Community Question Answering using Deep Learning

A DISSERTATION PRESENTED BY

VASUNDHRA DAHIYA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN THE SUBJECT OF

STATISTICAL COMPUTING IN **DATA SCIENCE**

TO

SCHOOL OF COMPUTER AND SYSTEM SCIENCES

JAWAHARLAL NEHRU UNIVERSITY

DELHI-110067, INDIA

JUNE 2021

Declaration

Intentionally left blank.

Certificate

Intentionally left blank.

Abstract

This dissertation aims to handle the task of Community Question Answering (CQA) through learning contextual representation of data by implementing deep neural models. Primarily, sequence modeling is experimented for effective deduction or selection or classification of "relevant" responses for a posed question. In CQA, open-domain user interaction creates abundance of dynamic data that is vital to the forum. Such websites require efficient, fast and accurate knowledge which is achieved by finding the most relevant answers for them and so forth, exhibiting them by their order of relevance to the users. The work done in this dissertation achieves the same using Natural Language Processing and Deep Learning techniques. As sequential networks have been helpful for understanding both the textual features and hidden context of the text, they are applied to acquire meaning from the forum's data.

Keywords: Community Question Answering (CQA), Open-domain Question Answering (QA), Natural Language Processing (NLP), Sequential networks, Long Short-term Memory, Word Embeddings, Answer Selection

Contents

Listing of Figures

[1] *A, B, C Subtasks; Letters for Model Name, 22*

Listing of Tables

Dedicated to

first generation female scholars who were told

ample times to give up on Higher studies.

Acknowledgements

….

"Future shock is a sickness which comes from too much change un too short a time.
It's the feeling that nothing is permanent anymore"
– Orson Welles

….

As the world went into a halt altogether this year and a new 'normal' became all our realities, I am prompted to take an unusual path to extend my acknowledgments. Words cannot contain my admiration and gratitude for people that have helped, supported and inspired me during the tenure of this dissertation. It has been a wild roller coaster ride and let me tell you, I am scared of those. The tenure was scary too nonetheless. I think we all saw humanity rise above huge despair during this pandemic to come together and help each other.

Perhaps, this is one of those pages I have wanted to write more than the *Conclusions* itself. Foremost, I would like to thank my supervisor Dr. Aditi Sharan who is truly an insightful supervisor and undoubtedly, a remarkable woman. I admired her patience, her perseverance and her cooperativeness throughout my time working with her. I can't thank her enough for being cool enough to tolerate and give her full attention unto my tedious experiments and read my lengthy drafts and explanation as we figure out a way to sail through. I believe had she not given me the time to perform and let my mind through spectrums as it did now, my work would not have been half as satisfactory. To the very last day, she stays to be a source of inspiration. It truly has been my pleasure to work with her.

Next, I extend my gratitude to a colleague of mine, Ashish Kumar *(Ashish Sir! my labmate. I guess he will be Dr. Ashish now).* He has entertained my doubts and neural models when I had no idea what I was working with and encouraged me to explore. And then explore some more. Whether it was my work or reading and discussing articles of NLP with me,

he never shied away from helping me, or anybody for that matter- literally, not even for one single error. His resilience to keep working on himself to build more skills and then, helping others do the same left me so much zeal. If somebody asks me what my favourite quality about him is that he will never make you feel stupid, no matter how wrong you are – and that's rare! (I sincerely hope his future to-be students appreciate it too).
I would like to another colleague, Ms. Sheeba Naz whose willingness introduced me to this project's domain with me and would try coding and studying alternatives with me whenever new ideas came up.

Now, I would like to address this unwavering support system with unique people with a beautiful array of qualities I have the honour of being friends with. Firstly, I would like to thank Abhibhav and Manish, two wonderful and extremely smart human beings whose camaraderie inspired me to reflect onto new dimensions of thinking as we enjoyed walks in the beauty that is our campus - whether it was discussions over social causes (we're in JNU, it is bound to come up) or for mathematical-and-philosophical.

My younger sister Lavanya has been of utmost support through my time in Master's and the extraordinary path she is on to becoming an amazing woman leaves me with specks of love and encouragement I deeply cherish. My family, a father who regularly steps up for me when I am undermined for choosing to study, and my mother who takes care of every little need of mine, and fought for me then so that today I have a voice.

Next, I would like to thank my *one-phone-call-away* people: Devika, Shalu, Urvashi and Vandita. These brilliant women have helped me imbibe all that is good and brave in this world. They have been my source of hope when I had none & confidence when I had too little. I would be a fool to not celebrate them every chance I get.

Finding the inspiration to not just be better but stay that way, is a hard thing to do. Writing this now as I am, is the least I could have done to share my gratitude towards them.

Outline

C*hapter 1* covers a quick know-how of the growth, uses and applications of the field NLP of which Question Answering (and so forth, Community QA (CQA)) is an important task. Following this is a briefing of problem description, contributions and how deep learning has brought changes in understanding said data. The chapter ends with a brief introduction to the problem description of the task of CQA. In *chapter 2*, the background of Community Question Answering domain is discussed along with a literature survey of the work done for the task of CQA so far. *Chapter 3* involves explanation and brief history of models and approaches that have been used for comprehending textual data. This chapter forms the basis of the approaches followed in this dissertation and are requisites to understand the experiments of the data discussed in *Chapter 4*. Here, proposed architecture and approach to handle the task has been discussed. Its performance and evaluation as opposed to several experimental settings have all been highlighted in the *Chapter 5*. The limitations and challenges with respect to work done on sequential networks has also been discussed in it. The dissertation ends with a brief discussion on the challenges and future scope of CQA.

1

Introduction

The abundance of data along with inexplicable dynamicity around the internet, the data requires to be explored, understood, and studied closely now. Be it in any format, speech, text, or visuals- the sole purpose is to incur human-like insights to develop an *intelligent* and *aware* system. Amongst these formats that build the language in the vastly expanding internet community, the most prominent interaction medium is text. Over the years, several approaches have been used to instigate meaning from text for professional purposes like industrial applications or academic publications, and personalized platforms like user forums and social media.

Intuitively, with **language**, the foremost question that arises is: how do you even make a machine understand language? How do machines understand a text? The fact is that computers understand binary information, so, how will they *perceive* text? Additionally, when studying textual data, there are language barriers such as varying writing styles, grammar or ambiguity conflicts, hybrid data format like use of both images and text together to convey something, or usage of slang or emoticons, etc. Even if structured, how can a machine understand the meaning behind such data uniformly? Such are the questions that the field Natural Language Processing (NLP) is dedicated to. NLP is an intersectional discipline formed by Artificial Intelligence, Computer Science, and Computational Linguistics. The fundamental idea behind NLP is to make machines understand text by training it with the rules of the language and processing the text. It

gives the machine the ability to read and more importantly understand - to derive meaning from human language. To achieve this, NLP deals with building computational algorithms and models. Its instrumental tasks include text classification, semantic analysis, named entity recognition, text summarization of documents, dialog generation, natural language inference and generation. Some of the widely used applications for these tasks are language translators, chatbots, plagiarism checker, spell checker, keyword search and auto-complete systems, targeted advertising, customer service automation, review analysis, survey analysis, sentiment analysis etcetera, and most recently under unfortunate social dynamics, even hate-speech detection.

This dissertation involves using deep learning and NLP to tackle the task *Question Answering (QA)*. A common practical example of a QA system is 'Siri', the intelligent conversational personal assistant, founded in 2010 by Apple. The long-standing application 'Question Answering' in NLP is an interesting mixture combining NLP and Information Retrieval. QA systems are deployed for automated answering or mapping retrieval of information from existing knowledge bases. It focuses on building techniques or systems that can automatically provide answers to a given question posed by users. Essentially, a Question Answering system deduces some internal (semantical & syntactical) representation for the questions and uses certain approaches to "understand" them and generate valid response(s) for the question asked by the user. Specifically, narrowing down the domain for QA applications, this dissertation explores Community Question Answering or CQA which is precisely discussed in the next sub-sections.

1.1 Problem Domain: Community Question Answering

CQA deals with generating relevant responses for community forums like Stack Overflow, Reddit, Quora like Lifestyle forums. Forums usually have *Questions* or *Queries* put on frequently by the users. These queries are subsequently answered by other community or users using the same web platform, thereby creating *Answers* or *Comments*. These responses are recorded and saved in a database that forms the knowledge base of the forum. The goal of CQA is essentially to harness this knowledge base, optimally and intelligently to devise an approach that automatically incurs valid responses to any new user [15]. As the user community expands on such forums, the automatic answering or retrieval of answers has become essential.

1.2 Problem Statement: Selection of Relevant Responses

In essence, CQA can be defined as *"given (i) any question and (ii) a large collection of question-comment threads created by a user community identify the useful comment for answering the question"*. Mathematically, this implies that for a Question Q with Answer set A_i (a_0 to a_n), the best of *n* answers has to be shown to the user i.e., we are looking for $\boldsymbol{a_s}$ where $1 <= s <= n$ or a set of candidates a_s. Inherently, this means for any *(Query, Response)* pair, we have to classify whether the response is relevant/useful or irrelevant/useless to the query.

CQA occurs in three different phases: the answers must be (i) retrieved and classified, and (iii) ranked – in essence, this implies that answers are displayed to the user by their order of relevance. The present challenge is to automate just this process. In SemEval 2016 [17], a similar CQA challenge was proposed and the dataset of *Qatar Living Forum* was released for the task. This dissertation chiefly focuses on the **former** phase i.e., identifying quality (or relevant) answers from the *Qatar Living Forum* data- formally known as, classification/selection of answers. We also perform experiments to examine and improve this classification of quality responses.

1.3 Motivation

Even though a posed question is new with respect to the collection, it is expected to be related to one or several questions. So, an effective CQA system *inspects the questions and its answers* enabled with its features, language or data-dependent to find the best answers. Features like the nature of question posed, its topic, its title, its description, the website traffic, popularity can be used for selection of quality answers. These features have been described in detail in Chapter 2, and some of these have also been deployed in our work for finding quality of answers. With representation learning in NLP, the capability to exploit language features and similarity features between the Question's text is also at our disposal. This clearly points towards a holistic way to traverse through knowledge base to provide answers. To experiment on this theory, SemEval proposed a challenge to rank answer in 2016 [16] which encouraged to work with better word representations, a non-IR based approach and use NLP to handle answer ranking.

1.4 Purpose of Deployment of Deep learning

Under Information Science, the data mostly used to be studied quantitatively or statistically. Both statistical and probabilistic approaches have been used to capture either syntactic or semantical or distributional information to be used as linguistic features. But to *qualitatively* understand text data, not only better language features are required but also effective modeling for better representation learning is required. From Predictive analytics for recommender systems, Sentiment analysis for brand monitoring, Time-series analysis for stock prediction and Network analysis for text (lexical or semantic) analysis- over the years, neural modeling has changed the fundamental ways in which the industry functions. As with Machine learning, the features are hand-picked, language models are often not as robust but Deep learning (DL) unlocks the possibility of 'thinking' or 'neural' machines and has imprinted its effect quite vividly. It is a major breakthrough in AI as it surpasses the leap from intelligent to truly self-aware i.e., self-learning systems. Because of deep learning, the NLP tasks like language modeling, natural language inference, speech recognition, machine translation, conversational agents and question answering have improved drastically over the course. The shift in NLP from statistical methods to neural methods with growing research in ML and DL on ways to understand text, and consequently *language*. With the ability to not only understand but also generate human-like responses, deep neural models have truly led to meaningful representation in AI.

In Information retrieval (IR), the focus lies on *retrieving* all documents of the queries, whereas for tasks like Question Answering, the focus lies on *extracting* the most relevant documents. For IR-based QA or Knowledge-Based QA, modern-day NLP requires an *aware* system that find answers effectively by understanding the *context* (i.e., meaning) rather than just matching best-suited keywords as done in traditional Information Retrieval methods. Such a system is also insightful for predictions of intuition (i.e., reason) behind data. This is why, to use deep learning models, namely, sequential networks will be deployed in this project to understand how they perform with this task. Sequential networks have been known to handle sequence data like time series and sentences too as they also a sequence, so we use to understand the meaning of our data.

1.5 Contributions

From the user perspective, automatic systems seldom work perfectly. It does not follow the *one-size fits all* dynamics. Nonetheless, the constant notion that most conversations in such forums is user-to-user interaction based and information flow, if not related is at least, a continuous thread-like structure, it deemed befitting to find their meaning and relevance via **contextualized** embeddings.

User-generated textual data plays a great role in deciding relevancy of responses on CQA forums. The information is often intertwined onto specific domains, subjects or categories. The dataset used in our forum data, for example, has a Question Subject and a Question Category that varies largely but also clusters into a certain topic being talked about on the forum. To be able to understand and predict the **quality** or relevance of the answers provided on forums, we make the assumption that data is inter-related, thereby, subject to contextual representation. Since, question and answer data are substantially available, their related columns are used for the three subtasks in which experiments have been carried out.

For the stated large-scale and long-length sequenced context capturing, sequential deep neural models like RNNs, specifically, LSTMs (Bi-Directional) are deployed in our work. Studying contexts for data as noisy and diverse as CQA forums requires niche observational settings, hence, various experiments using *Text* (Plain Text & Subject including; GloVe Word Embedding & Varied sentence length), *Data* (Upsampled for balanced classification), *Model* (Hyperparameters, Simple vs Stacked, Attention-based) features are used for training the data. To club data for convenience as ranked responses, data is sorted according the Cosine similarity scores of their contextualised word representation. The tasks have been carried in three divisions: A, B, C [16][17] which individually aim to learn context and classify each of *(Related Question, Answer)*, *(Question, Related Question)* and *(Question, Answer)* pair respectively into one of *0* (bad), *1* (useful), *2* (good) classes correspondingly. The final results for this multinomial classification are shown what classification has been done and how well they have been can be used for further analysis, if need be.

1.6 Summary

In chapter 1, a brief introduction of the task Community QA was provided. Along with that, how NLP, ML and DL is used understanding natural language is described. The domain, statement and the motivation behind the task-at-hand that has been carried out in this dissertation are explained. We end with a concise description of the contributions made in this thesis. In the next few chapters, all these are discussed further.

2

Background

In Chapter 2, the background of CQA is discussed. We explain how CQA forums usually work- in terms of the data it consists, the responses that are retrieved, and the process it follows. Chapter 2 ends with a literature survey that has been carried out in QA and CQA over the years.

2.1 Categorization of Data

Community Question Answering is a service for sharing and providing information that users seek on the community websites and platforms. User CQA platforms usually have a variety of data. Generally, in CQA, there are two categories questions and answers can be put into.

(a) Depending on nature of queries- Factoid and Non-factoid [15][26]: *Factoid* questions are questions based on factual knowledge, for example, some user might pose a question to find out the birth date of a celebrity, or like the capital city of a country. These questions have a definitive answer. *Non-factoid* ones are the questions that are usually looking for suggestions/opinions etc, for example, what did the someone think of a food store nearby, or a television show, a product, etc.

