic Incremental Program Evolution [31] are relatively a bit gradual because of the fact that gradient information is not present. But other than that, this algorithm can search in general algorithm spaces and can be useful at times.

Random guessing. For many uncomplicated criterion weight guessing technique gets us the answer and finds solutions a lot quicker than refined gradient algorithms [32]

2.3 Traditional LSTM

The elemental entity is mainly the memory block for the long short term memory network's hidden. The memory block customarily being used as a substitute for the conventional recurrent neural network's hidden units. A memory block sometimes incorporates a single and the other times many memory cells and a couple of robust, multiplicative gating entities which gate in turns, input and output to the entire cells along the block. With the consent of the task, memory blocks give consent to cells to divide among them the same gates, hence with respect to the emblem of adaptive criterion. All the memory cells have their essence, a recurrently self-bridged linear entity known as the "Constant Error Carousel" (CEC), whose provocation we declare the cell state. The constant error carousel is relatively helpful in the solution of the vanishing error obstacle. The confined error flow of constant error carousel continues to be in a constant state i-e neither increasing nor decreasing even if the current input or cell's error signals are not present. The constant error carousel is taken care of from both sides. The forward flowing activation is taken care of by the input gate and for backward flowing error they are protected by the output gates. When activation is almost insignificant that is, gates are

closed, insignificant inputs and noise do not have access to the cell, and the cell state would not fluster the rest of the network.

The cell state, S_c, is amended depending upon its ongoing or present state and their different sources of input namely net_c is input to the said cell whereas net_m and net_{out} are corresponding inputs to the input and output gates respectively.

In principle, as in the case of truncated back propagation through time, errors appearing at memory block's net inputs plus at the gates of memory blocks my not be affected by previous situations, however the incoming weights do get affected due to these errors. Specifically, earlier an error signal appears at an output of the memory cell, it gets evaluated and corrected to reach the required scale by the output gate and the output nonlinearity. After this happens then the discussed error signal penetrates inside the memory cell's linear constant error carousel, from where it can get back dubiously without ever being need to be adjusted. As a matter of fact, due to this, long short term memory neural network can link capricious time delays amid the input events and target signals. The one time when the error abdicates from the memory cell by means of an aperture input gate and the supplementary input nonlinearity, ahead of being pruned, it does adjust incoming weights and get scaled again. The repercussion of this pruning is that each long short term memory block depends on the type of output errors for its adjustment.
As far as this is concerned, the blocks do not barter error signals. So, it becomes very much difficult task for long short term memory to matriculate different tasks where one block has the privilege to handle other blocks without precisely abbreviating the output error. An example of such situation is a pointer that is pointing to a FIFO (first in first out) queue.

Horchieter as for now has clarified a comprehensive dimension of tasks with somewhat traditional long short term memory neural networks. First of the many is vastly recognized grammar criterion known as the embedded Reber grammar. Second of them is both full of noise and noise free sequences with delay up to and more than thousand

steps [33]. Third of the lot is the tasks that involves continuous values which feel necessity for the stock piling of values for longer duration of time and their respective summation and multiplication. Fourth and the important one is the temporal order predicaments with having inputs that are far-ranging disembodied.

Forget Gates: Forget gates are added as the modification to the existing long short term memory by Alex Graves. They have the ability to acquire information to reload memory cell composition erstwhile they are not required in the future. Forgetting may be materialized in cadence or in an input-dependent convention. Long short term memory acquiesces knowledge to be stocked with different and unusual time delays, and error signals to be transferred back through in time. This plausible firmness, nonetheless, can endow to as vulnerability in many different occasions. For instance, the cell state generally inclined to evolve linearly all along the demonstration of a time series. The nonlinear demeanor of sequence processing is forsaken to the compressing functions and the exceptionally nonlinear gates. If we exhibit a steady input stream, the cell states may well evolve in amaranthine convention, mustering imbibition of the output squashing function. This is materialized even if the condition and style of the complication advocates the cell articulates should be transformed frequently e.g. At the commencement of recent input sequence whose initiation are not unequivocally pinpointed. Concentration would tend to show the h's derivative fade away, hence thwarting the errors that are incoming and shape the cell output commensurate the output gate incitement. In such a way, the complete memory cell will be debauched into an accustomed back propagation through time unit, therefore the cell would eventually discontinue to function as a memory. Cell states inexplicitly goes back to initial state ahead of the commencement of every new coming sequence. The canonical method of weight decay, which is also known to benefit to incorporate the level of complete exertion inside the network, was entirely helpful to be established to bring about elucidations which were peculiarly susceptible to consummate various state advancements.

Our explanation to the complication described in the above paragraphs is to capitalize adaptive forget gates which have the ability to quickly grasp and find a way to reinitiate

memory blocks back when their compositions are obsolete and for that reason, counterproductive. By resets, it does not only anticipate prompt resets to initial state or zero but also it means the gradual resets correlating to gradually fading cell states. More definitely we reinstate traditional long short term memory's constant error carousel's unit weight with the activation of multiplicative forget gate.

Long short term memory's backward pass is a conducive combination of somewhat customized, truncated back propagation through time [34] and a modified adaptation of real time recurrent learning [35]. Output gates use somewhat, abridged back propagation through time. Truncated version of real time recurrent learning is being used by weights to cells, input gates and the novel forget gates. Truncations contemplates that entire errors are incised when they expose out of a memory cell or gate, despite the fact they are there to replace the incoming weights. The consequence is that the constant error carousels are the segment of the system by means of which errors can stream back enduringly. This makes long short term memory's amendments profitable without momentously influencing learning power: error flow outside of cells contributes to collapse exponentially nevertheless. [36]

To determine the computational complexity of enhanced long short term memory we must take into consideration as every weight to input influence entire cells in the memory block so therefore, weights to input gates and forget gates leads to additional extravagant updates as compared to others. We figure out a rather conventional topology practiced while performing the experiments. Each of the memory block is of the similar size; gates possess ongoing connections; output units and gates have a partial connection through a unit which have unity activation every time; alternative connections to output units emanate from memory blocks only; the hidden layer is completely affiliated.

Time Series. Time series touchstone complications present in the research articles, nonetheless, time and again are theoretically straightforward. They generally do not have the need to include recurrent neural networks under any circumstances, because all corresponding information related to the later event is guided by a certain latest event that

occurred during a small-time window. In this direction, we implement long short term memory to such theoretically straightforward tasks, to ensconce a reduction to the competence of the long short term algorithm in its most recent pattern. Grave has spotlighted two comprehensively discussed tasks. One of them is best prediction of the Mackery-Glass series [37] and tumultuous laser data from a competition at the Santa Fe Institute. [38]

Long short term memory is performed as a form of authentic autoregressive approach which can for once ingress the input from the recent time-step, unlike its adversaries checking single input at a time. In comparison, the other opponents to it concurrently check many different subsequent inputs in a conveniently decided time window. It is to be mentioned that time delay neural networks [39] are not essentially autoregressive, due to this they are permitted for explicit entrée to
previous events. We also consider in a series of distinct stages repeated cultivation as suggested by the famous machine learning expert Kuo [40] so that recurrent neural networks develop a compelling attractor in the place of regularly used commonly comparative output. Another machine learning guru Baker [41] tried to improve the repeated training scheme and got the result that it works well in comparison to stepwise training.

