**KATHFORD INTERNATIONAL COLLEGE OF ENGINEERING AND MANAGEMENT**

Balkumari, Lalitpur

****

A

Major Project Proposal

on

**“Duplicate Question Detection (DQD) in Quora using Machine Learning”**

[Subject Code: CT755]

**Project Members**

Mamata Shrestha (20/BCT/2072)

Mohit Dhungana (21/BCT/2072)

Subin Panta (39/BCT/2072)

Ujjwal Pudasaini (46/BCT/2072)

**DEPARTMENT OF COMPUTER AND ELECTRONICS & COMMUNICATION ENGINEERING**

**LALITPUR, NEPAL**

**JAN, 2019**

## ABSTRACT

Duplicate Question Detection (DQD) is a major problem encountered by Q&A forums like Quora, Stack-overflow, Reddit, etc. Answers to same questions get fragmented into different versions of the same question. These duplicate questions can be detected and using Natural Language Processing (NLP) and Machine Learning (ML). Dataset of more than 400,000 questions pairs was provided by Quora which will be used for feature extraction, these features will be fitted into then designed Artificial Neural Network (ANN). This project aims to automatically detect whether a new question entered by a user to a forum has been asked before in that forum and to help mark and remove it as a duplicate question.

**Keywords:** Duplicate Question Detection, Natural Language Processing, Machine Learning, Artificial Neural Network.

TABLE OF CONTENT

[ABSTRACT ii](#_Toc534964582)

[LIST OF ABBREVIATIONS v](#_Toc534964583)

[1 CHAPTER ONE: INTRODUCTION 1](#_Toc534964584)

[1.1 BACKGROUND 1](#_Toc534964585)

[1.2 PROBLEM STATEMENTS 1](#_Toc534964586)

[1.3 OBJECTIVE 2](#_Toc534964587)

[2 CHAPTER TWO: LITERATURE REVIEW 3](#_Toc534964588)

[2.1 Historical Background 3](#_Toc534964589)

[2.2 Review of related literature 5](#_Toc534964590)

[3 CHAPTER THREE: FEASIBILITY STUDY 7](#_Toc534964591)

[3.1 FINANCIAL FEASIBILITY 7](#_Toc534964592)

[3.2 TECHNICAL FEASIBILITY 7](#_Toc534964593)

[3.3 SCHEDULE FEASIBILITY 7](#_Toc534964594)

[3.4 OPERATIONAL FEASIBILITY 7](#_Toc534964595)

[3.5 RESOURCE FEASIBILITY 7](#_Toc534964596)

[4 CHAPTER FOUR: PROJECT METHODOLOGY 8](#_Toc534964597)

[4.1 BLOCK DIAGRAM OF PROPOSED SYSYTEM 8](#_Toc534964598)

[4.2 DEVELOPMENT MODEL 9](#_Toc534964599)

[5 CHAPTER FIVE: IMPLEMENTATION PLAN 10](#_Toc534964600)

[5.1 SCHEDULE 10](#_Toc534964601)

[5.2 HARDWARE AND SOFTWARE REQUIREMENTS 10](#_Toc534964602)

[6 CHAPTER SIX: EXPECTED OUTCOME 11](#_Toc534964603)

[7 References 12](#_Toc534964604)

**List of Figures**

[Figure 1 Block Diagram 8](#_Toc534923274)

[Figure 2 Prototype model 9](#_Toc534923275)

[Figure 3 Gantt Chart 10](#_Toc534923276)

# **LIST OF ABBREVIATIONS**

NLP: Natural Language Processing

ML: Machine Learning

DQD: Duplicate Question Detection

ANN: Artificial Neural Network

DL: Deep Learning

AI: Artificial Intelligence

GPU: Graphics Processing Unit

# CHAPTER ONE: INTRODUCTION

## BACKGROUND

Duplicate question detection (DQD) is a natural language processing (NLP) task that has recently become the focus of active research, where two interrogative segments are considered to be semantically equivalent, and thus duplicate, if they can receive the same answer. Much of the motivation for this research topic is coming from the usefulness of resorting to DQD to support online question answering community forums, and also conversational interfaces, in general.

