Predicting Paper Quality from Abstract

A Project Report Submitted

in Partial Fulfilment of the Requirements

for course work CS563

(Neural Network for NLP)

by

Saurabh Kumar
Mridul Jyoti Roy
Gyan Ratna
Mukesh Kumar

to

Dr. Amit Awekar

(Associate Professor)

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

[November 18, 2022]

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In last few decades with the advancement of machine learning(ML) and artificial intelligence(AI), the application of these techniques in almost all field of research increased dramatically. With this abrupt use of ML and AI in almost every field, the conferences, journals and preprint archives have expanded the dissemination of research and scholarly communications. According to AI index annual report 2021 [1], among the six fields of study related to AI on arXiv, the number of publications in Robotics (cs.RO) and Machine Learning in computer science (cs.LG) have seen the fastest growth between 2015 and 2020, increasing by 11 times and 10 times respectively(Fig. 1.1). In 2020, cs.LG and Computer Vision (cs.CV) lead in the overall number of publications, accounting for 32.0% and 31.7%, respectively, of all AIrelated publications on arXiv. Between 2019 and 2020, the fastest-growing categories of the seven studied here were Computation and Language (cs.CL), by 35.4%, and cs.RO, by 35.8%.

With this tremendous growth in research, for a researcher it is very difficult to

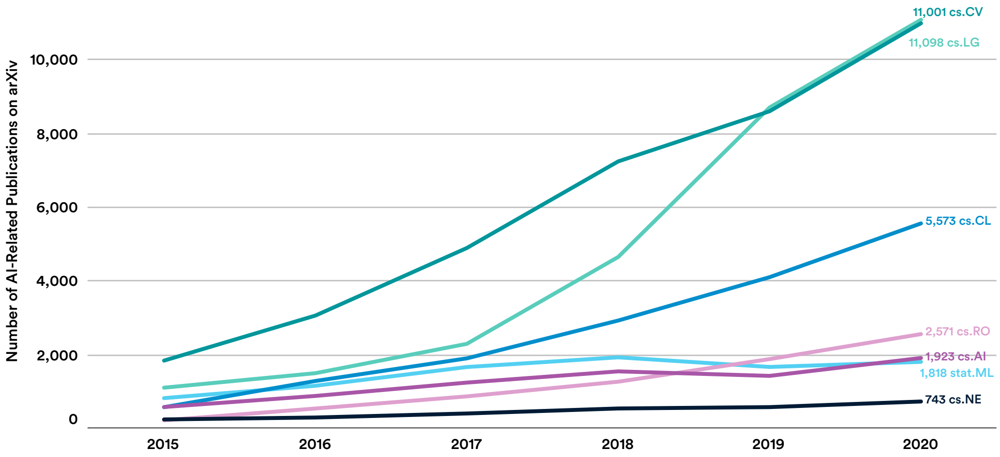


Figure 1.1: Number of AI-Related publication on arXiv by field of study 2015-20 [Source : https://aiindex.stanford.edu/ai-index-report-2021/]

decide to choose a particular conference or journals for his/her research publication. Currently no such system exist that predict where to send the research work for publication.

Motivated by the aforementioned problem a framework has proposed that predict the domain of the research among five most growing domain i.e. Computer Science, Mathematics, Physics, Electrical Engineering and Systems Science, and Quantitative Biology. Moving further the framework also predict the subdomain of research among top 16 growing domain of Computer Science along with the quality of the paper just by taking the abstract of paper as input. Considering the subdomain and the quality of paper it is easy to decide where to send the research work for the publication.

The proposed framework is consist of three machine learning model trained over large number of abstracts collected from arXive and S2ORC. For quality tagging of the abstract CORE ranking has been considered. The proposed frame work is predicting the subdomain and quality of paper only for Computer Science domain.

Chapter 2

Data Collection and Preprocessing

Data is the prerequisite to train any machine learning model to solve any task. Here the task is to predict the quality of the paper just seeing the abstract. And to train a particular machine learning model for this task, data set consists of abstract along with some quality tag is required.

2.1 Data Collection

Data is collected from two publicly available sources, namely arXiv[1](a digital archives) and The Semantic Scholar Open Research Corpus(S2ORC) [3].