(b) Depending on nature of forum- Open and Closed domain [15][26]: *Open* domain systems contain questions just about anything (like Quora) whilst the *closed* domain

systems focus on a limited/specific domain, field or topic (like Stack Overflow, WebMD etc).

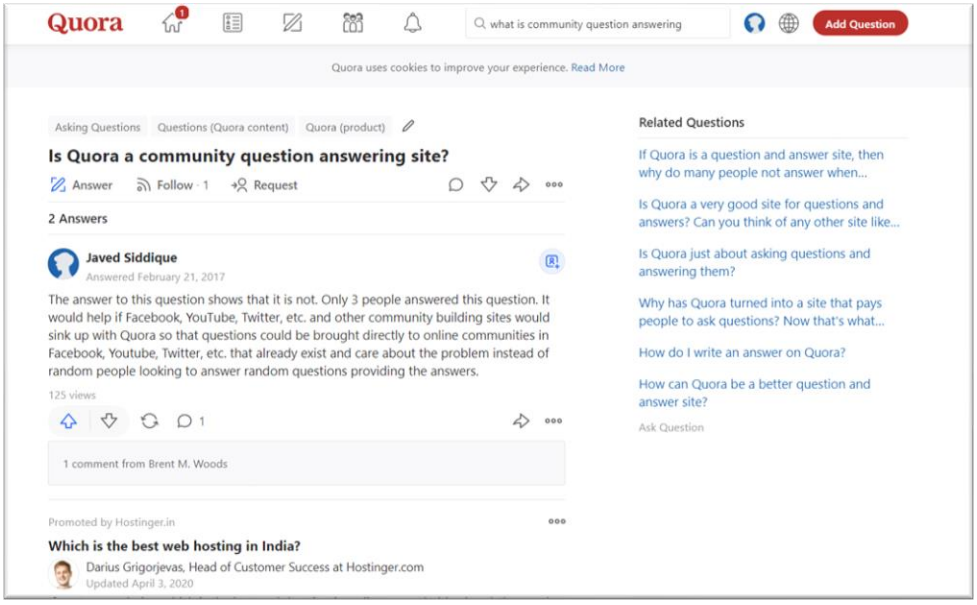


Fig 1: Example of a Community Question-Answering website.

2.2 Lifecycle of CQA

In general terms, CQA lifecycle is as follows: [15]

- Question Creation
- **Question Answering**
- Question Closing
- Question Search

Once the question is asked, the answer is accepted and the query is closed by the user; consequently, the CQA task begins. This involves traversing through the knowledge database to look for high-quality responses. These responses either can be a specific relevant answer or a list of relevant questions and corresponding answers. [15]

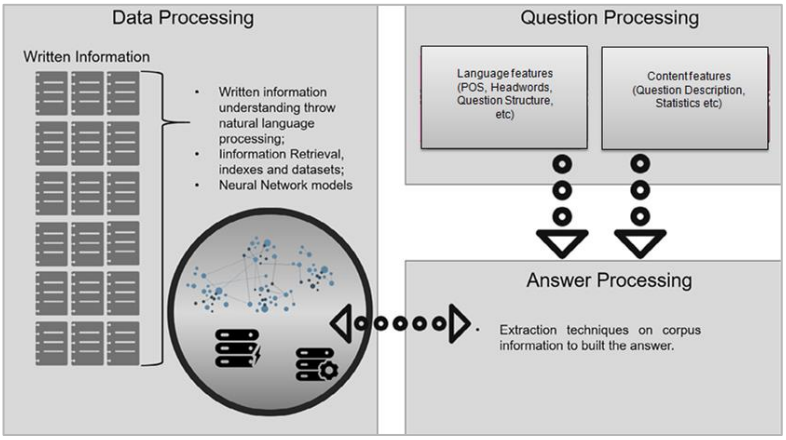


Fig 2: Lifecycle of CQA [15]

2.3 Deciding Relevancy of responses

It is difficult to just *find* the best or similar or the relevant answers let alone align them in a type and order that a user might like. Depending on the nature of data, the modus operandi to deal with data shifts too. The usage or goal of said data also decides what corresponds to 'relevancy' i.e., how the relevancy will be defined as and so forth, be calculated to find the best of them for the user.

Depending on the nature of the *content* on websites, the types of question-answers on stack overflow would be different from answers on tourist or say, daily lifestyle forums. To find the best answer amongst it, the intersectional nature of data would depend not only on correctness but might also take in account the modeling of its popularity, e.g., number of likes/ upvotes on such forums. Such data also has dependent attributes such as its topic, its title; its description, etc that can be used to look for responses [15]. Furthermore, the best answers can come from different question-comment threads, like similar questions and their responses especially if they answer the original question as well. The question's features themselves can be inspected. The popularity features can be taken in account too [17].

Additionally, data as per nature of the *forum*, open-domain question answering has been seen rather easier to deal with than closed-domain question answering. The observable reason behind this is the availability of larger knowledge base. The results may vary with

the type of *queries*. Information can be either factual, opinions, or even stories. What is rich about the factoid and non-factoid data is not only the challenge to find better responses (keeping in mind, its uniqueness) but also its linguistic challenges. For example: data especially in natural language, contains slangs, wrong informal sentence structures, usage of URL links or images in between text etc. In non-factoid particularly, the ground truth is unknown or unverified, thereby making user's expertise and efficient text analysis of its answers, the best way to find a good answer. Undoubtedly, all factors together contribute towards well enough classification or ranking of relevant responses. Usually, CQA is open domain and non-factoid natured which makes it computationally as well as linguistically challenging.

It is imperative to take note that fundamentally, CQA is still a standing NLP problem. Indeed, as any other NLP task, CQA also suffers from both model-driven and data-based limitations. Herewith, former means includes the lack of good representation learning for words or modeling the words; whereas the latter usually means the unavoidable inconsistency and noisy incomprehensible data. The inconsistency of data in a forum data comprises cases like the absence of systematic classification of problems, unbalanced or missing data values etc. This is because majority of the users look for immediate answers on the forums, thereby causing a lack of clarity with attributes like the category or domain of question being asked. Forums also use votes-based approach to classify and recommend responses to the user- in lack of data or meaningful representation for it, the quality of responses is bound to deteriorate. Traditional IR methods where the answers are computed based on the frequency or similarity of words occurring in the query and the knowledge base are still widely used. However, as mentioned earlier, rather than just quantitatively linking or connecting words, a smarter approach is to find features and the meaning behind text, for machines to self-learn and generate better human-like responses. This shift in representation learning of words has led to dealing with former problems like lexical and semantical gap and ambiguity which are long-standing language modeling subjects.

CQA is to provide a collaborative learning platform for efficient knowledge sharing among the users. In real-time, it is noticeable that retrieval of best responses to asked questions too, is time sensitive. The goals of CQA thus can vary according to the need of the platform- with time sensitive and quick services, prioritizing user need. Otherwise,

most platforms build a content-driven platform, considering answer 'text' quality a priority which is decided through context or through semantics.

2.4 Literature survey for CQA

[15] provided a comprehensive survey of 265 articles published between 2005 and 2014 to narrow down all the approaches and classifying these approaches used for CQA, until then. Pointing out, the best features were the ones that were mutual for all CQA approaches were properties like question quality, answer quality, user expertise, user modeling, etc. It shows classifying the type of problem as prediction, ranking or analytical problem solving with which kind of algorithm (Machine learning or a traditional matching method) can be helpful. Other more data-dependent features are textual (length, structure, style of writing/ readability), non-textual (community feedback, temporal features), thread features (relevance/ similarity, thread statistics), topic features (user assigned topic, language or topic model used). Other than this, user and content analysis achieved with studying the quality of the question, answers, topic and analysing the user's details answering the questions can be done for a better ranking. User details might range from his expertise level (global representation, topical expertise, etc) to his profiling (response's popularity in the community, etc). Primarily, the quality of question-and-answer ranking is dependent on how relevant the answer is to the question. Measuring this can include accuracy of the answers(s) for the question, responsiveness, comprehensiveness, completeness, objectivity, and originality of the answers. Apart from this, ranking techniques differ based on if the problem is considered as Classification (binary or not; implemented with Regression, SVM, etc) or a Ranking problem (like Learn-to-Rank). [15] certainly emphasized the importance of data-dependent features and general factors for CQA.

However, with Deep learning, initially, the work performed was model dependent only. [11] focused on non-factoid questions treating the problem as a binary classification problem. With no linguistic tools to inspect the data, BOW and IR-Weighted dependency as baseline models, implementation of CNN was done. The aim was to find better (Question, Answer) pair, the approach was rather a data-independent and more model-focused. [11] proved to provide a new direction for CQA with the help of deep learning. Even without feature engineering, a good 65.3% was improved over the baseline models.

As an improvement, on the same dataset i.e., InsuranceQA dataset, [12] used LSTM based (variant BiLSTM) deep learning models for Answer selection. Here, baselines were the same as [11] with two additional CNN architectures that they introduced. [12] significantly improved the embeddings by learning the questions and answers together. As a simple pooling layer may fail to preserve the local linguistic information, this proposed an additional layer of CNN on top of LSTM. Also, an attention model was introduced, to focus more on candidate answers embeddings according to the question context.

With the same deep learning model, but treating CQA as a ranking problem, [13] used Learn-to-Rank to learn a similarity metric for deep learning-based question-answer pairs. However, here the Bi-LSTM-CNN was only a sentence model. They proposed a hypernym strategy to get more general text pairs. The distributional sentence model maps query and answers to their vectors, which are then used to learn the semantic similarity matching between them.

[14] proposes a novel approach for CQA based on the assumption that if two questions have similar answers (ranked), the questions are semantically similar too, and vice versa. This deeply integrates the two tasks, (i) Question Retrieval and (ii) Answer Ranking, rather than treating them as independent or sequential (as traditionally done). While the former dealt with finding similar questions from CQA archives by estimating the degree of semantic similarity of Q-Q pair, the latter dealt with how well an answer responded to the question according to the degree of semantic relatedness of Q-A pair. The advantage of the first step is that it filters out the unrelated answers (with respect to the question). This way only top N answers are available to be ranked later on. The word features Word Matching feature, Translation based features, Topic model-based feature (LDA), Lexical semantic similarity feature (LSS). Additional features like question's length, position and answer's length, headwords, user id, special characters like question marks, emoticons, etc turned out to be very helpful. A comparison of ranking was done with Learn-to-rank versus Supervised methods (SVM, Logistic regression).

In [22], through statistics and neural networks, CQA was implemented for Ranking. For Question-to-Question similarity, features were based on data statistics i.e., the number of words, number of sentences, bag-of-words, word overlap, noun overlap and name-entities overlap. Other than this, question category features were used from the data and

a SVM classification model was applied to predict this question category. The result is used to get question to question information on being fed to a CNN with a MLP followed for binary classification, and the result is forwarded into a BiLSTM for Question to answer pairs.

[23] also used data features to find quality answer and rank them. On Stack exchange dataset, a new tab named 'promising answers' tab was introduced which encompassed answers which were useful. It also benefits the idea of minimizing Matthew effect. The criteria for usefulness were text, user details (answerer's reputation) and their usage (activeness of answers). Other than this, NLP linguistic & statistical (such as answer ratio, topic similarity score, q-a cosine and q-q average cosine) was used. For classification, Naïve bayes, Random forest and Gradient boosting was used whereas for regression, RMSE was used and only Class 2 (which was defined as good) labels.

Another interesting approach to ranking for CQA is used in [18] and [4], which used positional information to find high quality answers. In [18], ranking is performed in three steps. First, the detection then a CNN and GRU model, and then aggregation for analysis. First step detection of a relevant location, second builds up its context to detect relevance locally, and third aggregates for output query and global wise for analysis. In [4] however structural information is used. Probabilistic relational modeling uses Bayesian networks to reflect objects along with their properties and relations. The analogy made is that measure of similarity between structure of related objects (here, question-question) and relation between structure is found. With only positive data pairs of (q, a), there are two tasks namely (i) learning link predictions through Bayesian logistic regression model and learning gaussian prior weights, (ii) knowledge mining by studying cluster structure. First step produces link between objects (question and answer) and what part of subpopulation comes under this latent linkage is found. A good candidate in this linkage is found by gaussian priors.

3

Pre-requisite knowledge

To understand the task, answer selection and so forth, ranking through classification in Community Question Answering, few NLP concepts have to be understood. With text, all approaches that have been used till now play a major role in deciding why and how the current state-of-the-art methods help in understanding human language is perceived by the computer. To discuss, this chapter focuses on the representations used for text. The process is technically known as **Representation Learning**. [25]

Perceptibly, for machines, dealing with human language is harder. Unstructured text data must be dealt with cognitively. That means, not just the interpretation is to be focused but also the meaning behind it. Human language is very tricky because it goes beyond just words. The process of understanding and manipulating language is not always easy. Everything we say, verbally or in written form carries so much information. Our words, the topic, even our tone constitutes to some value in the data. It is subjective to the speakers, his intent. It could be categorical if it includes gestures. It is highly ambiguous in its linguistics, situational, contextual or visual-based. With language, the data is often noisy. For example, words like 'the', 'a', 'an' etc (called stop words) are removed from the text to be processed and this is called Stemming. Words like "cry" and "crying"; "go" and "went" mean the same but can increase the frequency. To handle this, NLP has *lemmatization* where the words are reduced to their base form. After processing, only *tokenized* unique words are left. Such words usually work as noise in the text data, adding

computational space time complexity, and not contribute to 'meaning' and 'insight'. This is why text data has to be pre-processed efficiently and strategically to make the best of it. [10][26]

Following are some representations used for words. Under following two subsections discusses how 'understanding' of text is done, and how is it modelled for the computer to understand and capture its features and context fully.

3.1 Early word representations

Starting out, words' meaning was analysed quantitatively. Being the atomic unit of language, the easiest way to represent words on a computer is One-hot encoding. It only involves creating a vocabulary filled with 0's and 1's i.e., 0 if a word is absent, and 1 if present. The main drawback of this representation is that it suffers from sparsity and no semantic relation or similarity between words is captured as it loses its positional relationships.

For example : He has a cat.