Arithmetic Operations. Common characteristic real circumstances sequence processing problems require the input streams that are continual, input representations that is distributed. All the states including targets, inputs and internal states should be continuous valued, and even longer time delays among the events that correspond. That's because of which the developers decided to be architecture as many different factitious nonlinear problems that help to associate these factors.

Because of its architectural design, traditional long short term memory is tailor made for problems engrossing addition, subtraction and integration. All such tasks are very important for many real circumstances problems. But when it comes to multiplication which is also a very important essential arithmetic operations, we must face certain

problems. Forget gates, nonetheless, as a matter of fact were pioneered to exculpate those memory contents which are not required anymore, greatly put a positive affect the long short term memory's performance on problems that include multiplication.

Context free and Context Sensitive languages. We know by various experiments performed by different scientists that long short term memory performs better than the traditional recurrent neural networks on learning regular languages from commendable training sequences. Long short term memory's far better results and achievements with context free language touchstones for recurrent neural networks. As far as we know long short term memory's alternatives are the source for first recurrent neural networks that should learn an uncomplicated and elementary context sensitive language.

Chapter 3:

DESIGN AND FUNCTIONALITY

During the recent era, there has been a flourishing enthusiasm and concern to consider bulk volumes of visual data and for that there is an equal amount of interest in the research associations as well as in the business surroundings. Business communities have the need to deal with millions of documents that are either scanned or digital. Hence there is a real-time requirement to implement classification and retrieval tasks on the huge corpora. For instance, segregating an incoming document might be really convenient to automatically determine the workflow where the document ought to be stored relying on the category to which it belongs. Also, we are concerned in retrieving current documents in a dataset that are somewhat identical to the incoming document. For instance, if we have a set of annotated documents, retrieving the rather comparable documents might also be advantageous for extracting certain metadata from those documents and then in the process delegate it to the incoming document. The real-time problems that involve classification and retrieval tasks have conventionally been solved according to the given textual information. Nonetheless, taking the excerpt of that information may sound difficult or impractical, because the documents can be outdated, or maybe of insignificant quality. There may be a case in which it contains very less textual information. It might be including languages that are not known or similar. In all the cases discussed above, it has now become important to rely on visual features.

Methodology

In our case, we started with breakdown into lines using OpenCV. Text is extracted using tesseract. Using this text, extraction of textual features is done used python and C++. It is stored in the form of histogram. After that, visual features are extracted using convolution neural networks. The mean and max features are the ones we used for our technique. The concatenation of both visual and textual features is done afterwards. Then LSTM is applied on different preprocessed datasets. The dataset we have used for these experiments is Medical Articles Records Ground truth (MARG). Below figure shows the flowchart description of our overall technique.

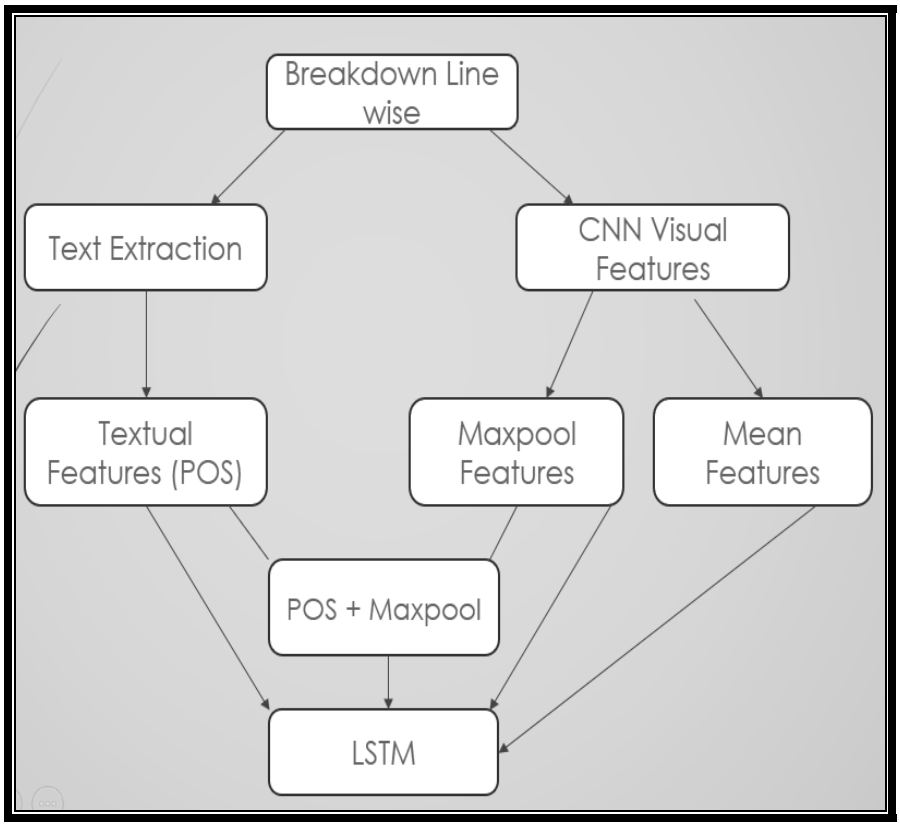
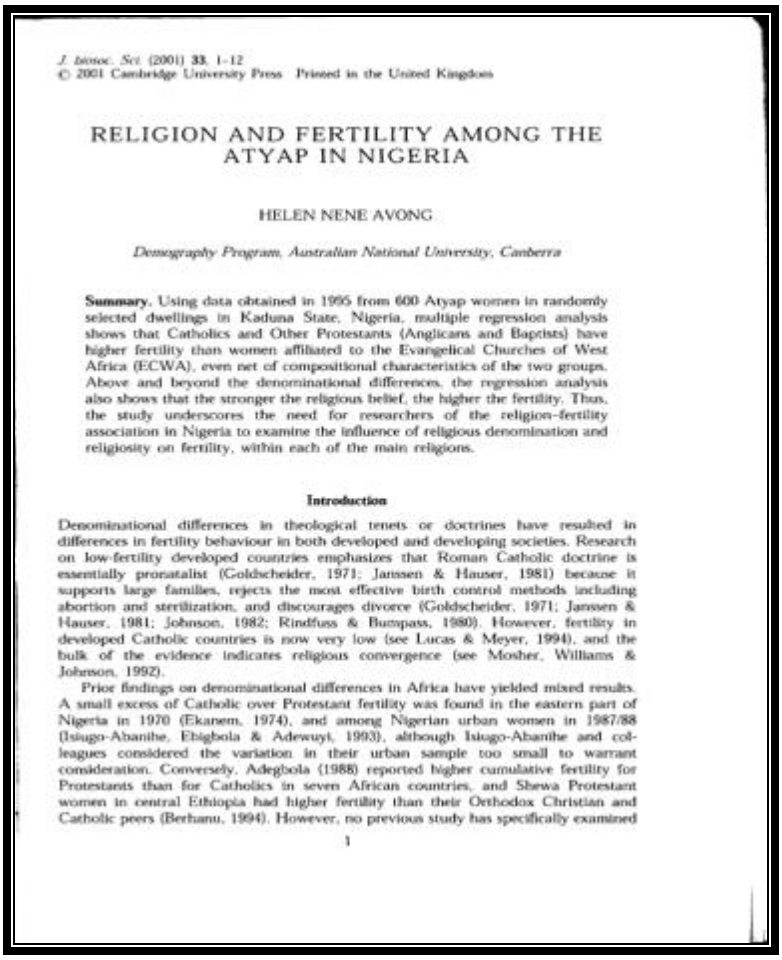


Fig 3.1: Flowchart Description

3.1 Textual Feature Extraction

The MARG dataset contains the title pages of medical journals. The technique which we are applying needs the textual features to be extracted first. The process of textual feature extraction completes in three steps. Let us take one image document from the dataset. For instance, we have taken an image from Type-A shown in the figure below.