Two sets of keywords are used to fetch the data from arXiv using the provided API. One set of the keywords is specific to common scientific research domain from Physics(ph), Electrical Engineering and Systems Science(eess), Mathematics(math),

[1] https://arxiv.org

Computer Science(cs) and Quantitative Biology(q-bio) and another set of keywords is specific to only Computer Science domain. While collecting the data for a specific keyword, latest 10,000 data are considered. Data collected from the arXiv contains nine fields namely unique *id* provided by arXiv, *title* of the paper, *abstract* , $journal\_ref$ if present, published *date*, *categories* (domain of research), *doi* (a URL for the resolved DOI to an external resource if present),*comment* of author if present, and $authors$.

Although S2ORC contains 81.1M English-language academic papers spanning many academic disciplines, for this project only a subset containing 6.1M data is downloaded.

2.2 Data Preprocessing and Annotation

Both of collected data sets are processed separately, and three different data sets are built for each of the task mentioned, i.e. prediction of main domain of research, followed by prediction of subdomain of research and prediction of quality of paper in Computer Science domain. Statistics of different datasets is tabulated in Table 2.1.

Firstly, duplicate data are removed from the collected data from arXiv. Then for creating the dataset, named **Main_Domain_DS**, for main domain prediction, data belongs to same domain are merged and relabelled with the main research domain, i.e. ph, $eess$, $math$, $q-bio$ and cs.

For subdomain prediction in domain of Computer Science, a separate data set is built by considering top 16 subdomain of Computer Science namely, 'Computation

Table 2.1: Statistics of different datasets

Dataset	#data	#label	words count		
			#max	#min	#average
Main_Domain_DS	82657	5	530	6	163
CS_Domain_DS	131678	16	552	7	173
Quality_CS_DS	520	4	540	29	168

and Language', 'Machine Learning', 'Computer Vision and Pattern Recognition', 'Artificial Intelligence', 'Cryptography and Security ', 'Networking and Internet Architecture', 'Distributed, Parallel, and Cluster Computing', 'Information Theory', 'Robotics', 'Information Retrieval', 'Databases', 'Human-Computer Interaction', 'Social and Information Networks', 'Data Structures and Algorithms', 'Software Engineering', 'Computers and Society'. The default label given by arXiv is used to build this dataset, named **CS_Domain_DS**.

A third data set named **Quality_CS_DS** is built for paper quality prediction. S2ORC dataset and arXiv data having reference are considered to build this dataset. CORE Conference Ranking[2] database is used as reference to give quality label to each abstract based on which conference or journal the paper got published. The data is labelled with A^*, A, B, and C considering A^* to be the high-grade publication and C the low-grade publication.

[2] https://www.core.edu.au/conference-portal

Chapter 3

Proposed Framework

A framework is proposed to predict the quality of the paper in Computer Science domain. This framework consists of a pipeline of three individual trained machine learning model. In this pipeline, first model predict the main domain of the research, i.e. Physics, Electrical Engineering and Systems Science, Mathematics, Computer Science and Quantitative Biology. After predicting the main domain if the paper belongs to Computer Science, the next two parallel models pipelined to first model predict the subdomain of research and the quality of the paper. The proposed pipelined model is shown in Fig. 3.1.

3.1 Model Description

Each model is trained to perform particular task, i.e. prediction of main research domain, prediction of subdomain in Computer Science research domain and prediction

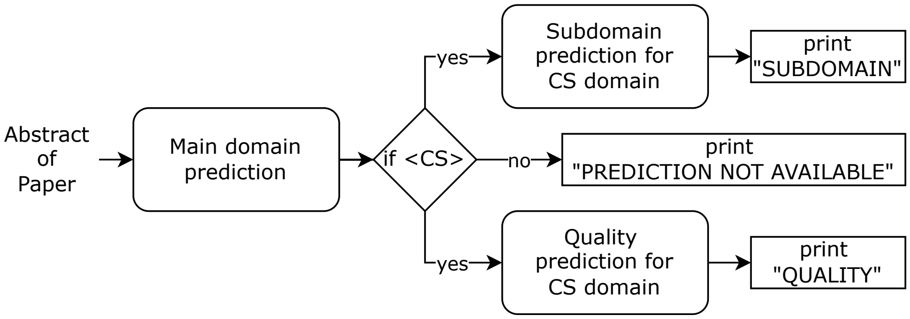

Abstract of Paper
Main domain prediction
if <CS>
yes
no
yes
Subdomain prediction for CS domain
print "SUBDOMAIN"
print "PREDICTION NOT AVAILABLE"
Quality prediction for CS domain
print "QUALITY"