Over 100 million people visit Quora every month and more than 14 million questions have been asked so far. Therefore, there is a high chance that many people ask similar questions that may be in different forms. This is a severe issue and hence Quora published its dataset for the first time in Feb 2017. These dataset consists of 404,290 question pairs along with the is\_duplicate parameter. Collection and visualization of the dataset is done before further processing of the dataset. Preprocessing of dataset consists of tokenization, stemming and removal of stop words.

when applied to the online question answering community forums, DQD can be used to automatically detect whether a new question entered by a user to a given forum has been asked before in that forum, and to help mark and eventually remove it as a duplicate question, thus mitigating the proliferation of duplicate questions that is a major issue hindering the usability of such forums.

And when embedded in the conversational interfaces, DQD can be used to compare a newly entered question against a database of previous question-answer pairs and if a similar question is found, to reply by delivering the corresponding stored answer, thus possibly avoiding to resort to a human operator.

## PROBLEM STATEMENTS

Nowadays people use question answering forums on a large scale. With the increased use of such forums there is a high chance that many people may ask similar questions that may be in different forms. The questions we ask may have already been answered or asked before. These similar questions but having different forms are treated as separate due to which same question needs to be answered repeatedly. This degrades the performance of such question answering forum. Our system aims to detect such duplications and helps to flag them. Doing so will make it easier to find high quality answers to questions resulting in an improved experience for Quora users.

## OBJECTIVE

* To detect whether a new question entered by the user in a question answering forum has been asked before and to help mark and remove it as a duplicate question.

# CHAPTER TWO: LITERATURE REVIEW

## Historical Background

The DQD (Duplication Question Detection) makes a high use of deep learning to detect questions which are semantically similar. Deep Learning, is a branch of Machine Learning, employs algorithms to process data and imitate the thinking process, or to develop abstractions. The history of Deep Learning can be traced back to 1943, when Walter Pitts and Warren McCulloch created a computer model based on the neural networks of the human brain. They used a combination of algorithms and mathematics they called “threshold logic” to mimic the thought process. In 1960 Henry J. Kelley developed the basics of a continuous Back Propagation Model. In 1962, a simpler version based only on the chain rule was developed by Stuart Dreyfus. The earliest efforts in developing Deep Learning algorithms came from Alexey Grigoryevich Ivakhnenko and Valentin Grigorʹevich in 1965. They used models with polynomial activation functions, that were then analyzed statistically. From each layer, the best statistically chosen features were then forwarded on to the next layer. In the 1970s the first AI winter kicked in when the result of promises couldn’t be kept. The impact of this created lack of funding limited both DL and AI research. [1]

The first convolutional neural networks were used by Kunihiko Fukushima. Fukushima designed neural networks with multiple pooling and convolutional layers. In 1979, he developed an artificial neural network, called Neocognitron, which used a hierarchical, multilayered design. This design allowed the computer the “learn” to recognize visual patterns. [2]

In the 1970 back propagation, the use of errors in training Deep Learning models, evolved significantly. This was when Seppo Linnainmaa wrote his master’s thesis, including a FORTRAN code for back propagation. This was when Rumelhart, Williams, and Hinton demonstrated back propagation in a neural network could provide “interesting” distribution representations. In 1989, Yann LeCun provided the first practical demonstration of back propagation at Bell Labs. He combined convolutional neural networks with back propagation onto read “handwritten” digits. This system was eventually used to read the numbers of handwritten checks. This time is also when the second AI winter (1985-90s) kicked in, which also effected research for neural networks and Deep Learning. Various overly-optimistic individuals had exaggerated the “immediate” potential of Artificial Intelligence, breaking expectations and angering investors. The anger was so intense, the phrase Artificial Intelligence reached pseudoscience status. Fortunately, some people continued to work on AI and DL, and some significant advances were made [2].