J. biosoc. Sci. (2001) **33**, 1–12
© 2001 Cambridge University Press Printed in the United Kingdom

RELIGION AND FERTILITY AMONG THE ATYAP IN NIGERIA

HELEN NENE AVONG

Demography Program, Australian National University, Canberra

Summary. Using data obtained in 1995 from 600 Atyap women in randomly selected dwellings in Kaduna State, Nigeria, multiple regression analysis shows that Catholics and Other Protestants (Anglicans and Baptists) have higher fertility than women affiliated to the Evangelical Churches of West Africa (ECWA), even net of compositional characteristics of the two groups. Above and beyond the denominational differences, the regression analysis also shows that the stronger the religious belief, the higher the fertility. Thus, the study underscores the need for researchers of the religion–fertility association in Nigeria to examine the influence of religious denomination and religiosity on fertility, within each of the main religions.

Introduction

Denominational differences in theological tenets or doctrines have resulted in differences in fertility behaviour in both developed and developing societies. Research on low-fertility developed countries emphasizes that Roman Catholic doctrine is essentially pronatalist (Goldscheider, 1971; Janssen & Hauser, 1981) because it supports large families, rejects the most effective birth control methods including abortion and sterilization, and discourages divorce (Goldscheider, 1971; Janssen & Hauser, 1981; Johnson, 1982; Rindfuss & Bumpass, 1980). However, fertility in developed Catholic countries is now very low (see Lucas & Meyer, 1994), and the bulk of the evidence indicates religious convergence (see Mosher, Williams & Johnson, 1992).

Prior findings on denominational differences in Africa have yielded mixed results. A small excess of Catholic over Protestant fertility was found in the eastern part of Nigeria in 1970 (Ekanem, 1974), and among Nigerian urban women in 1987/88 (Isiugo-Abanihe, Ebigbola & Adewuyi, 1993), although Isiugo-Abanihe and colleagues considered the variation in their urban sample too small to warrant consideration. Conversely, Adegbola (1988) reported higher cumulative fertility for Protestants than for Catholics in seven African countries, and Shewa Protestant women in central Ethiopia had higher fertility than their Orthodox Christian and Catholic peers (Berhanu, 1994). However, no previous study has specifically examined

1

Fig3.2: Image of a Journal from MARG Dataset

First of all, we divided the image line-wise i-e each line is separated and stored as another image files. There were different types of documents in the dataset. Some were single columned, some included the two columns and others were triple columned. So, we had to adjust the OpenCV code accordingly. We have named these images which we have obtained now like 001-4-17. In this 001 is the name of the original image, 4 is the class to which it belongs, 17 is the line number of this image on the journal paper. We have 4 classes to be classified. First one is Title, second is Author, third is Affiliation and fourth is Abstract. This means this image is taken from 001 image, it belongs to abstract class and it is 17th line on the original journal. Below figure shows different image excerpts that are

broken down from the original image of the journal's title page which we have used as an example.

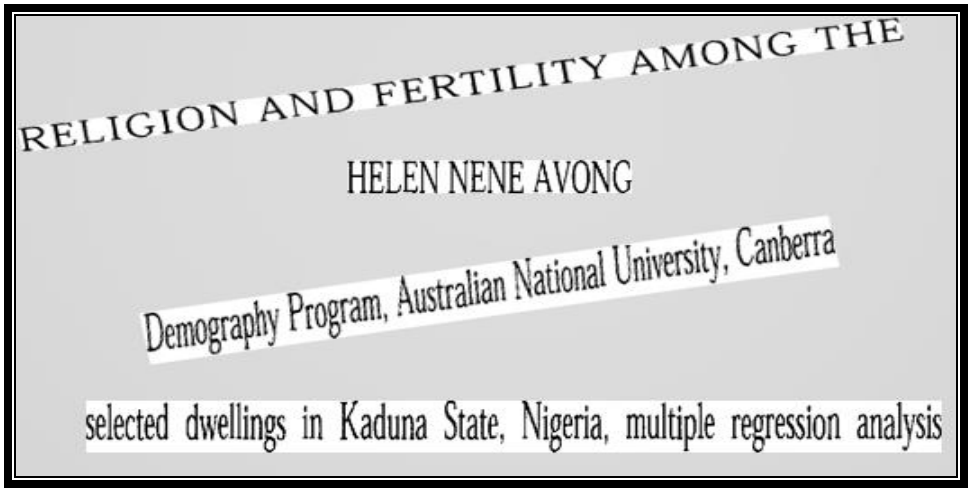


Fig 3.3: Line-wise Breakdown

When we have obtained the line-wise breakdown then we extract the text in those image excerpts. This is done using tesseract and stored in a text files. For naming of these files also the same convention is followed i-e original image name, class and line number. Below are the screenshots of some of the text files.

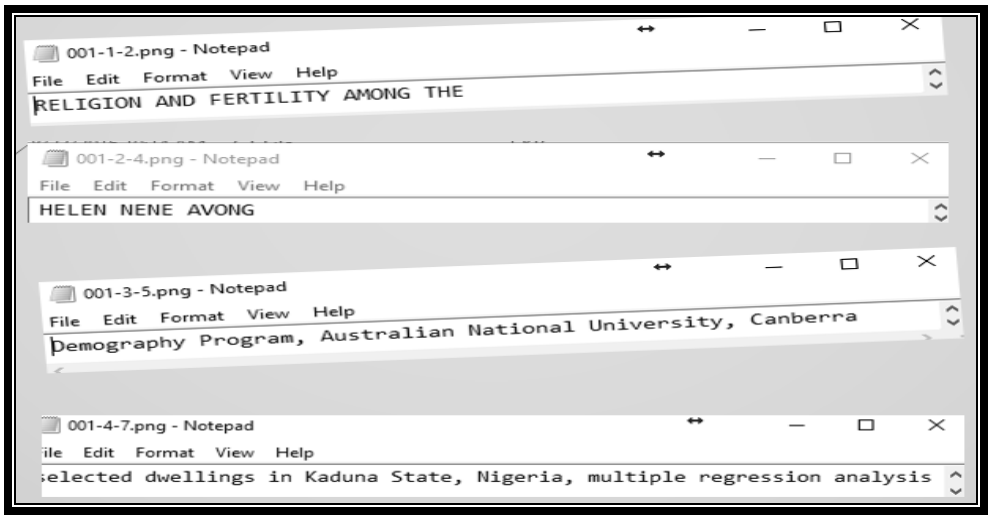


Fig 3.4: Screenshots of Text files

After the text is extracted, we apply the python code on that text. By using nltk (natural language tool kit) library of python we find out the textual features. These features are stored as a comma separated file. The textual features are the parts of speech. That includes conjunction, preposition, adjective, adverb, verb, proper-noun, others, noun and pronoun. These features are then normalized to 1 i-e. in the form of histogram.