Figure 3.1: Proposed framework for research domain prediction and paper quality prediction using the abstract of the paper

of quality of paper. Each of these three machine learning models is trained using the aforementioned datasets. Two deep learning model, one is based on Convolutional Neural Network (CNN) and another is based on Bidirectional Long Short-Term Memory (BiLSTM), are trained for each task. **Main_Domain_DS** dataset is used to train the model for predicting main research domain, **CS_Domain_DS** is used to train the model for predicting subdomain in Computer Science and **Quality_CS_DS** dataset is used to train the model for quality prediction.

3.1.1 Data Preprocessing

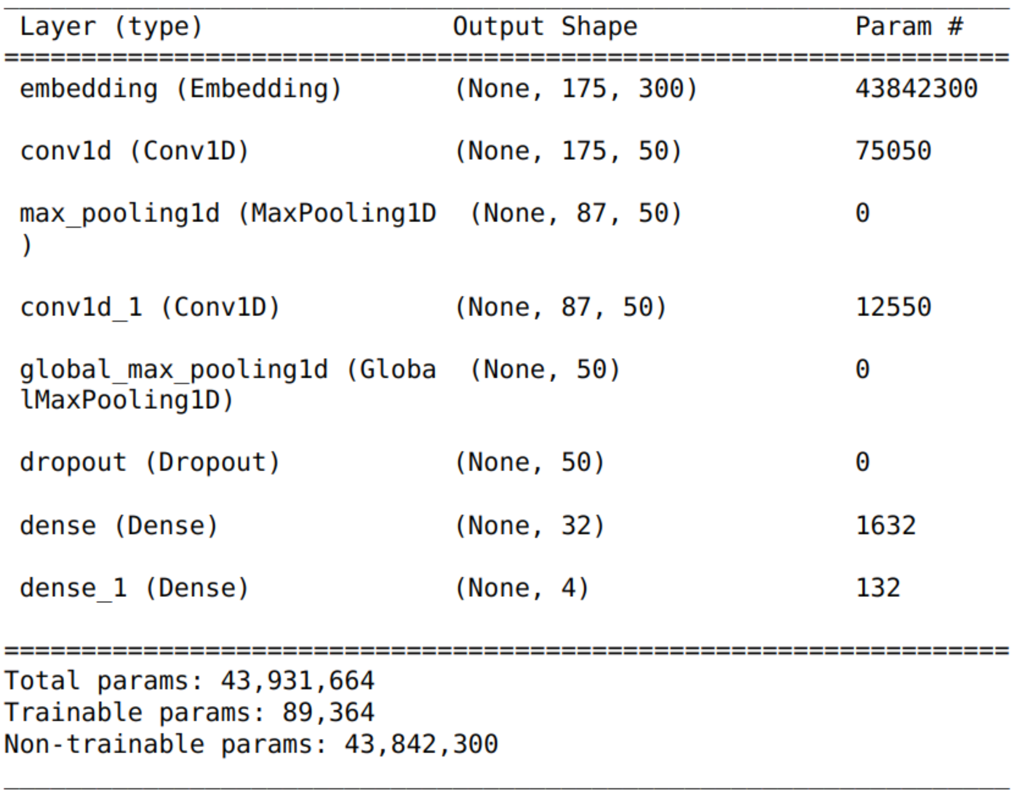
Each dataset is divided into Training and Testing set in the ratio of 8:2. Further, during training, 10% of Training set is kept reserved for validation. As while training CCN and BiLSTM required fixed size input, abstract of each data is converted to fixed size. It is found that more than 50% of data has abstract of length less than 170 words. Considering this analysis, a fixed size of 175 words are taken as input to train and test the model. Each data is need to be converted into some numerical form, i.e. in vectors before giving to any machine learning model. Pretrained FastText [4] embedding model trained for English language is used for this word embedding. Each word in a sentence is represented using a 300-D vector. Taking this, each input will be of [175×300]-D.