In 1995, Dana Cortes and Vladimir Vapnik developed the support vector machine (a system for mapping and recognizing similar data). LSTM (long short-term memory) for recurrent neural networks was developed in 1997, by Sepp Hochreiter and Juergen Schmidhuber. [3]

The next significant evolutionary step for Deep Learning took place in 1999, when computers started becoming faster at processing data and GPU (graphics processing units) were developed. Faster processing, with GPUs processing pictures, increased computational speeds by 1000 times over a 10 year span. During this time, neural networks began to compete with support vector machines. While a neural network could be slow compared to a support vector machine, neural networks offered better results using the same data. Neural networks also have the advantage of continuing to improve as more training data is added. [3]

Around the year 2000, The Vanishing Gradient Problem appeared. It was discovered “features” (lessons) formed in lower layers were not being learned by the upper layers, because no learning signal reached these layers. This was not a fundamental problem for all neural networks, just the ones with gradient-based learning methods. The source of the problem turned out to be certain activation functions. A number of activation functions condensed their input, in turn reducing the output range in a somewhat chaotic fashion. This produced large areas of input mapped over an extremely small range. In these areas of input, a large change will be reduced to a small change in the output, resulting in a vanishing gradient. Two solutions used to solve this problem were layer-by-layer pre-training and the development of long short-term memory. [4]

By 2011, the speed of GPUs had increased significantly, making it possible to train convolutional neural networks “without” the layer-by-layer pre-training. In 2012, Google Brain released the results of an unusual project known as [The Cat Experiment](http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html). The free-spirited project explored the difficulties of “unsupervised learning.” Deep Learning uses “supervised learning,” meaning the convolutional neural net is trained using labeled data (think images from ImageNet). Using unsupervised learning, a convolutional neural net is given unlabeled data, and is then asked to seek out recurring patterns. The Cat Experiment works about 70% better than its forerunners in processing unlabeled images. However, it recognized less than a 16% of the objects used for training, and did even worse with objects that were rotated or moved. [4]

Currently, the processing of Big Data and the evolution of Artificial Intelligence are both dependent on Deep Learning. Deep Learning is still evolving and in need of creative ideas [4]

## Review of related literature

Y. Wu, Q. Zhang, and X. Huang in their paper titled “Efﬁcient near duplicate detection for Q&A forum” approached DQD with a Jaccard coefﬁcient to measure similarities between two interrogative segments in a pair. A DQD dataset was created resorting to the Baidu Zhidao, a question and answer forum in Chinese provided by the Baidu search engine. Training with 3M pairs and testing on 3k pairs, the system scored an f-score of 60.29. [5]

D. Bogdanova, C. N. dos Santos, L. Barbosa, and B. Zadrozny introduced the architecture of convolutional neural network to tackle DQD in their paper “Detecting semantically equivalent questions in online user forums”. This network obtains the vectorial representation of the words in the two input segments, and the next convolutional layer constructs a vectorial representation for each one of the two segments. Finally, the two representations are compared using cosine similarity. This system was reported to score over 92% accuracy, resorting to 30k data taken from the Meta forum in StackExchange and AskUbuntu forum, with an 80%/20% train/test split. [6]

A deep neural network approach to DQD in N. Afzal, Y. Wang, and H. Liu system titled “MayoNLP at SemEval-2016 task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model” obtained the best accuracy in the SemEval-2016 Task 1 “Question Question” subtask, namely 0.73035 in terms of Pearson correlation coefﬁcient, which had the objective of determining the degree of similarity, on a 0–5 scale, between two interrogative sentences. [7]