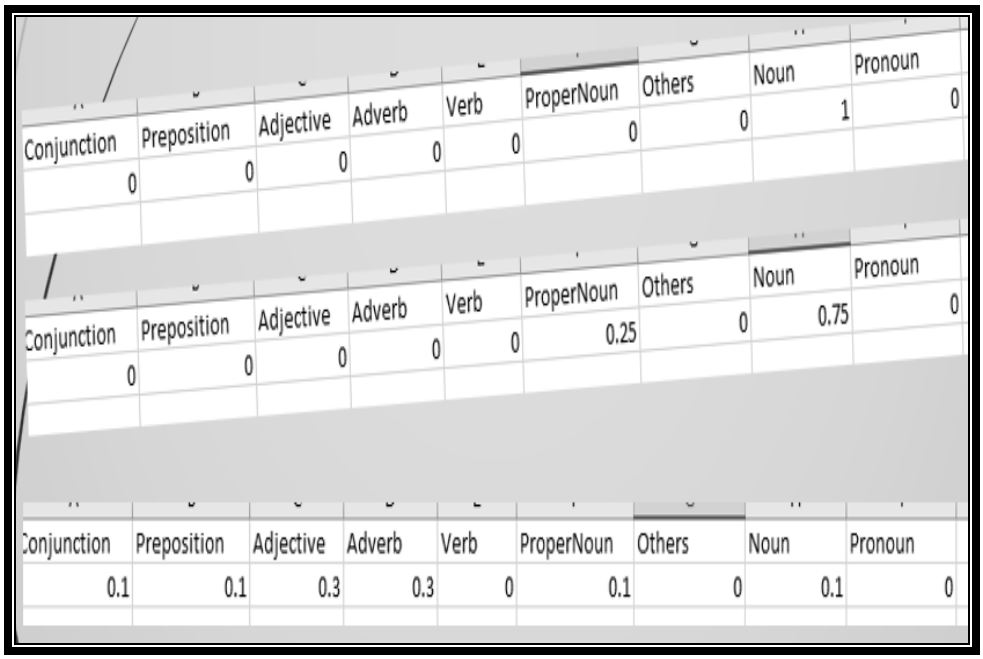


Fig 3.5: Screenshot of Histogram Features

3.2 Visual Feature Extraction

The next features which we dealt with were the visual features. These features were extracted using concurrent neural networks (CNN). The visual feature we obtain are the 512-D vector. The features are extracted from the final convolutional layer of VGGNet i.e. Conv5_3. As these features have dimension 7x7x512 for input of size 224x224. After that we combine them by either taking the max or mean of these 49 vectors. These features are also stored as comma separated files for each title page of the journal from the dataset. The naming convention for each csv file remained the same as discussed before.

3.3 Concatenation of Visual and Textual Features

After the visual features are extracted, we had textual features which we extracted using different programming languages and special libraries. Also with visual features, we had the preprocessed dataset ready to apply certain machine learning algorithms to check their usefulness for achievement of our solution to the given problem. We thought it might be advantageous to concatenate the visual and textual features. So, by using python language and its libraries we concatenated these two important features. It happened to help to solve our problem in a certain way.

Chapter 4

IMPLEMENTATION AND RESULTS

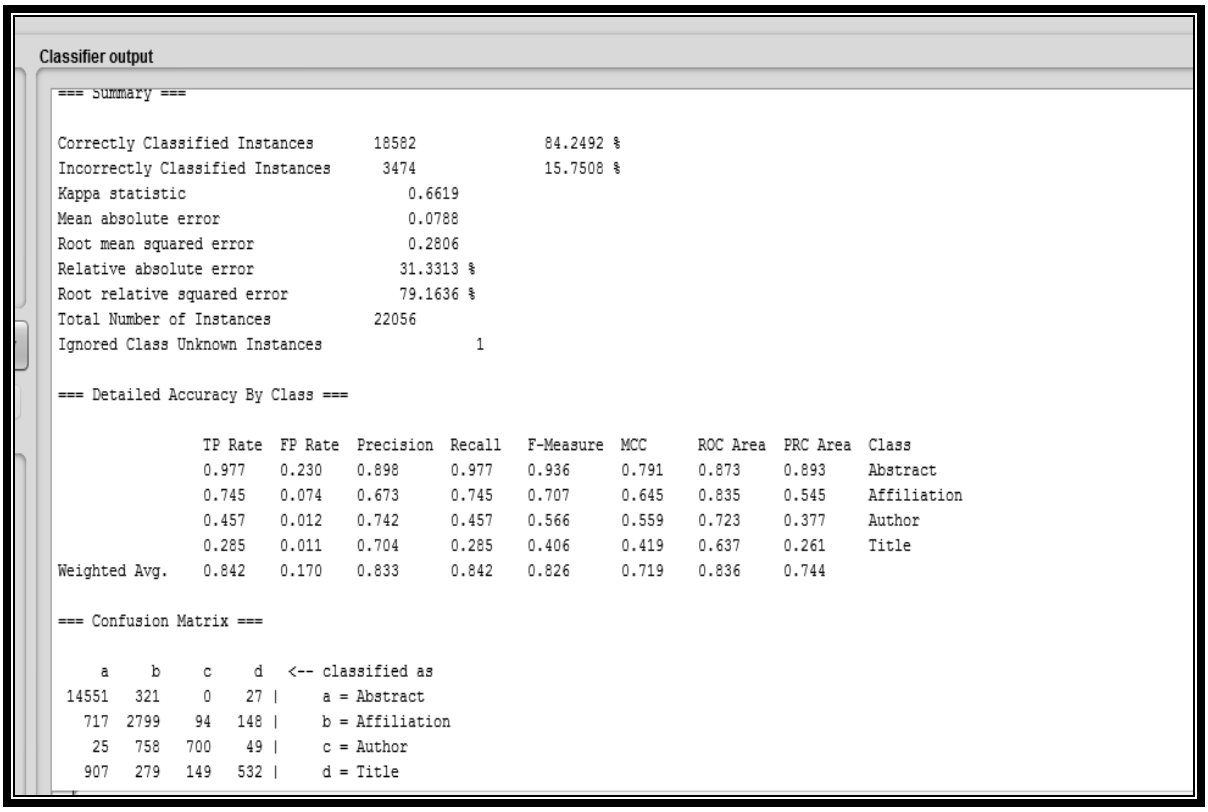
In this Chapter, we will provide the results that are generated after LSTM is applied. Several other techniques were also applied during the course of time throughout the thesis. Although the best results were found using LSTM (Long Short Term Memory) neural networks, yet we applied other techniques so that we can make the comparison and it came out good as our assumptions were correct. LSTM stands out among its competitors. The screenshots of all the results are given in each of the sections below.

For this experiment, we have 4 classes to be classified in the classification problem. These include Title, Author, Affiliation and Abstract which are labeled as 1,2,3 and 4 respectively for purpose of training our algorithm. We have used Linux with certain libraries for LSTM training. For other techniques, we have used weka software on windows. Now we will look at the results on textual, visual and concatenated features one by one.

4.1 Results on Textual Features

As discussed in the last chapter, textual features are in the such a format that these are 9-valued csv (comma separated values) files. So, we applied SVM and LSTM on it using default parameters. The result at first was not good because all the files were classified as others class. This was due to the fact that others class is quite a huge class. The classifier thought it to be the only class during the training. So, for further trainings we removed the others class which was not required but was a part of the journal's title page in most of the cases.