3.1.2 Model Architecture

Two deep learning model, one is based on CNN and another is based on BiLSTM, are trained for each task. The architecture for each model is described subsequently.

The CNN Model

Same model architecture is used to train CNN model for all three tasks taking input from different dataset . The output layer is also different for different task. Each architecture starts with Embedding layer to take input. Upon that Conv1D layer is added to perform convolution operation on the input vectors. Then MaxPooling1D is added to apply pooling operation over the output of Conv1D. One more Conv1D layer is added to the output of MaxPooling1D. Then a GlobalMaxPooling1D layer is added to make the the output one dimensional. To overcome the over-fitting while training a dropout layer is added to the output of GlobalMaxPooling1D. Upon this one dense layer of 32 node is added and finally another dense layer(number of node = number of classes) is added as output layer. Details of each layer along with number of parameter of model architecture to train CNN model for quality prediction is shown in Fig. 3.2. The model is compiled taking *binarycrossentropy* as loss function and *adam* as the optimizer. Same architecture(with different output layer) is used for training CNN model for other task also. Model parameters for different layer to train CNN model are tabulated in Table 3.1.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding (Embedding)       (None, 175, 300)          43842300  
                                                                 
 conv1d (Conv1D)             (None, 175, 50)           75050     
                                                                 
 max_pooling1d (MaxPooling1D  (None, 87, 50)           0         
 )                                                               
                                                                 
 conv1d_1 (Conv1D)           (None, 87, 50)            12550     
                                                                 
 global_max_pooling1d (Globa  (None, 50)               0         
 lMaxPooling1D)                                                  
                                                                 
 dropout (Dropout)           (None, 50)                0         
                                                                 
 dense (Dense)               (None, 32)                1632      
                                                                 
 dense_1 (Dense)             (None, 4)                 132       
                                                                 
=================================================================
Total params: 43,931,664
Trainable params: 89,364
Non-trainable params: 43,842,300
_________________________________________________________________
```

Figure 3.2: Model architecture to train CNN model for Quality Prediction