In what concerns the Quora dataset, released in 2017 January, a paper concerning DQD was released by Z. Wang, W. Hamza, and R. Florian titled “Bilateral multiperspective matching for natural language sentences” .It proposes a multi-perspective matching (BiMPM) model. The resulting DQD resolver is reported to reach an accuracy of 88.17% when evaluated upon a 96%/2%/2% train/dev/test split. [8]

Chakaveh Saedi, Jo˜ao Rodrigues, Jo˜ao Silva, Ant´onio Branco, Vladislav Maraev released a paper in 2017 concerning DQD titled “Learning Proﬁles in Duplicate Question Detection” where they use Jaccard Index over all sets of n-grams of each question, with n ranging from 1 to 4 .Two questions are considered similar if their Jaccard index is above a certain value. For each dataset size, this similarity threshold is determined by picking the value that achieves the best accuracy on the training data. To ﬁnd the best threshold, the training set is used in a series of trials whose outcome is applied to the test set. [9] The result of the data is shown in the table below:

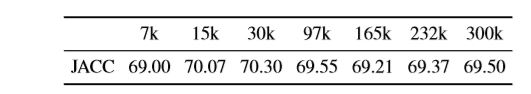


Table 1:Accuracy of DQD resolver based on the jaccard index

# CHAPTER THREE: FEASIBILITY STUDY

This project is highly feasible to the real world and can be realistically accomplished. The different types of feasibility studies that were analyzed before moving to the project are discussed below:

## FINANCIAL FEASIBILITY

This system is software-based system so it is financially feasible. It is cheaper in comparision to other hardware related systems.

## TECHNICAL FEASIBILITY

The technical resources are not difficult to acquire so this application is also technically feasible.

## SCHEDULE FEASIBILITY

The minimum time for the completion of this project is about six to seven month whereas testing and optimization can take a month. Thus this system can be developed in the specified time.

## OPERATIONAL FEASIBILITY

This system is simple and easily accessible. This does not require any new hardware or software system to operate.

## RESOURCE FEASIBILITY

Since all the resources in this project are easily available, this project is feasible in terms of resources**.**

# CHAPTER FOUR: PROJECT METHODOLOGY

## BLOCK DIAGRAM OF PROPOSED SYSYTEM

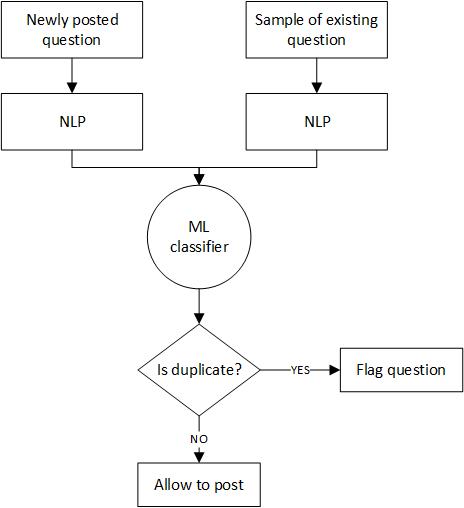


Figure 1 Block Diagram

## DEVELOPMENT MODEL

For this project, we have used the Prototype Model of Software Process Model.