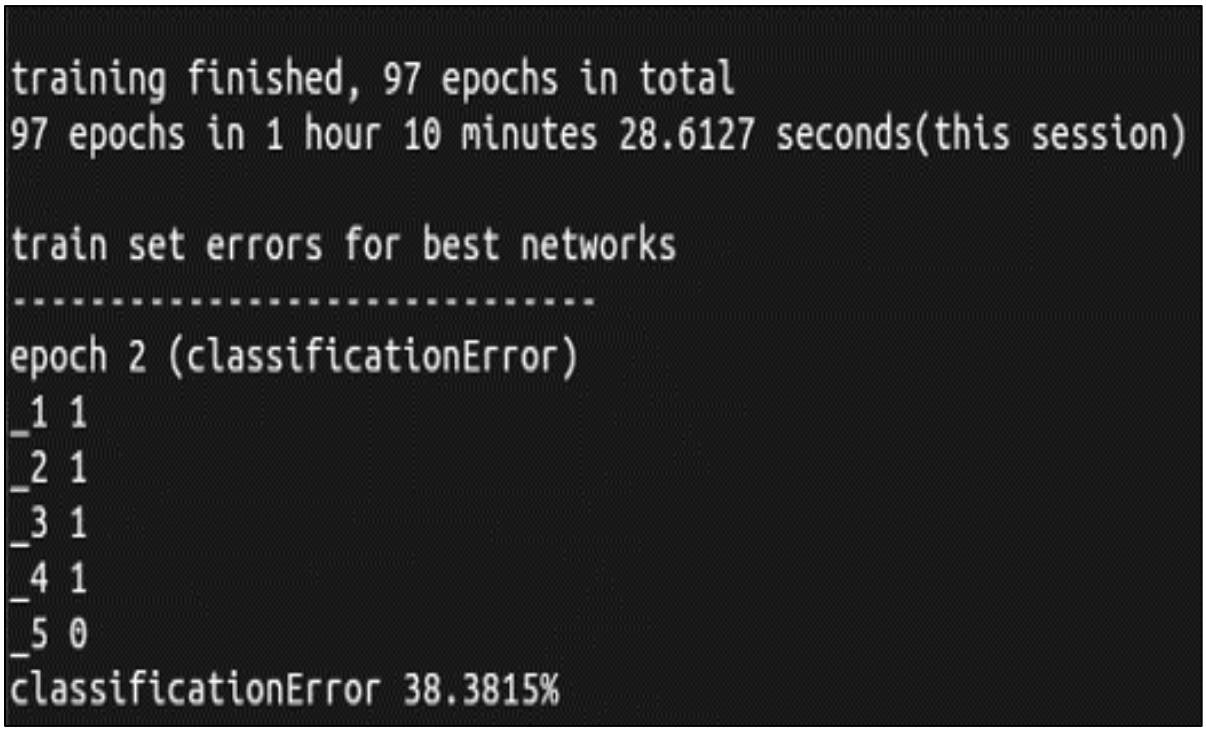
After removing the other's class, we got comparatively good results. With LSTM, we got 62% accuracy. The results for both SVM and LSTM is given below as screenshots.



Classifier output

```
=== Summary ===

Correctly Classified Instances       18582               84.2492 %
Incorrectly Classified Instances      3474               15.7508 %
Kappa statistic                          0.6619
Mean absolute error                      0.0788
Root mean squared error                  0.2806
Relative absolute error                 31.3313 %
Root relative squared error             79.1636 %
Total Number of Instances            22056
Ignored Class Unknown Instances                1

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.977    0.230    0.898      0.977    0.936      0.791    0.873     0.893     Abstract
               0.745    0.074    0.673      0.745    0.707      0.645    0.835     0.545     Affiliation
               0.457    0.012    0.742      0.457    0.566      0.559    0.723     0.377     Author
               0.285    0.011    0.704      0.285    0.406      0.419    0.637     0.261     Title
Weighted Avg.  0.842    0.170    0.833      0.842    0.826      0.719    0.836     0.744

=== Confusion Matrix ===

     a     b     c     d   <-- classified as
 14551   321     0    27 |     a = Abstract
   717  2799    94   148 |     b = Affiliation
    25   758   700    49 |     c = Author
   907   279   149   532 |     d = Title
```

Fig 4.1: SVM on Textual Features



```
training finished, 97 epochs in total
97 epochs in 1 hour 10 minutes 28.6127 seconds(this session)

train set errors for best networks
------------------------------
epoch 2 (classificationError)
_1 1
_2 1
_3 1
_4 1
_5 0
classificationError 38.3815%
```

Fig 4.2: LSTM on Textual Features

4.2 Results on Visual Features

For visual features, we discussed in chapter 3 that it is also a csv (comma separated file). Each file is 512-valued. Visual Features were extracted as mean and max vectors. So, we applied LSTM on both visual features. Here also we ignored other's class. The results on max came out to be good as comparison to mean vectors.

With LSTM, we got 70% accuracy. The results for both SVM and LSTM is given below as screenshots