The BiLSTM Model

Like wise CNN same model architecture is used to train BiLSTM Model also for all three tasks taking input from different dataset . Only difference is in output layer. The node in output layer depends on the number of the classes for different task. Each architecture starts with Embedding layer to take input. Upon that Bidirectional layer is added and LSTM is given as the parameter input to this Bidirectional layer making the layer act as BiLSTM. Upon this Bidirectional layer a Dense layer is added. To overcome the over-fitting while training a dropout layer is added to the output of Dense layer. Upon this one dense layer of 64 node is added and fi-

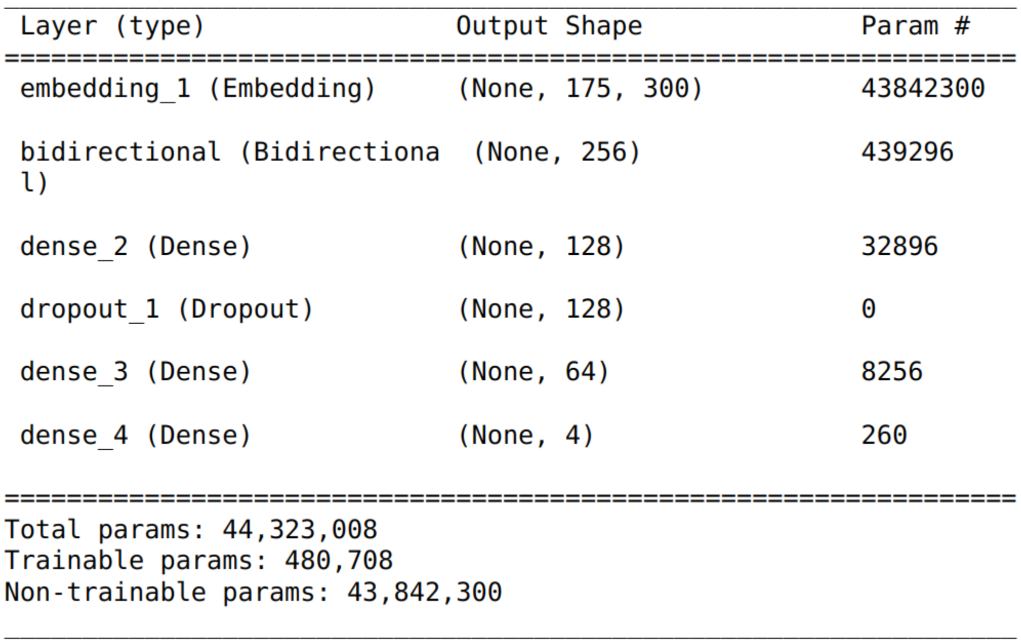
Table 3.1: Model parameters for different layer to train CNN model

layer	parameters	value
Conv1D	# filter	30
	kernel_size	5
	activation	relu
MaxPooling1D	pool_size	2
Dropout	rate	0.5
Output	units	#class
	activation	softmax

nally another dense layer(number of node = number of classes) is added as output layer. Details of each layer along with number of parameter of model architecture to train CNN model for quality prediction is shown in Fig. 3.3.The model is compiled taking *binarycrossentropy* as loss function and *adam* as the optimizer. Same architecture(with different output layer) is used for training BiLSTM model for other task also. Model parameters for different layer to train BiLSTM model are tabulated in Table 3.2.

Table 3.2: Model parameters for different layer to train BiLSTM model

layer	parameters	value
Bidirectional	layer	LSTM
LSTM	output_units	128
	activation	tanh
	recurrent_activation	sigmoid
	dropout	0.2
Dense_1	units	128
	activation	relu
Dropout	rate	0.5
Dense_2	units	64
	activation	relu
Output	units	#class
	activation	softmax



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding_1 (Embedding)     (None, 175, 300)          43842300  
                                                                 
 bidirectional (Bidirectiona  (None, 256)              439296    
 l)                                                              
                                                                 
 dense_2 (Dense)             (None, 128)               32896     
                                                                 
 dropout_1 (Dropout)         (None, 128)               0         
                                                                 
 dense_3 (Dense)             (None, 64)                8256      
                                                                 
 dense_4 (Dense)             (None, 4)                 260       
                                                                 
=================================================================
Total params: 44,323,008
Trainable params: 480,708
Non-trainable params: 43,842,300
_________________________________________________________________
```