He	=	[1 0 0 0]
Has	=	[0 1 0 0]
A	=	[0 0 1 0]
Cat	=	[0 0 0 1]

Another early representation, n-gram modeling, follows the logic that each next word can be predicted by looking at the previous words i.e., n^{th} term to be predicted using *(n-1)* term. For n-grams, words were taken in pairs (unigram, bigram, trigram) to study statistical properties of text. Built on similar reason is distributional hypothesis. It says words following similar probability distribution have similar meanings [25]. Another representation named BoW (Bag of Words) treats words as a collection of words. Of course, the order here is lost. In Bag of Words (BoW) approach, all words in a document are counted in as a collection. Since only the presence of words is found, it fails to tell where in the document, the word can be found. This happens independent of the grammar or even, the word sequence. This approach although solved the similarity problem, wasn't able to solve the problem of high sparsity. To solve the ordering problem distributional representation in 1986, focused on offered to study pattern in text. So, the

syntactic relations (using parsing, part-of-speech tagging etc) were taken in account. Some representations were based on building concepts and relations, using predicate logic to build semantic nets or form conceptual dependencies using parsing. This generally is termed under *Distributional semantics*. [25][26]

These approaches have been helpful in spam filtering, text classification etc. It is purely occurrence based and is still used for NLP tasks. One of frequency-based popular schemes for weighing the contribution of words is Term frequency and an Inverse document frequency (TF- IDF). These methods lacked consideration of both semantics and context and therefore, failed to have a fine understanding of language. However, TF-IDF is used as baseline nowadays too, depending on the word-representation being used. One of the pioneer NLP moments is when 'learning' was introduced. These were essentially probabilistic models to perform Predictive modeling. These predictive models inspected text with its context i.e., neighbouring words to predict words. To achieve this, neural language models was proposed. These models learned the distribution of words over the documents. This improved representation of words called *Word Embeddings* solved the same [6]. Initial idea was to use a simple neural network and learn embeddings values by back propagation algorithm. However, embeddings weren't just introduced to handle dimension, space and complexity reduction but also, incur meaning. Hence, leading NLP to efficient pre-trained embeddings which accounted for context. These prediction-based embeddings are just words represented with vectors of real valued numbers that defines some relationship with words in context, hence, providing the meaning of the word.

3.2 Word Embeddings

Learned Word embeddings uses the idea that words having similar meaning have the same representation. One of the popular word-embedding techniques is **Word2Vec.** This model trains on a task where the objective is to *predict* a word based on its context, typically using a shallow neural network. This means, it attempts to predict the probability that a chosen word would appear in nearby context. According to [6], words can be semantically connected. For example: 'wheel' and 'engine' are related to the word 'car' because of the similarity of their meanings, the word 'banana' must be different. It is very interesting too

because, with such a mathematical representation, words can have mathematical operations too with vectors, such as *king – man + woman = queen*.

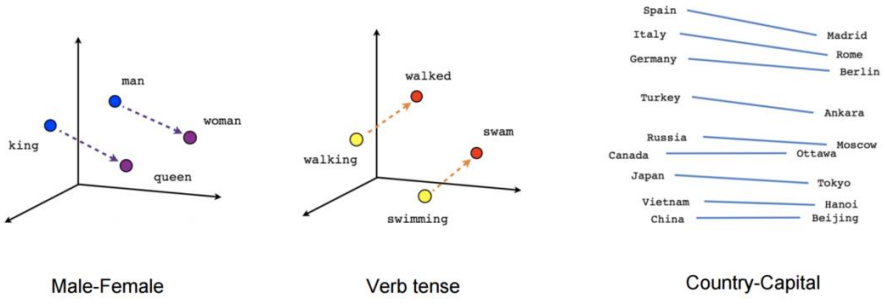


Fig 3: Relations captured by word2vec [8]

Word2Vec model can be implemented using two neural models- Continuous BOW (CBOW) and Skip gram model. These models worked with the context of words within a window size. While the former in Fig 4 computes the conditional probability of a target word given contextual words in window size, the latter predicts the surrounding context words given a central target word. [6]

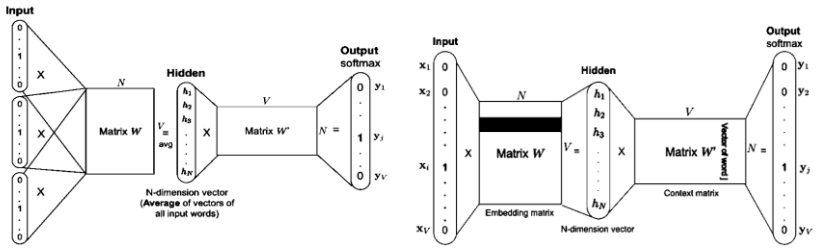


Fig 4: CBOW and Skip Gram Model

Neural models input the word in form of one-hot representation which is then fed into the hidden layer. The output is fed to the SoftMax layer to normalize the range of probability the results may output. Once the model is trained, learned weights of the hidden layer can be used to represent the dense representation of that word, a.k.a word-embedding. Word2Vec and other variants have been used for tasks like language modeling. Developed in 2013 to make the neural-network-based training of the embedding more efficient, it still is considered standard for developing pre-trained word embedding as they tend to embed syntactical and semantic information. [6]

Another embedding model named **GloVe** (Global Vector) representation. GloVe's intuition essentially is that co-occurrence of words encodes some meaning. Using the word-context matrix, it captures the statistics of words co-occurrences. This training is performed on global word-to-word co-occurrence to find some meaning. It combines the matrix factorization method and skip-gram model to optimally capture global statistics of the data. Since this method captures the global statistics in the corpus, it is named Global Vectors (in short GloVe). In LSA (Latent Semantic Analysis), it also uses matrix factorization to perform dimensionality reduction. These methods have a very good performance on various tasks like word analogy, word-similarity and named entity recognition. If implemented solely, these methods fail to work as effectively, word analogy task and leverage statistical information respectively. But GloVe uses an unsupervised global bilinear regression model and takes care of these problems. [7]

Advances in neural modeling modified use of language models from shallow to deep neural networks. With respect to word-window, sequence models, transformer modeling, NLP has shifted and come a long way. However, their applications are yet to be exploited. For examples: Models like ULMfit, ELMo, and BERT did significantly raise the bar for previous neural models, they are still widely used and often used in solving different NLP tasks and applications.

3.3 Deep learning for Representation learning

While traditional machine learning methods worked well because of human-designed features and representations, deep learning improved that by learning 'good' features automatically. With data growing at such a high rate, self-learning, and adaptive models becomes all the more important. Sequential deep learning models have led to ground-breaking results in NLP. Recurrent Neural Networks or RNNs are specialized neural-based approaches for sequential information. On every instance for the input, based on the previous results i.e., 'memory', a computation is applied recursively. [10]

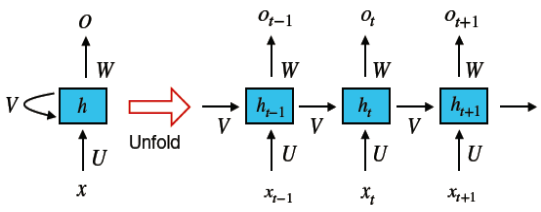


Fig 5: Simple RNN network [8]

The main strength of such a network is to **memorize** the result of previous computation and use this information for upcoming computation. This effectively improves performances for sequence i.e., context-based relations. What makes RNNs special is that they allow operations of a sequences of vectors. They can even be used in batched for training and BTT can be applied batch-wise for effective learning process. It is also notable that even if the data is not in form if sequences, RNNs can still formulate and train pretty well in a sequential manner itself. However, RNNs have been ineffective with long term dependencies because it suffers from vanishing gradient problem. To take care of this, variants like GRUs, LSTMs were introduced.

LSTMs are a variation of RNNs that outperformed simple RNNs with its capability to learn *long term dependencies*. LSTMs have since been one of the most effective variants of RNNs [26]. It follows a simple chain-like structure like RNNs but the trick is every module has a different structure or function. Instead of one neural layer as in RNN, it has four interacting in a special manner. LSTM works with three gates: input, forget, and output gates. [1]

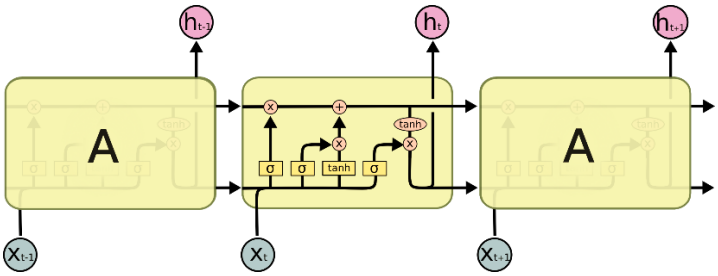


Fig 6: LSTM cell [9]

These gates decide which information has to be thrown away from the cell state carried out by a *sigmoid* layer (0 cell state to forget, 1 cell state to keep). With input layer, what has to be stored is decided. This is done with a *sigmoid* layer for input for new updates and *tanh* layer to create a new candidate vector which replaces the nodes to be forgotten and saved into memory. Here, tanh also distributes the gradients in a way that regulates the values flowing through the network. Further is the output layer that applies a *sigmoid* operation and another *tanh* operation on cell states, and multiplies this to determine which values to output. Because of such interaction between these four layers and not just one NN layer, LSTM is more resilient than a vanilla RNN. [9]

$$\hat{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$
$$i^{<t>} = \sigma(W_i[a^{<t-1>}, x^{<t>}] + b_i)$$
$$f^{<t>} = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$
$$o^{<t>} = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$
$$c^{<t>} = i^{<t>} \odot \hat{c}^{<t>} + f^{<t>} \odot c^{<t-1>}$$
$$a^{<t>} = o^{<t>} \odot \tanh(c^{<t>})$$

Here, σ for logistic sigmoid function, ⊙ for element-wise multiplication function, c for memory cell states, a for activation (hidden) state, W for weight matrices, b for bias vectors of different gates for input x. [1]

Sometimes, for improved self-learning through this, these layers are stacked together in order to retain more information. These are known as **Stacking** RNNs models. As LSTMs are variation of recurrent networks, they also are used by Stacking. For eg: x number of LSTM layers are stacked when final outputs from each layer is used as input in another, and goes so-forth until x layers are completed.

Another improvement to deal with sequential data is Bidirectional LSTM (BiLSTM). It uses memory information from both past and future i.e., previous and future sequence to preserve the context of text data. To achieve bi-directionality, two separate unidirectional LSTMs are used. One for capturing past context (Forward LSTM) and another to capture future context (Backward LSTM). In forward LSTM, sequential inputs are fed in the same order as they occur but in backward LSTM, reversed inputs are fed. This way the context (both past and future) is preserved. [26]

The SOTA Transformer model uses **Attention** to boost the training speed of its encoder-decode architecture model. Through, self-attention it claims to achieve faster and efficient sequence modeling. An attention function is nothing but a mapping function of query and a set of *key-value* pairs to an output where all of these elements outputs and vector themselves [19]. With the aim to understand what to "attend" to, this mechanism is used to find which part of the sequences to consider. For instance- If a minor change in an LSTM network is made, say, instead of directly using the learned hidden states for the dense layer for classification- the values are used to compute a weighted sum, or any such function. Then this operation is what we call an attention-based approach. Special *attention* (literally, not the mathematical equation I mean) onto such weights, better representations can be determined. Although sometimes because of positional encoding, attention might sometimes not be the best approach for finding 'meaning'. However, this can be resolved

by using meaningful embeddings. Attention has helped relieve computational complexity, preserve more and better long-range dependencies in sequences and facilitated parallel computational learning in Sequential modeling.

4

Proposed Work

Aforementioned in Contributions in chapter 1, we explained how contextual representations are important for **quality** response selection for CQA. The task is a classification task through which, later- the distribution of classes of 0, 1, 2 is obtained and contextualised ranks through these meaningful word-representation are found. On experimenting with the given *Qatar Living forum* data [16][17] to improve this class distribution, several flaws were encountered. The reasons behind conducting such combinations of parameters to handle them and enhance feature selection for text sequences is covered in this Chapter. We work not only with the focus to identify the quality or *relevancy* of the text but also to identify the factors under which sequential networks can perform well for *context-based answer selection* in open-domain question answering.

4.1 Task description

The task has been divided into three subtasks: Subtask **A**, Subtask **B** and Subtask **C**. The goal is to classify the relevance of the data pair into Bad (0), Good (2) or Potentially Useful (1) class i.e., classify corresponding data pair *like (x, y)* into a class label which depicts the relevance of the pair.

The following table shows how three subtasks have been divided. The third column is added just to facilitate visualization of the combination of first-entry of our data.

Subtask	Entity to check Relevance	Pair for Classification Label (0,1,2)
Subtask A	(Related Question, Answer)	1 relatedQuestion - 10 answers
Subtask B	(Original Question, Related Question)	1 originalQuestion - 10 relatedQuestions
Subtask C	(Original Question, Answer)	1 originalQuestion - 100 answers

Table 1: Task description

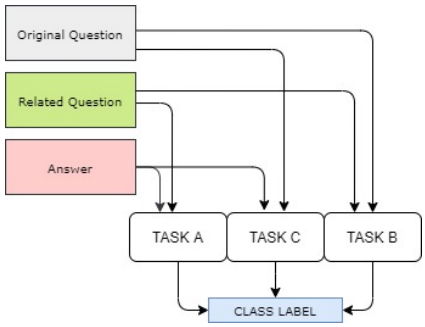


Fig 7: Task description (This label (0, 1 or 2) depicts the 'relevance' of the text pair.)

4.2 Dataset Description

The IR part i.e., the retrieval of CQA knowledge base already carried out by SemEval. It provided Qatar Living Forum's dataset training data for English and Arabic languages. Qatar living forum is an open-domain forum for people to pose questions from multiple aspects of their daily life. Provided by SemEval 2016 in this collection, the threads are independent of each other and the lists of comments are chronologically sorted and contain additional information (e.g., date, user, topic, etc.) [16]. Here's a data snippet.

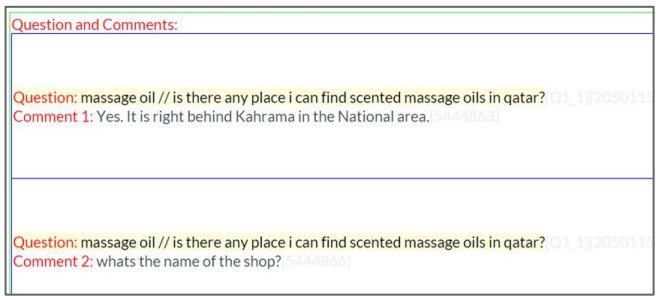


Fig 8: Screenshot CQA

```
<OrgQuestion ORGQ_ID="Q1">
    <OrgQSubject>Massage oil</OrgQSubject>
    <OrgQBody>Where I can buy good oil for massage?</OrgQBody>

    <Thread THREAD_SEQUENCE="Q1_R1">
        <RelQuestion RELQ_ID="Q1_R1" RELQ_RANKING_ORDER="1" RELQ_CATEGORY="Qatar Living Lounge"
        RELQ_DATE="2010-08-27 01:38:59" RELQ_USERID="U1" RELQ_USERNAME="sognabodl"
        RELQ_RELEVANCE2ORGQ="PerfectMatch">
            <RelQSubject>massage oil</RelQSubject>
            <RelQBody>is there any place i can find scented massage oils in qatar?</RelQBody>
        </RelQuestion>

        <RelComment RELC_ID="Q1_R1_C1" RELC_DATE="2010-08-27 01:40:05" RELC_USERID="U2"
        RELC_USERNAME="anonymous" RELC_RELEVANCE2ORGQ="Good" RELC_RELEVANCE2RELQ="Good">
            <RelCText>Yes. It is right behind Kahrama in the National area.</RelCText>
        </RelComment>

        <RelComment RELC_ID="Q1_R1_C2" RELC_DATE="2010-08-27 01:42:59" RELC_USERID="U1"
        RELC_USERNAME="sognabodl" RELC_RELEVANCE2ORGQ="Bad" RELC_RELEVANCE2RELQ="Bad">
            <RelCText>whats the name of the shop?</RelCText>
        </RelComment>
```

Fig 9: Dataset description

4.2.1 Data Preparation for Subtasks

The dataset contains about 200 Original questions. Each Original Question *Oq* has been given 10 related Questions, and these 10 related questions have 10 answers each respectively. As Each of these combinations come with its respective retrieved Relevance label. The labels are shown in following table.