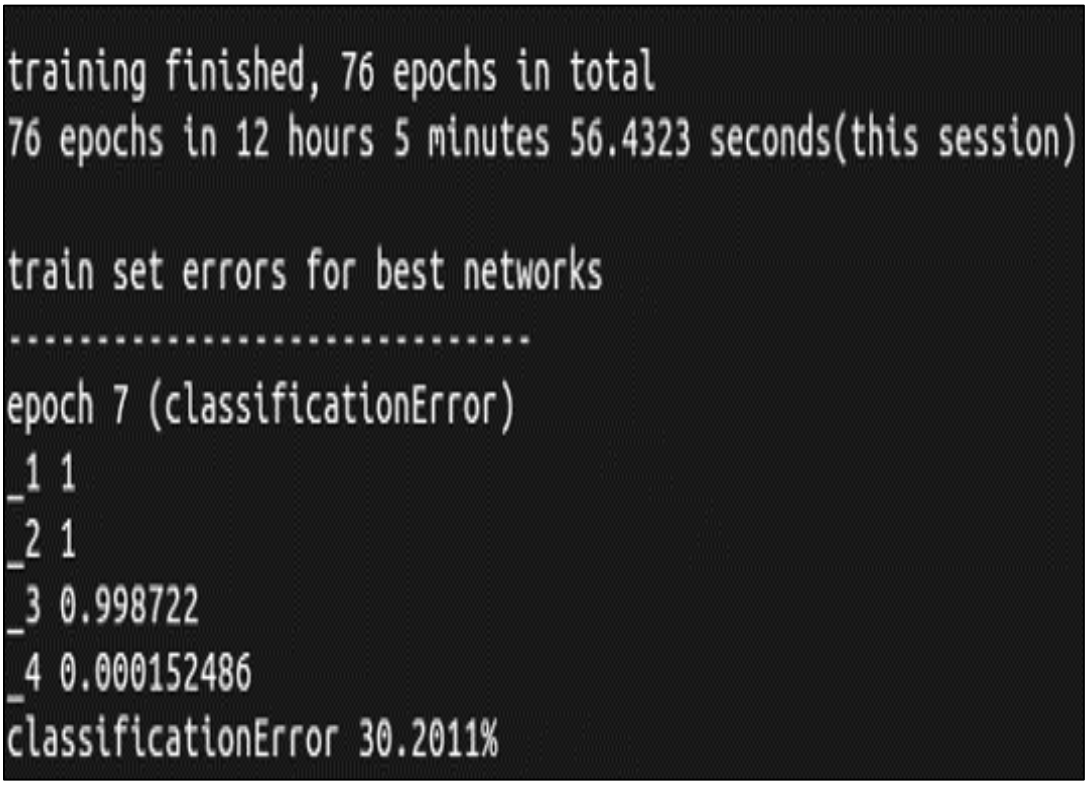


Fig 4.3: LSTM on Visual Features

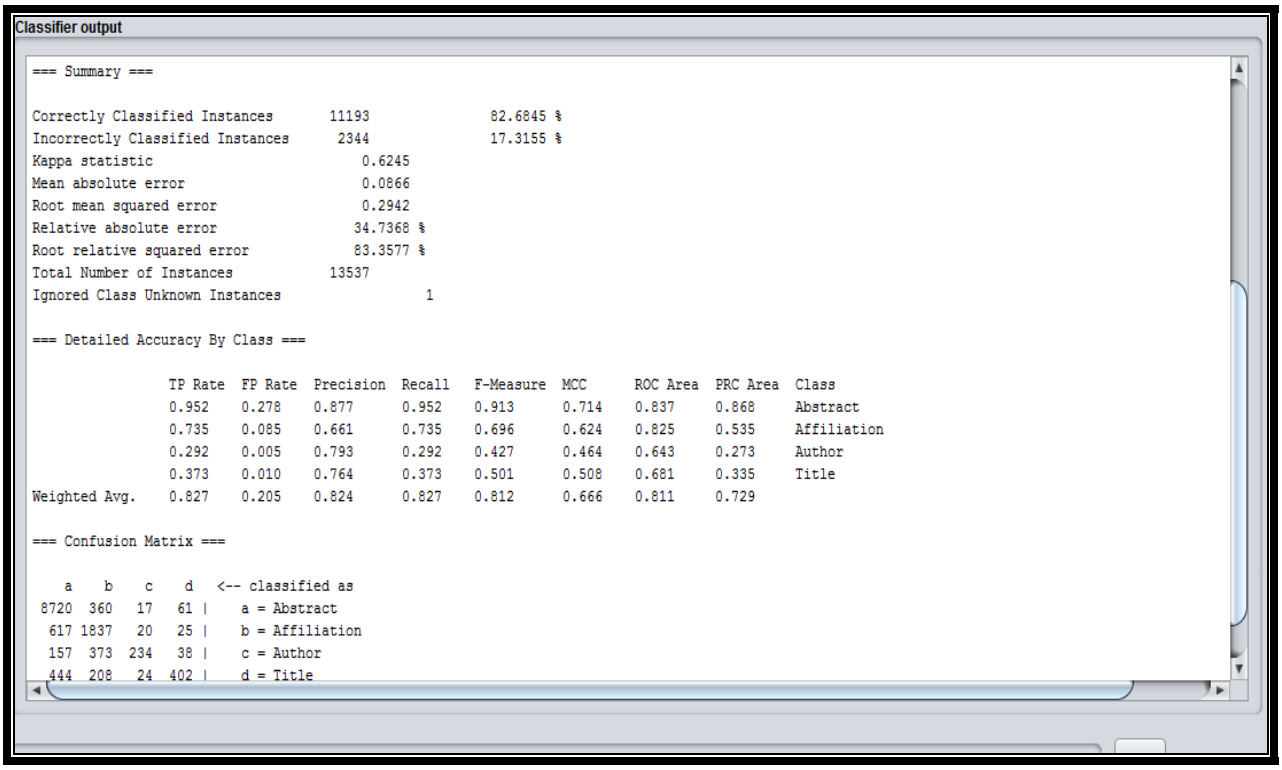


Fig4.4: SVM on Visual Features (Max)

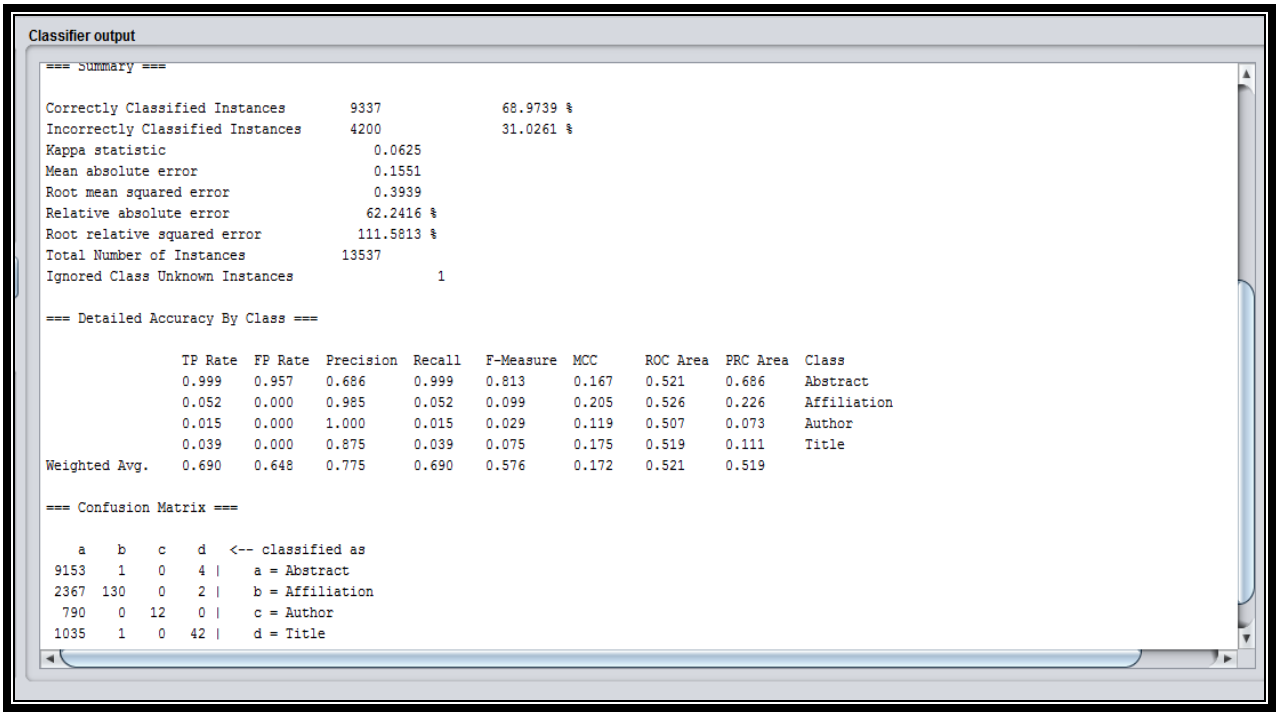


Fig 4.5: SVM on Visual Features (Mean)

4.3 Results on Concatenated Features

The concatenated features were obtained by combining 9-valued textual and 512-valued visual features. The results on combined were far superior than the results on them independently. We applied both SVM and LSTM on these concatenated features.