Figure 3.3: Model architecture to train BiLSTM model for Quality Prediction

3.2 Model Training

Keras framework is used to compile and train the aforementioned models. The model is trained taking the appropriate dataset. The batch size for training is taken considering the size of the data set. Batch size of 256 is taken while training the CNN model and BiLSTM model for main domain prediction and subdomain prediction. In case of training the model for quality prediction a batch size of 32 is taken. Number of epoch is set to 100. Early stop technique is used to stop the training process just before the over-fitting. Training accuracy, validation accuracy and test accuracy are described in Chapter 4.

Chapter 4

Results and Evaluation

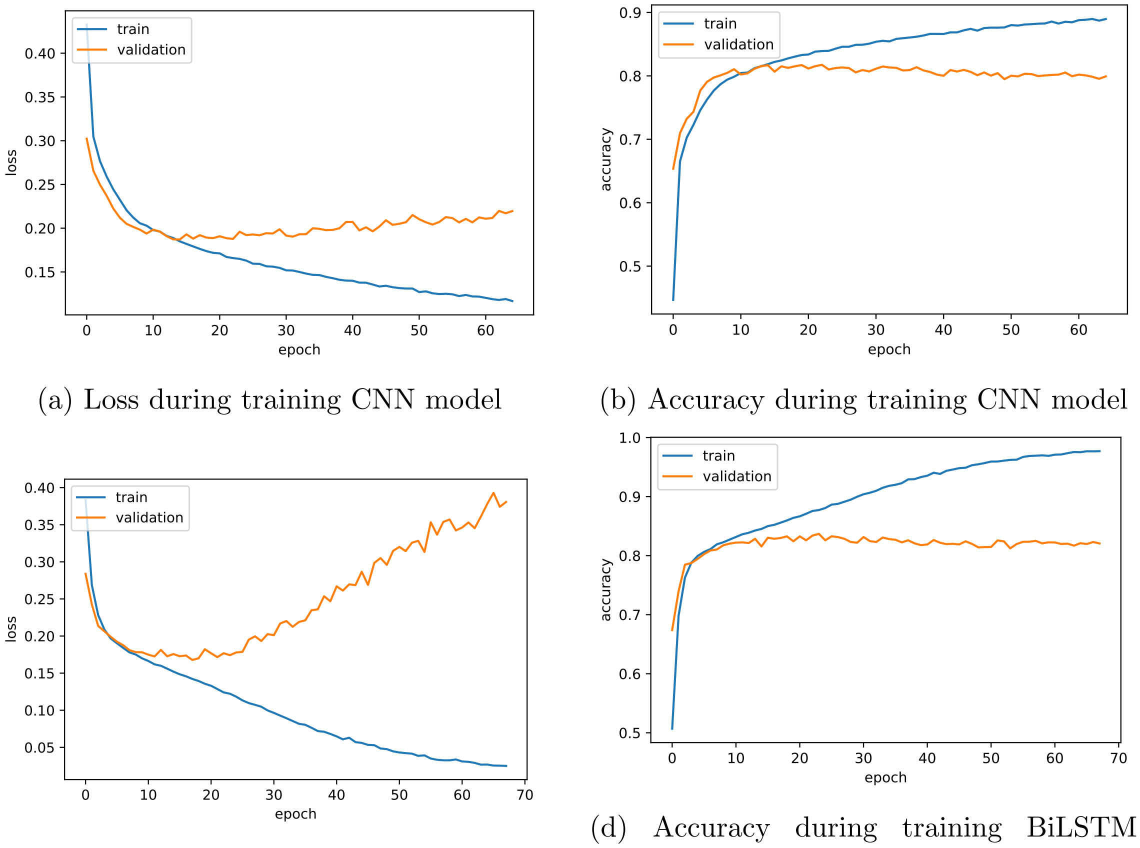
Taking all aforementioned dataset and the parameters tabulated in Tables 3.1 and 3.2, two model one based on CNN and another based on BiLSTM for each task are trained. Accuracy is taken as the performance evaluation metric for the different models while training and testing. Accuracy of both CNN and BiLSTM model during training and testing for different tasks are summarized in Table 4.1.

Table 4.1: Accuracy of CNN and BiLSTM model during training and testing for different tasks

Task	Model	Accuracy		
		Train	Validation	Test
main domain prediction	CNN	0.921	0.801	0.805
	BiLSTM	**0.978**	**0.820**	**0.816**
subdomain prediction	CNN	0.763	0.678	0.658
	BiLSTM	**0.922**	**0.756**	**0.734**
quality prediction	CNN	**0.954**	0.619	**0.644**
	BiLSTM	0.909	**0.630**	0.635

4.1 Main Domain Prediction

The **Main_Domain_DS** dataset is used to train the Main Domain Prediction model. The best training accuracy is obtained at 65^{th} (Fig. 4.1b) epoch while training the CNN model for this task and for BiLSTM model the best training accuracy achieved at 68^{th}(Fig. 4.1d) epoch. The losses during training the CNN model and BiLSTM model are represented in Fig. 4.1a and 4.1c respectively. It is evident from Table 4.1 that the BiLSTM model outperforms the CNN model.