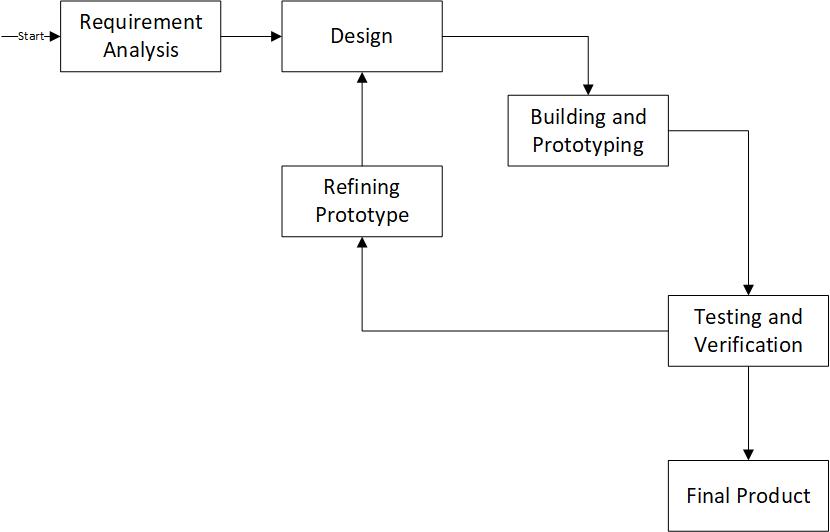


Figure 2 Prototype model

# CHAPTER FIVE: IMPLEMENTATION PLAN

## SCHEDULE

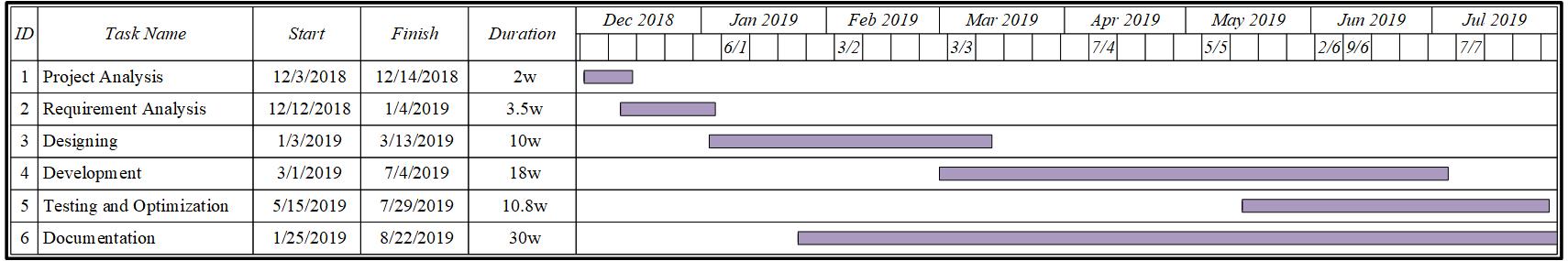


Figure 3 Gantt Chart

## HARDWARE AND SOFTWARE REQUIREMENTS

**Software:**

1. Python

# CHAPTER SIX: EXPECTED OUTCOME

This project will develop software to detect duplicate questions which will make it easier to find the answers to questions resulting in an improved experience for Quora users.

# REFERENCES

|  |  |
| --- | --- |
| [1] | D. W. Kramer, How AI came to be, Washington: Science Revelation, 2000. |
| [2] | F. Chollet, Deep Learning with R, 2003. |
| [3] | P. Norvig, Artificial Intelligence: A Modern Approach (3rd Edition), 2012. |
| [4] | Egon Börger,Dines Bjørner, Machine Learning, Mind Warehouse, 1997. |
| [5] | Y. Wu, Q. Zhang and X. Huang, "Efﬁcient nearduplicate detection for Q&A forum," *Proceedings of the 5th International Joint Conference on Natural Language Processing,* pp. 1001-1009, 2011. |
| [6] | D. Bogdanova, C. N. dos Santos, L. Barbosa, and B. Zadrozny, "Detecting semantically equivalent questions in online user forums," *roceedings of the 19th Conference on Computational Natural Language Learning,* pp. 123-131, 2015. |
| [7] | N. Afzal, Y. Wang, and H. Liu, "MayoNLP at SemEval-2016 task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model," *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016),* pp. 1258-1263, 2016. |
| [8] | Z. Wang, W. Hamza and R. Florian, "Bilateral multiperspective matching for natural language sentences," 2017. |
| [9] | Chakaveh Saedi, Jo˜ao Rodrigues, Jo˜ao Silva, Ant´onio Branco, Vladislav Maraev, "Learning Proﬁles in Duplicate Question Detection," 2017. |
|  |  |
|  |  |