Also, in this case again, we have removed the others class. With LSTM, we got approximately 83% accuracy. The results for both SVM and LSTM is given below as screenshots

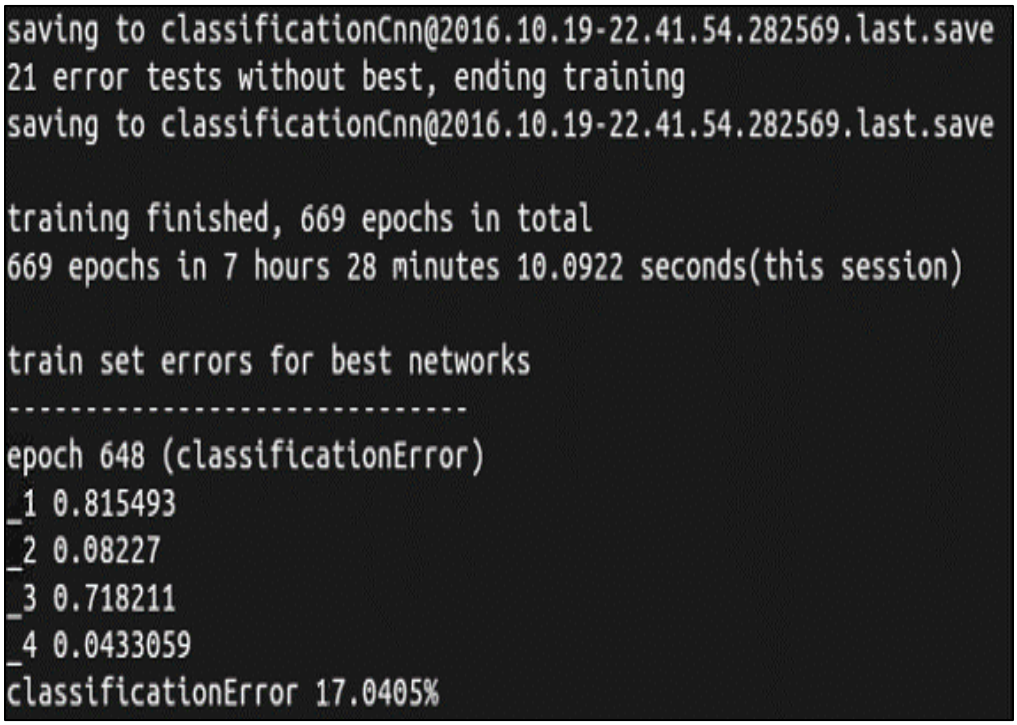


Fig 4.6: LSTM on concatenated Features

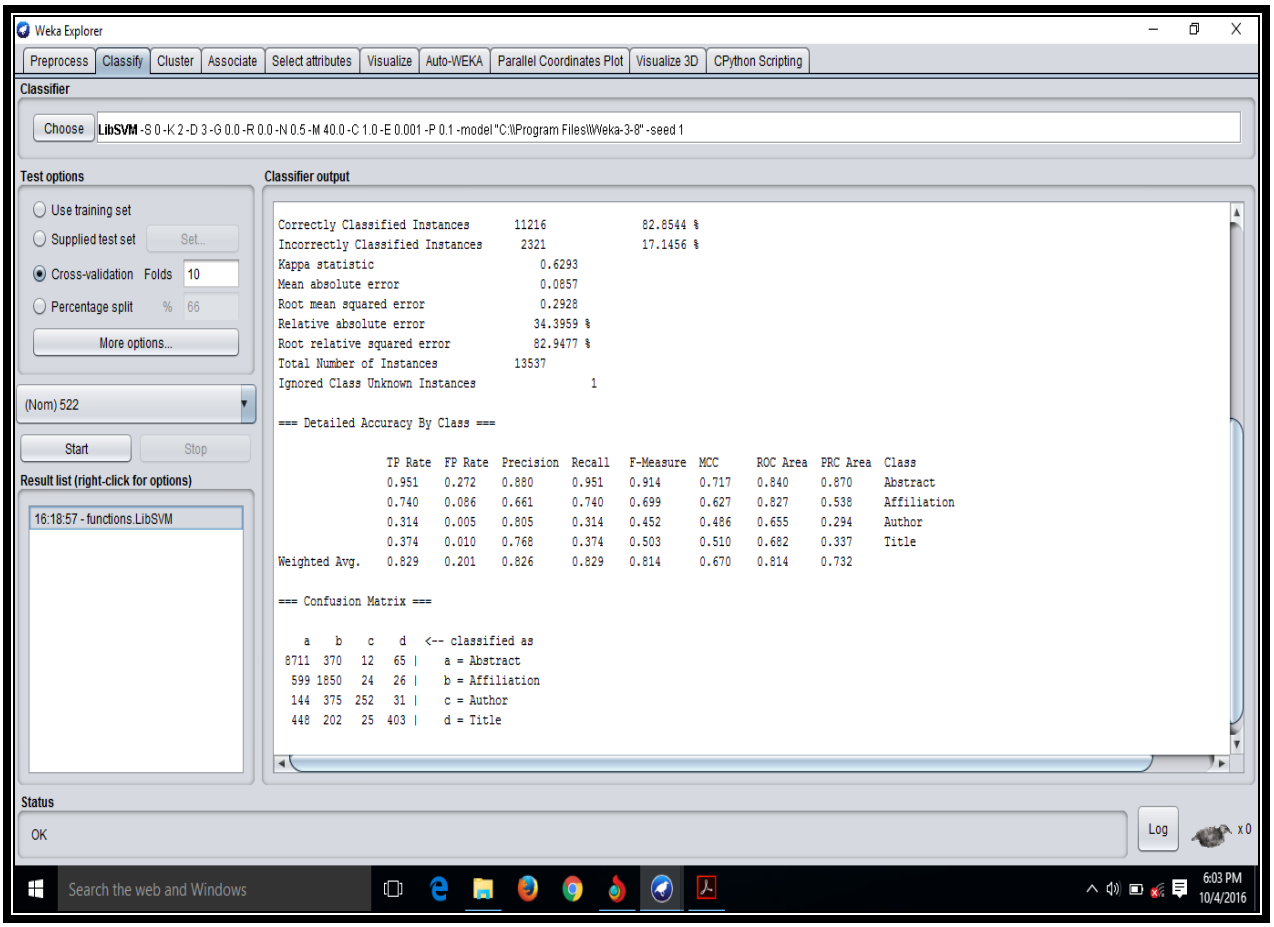


Fig 4.7: SVM on concatenated Features

Chapter 5

CONCLUSION AND FUTURE RECOMMENDATIONS

5.1. Conclusion

In short, our research can become a building block in the domain of document understanding systems in the coming future. We have used the best learning algorithm LSTM (Long Short Term Memory) neural networks which have proved its significance in many other domains. So, we applied that algorithmic technique for document understanding system. There were other techniques available in the literature. For instance, we checked SVM (support vector machine) results and found out that LSTM neural networks produced the best results for learning and classification. In future document understanding systems will be used in different domains such as bank invoices, postal services etc.

We first extracted the textual features by using natural language toolkit. It helped in classifying the author and affiliation class. After that we extracted the visual features. The visual features helped us in classifying the title as visual features are very useful in locating where that text is located on the page. But, the best results were achieved when we concatenated the results of visual and concatenated features.