TASK	PAIR	RELEVANCE LABEL COLUMN NAME (/0 = i^{th} col)
A	Related Question and Answer	Thread/RelComment/0/_RELC_RELEVANCE2RELQ
B	Original Question and Related Question	Thread/RelQuestion/_RELQ_RELEVANCE2ORGQ
C	Original Question and Answer	Thread/RelComment/0/_RELC_RELEVANCE2ORGQ

Table 2: Relevance Label detail

However, since the labels of *Questions to Answers* and *Question to Questions* had different names in the dataset, they have been renamed as such for classification task.

Question and Answer (Q, A)	Question and Question (Q, Q)	New Replaced Label
Good	Perfect Match	2
Bad	Irrelevant	0
Potentially Useful	Relevant	1

Table 3: Changed Label data

Other than the Body i.e., question Text and answer Text Attributes that describe a question and thus can be used as features as 'content' itself are Question (original and related) Subject, Question (only related) Category. Additional columns like retrieval ids or user details like username, user id etc were not taken into consideration here as it does not facilitate contextual learning. A snippet of first few columns from the retrieved dataset look as such.

OrgQSubject	Thread/RelQuestion/RelQSubject	Thread/RelQuestion/_RELQ_CATEGORY	OrgQBody	Thread/RelQuestion/RelQBody
Massage oil	massage oil	Qatar Living Lounge	Where I can buy good oil for massage?	is there any place i can find scented massage ...
Massage oil	Philipino Massage center	Advice and Help	Where I can buy good oil for massage?	Hi;Can any one tell me a place where i can hav...
Massage oil	Best place for massage	Qatar Living Lounge	Where I can buy good oil for massage?	Tell me; where is the best place to go for a m...
Massage oil	body massage	Qatar Living Lounge	Where I can buy good oil for massage?	hi there; i can see a lot of massage center he...
Massage oil	What attracts you more ?	Qatar Living Lounge	Where I can buy good oil for massage?	What attracts you more ?

Fig 10: Question Data Columns

Thread/RelComment/0/RelCText	Thread/RelComment/1/RelCText	Thread/RelComment/2/RelCText	Thread/RelComment/3/RelCText	T
Yes. It is right behind Kahrama in the Nationa...	whats the name of the shop?	It's called Naseem Al-Nadir. Right next to the...	dont want girls;want oil	
Most massages in Qatar are a waste of money. A...	my masseuse is very good. calling her from to ...	there is a massage center near mall roundabout...	Try Magic Touch in Abu Hamour (beside Abu Hamo...	(
"have you try lady siam massage next to toy ""...	I have never been for a massage! Can you belie...	Massage n ramada is not bad... If it's true th...	never oryx?? well how about we make a date and...	y
So you are looking for a message !!! hmmmm i d...	Gringer; You can also try out the Internationa...	"There was a good Thai massage place next to t...	If you're female (both places only service wom...	
none: youth	massaging stones !	believe	Girls with short hair cuts .	

Fig 11: Answers Data Columns

Thread/RelComment/0/_RELC_RELEVANCE2ORGQ	Thread/RelComment/1/_RELC_RELEVANCE2ORGQ	T
Good	Bad	
Bad	Bad	
Bad	Bad	
PotentiallyUseful	Good	

Fig 12: Label Data Columns

For convenience, we use the following abbreviations here on out for the data columns available: *oQ* for original question, *relQ* for related question and *A* for answers; *F* stands for 'feature'. As per Table 4, it is apparent how many *text Pairs* must be made from these attributes in order for the data to be fully coherent, for eg: In task C for 1 original question, 100 pairs of it have to be made for 100 answers; but in task B, for 1 original question 10 pairs of it are required for its 10 relative questions; and in task A, each of these related Qs have 10 answers so every entry will form a pair with each of these 10 pairs correspondingly.

Original Question oQi *- Q - Body* *- F - Subject*	**10 Relevant Questions relQj** (0-9) for every oQi *- Q -Body* *- F - Subject, Category (Merged in Data)*	**10 Relevant Answers Aij** (0-9) for every related question i.e., **100 Relevant Answers Aij (0-99)** for every original question *- A - Body*

Table 4: Subtask description

(Please note that in experiments, the final pair count may vary as duplicate and missing values removed.)

In experiments, the final count in analysis may vary as duplicate and missing values were removed in Pair generation. Once the needed columns were retrieved for each subtask, the (q,a) or (q,a) i.e., (text1, text2) pairs were formed and separately stored in csv files for ease of experimentation. Following Dataframes shows the prepared data here have *same* (& renamed from original) columns names for analysis, convenience and to avoid confusion. As mentioned earlier in table 4, feature means question subject or category. It

is to be noted that labels are left the same at this stage. They are replaced later when embeddings are prepared.

```
train_base.head(15)
```

	feature	question	answer	labels
0	massage oil	is there any place i can find scented massage ...	Yes. It is right behind Kahrama in the Nationa...	Good
1	massage oil	is there any place i can find scented massage ...	whats the name of the shop?	Bad
2	massage oil	is there any place i can find scented massage ...	It's called Naseem Al-Nadir. Right next to the...	Good
3	massage oil	is there any place i can find scented massage ...	dont want girls;want oil	Bad
4	massage oil	is there any place i can find scented massage ...	Try Both ;) I'am just trying to be helpful. On...	Good
5	massage oil	is there any place i can find scented massage ...	you mean oil and filter both	Bad
6	massage oil	is there any place i can find scented massage ...	Yes Lawa...you couldn't be more right LOL	Bad
7	massage oil	is there any place i can find scented massage ...	What they offer?	Bad
8	massage oil	is there any place i can find scented massage ...	FU did u try with that salesgirl ?	Bad
9	massage oil	is there any place i can find scented massage ...	Swine - No I don't try with salesgirls. My tas...	Bad
10	Philipino Massage center	Hi;Can any one tell me a place where i can hav...	Most massages in Qatar are a waste of money. A...	Bad

Fig 13: Subtask A Data Columns

```
train_base
```

	feature	question	rel_question	labels
0	Massage oil massage oil	Where I can buy good oil for massage?	is there any place i can find scented massage ...	PerfectMatch
1	Massage oil Philipino Massage center	Where I can buy good oil for massage?	Hi;Can any one tell me a place where i can hav...	Relevant
2	Massage oil Best place for massage	Where I can buy good oil for massage?	Tell me; where is the best place to go for a m...	Irrelevant
3	Massage oil body massage	Where I can buy good oil for massage?	hi there; i can see a lot of massage center he...	Relevant
4	Massage oil What attracts you more ?	Where I can buy good oil for massage?	What attracts you more ?	Irrelevant
...	...	...	...	...
1953	How to get rid of cats? Curiosity and the cat:...	I have a number of cats constantly roaming in ...	Everyone says curiosity killed the cat. When w...	Irrelevant
1954	How to get rid of cats? I need every woman advice	I have a number of cats constantly roaming in ...	I have cat at home very clean and healthy...i ...	Relevant
1955	How to get rid of cats? HOW TO KILL ONE RAT AT...	I have a number of cats constantly roaming in ...	Just recently I saw one rat in our kitchen. I ...	Irrelevant
1956	How to get rid of cats? How to get rid of rats...	I have a number of cats constantly roaming in ...	let me know what to buy and from where to buy ...	Irrelevant
1957	How to get rid of cats? How to catch rat in home?	I have a number of cats constantly roaming in ...	Can any one suggest me how to catch rat which ...	Irrelevant

1958 rows × 4 columns

Fig 14: Subtask B Data Columns

```
train_base
```

	feature	question	answer	labels
0	Massage oil	Where I can buy good oil for massage?	Yes. It is right behind Kahrama in the Nationa...	Good
1	Massage oil	Where I can buy good oil for massage?	whats the name of the shop?	Bad
2	Massage oil	Where I can buy good oil for massage?	It's called Naseem Al-Nadir. Right next to the...	Good
3	Massage oil	Where I can buy good oil for massage?	dont want girls;want oil	Bad
4	Massage oil	Where I can buy good oil for massage?	Try Both ;) I'am just trying to be helpful. On...	Bad
...	...	...	...	...
19985	How to get rid of cats?	I have a number of cats constantly roaming in ...	Try Rat Glue; Cheese trap or a KAT.	Bad
19986	How to get rid of cats?	I have a number of cats constantly roaming in ...	Rat Trap don't work.. i tried it; but the Ba$$...	Bad
19987	How to get rid of cats?	I have a number of cats constantly roaming in ...	but find a big rock for it.	Bad
19988	How to get rid of cats?	I have a number of cats constantly roaming in ...	panda this is a stone age technique.....hope...	Bad
19989	How to get rid of cats?	I have a number of cats constantly roaming in ...	every_mothers_n :)	Bad

19990 rows × 4 columns

Fig 15: Subtask C Data Columns

4.3 Proposed Approach for Selection of 'Relevant' responses

In coherence with aforementioned notion that community forum's data is usually related and contextual in manner, and has the capacity to span the spectrum of a particular question or domain or category (which ever dependent variable might be)- the focus here is to learn **contextualized** embeddings using **sequential** networks as advantageously and accurately as possible [20]. With efficacious enough implementation, the *Quality* or *Relevancy* of retrieved answers can certainly be judged using predictions of class itself. Details are discussed in upcoming sub-sections.

The flow for the task (even all subtasks) is as follows:

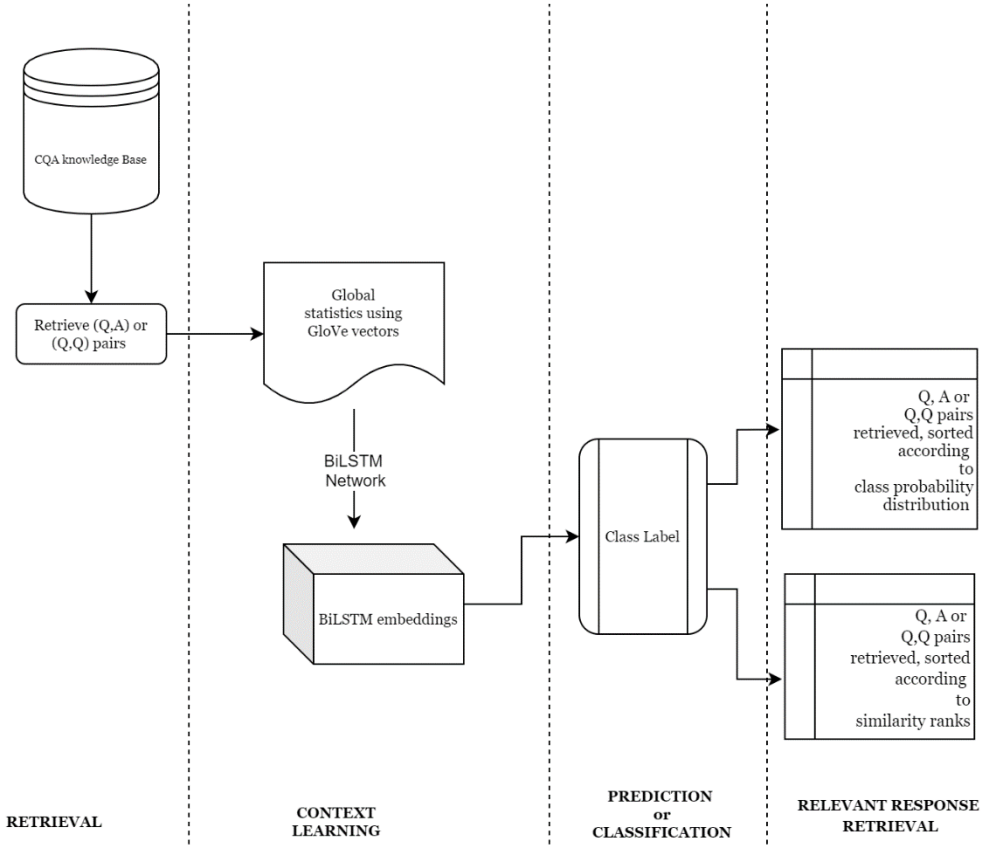


Fig 16: Proposed work flowchart

4.3.1 Learning Representations for Predictions

Motivated from the formerly explored methods, we experiment on BiLSTM model as it has been promising for sequence modeling, so forth, open-domain QA task [20][12]. The network implementation is rather simple. Vector representations for texts, say, input question and related questions/answers (or other pairs as per subtask) are generated separately here to capture sequential information attribute wise. Essentially, the problem has been treated as a Classification Problem so for each of (oQ, relQ), (relQ, A) and (oQ, A) new embeddings are computed to label them as good, bad or useful [11][12]. For example, for subtask C it goes like this:

Original Question -> *BILSTM* -> *question embedding*

Answer -> *BILSTM* -> *answer embedding*

Mathematically, the learned vector of a paragraph of x tokens is fed into LSTM to obtain the paragraph vector: $q=\{q_1,\ldots,q_x\}$ where $q_{\sim i}=LSTM(\{q_{\sim 1},\ldots,q_{\sim x}\})=\{Q_{embed}\}$
(here q means question). The other text vectors, answer and feature are encoded similarly only to be concatenated together later.

We assume such content-rich embeddings (or data-dependent features) are bound to improve the Q-A pair information hence, better feature engineering as possible. For eg. Subject and Category words such as "Visa" and "Month" will appear in responses (questions or answers) and having such words can be treated as good aspects or features for an answer about booking and locations of visa appointments and hearings. This also infers that topics which are talked about can be found out and with such feature columns (subject/category here), it can be fruitful to predict *subset of documents* inside that topic. Traditionally comparing, you can see it is not much different than using LDA (Latent Dirichlet Allocation) which is known for topic modeling. Naturally, the aim is to find good results (just) for the given dataset to facilitate better learning question-dependent attributes as **feature** themselves, data attributes are used as separate features, hence have their own embeddings.