(a) Loss during training CNN model

(b) Accuracy during training CNN model

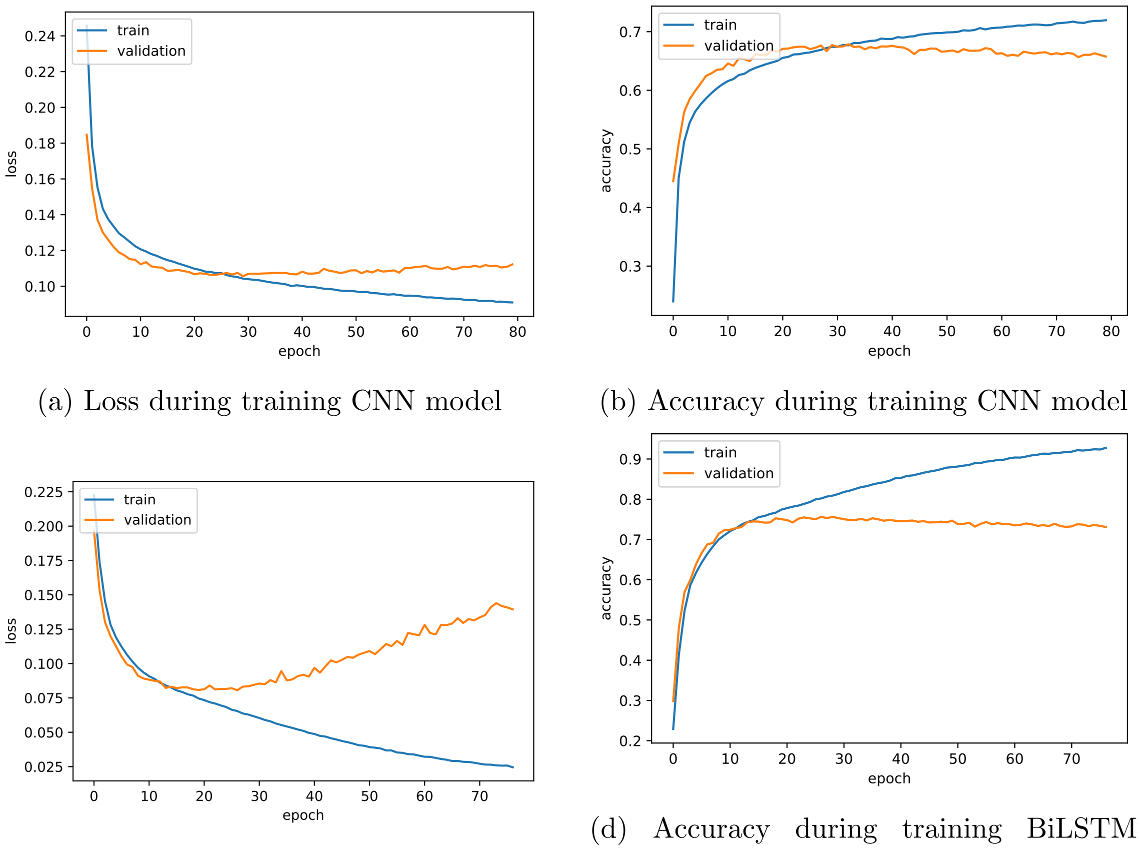
(c) Loss during training BiLSTM model

(d) Accuracy during training BiLSTM model

Figure 4.1: Loss and accuracy at different epoch during training the models for Main Domain Prediction

4.2 Subdomain Prediction

The **CS_Domain_DS** dataset is used to train the Subdomain Prediction model. The best training accuracy is obtained at 80^{th} (Fig. 4.2b) epoch while training the CNN model for this task and for BiLSTM model the best training accuracy achieved at 77^{th}(Fig. 4.2d) epoch. The losses during training the CNN model and BiLSTM model are represented in Fig. 4.2a and 4.2c respectively.