The results we achieved were not up to the mark. This was due to the fact that preprocessing what we done was not that sufficient as we thought that it would have been. We have got the accuracy less than the state of the art method that is the case based reasoning method but I am sure that LSTM can give better results if preprocessing of the dataset results in a refined one.

5.2. Future Recommendation

We will recommend if taken further, this project can be made more advanced by certain improvement in the Natural Language Toolkit (nltk). There are certain limitations in existing nltk. It cannot differentiate between the proper nouns that are verb as well or any word that can be classified as two different parts of speeches.

Secondly, further refinement of visual features is required. We can concatenate the font size and boldness with the other features. Also, we can append the spacious location to the visual features. Also, we can use different layers of convolution neural network for better refining of visual aspects of the documents. These both will be useful in successfully classifying titles in the MARG dataset.

Thirdly, there were less documents available in the existing MARG dataset. The advanced level of MARG dataset is available now. It has almost double images of the journal title pages. So, our solution can be applied to that dataset to improve the results in the future. Because it is an improved dataset itself. This project can be complimented with advanced algorithms as to make use of huge volumes of scanned image data to predict future trends. The dynamic bibliographic extracted data created through our designed can help in the Big Data domain as well.

BIBLIOGRAPHY

1 Mozer, M. C. (1993): Neural network architectures for temporal pattern processing. In A. S. Weigend & N. A. Gershenfeld (Eds.), Time series prediction: Forecasting the future and understanding the past (pp. 243-264). Redwood City

2 F. Gers, N. Schraudolph, and J. Schmidhuber (2002): Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research, 3:115–143

3 A.Graves, " Supervised sequence labeling with recurrent Neural Networks" in Handbook of Document Image Processing and Recognition pg 1-137

4 S Hochreiter, J Schmidhuber (1997): Long Short-Term Memory: Neural Computation 9(8):1780.

5 A.Dengel, F Shafait, "Analysis of logical layout of Documents." in Handbook of Image Processing and Recognition pg 177-222

6 Mc-Culloch and Pitts (1988)

7 B. Hammer and Jochen J. Steil (2000): Perspectives on Learning with RNNs.

8 G Ford, G R. Thomas Ground Truth Data for Document Image Analysis.

9 A Graves, S Fernandez, J Schmidhuber (2003): Multi-Dimensional Recurrent Neural Networks.

10 Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner (1998): Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, November.

11 S Hochreiter (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies

12 Y Rangoni · A Belaïd · S Vajda (2001): Labelling logical structures of document images using a dynamic perceptive neural network

13 L Candela, D Castelli, Pagano, (2007): A reference architecture for digital library systems: principles and applications.

14 P Lervik, S Brygfjeld (2006): Search engine technology applied in digital libraries. ERCIM News 1(66), 18–19.

15 J. Elman (1990): Finding structure in time. Cognitive Science. 179-211.

16 D. Rumelhart, E. Hinton, R. Williams (1986): Learning internal representation by error propagation. Cambridge, MAL MIT Press.

17 P. Hafner, A. Wiebel (1992): Multistate time delay networks for continuous speech recognition.

18 T. Lin, B. Horne, P. Tino C.L. Giles (1996): Learning long-term dependencies in NARX recurrent neural networks. IEE Transactions on Neural Networks 1329-1338.

19 M. C. Mozer (1992): Induction of multiscale temporal structure. Advances in Neural Information Processing Systems. Pg. 275-282.

20 G. Sun, C.L. Giles, C.Y. Lee (1993): The neural network pushdown automation. Technical Report No. CS-TR-3118, University of Maryland, College Park.

21 M.B Ring (1994): Continual learning in reinforcement environments. University of Texas at Austin, Austin, Texas 78712.

22 J.Schmidhuber (1989): The neural Bucket Brigade, a local learning algorithm for dynamic feedforward and recurrent networks. Connection Science 403-412.

23 L. Lee (1996): Learning of context-free languages. A survey of the literature. Centre for Research in Computing Technology. Harvard University, Cambridge, Massuchussets.

24 Y. Sakakibara (1997): Recent advances of grammatical inference. Theoretical Computer Science.

25 M. Osborne, E. Briscoe (1997): Learning stochastic categorical grammars. In Proceedings of the Association for Comp. Linguistics, Workshop Pg. 80-87 Madrid.

26 Y. Kalinke, H. Lehman (1998): Computation in recurrent neural networks. From counters to iterated function systems.

27 P. Rodriquez, J. Wiles (1998). Recurrent neural networks can learn to implement symbol-sensitive counting. In Advances in Neural Information Processing Systems. Pg. 87-93.

28 F. Gers, J. Schmidhuber (2000): Learning to forget. Continual prediction with LSTM. Neural Computation Pg. 2451-2471.

29 Y. Bengio, P. Fresconi (1995): An input output HMM architecture. In advances in Neural Information Processing Systems. San Mateo CA. Morgan Kaufman.

30 J.R. Koza (1992): Genetic Programming. Cambridge, MA.

31 R.P. Salustowicz, J. Schmidhuber (1997): Probabilistic incremental program evolution. Stochastic search through program space. Machine Learning. ECML-97. Lecture Notes in Artificial Intelligence. Pg. 318-362.

32 S. Horcheiter, J. Schmidhuber (1997): Long Short Term Memory. Neural Computation 1735-1780.

33 Y. Bengio, P. Frasconi (1995): learning long term dependencies with gradient descent is difficult. IEEE Transactions on Neaural Networks Pg. 157-166.

34 R. Williams, J. Peng (1990): An efficient gradient-based algorithm for online training of recurrent network trajectories. Neural Computations. Pg. 490-501.

35 A.J. Robinson, F. Fallside (1987): The utility driven dynamic error propagation network. Cambridge University Engineering Department.

36 S. Horchieter (1991): Undergoing dynamics. Technical University Munich.

37 F.A. Gers, J. Schmidhuber (2001): Long Short Term Memory learns context free languages and context sensitive languages. IEEE Transactions on Neural networks.

38 M. Mackey, L. Glass (1977): Oscillations and chaos in a physiological control system. Science.

39 J.C Principe, J.M. Kuo (1995): Dynamic modeling of chaotic time series with neural networks. Advances in Neural Information Processing Systems. Vol 7. Pg. 311-318. The MIT Press.

40 Rathie, J.C. Principe, J.M. Kuo (1992): Prediction of chaotic time series with neural networks and the issue of dynamic modelling. Int J. Bifurcation and Chaos. 2(4). Pg. 989-996.

41 R. Baker, J.C. Schouten, C. Giles, F. Takens, C.M. Bleek (2000): Learning chaotic attractors by neural networks. Neural Computations.

42 J. Schmidhuber (1992a): Learning complex, extended sequences using the principle

of history compression. Neural Computation. Pg. 234-242.

43 D. Zipser, R.J. Williams (1992): Gradient based learning algorithms for recurrent networks and their computation complexity. In Y. Chauvin & D.E. Rumelhart, Back-propagation: theory, architectures and applications.

44 F.S. Tsung, G.W. Cottrell (1989): A sequential adder using recurrent networks. In proceedings of the first international joint conference on neural networks, Washington DC, San Diego: IEEE TAB neural network committee.

45 Y. Kalinke, H. Lehman (1998): Computation in recurrent neural networks. From counters to iterated fuction systems. Advanced topics in artificial intelligence. Proceedings of the 11th Australian joint conference on artificial intelligence. Berlin Heidelberg. Springer.