Feature -> *BILSTM* -> *feature embedding*

The BiLSTM is not randomly initialised but uses GloVe vector weights/ embeddings (50 dimension) to mainly begin with a simple document-level statistic of the data. Once new

weights are computed through this sequential network, they are concatenated together i.e., Q_{embed}, A_{embed},F_{embed} vectors into Z_{embed} to be fed to Dense layer of SoftMax activation for Classification. The function $Softmax(Z_i) = e^{zj} / k\ e^{z} k$ [26] produces label into one of the three classes.

The output retrieved is essentially, a vector of probabilities of each class for the given particular sequence pair or text pair. They imply how likely the (q, a) or (f, q, a) pair belongs to the Classes (0, 1 and 2 here).

The common model hyperparameters that have been fixed across conducted experiments are: For *(0, x)* set of class labels where *x* is the maximum number of classes, we use *Softmax* as it is best for multinomial classification*; Sparse Categorical Cross Entropy* is chosen because it works well for categorical data. Unlike categorical cross entropy that generate one-hot categorical encodings for labels- this entropy saves a lot of space by skipping that. Also, it facilitates the idea of text pairs having only one class label exhibiting a mutually exclusive relationship- therefore, making the classification better for our task. However, if multi-label classification is to be entertained, Categorical cross entropy comes in handy; Lastly, *RMSprop* was chosen as it an optimization technique that is especially gradient based. It even takes care of the learning rate itself. It is good for handling vanishing gradients and have worked well for LSTMs in general.

The **training or learning flow** for Selection/Prediction task for a *(text, text, label)* pair is as such:

1. Got *GloVe* initialized embedding layer for the text pair.
2. Computed *BiLSTM* embeddings of each text attribute of the text pair.
3. Concatenated these embeddings and flattened them along the *Embedding* dimension (50). It can also be done at Text length for matching sentence dimensions as it is done for Subtask b (Q-Q).
4. Used Dense layer with SoftMax as activation function.
5. Compiled (and trained) this model using Sparse Categorical Cross Entropy as Loss function, RMSProp for Optimizer (*Other hyperparameters varying, as mentioned in table ahead in section 4.3.2 Experiments*)
6. Evaluated it on test data and Stored predictions.

After 6th, we fetched the most probable class label for each text sample and saved these newly made **classifications** or predictions. On the contrary, another method was used with the aim to exhibit results from a similarity-based-rank measure. For this, the cosine similarity of *contextualized embeddings* itself was computed (Pooled Embeddings as per Step 3 were fetched from intermediate layers, rather than just using initial GloVe vectors) and test samples were sorted according to these scores. This attempt was made for **improved** feature embeddings for representation of data via these globally contextual and sequentially-trained context-based embeddings. Thus, these relevant scores unpack the opportunity for both *context-driven retrieval* and *Ranking of answers* through classification.

In our work specifically, we use the class predictions as the main aspect to be retrieving and/or sorting response data. To sort, general indexing for unique set of questions was done. Also, using cosine similarity for relative ranks to take benefit from these globally and sequentially trained context-based embeddings. The information retrievable after this comprises some of the attributes in the following table.

Feature sequence	*Question (or A) sequence*	*Answer (or Q) sequence*	*Predicted Label*	*Cosine Scores*	(Optional)
Text	Text	Text	Max probability for Label	Cosine similarity	*Saving 0/1/2 separately to facilitate targeted selection of responses to be retrieved.*

Table 5: Output format

To find the most useful training approach many experiments to exploit properties of BiLSTMs, operations in deep learning and the Data itself (as shown in upcoming tables 6, 7 and 8) were conducted over the course of this dissertation. The goal, performance and results of all these are shown in the next subsection.

4.3.2 Experiments

For classification of quality/relevance/label of a question-question or question-answer pair or their relationship or their context, experiments with following **aspects** in mind were carried. *Data Attributes* can be used in order to make most use of data. In conjunction to this 'featured' data, other factors that are responsible to train a meaningful

representation are Word Embeddings, Nature of the Neural model, Nature of data being used for model, and lastly model's corresponding *Hyperparameters* are to pondered upon. These aspects can be judged using the model's metrics: loss and accuracy, and so forth, by comparing the difference in **true and predicted** label/classes of the given data.
To find the best model for learning representation and so forth, have right predictions of quality answers i.e., relevant responses, we chose to stay investigational for it and deployed several combinations playing with these constraints.

In our dataset, to understand these factors closely, we **monitor**:

1. what textual information must be used i.e., only body/text, or category/subject question features.
2. what must be the length of answers/questions/features be i.e., how many 'words' or lists from the textual columns be used for which GloVe vectors will be read. This also helps in having better (less probably) dimension for simply encoded word representation.
3. what model must be capturing context well i.e., how to deploy BiLSTMs in the best manner possible; for eg: through vanilla model, tuned model, stacked model, or hybrid model.
4. what must be the hyperparameters i.e., the number of hidden layers, the dropouts, the recurrent dropouts and batch size for used LSTM-based models.
5. what to do with **imbalanced** class distribution of label data i.e., what to do about huge gaps in number of training labels for each class. For instance, Bad class has 100 labels but Good has only 10 etc. This will lead to skewness and inaccurate learning.

The basis to monitor these factors were found in the tenure of this research. To address these, following are the conditions under which they are implemented. Even though, many combinations for aspects under experiments 1, 2 and 3 were carried out; they are not written in this dissertation as they were executed with changes uncommon across tasks. However, they helped in setting up constraints and models to compare and study.

Experiment 1 uses a Three Pair data to implement three kinds of BiLSTM Models- Simple (1-layer), Stacked (4-layered) and a 1-layer attention model *(see M1, M2, M3 names*

used for convenience). The first simple model is used to examine the performance of this sequential model, while others are used as they have known to be powerful and with data such as ours, which depends on *communication* or *words* or *sequences* across the database. In hope to capture as much information, these models were implemented. Additionally, (M3) for finding cosine similarity of BiLSTM-learned weights is only implemented under simple 1-layer model. Here, attention was deployed over text pairs. These are properly shown in following table.

Experiment	**Data (train pair – ([f,q,a],label)**	**BiLSTM Model ->** *Hyperparameters*	**Simple (M1)**	**Stacked (M2)**	**Attn Cosine (M3)** *(common sentence length 50)*
E-1	**F, Q, A** *for task A and C* (sentence length 10, 20, 50) Or **F, Q, Q** *for task B* (sentence length 50)	Hidden Units	128	64	64
		Dropout	0.2	0.3	0.4
		Recurrent Dropout	0.15	0.2	0.2
		Batch Size	64	32	32

Table 6: Experiment-1 settings

Here's a simple plotted model of task A with two text pair (q, a) as input. At each level, just another is increase if feature too is used i.e., the three-text pair (f, q, a) pair is used.

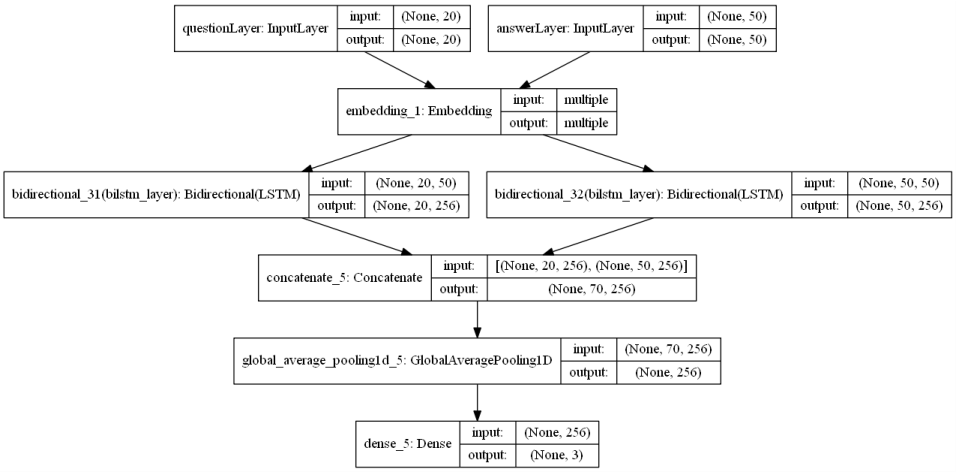


Fig 17: Simple one-layer (q, a) model

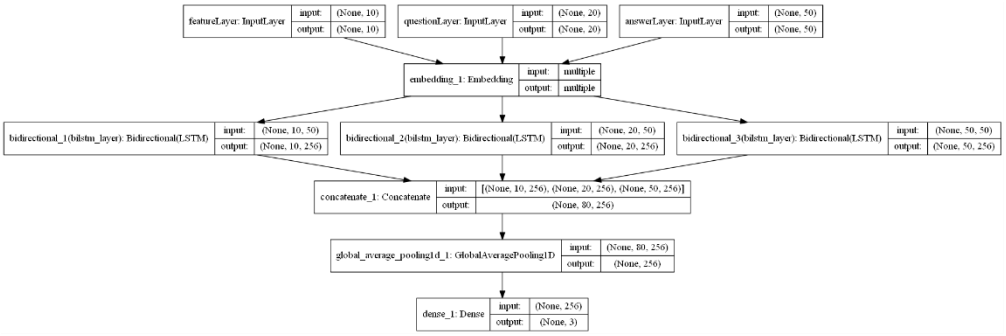


Fig 18: Simple one-layer (f, q, a) model

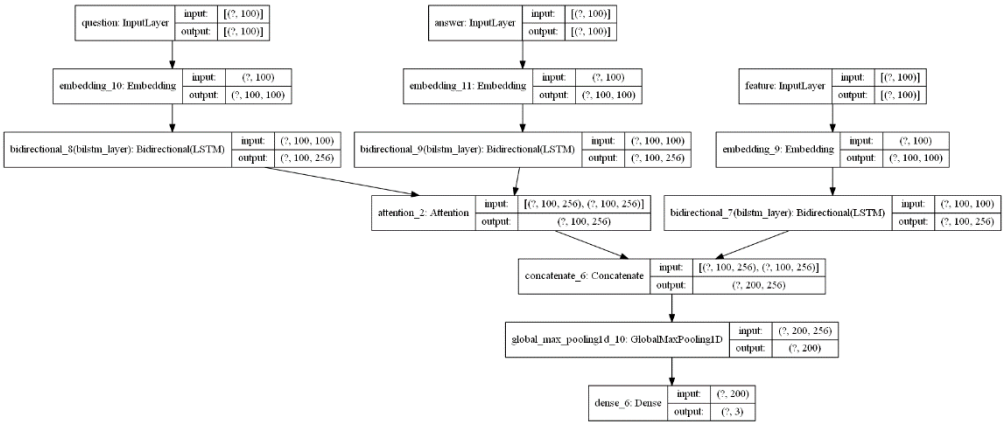


Fig 19: Cos-BLSTM model

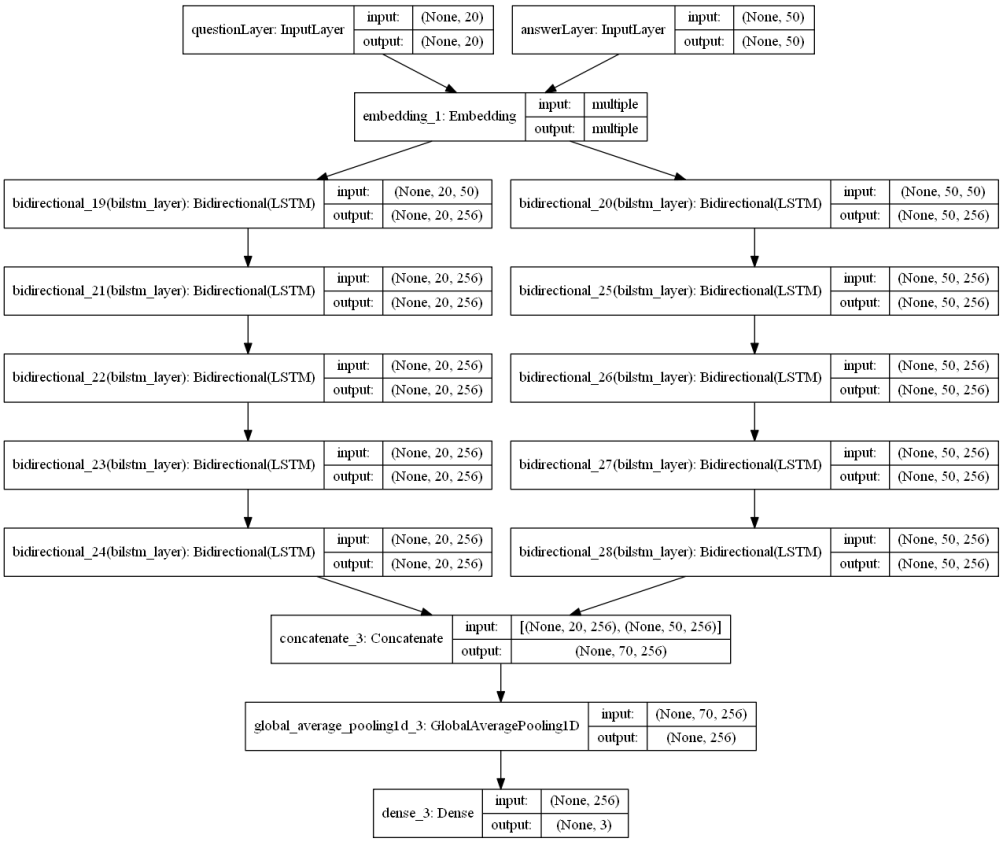


Fig 20: QA stacked model

Experiment 2 is shown for only Two text pair i.e., question body text pair & we use only Stacked BiLSTMs.

Experiment	**Data**	**BiLSTM Model ->** *Hyperparameters*	**Stacked (M2)**
E-2	Q, A *for task A and C* Or Q, Q *for task B*	Hidden Units	128
		Dropout	0.4
		Recurrent Dropout	0.15
		Batch Size	64

Table 7: Experiment-2 settings

Experiment 3 is for dealing with *imbalanced* class or label data which perhaps is the roadblock. In our dataset, the irrelevant or bad or 0 class was in abundance and in contrast, class 1 and 2 were few. When implementing Experiment 1 and 2, it was observed

that predictions too were made for Class 0, and were not at all accurate. We believed *balancing* class distribution might yield interesting results so we performed **Upsampling**. This was carried out by writing a simple function that calculate label counts for the set to be sampled and increase the Good (i.e., 2) and Useful (i.e., 1) classes samples to the maximum or *Highest*-class count, which in our data was Bad (i.e., 0). We tried various data subsets to upsample to train for classifications, for instance, only training set was upsampled and the given validation set was used for the task completion; or all train, validation and test were upsampled and trained for classification. But none showed as stable and close a performance as what we include in this project. Hence, in experiment 3 here, the training and validation sets were Upsampled and new training, testing and validation data were created using sklearn's *train_test_split()* function. This new Upsampled database followed usual course of the Representation approach after this i.e., once creation of these **data pair** was done, both experiment 1 and experiment 2 were carried out[2]. Abbreviation used S2 where S stands for Sampling. If you must foresee, at least seven models for one task are mentioned in our work. Summing up, experiments sum out to be as such.