(a) Loss during training CNN model

(b) Accuracy during training CNN model

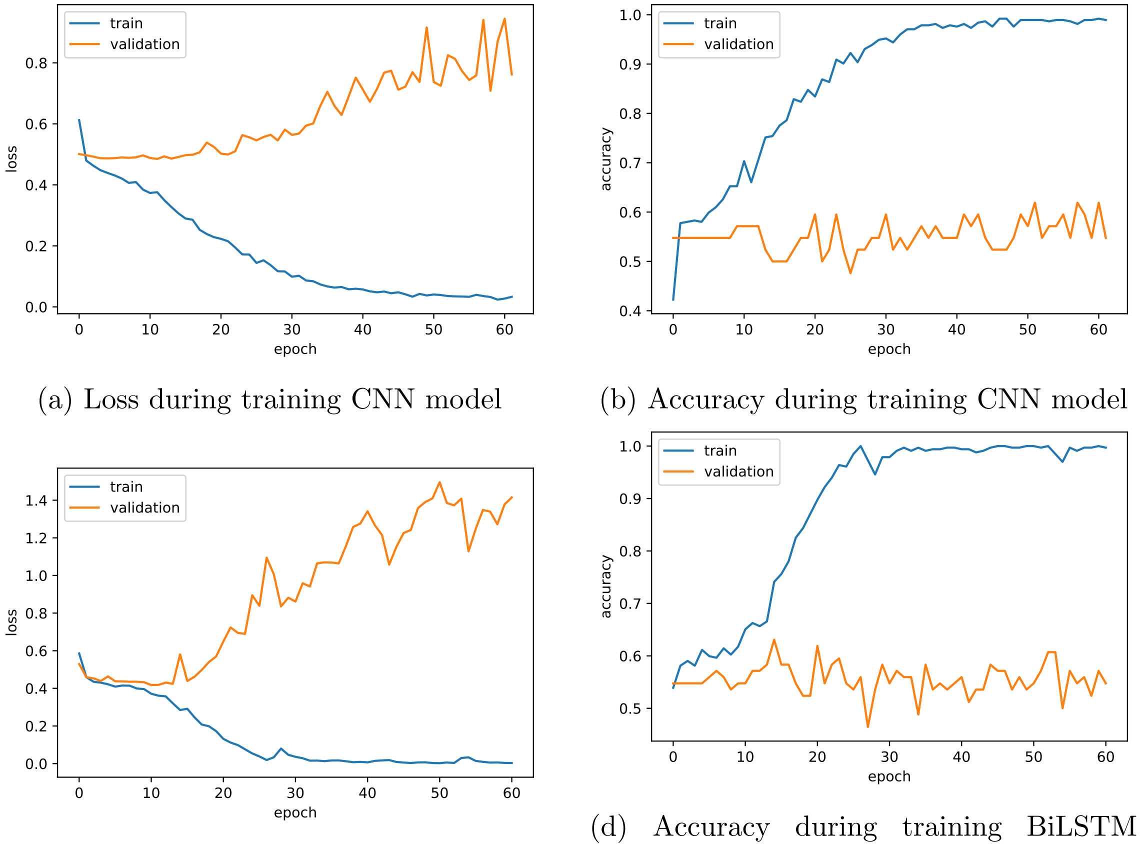
(c) Loss during training BiLSTM model

(d) Accuracy during training BiLSTM model

Figure 4.2: Loss and accuracy at different epoch during training the models for Subdomain Prediction

4.3 Quality Prediction

The **Quality_CS_DS** dataset is used to train the Quality Prediction model. Due to small size of the dataset, models for Quality Prediction are not trained properly. Losses and the accuracy during the model training are shown in Fig. 4.3. In this case the CNN model outperforms the BiLSTM model (Table 4.1).



(a) Loss during training CNN model

(b) Accuracy during training CNN model

(c) Loss during training BiLSTM model

(d) Accuracy during training BiLSTM model

Figure 4.3: Loss and accuracy at different epoch during training the models for Quality Prediction

Chapter 5

Conclusion and Future Work

A system is developed to predict the quality of paper from the abstract of the paper along with the the research sub domain prediction in Computer Science research domain. Three individual datasets are curated to train the CNN and BiLSTM model for each of the three modules of the system. Overall BiLSTM performs well in all the cases.

Particularly the **Quality_CS_DS** dataset is too small to train the quality prediction module properly. There is a scope to curate such dataset. There is also the scope to implement of transfer leraning concepts like zero-shot learning, one-shot learning, and few-shot learning to overcome the the lack of dataset. The use pre-trained models based on transformer [6] like BERT [2] and GPT [5] is going to improve the prediction accuracy as well. Moreover the system can be extended beyond the Computer Science domain.

Bibliography

[1] Saurabh Mishra Daniel Zhang et al. “The AI Index 2021 Annual Report”. In: *THE AI INDEX REPORT-Measuring trends in Artificial Intelligence*. AI Index Steering Committee, Human-Centered AI Institute, Stanford University, Stanford, CA: Stanford University, Mar. 2021. URL: https://aiindex.stanford.edu/ai-index-report-2021/.

[2] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[3] Kyle Lo et al. “S2ORC: The Semantic Scholar Open Research Corpus”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4969–4983. DOI: 10.18653/v1/2020.acl-main.447. URL: https://www.aclweb.org/anthology/2020.acl-main.447.

[4] Tomas Mikolov et al. “Advances in Pre-Training Distributed Word Representations”. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*. 2018.

[5] Alec Radford et al. “Improving language understanding by generative pre-training”. In: (2018).

[6] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).