SubTask name	**Experiment notation**		**Model notation**
A **B** **C** ***A)B)C) Reference***	**E1** (fqa) or (fqq)		M1 (simple) M2 (stacked) M3 (cosine)
	E2 (qa) or (qq)		M2 (stacked)
	E3 (S2) for sampling	*FQA/FQQ* *QA/QQ*	S2-M2 (S2- sampled stacked) S2-M1 (S2- sampled simple) E2-M1 (S2- sampled simple)

Table 8: [3] Notations for all Experiments

The results for all tasks with their model performance and its classification report, and retrieving all data pairs with their respective labels as required; visualized through pandas' functions are shown in the next chapter.

[2] Please note that all models were implemented for Subtask A, B and C.
[3] Refer 5.1.2 **5.1.2 Performance** Report

5

Results and Analysis

By the end of this chapter, it will be clear how sequential networks have performed, what problems occurred and how they can be acknowledged for such tasks. Up until now, to the best of our knowledge, no work has been done for *all* subtasks together- in classification or selection, or ranking through classification. Hence, in this final chapter, we demonstrate results to report all the results in as comprehensive a manner.

5.1 Experimental Results

Firstly, there is how retrieval was visualized. Following it is a collective performance report of investigatory models in tabular form. And afterwards, a comprehensive analytical report is shown that was being used to judge experiments to not only tune parameters but also devise new experiments along the way.

5.1.1 Retrieval of Relevant responses

Using *Pandas'* simple & multi-level indexing, a simple and efficient demonstration of classified data of each subtask has been performed to visualize responses after classification by sorting by computed cosine similarity. Following there are **snippets** of

all tasks data upon simply retrieving with and without classifications. [4] With the former, retrieval with sorting through unique pairs of rows and columns or even column pair like using (feature, cosine similarity) can be done; like for combination of task A, first 10 rows will be aligned with respect to an attribute, in an orderly manner. Here, attribute refers to any column upon whose selection, it can be used as an index in a frame. What is great about this process is that in all its simplicity it present results as easily readable and accessible information.

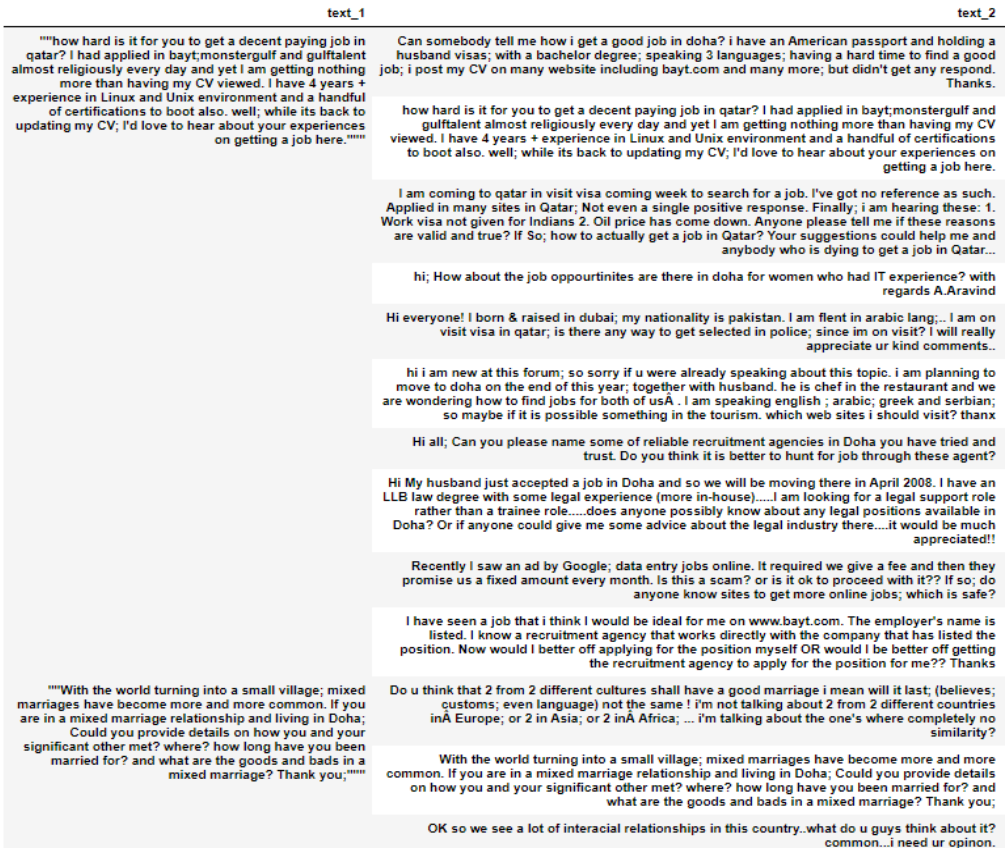
feature	text_1	text_2
"""HMC hospital; Qatar""	""Hi.... I was just wondering if anyone had any experience of working for Hamad Medical Corporation in Qatar? I have questions such as; how are they as an employer? What's the male staff accommodation like? Is the hospital and staff accommodation central to Doha? Are staff looked after? Is the work secure? Any experiences / knowledge / tips would be hugely appreciated. Thank you.""""	I have no idea about the salary; but my real estate agent showed me the new housing for the Asian Games. She told me that after the games; the housing would used for hospital employees. In a guide book; it's listed as Married Dr. Housing. Looked pretty nice. Good luck... I'll be working at WCMC-Q and you may be working with some of our students.
		hello; I'm a Biomedical Engineer hired by unistaff also last June 12; 2012. I got a call from unistaff that my visa is already with them. I went for my medical last Sept. 20; 2012. They told me that they sent my passport to the Qatar Embassy for Visa Stamping. After 1 week they called me and told me that my passport with the visa stamp is already in POEA for OEC. Today (Oct. 9; 2012) they called me and asking me to go to their office for the PDOS. Tom. (Oct. 10; 2012) will be my PDOS. My colleagues (also Biomedical Engineers) already have their plane tickets; they will leave on Oct. 13; 2012. I am hoping to leave sooner also (CROSSED FINGERS! :D). Patience is virtue.. I believe whining/complaining/getting angry/making endless calls to Ms. Marie will do us any good. Let's just wait for them to call us. Holding our papers/visa/passports won't make them richer neither won't make them any happier; those papers will just stack up in their office. They are just following the right process so.....
		Who asked you to go for an MRI; that doctor can give a referral for hamad hospital; requesting a MRI. Appointment can take upto a month. I don't think it's necessary for the phc dr to give a referral or that you need to check again with a dr at hamad. I had a referral once directly from A private clinic for an MRI. but rules change. You can check with the dr you are following up your problem with; ask him or her for a referral and then go to hamad hospital X-ray MRI unit and check about the appointment. They are usually helpful; and if they appear uncooperative; please complain since their customer service takes such things seriously; but I don't think you will get such a problem. Their staff is usually helpful; sometimes it can take more than 3 staff members to give you the right picture but they will try and help you and guide you.
		Ahli hospital
		what else you expect with a medical card of QR. 100 ? the pvt clinics ; which would have taken 75 for consultation and 75 for medicines.. (for a normal light illness) here; the health centers under HMC; give FREE consultation and the medicines costing 2 to 8 QR.. Pls check once and satisfy yourself

Fig 21: Subtask C retrieved responses

feature	text_1	text_2
"""" Imagine a world without Filipinos""	"" Imagine a world without Filipinos"" -Arab journalist Abdulla Al-Maghlooth come to think of it?... hmmmmm... http://www.arabnews.com/?page=13&section=0&article=110923&d=16&m=6&y=2008"	Already imagined www.qatarliving.com/node/121833
		wow! i feel so gratefull ginaganahan ako magtrabaho!!!!!
		walang gwapo at magaganda.....;; chaka lahat ng tao ...:)
		Yes the Qube would shutdown if their were no Pinoys..:) ---------- HE WHO DARES WINS
		Exilesaint; I thought I was the only one to see how much patronizing the article is for Filipinos!
...	...	...
working visa for female	why its very hard to get an approval for female working visa here in Qatar?	Jesus Christ...with Zhel giving a body massage looks like i'll need to keep a nurse on standby...:P PS: i'm deleting my pm draft to you :P
		Indian female visa are simple...locate indian female; marry her bring her here on your visa. Single Indian lady's visa: Locate Father first. Other Indian lady's: Locate Job First.
		Why cant you just give us the info here so everyone can benefit of your info :P
		hey funbyhobby i know what u like;pls stop sending private message;if u need nurse and body massage; go to hospital or salon :)
		im totally agree to your opinion metoyou :) its so weird because we all knew that Qatar is Booming; logically they need workers.. ""sigh""

Fig 22- Subtask A retrieved responses.

[4] Note that, cosine score is close to -1 for a good answer as Tensorflow's CosineSimilarity Losses function was used which depict -1 for higher similarity, 0 to orthogonality and 1 for most dissimilarity.



text_1	text_2
"""how hard is it for you to get a decent paying job in qatar? I had applied in bayt;monstergulf and gulftalent almost religiously every day and yet I am getting nothing more than having my CV viewed. I have 4 years + experience in Linux and Unix environment and a handful of certifications to boot also. well; while its back to updating my CV; I'd love to hear about your experiences on getting a job here."""	Can somebody tell me how i get a good job in doha? i have an American passport and holding a husband visas; with a bachelor degree; speaking 3 languages; having a hard time to find a good job; i post my CV on many website including bayt.com and many more; but didn't get any respond. Thanks.
	how hard is it for you to get a decent paying job in qatar? I had applied in bayt;monstergulf and gulftalent almost religiously every day and yet I am getting nothing more than having my CV viewed. I have 4 years + experience in Linux and Unix environment and a handful of certifications to boot also. well; while its back to updating my CV; I'd love to hear about your experiences on getting a job here.
	I am coming to qatar in visit visa coming week to search for a job. I've got no reference as such. Applied in many sites in Qatar; Not even a single positive response. Finally; i am hearing these: 1. Work visa not given for Indians 2. Oil price has come down. Anyone please tell me if these reasons are valid and true? If So; how to actually get a job in Qatar? Your suggestions could help me and anybody who is dying to get a job in Qatar...
	hi; How about the job oppourtinites are there in doha for women who had IT experience? with regards A.Aravind
	Hi everyone! I born & raised in dubai; my nationality is pakistan. I am flent in arabic lang;.. I am on visit visa in qatar; is there any way to get selected in police; since im on visit? I will really appreciate ur kind comments..
	hi i am new at this forum; so sorry if u were already speaking about this topic. i am planning to move to doha on the end of this year; together with husband. he is chef in the restaurant and we are wondering how to find jobs for both of usÂ . I am speaking english ; arabic; greek and serbian; so maybe if it is possible something in the tourism. which web sites i should visit? thanx
	Hi all; Can you please name some of reliable recruitment agencies in Doha you have tried and trust. Do you think it is better to hunt for job through these agent?
	Hi My husband just accepted a job in Doha and so we will be moving there in April 2008. I have an LLB law degree with some legal experience (more in-house).....I am looking for a legal support role rather than a trainee role.....does anyone possibly know about any legal positions available in Doha? Or if anyone could give me some advice about the legal industry there....it would be much appreciated!!
	Recently I saw an ad by Google; data entry jobs online. It required we give a fee and then they promise us a fixed amount every month. Is this a scam? or is it ok to proceed with it?? If so; do anyone know sites to get more online jobs; which is safe?
	I have seen a job that i think I would be ideal for me on www.bayt.com. The employer's name is listed. I know a recruitment agency that works directly with the company that has listed the position. Now would I better off applying for the position myself OR would I be better off getting the recruitment agency to apply for the position for me?? Thanks
"""With the world turning into a small village; mixed marriages have become more and more common. If you are in a mixed marriage relationship and living in Doha; Could you provide details on how you and your significant other met? where? how long have you been married for? and what are the goods and bads in a mixed marriage? Thank you;"""	Do u think that 2 from 2 different cultures shall have a good marriage i mean will it last; (believes; customs; even language) not the same ! i'm not talking about 2 from 2 different countries inÂ Europe; or 2 in Asia; or 2 inÂ Africa; ... i'm talking about the one's where completely no similarity?
	With the world turning into a small village; mixed marriages have become more and more common. If you are in a mixed marriage relationship and living in Doha; Could you provide details on how you and your significant other met? where? how long have you been married for? and what are the goods and bads in a mixed marriage? Thank you;
	OK so we see a lot of interacial relationships in this country..what do u guys think about it? common...i need ur opinon.

Fig 23: Subtask B retrieved responses

5.1.2 Performance Report

We assume that most innate way to understand the main goal of classification can be done so by observing both the model learning performance and final count of predicted labels as opposed to their original ones. One way for experiments to be judged is as the following format. The training-testing performance tells us the model's performance during the learning process. The metrics judges the classification and for evaluation of these in intuitive manner, is the difference between Number of labels.

Experiments	**Training and Testing performance**	**Metrics Report**	**True vs Predicted Labels**
Experiment Refer	Loss and Accuracy	Precision, Recall and F1 score	#Label Count for each class

Table 9: Result format

So, we show the collective report with measures as devised experiments as below. [5]

A) Results for A [4]

FQA

				loss	acc	epoch			precision	recall	f1			TRUE	predicted
simple	M1	Model	training	0.17	0.93	100	Report	0	0.76	0.86	0.81	Count	0	3053	3430
			testing	1.02	0.74			1	0.6	0.59	0.59		1	1123	2305
			validatio	1.67	0.66			2	0.8	0.69	0.74		2	2664	1105
								acc			0.75				
stacked	M2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
		Model	training	0.49	0.8	80	Report	0	0.76	0.84	0.8	Count	0	3053	3360
			testing	0.78	0.72			1	0.54	0.5	0.52		1	1123	1030
			validatio	0.97	0.67			2	0.76	0.7	0.73		2	2664	2450
								acc			0.73				
cosine	M3			loss	acc	epoch			precision	recall	f1			TRUE	predicted
		Model	training	0.92	0.58	53/100	Report	0	0.5	0.83	0.62	Count	0	3052	5142
			testing	1.02	0.49			1	0.34	0.1	0.15		1	1123	326
			validatio	1.06	0.49			2	0.55	0.28	0.37		2	2664	1372
								acc			0.5				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	m1	Model	training	0.18	0.93	100	Report	0	0.79	0.76	0.77	Count	0	8149	7912
			testing	0.99	0.77			1	0.77	0.77	0.77		1	7327	7337
			validatio	1.0018	0.77			2	0.76	0.78	0.77		2	8094	8321
								acc			0.77				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	m2	Model	training	0.5	0.79	80	Report	0	0.76	0.75	0.75	Count	0	8149	8105
			testing	0.77	0.72			1	0.72	0.65	0.68		1	7327	6643
			validatio	0.77	0.72			2	0.7	0.76	0.73		2	8094	8822
								acc			0.72				

Table 10: Results for Subtask A for three-pair data i.e., (feature, question, answer)

QA

				loss	acc	epoch			precision	recall	f1			TRUE	predicted
stacked	M2														
		Model	training	0.12	0.95	100	Report	0	0.79	0.81	0.8	Count	0	3053	3122
			testing	1.16	0.73			1	0.54	0.59	0.57		1	1123	1222
			validatio	1.85	0.64			2	0.76	0.72	0.74		2	2664	2496
								acc			0.74				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	M1	Model	training	0.42	0.83	50	Report	0	0.8	0.68	0.73	Count	0	8189	6998
			testing	0.76	0.73			1	0.7	0.74	0.72		1	7324	7745
			validatio	0.77	0.73			2	0.71	0.78	0.75		2	8057	8827
								acc			0.73				

Table 11: Results for Subtask A for two-pair data i.e., (question, answer)

[5] See the Table 8 (Experiments Section) for reference to Model names.

B) Results for B [6]

FQQ

				loss	acc	epoch			precision	recall	f1			TRUE	predicted
simple	M1	Model	training	0.37	0.85	91/100	Report	0	0.68	0.76	0.72	Count	0	454	510
			testing	1.41	0.55			1	0.23	0.23	0.23		1	149	154
			validatio	1.36	0.55			2	0.15	0.04	0.06		2	81	20
								acc			0.56				
stacked	M2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
		Model	training	0.73	0.68	46/80	Report	0	0.67	0.92	0.78	Count	0	454	627
			testing	0.97	0.63			1	0.31	0.11	0.16		1	149	51
			validatio	1.03	0.55			2	0	0	0		2	81	6
								acc			0.64				
cosine	M3			loss	acc	epoch			precision	recall	f1			TRUE	predicted
		Model	training	0.36	0.87	200	Report	0	0.69	0.58	0.63	Count	0	454	386
			testing	1.33	0.48			1	0.24	0.37	0.29		1	149	226
			validatio	1.52	0.44			2	0.17	0.15	0.15		2	81	72
								acc			0.49				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	m1	Model	training	0.25	0.9	100	Report	0	0.7	0.73	0.72	Count	0	1071	1126
			testing	1.017	0.72			1	0.7	0.56	0.62		1	977	1064
			validatio	1.03	0.73			2	0.76	0.87	0.82		2	930	788
								acc			0.72				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	m2	Model	training	0.45	0.81	80	Report	0	0.75	0.51	0.61	Count	0	1071	725
			testing	1.007	0.65			1	0.56	0.61	0.58		1	977	1064
			validatio	1.04	0.65			2	0.68	0.87	0.76		2	930	1189
								acc			0.65				

Table 12: Results for Subtask B for three-pair data i.e., (feature, question, related-question)

QQ

				loss	acc	epoch			precision	recall	f1			TRUE	predicted
stacked	m2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
		Model	training	0.36	0.85	100	Report	0	0.69	0.71	0.7	Count	0	454	464
			testing	1.59	0.54			1	0.26	0.33	0.29		1	149	188
			validatio	1.61	0.53			2	0.16	0.06	0.09		2	81	32
								acc			0.55				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	m1	Model	training	0.61	0.74	50	Report	0	0.69	0.48	0.56	Count	0	1102	770
			testing	0.88	0.61			1	0.5	0.66	0.57		1	967	1264
			validatio	0.89	0.62			2	0.71	0.73	0.72		2	909	770

Table 13: Results for Subtask B for two-pair data i.e., (question, related-question)

[6] See the Table 8 (Experiments Section) for reference to Model names.

C) Results for C [7]

FQA																
				loss	acc	epoch			precision	recall	f1			TRUE	predicted
simple	M1	Model	training	0.22	0.91		Report	0	0.86	0.93	0.9	Count	0	5943	6422
			testing	0.86	0.81			1	0.12	0.06	0.08		1	403	193
			validatio	0.95	0.77			2	0.26	0.15	0.19		2	654	385
								acc			0.8				
stacked	M2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
		Model	training	0.42	0.84		Report	0	0.85	0.95	0.9	Count	0	5943	6623
			testing	0.58	0.81			1	0.04	0	0.01		1	403	47
			validatio	0.59	0.8			2	0.18	0.09	0.12		2	654	330
								acc			0.82				
cosine	M3			loss	acc	epoch			precision	recall	f1			TRUE	predicted
		Model	training	0.53	0.81		Report	0	0.84	0.88	0.86	Count	0	5943	6200
			testing	0.68	0.75			1	0	0	0		1	403	0
			validatio	0.68	0.75			2	0.88	0.09	0.08		2	654	800
								acc			0.75				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	m1	Model	training	0.23	0.91		Report	0	0.91	0.82	0.86	Count	0	15170	13753
			testing	0.48	0.86			1	0.83	0.89	0.86		1	12501	13368
			validatio	0.47	0.86			2	0.86	0.9	0.88		2	12491	13041
								acc			0.87				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	m2	Model	training	0.2	0.92		Report	0	0.92	0.78	0.84	Count	0	15170	13848
			testing	0.53	0.85			1	0.83	0.89	0.86		1	12501	13422
			validatio	0.53	0.86			2	0.83	0.92	0.87		2	12491	12892
								acc			0.86				

Table 14: Results for Subtask C for three-pair data i.e., (feature, question, answer)

QA																
				loss	acc	epoch			precision	recall	f1			TRUE	predicted
stacked	M2														
		Model	training	0.21	0.92	100	Report	0	0.87	0.89	0.91	Count	0	5943	6238
			testing	0.91	0.79			1	0.89	0.89	0.89		1	403	401
			validatio	0.94	0.77			2	0.28	0.15	0.2		2	654	361
								acc			0.79				
sampled	S2			loss	acc	epoch			precision	recall	f1			TRUE	predicted
	M1	Model	training	0.36	0.85		Report	0	0.84	0.84	0.84	Count	0	15206	15072
			testing	0.47	0.84			1	0.82	0.83	0.83		1	12454	12681
			validatio	0.47	0.84			2	0.86	0.86	0.86		2	12502	12409
								acc			0.84				

Table 15: Results for Subtask C for two-pair data i.e., (question, answer)

[7] See the Table 8 (Experiments Section) for reference to Model names.

5.1.3 Comprehensive Analysis

This section is essentially is a **comprehensive** walk-through of performance & observations made for analysis across experiments during the course of implementation.

Rather than dividing it by experiments alone, the observations are explained in by the variations of **DATA ⇔ MODEL** parameters in all experiments and across subtasks. So, the lens to envision the performance is:

1. *Data to Data*: Includes FQA/QA or FQQ/QQ; And its Sequence length change.
2. *Model to Model*: Includes Simple vs Stacked vs Attention Sequential Model
3. *Data to Model*: Includes Model performance as per Sampling as well as Hyperparameter Tuning.

First, we see **Data to Data** Changes. When applied QA and FQA with long-sequence length, the model not only trained too slow but it didn't learn at all. It drastically overfit. Which is why, for common experiments we used short sequences (10, 20, 50) for (feature, question, response) respectively. These also overfit on validation data. The difference was with *feature* the learning was progressing well.

For Example:

For subtask A, FQA for model M1, so is the training graph.

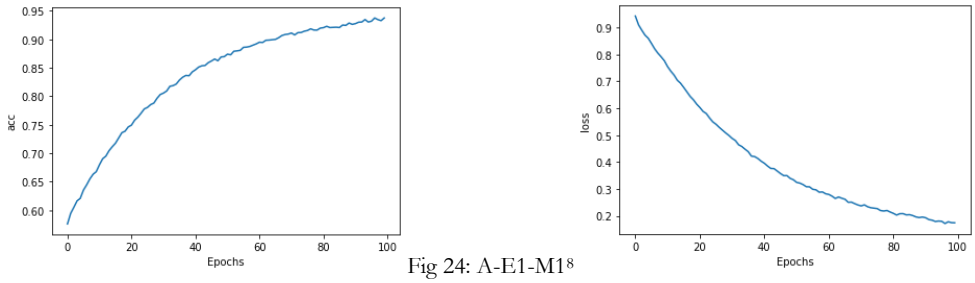


Fig 24: A-E1-M1[8]

But it fails spectacularly on validation data as following.

[8] Table 8 Refer Notations; Named as {SubtaskName-ExperimentNo-ModelName}

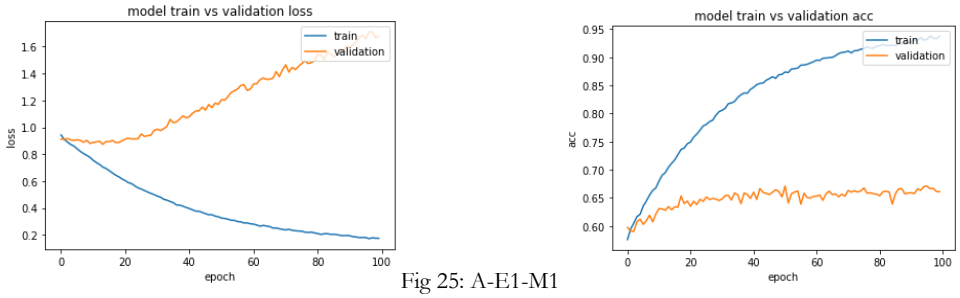


Fig 25: A-E1-M1

This happened in all three subtasks B and C. The difference between subtasks was the C trained way slower, then came B and then A in terms of speed. The fluctuation can also be seen here. Also, QA showed more of such and hence, it deemed fit to say it an ineffective approach to make classifications, especially through simple models. The possible reason for such a result found was that the pairs' first attribute i.e., question or feature, question was repetitive information [9] Other than this, hyperparameters and models were used over same data which is described in upcoming point.

In **model-to-model changes** to judge simple and stacked models performed with CQA and QA, the results were rather interesting. Stacked two-pair texts (QQ or QA) did not perform well. It overfit and even with stacked, the performance did not change. The speed was obviously lower as number of operations increased, however if performance report is considered the training vs validation vs testing performance difference within 100 epochs did show potential of 'learning' better than simple, no matter how unsuccessful it turned out to be for these tasks. This is exactly why they were experimented again with Sampling Data. Another model, deploying attention, that was taken only to compute Cosine did not perform well in the same settings. The predictions made were too inaccurate.

For example:
It is best to explain these observations using Subtask C's training models. Here for FQA simple mode (M1), overfitting happened too.

[9] Refer Table 8 to recall repetitive value pair in attribute of a datapair.

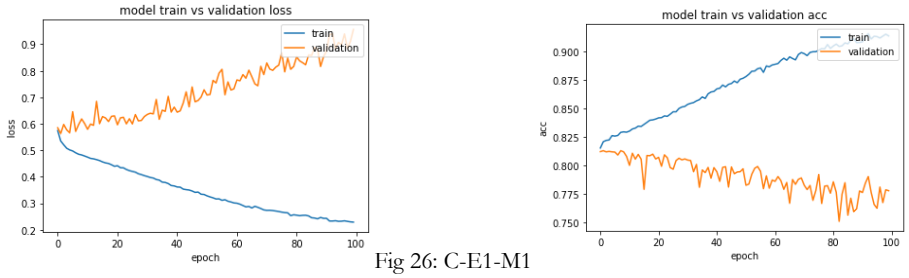


Fig 26: C-E1-M1

But in Stacked model (M2), the difference in fluctuations between were seen.

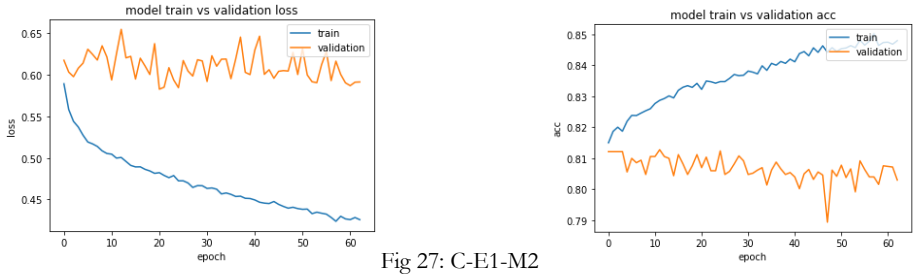


Fig 27: C-E1-M2

On the contrary, Cosine model (M3) performed almost the same computations as above but half in time, and a stagnant learning all too fast.

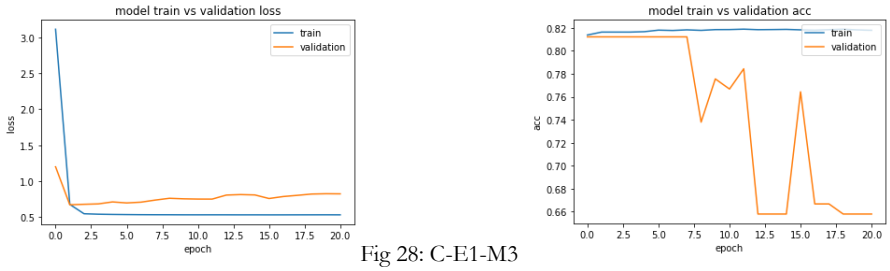


Fig 28: C-E1-M3

It is notable that (For E-1) M1 and M2 other than stacking also used different **hyperparameters**. M1 has 128 hidden units but M2 had 64 units, and so was the batch size and dropouts. The reason was both the lack of better storage space and powerful system, and importantly, using 128 cells with Simple Model and 64 (lesser) with Stacked was to ensure a stable time and space performance. Model with heavier sequences and BiLSTM layers were kept lighter on hyperparameters for lighter network. And with lesser information and a single layer network, hyperparameter were changed to accommodate more learning nodes to facilitate more information learning within the two attributes.

Perhaps many observations like hyperparameters with types of model can be included across experiments in bifurcation, but we keep focus on the most important factor being the change in data distribution itself, which is what we include in analysing for **data-to-model** changes interactions. The observations here are interesting in the manner that it performed well in some and failed in others. With **sampling**, predictions showed a good and relatively, a stable performance with 87% accuracy for subtask C's model M1; whereas 72% for subtask B and 77% for subtask A. With stacked models, this was lesser. This is also observable through the graphs so here's an example snippet.

With Sampling, for Subtask C's three-pair text (i.e., feature, question and its answer) as it can be seen in the following, the graph which of sampled-simple model (M1) performed well.

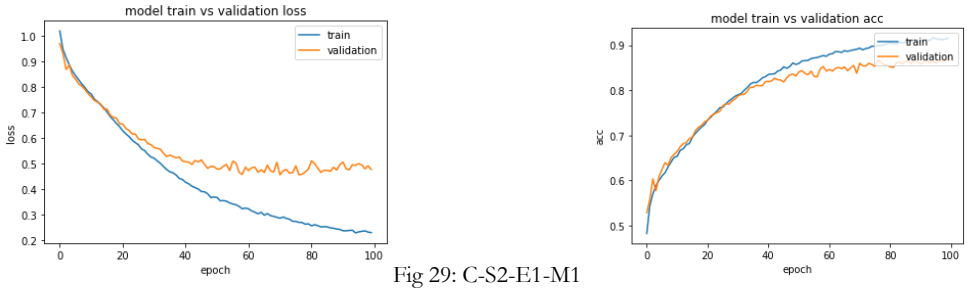


Fig 29: C-S2-E1-M1

However, the sampled-stacked (model M2) shown below had *fluctuations* so drastic from the above graph of simple-model (M1).

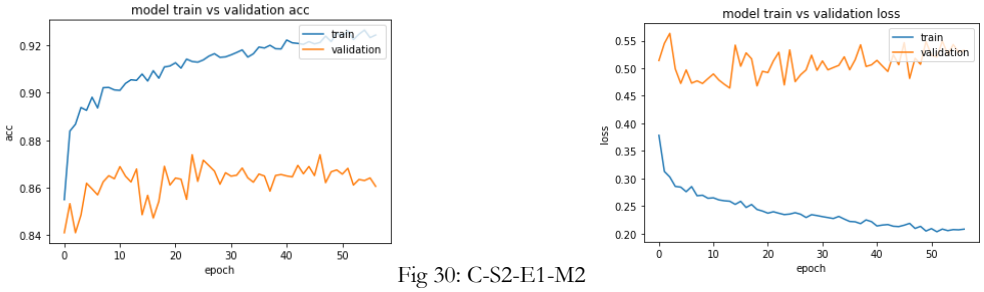


Fig 30: C-S2-E1-M2

The three-text pair performed well on sampled data whereas for two-text pair sequence, learning was not as smooth.

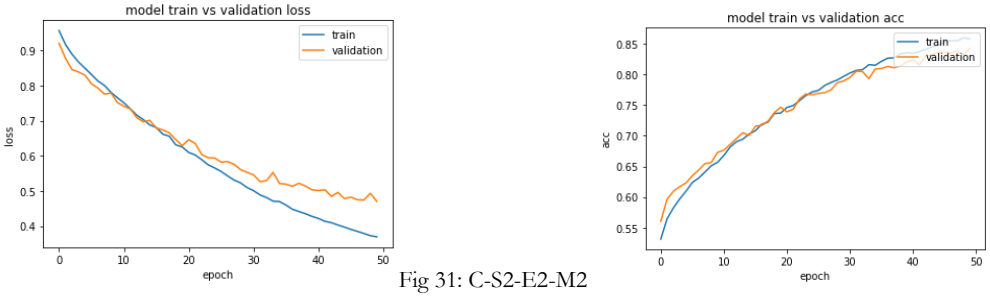


Fig 31: C-S2-E2-M2

There were fluctuations in subtask B and A. By extension, the performance of two-pair information of subtask B and A was inferior to two-pair performance of subtask C's model. Here, is the performance of Subtask B for comparison. In following, for FQQ with a single model layer (M1) on sampled data was implemented.

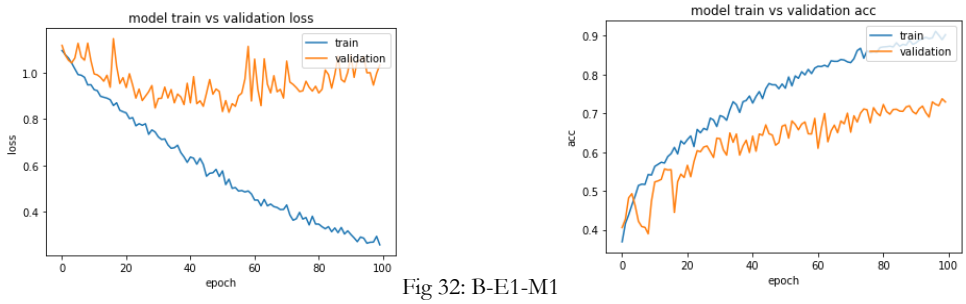


Fig 32: B-E1-M1

Subtask A also performed the same except that the difference between fluctuation frequency and gap between training and validation data was seen less. In following graph of subtasks A's two-pair sampled model M1, it can be how two-pair information has a difference in both fluctuation frequency and gap between training and validation.

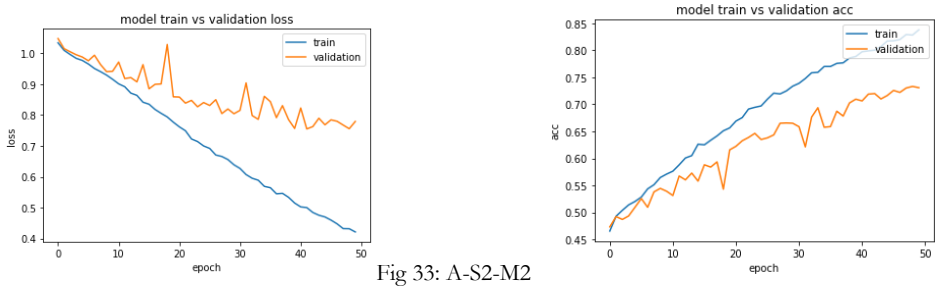


Fig 33: A-S2-M2

5.2 Remarks

The experiments devised helped narrow down the best approach for classifications and make quality predictions, hence improving relevant response selection task.

Our hypothesis that **contextual information** is captured well with **sequential networks** were derived true. With training performance of two pair data usually overfit and rather smoothened with extra text as the three-pair (f, q, a), hence reaffirming that extra textual attribute i.e., as the three-pair (f, q, a) here, more coherent sequences thus, context can be learned. Not only this, the former fails to learn and keeps a slower pace but the latter does not. To handle said hurdle with hyperparameter tuning and by changing number of model layers according to data being processed, the attempts made were fruitful too. However, to improve the lags and performance issue, the experimental settings under Experiment 3's results can be interpreted on a spectrum of dimensions, as aforementioned.

Through experiments 3 and above mention analysis of data-to-model, it can be summed that sampling works well with one-layered model than in stacked model, and better for three pair i.e., (f, q, a) than in two-pair (q, a). In contrast to it, without sampled data, performance for stacked sequential network is better than simple one-layer sequential network. In addition to this, we see LSTMs fail in simpler models. Now it is well established that learning in LSTMs takes a rather good amount of time as it has so many computational gates and cell. But here, it also failed for long sequences. In experiments (not shown in thesis) when the length was 100 for Question and 200 for Answer pair, the model was not only slower in speed but also in learning.

It must be noted that the sampling was done with respect to having **equal class distribution** and seeing the true vs predicted class count from report in the above section, it is apparent it has improved in classification for each class too. In other cases, it is observable that while even if 0 class is fairly predicted, class 1 and 2 struggles. In class 2 too, under some models the prediction count was close to true count of class 2 but 1 still failed to be considered.

Another observation is how feeding the same first question (repetitive 10 texts for task B or 100 texts for task C as opposed to its on one text response) sequences input to the

network, it performs poorly. In experiments other than the mentioned, it was observed the same data would perform better, if not, overfit when the data is rather random. This is probably why subtask A where there was no repetitive pair of information, did not overfit as easily as subtasks B and C.

Last but not the least, through experimental analysis only the difference of performance of subtask A, B and C were found. While balanced data (sampled in experiment 3) did great on classification for subtask C, both stable learning and prediction wise; it did not perform well at all on B and A. The learning of task C was slower than of A and B. Task C did good with balance distribution eventually but not for subtask A and B.

5.3 Limitations

It must be known that CQA is both quality retrieval and ranking problem. With the model experiments in our work, it was observed that even though the notion of Sequential learning is to facilitate contextual way of learning, it happens to fail until specific data is experimented with or hyperparameter tuning is done here. The problem can be both *data* (content and word embedding representation) and the *network* deployed for classification or regression tasks.

Even though changes were made to data with sampling to balance, sentences were augmented to short and long lengths manually, and through context-statistic based & varying dimension for embeddings- the sequence were improved and tested against; the model did not do well *until* it was sampled, new sets were created which had each (q, a) randomly indexed, and followed by the same representation models. Hence, it also becomes a sequential network problem.

Statistically, data happened to be the problem as it had imbalanced class distribution. Repetitive sequence-pair in task C and B was a hindrance. Same sequential networks would perform differently in both learning and speed for each subtask. This could be the reason for Overfitting too. Overfitting on validation data was observed almost in all models until the given set was dissolved, yet again in task B and A it started fluctuating. Reasons behind which when explored came out to be the data splitting or hyperparameters or regularization. To solve this, hyperparameter tuning can be done. In

our work, however, we changed them manually and experimented these models against varying data combinations (as seen in M1, M2 models).

6

Conclusion

The motive of this dissertation to apply classification for selection of relevant responses in community QA forms was achieved aptly through Sequential networks, namely, Long Short-Term Memory network. Though, they might require tuning for better representation learning. Upon investigating for solutions over it, we found that it is common for LSTMs to fail for very large sequences. To handle these, techniques that are known to be helpful are data augmentation through truncated sequences using algorithms like TBTT (Truncated Back Propagation) can work. To conclude through confirmed experiments over sequence modeling, we now believe that Contextual learning with Sequential (LSTM) Networks is an effective approach to handle CQA classification task.

Apart from solving CQA not by contextual representations- the other two dimensions to understand and address problems in CQA task i.e., linguistic and computational. It is apparent that textual data suffers from **lapses** in understanding and preserving linguistic features when being compared. In forums especially, data can have lexical gaps and ambiguity problems easily. Moreover, language features can have structural vs. syntactic and lexical vs. semantic dimensions too. With mathematical models being used as Feed ranking methods, certainly, more models like LDA and LSS can be used in combination with features for refined deep neural models. Inclusion as such would definitely improve current systems to become more semantically proficient.

Moreover, if word level is not considered but also semantic relations of sentences themselves are to be seen; CQA will cease to perform the way it does. This way, the representation of words also gets trickier, e.g., learning feature-rich embeddings with shallow or deep neural networks. Pre-training the embedding context, hence, was proposed by Google by BERT. It handles embeddings contextually. It is possible to lessen the language features and can still get a better ranking with BERT. Variants of BERT have now been explored but it has not been so imminent for CQA yet. Recently launched sentence transformers have proved to be working well for retrieval tasks, so experimenting on those using our prepared data will be fascinating.

With CQA, not only the challenge is the language but the features of the answers too. Since all responses are user-based, it is tough to know the ground truth of the queries. Adding the *answerer's* details to weigh in the responses can help improving quality of responses. For example, a review of a camera will more reliable if given by a photographer rather than a common person. Generally, depending on the tasks given, the analysis of *all* possible comments and similar questions, although helpful, fails to provide answers for new questions and might not have enough contextual information. In such cases, the results would either just be null or potentially relevant; since the context of the question may or may not have semantic relations. As CQA has not been explored in real-time, any approach to deal with dynamicity will certainly lead to an expert answering system.

7

References

1. Hochreiter, Sepp and Schmidhuber, Jurgen, "Long short term memory", Neural computation Vol-9, No.8, Pg(1735-1780), MIT Press, 1997.
2. Yoshua Bengio, Réjean Ducharme et al., "A Neural Probabilistic Language Model"; Journal of Machine Learning Research 3 (2003) 1137–1155 Submitted 4/02; Published 2/03.
3. Ronan Collobert, Jason Weston., "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning"; Appearing in Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 2008.
4. Xin-Jing Wang et.al, "Ranking Community Answers by Modeling Question-Answer Relationships via Analogical Reasoning", SIGIR 2009.
5. Tomas Mikolov et al., "Distributed Representations of Words and Phrases and their Compositionality" (2013).
6. Tomas Mikolov., "Efficient Estimation of Word Representations in Vector Space"; arXiv:1301.3781v3 [cs.CL] (2013).
7. Jeffrey Penington, Richard Socher, Cristopher Manning et.al "GloVe: Global Vectors for Word Representation", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), ACL, October 2014.
8. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.

9. Christopher Olah, "Understanding LSTM Networks", *August 27, 2015* [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/
10. Andrej Karpathy, "The unreasonable effectiveness of Recurrent Neural networks", May 21, 2015 [Online] Available: http://karpathy.github.io/2015/05/21/rnn-effectiveness/
11. Minwei Feng, Xiang et. al, "Applying Deep learning to Answer selection: a study and open task"; arXiv: 1508.0158v2 [cs. CL], October 2015.
12. Ming Tan, Xiang et. al, "LSTM based deep learning models for non-factoid Answer selection"; arXiv: 1511.04108v4 [cs. CL], March 2016.
13. Zhenzhen Li, Huang et. al, "LSTM based deep learning models for Answer ranking"; IEEE First International Conference on Data Science in Cyberspace, 2016.
14. Man Lan et. al, "Building Mutually Beneficial Relationships Between Question Retrieval and Answer Ranking to Improve Performance of Community Question Answering"; IJCNN, 2016.
15. Ivan Srba and Maria Bielikova, "A comprehensive survey and classification of approaches for Community Question answering"; ACM Transactions on the Web, August 2016.
16. International Workshop on Semantic Evaluation (SemEval- 2016) [Online]. Available at http://alt.qcri.org/semeval2016/task3/
17. Preslav Nakov et. al, "SemEval-2016 Task 3: Community Question Answering"; Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval-2016.
18. LiangPang, Yanyan Lan et.al, "DeepRank: A new Deep architecture for Relevance Ranking in Information Retrieval", CIKM 2017, ISBN: 978-1-4503-4918-5/17/11, 2017.
19. Ashish Vaswani, Noam Shazeer et. al, "Attention is all you need", 2017. Available: https://arxiv.org/pdf/1706.03762.pdf
20. Danqi Chen, et.al "Reading Wikipedia to Answer Open-domain Questions", Stanford, ArXiV: 1704.0051v2, April 2017.
21. Sebastian Ruder, "*A Review of the Neural History of Natural Language Processing",* Oct 2018. [Online]. Available: http://ruder.io/a-review-of-the-recent-history-of-nlp/
22. Van-Tu Nyugen and Anh-Cuong Le, "Deep neural network-based models for ranking Question-Answering Pairs in Community Question-Answering Systems", Springer, 2018.

23. Roy, P.K., Ahmad, Z., Singh, J.P. et al. Finding and Ranking High-Quality Answers in Community Question Answering Sites. Glob J Flex Syst Manag 19, 53–68 (2018). https://doi.org/10.1007/s40171-017-0172-6

24. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"; arXiv:1810.04805v2 [cs.CL] (2019).
25. Liu Z., Lin Y., Sun M. (2020) "Representation Learning and NLP". In: Representation Learning for Natural Language Processing. Springer, Singapore. https://doi.org/10.1007/978-981-15-5573-2
26. Dan Jurafksy, James H. Martin, "Speech and Natural Language Processing", 2020 Draft, MITs

---

"Oh, so many Views! Quality Check?"

"No likes? How irrelevant!
No, wait. That comment actually makes sense."

"LOL. If only there was more to this answer."

-things an average user thinks
when browsing forums
