MULTI MODAL SENTIMENT ANALYSIS USING ATTENTION

Nikhil Kala

Introduction

Sentiment Analysis can be defined as the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. A person's opinion or feelings are for the most part subjective and not facts. Which means to accurately analyse an individual's opinion or mood from a piece of text can be extremely difficult. With Sentiment Analysis from a text analytics point of view, we are essentially looking to get an understanding of the attitude of a writer with respect to a topic in a piece of text and its polarity; whether it's positive, or negative.

Sentimental Analysis is quite useful tool for understanding the user reviews towards certain product. For example, we can analyse whether the people liked a particular movie or not based on the sentiments in the reviews given by them. We can also apply sentimental analysis over the reviews of a newly launched headphone and know if its customers are satisfied with it or not. Similarly, if we want to start reading a new book, we can check if it is interesting or not by doing sentimental analysis of its review.

It becomes evident that the process of sentimental analysis depends upon the domain of the text that is being analysed. The words and their sentiments are not unique. They depend upon the context in which they are being used. If we use the word "cheap" it may be a positive word for a situation while it may be negative for another. This can be elucidated by the following example. "The movie can be said to be cheap and vulgar" is a negative sentence in the movie domain while "The smartphone is excellent and cheap" is a positive sentence in the electronics domain. The objective of this project is to apply sentimental analysis on multiple domains and identify the reviewer's opinion for it. This will allow an unbiased evaluation of the sentiments regarding products and services, thus allowing this model to be used ubiquitously. This is a mutually beneficial framework as the customer and the provider both will get to know how well the product is doing in the market. The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organisations across the world. The knowledge gained from the sentiments is then used for marketing of products and advertisement placement.

There are different ways to address the multi domain aspect of the sentiments. The most basic way is to train a different model for every domain. But this will require a lot of training dataset as well as time. There are many words that will inherently have a particular meaning and those words links these domains. Thus, it will be computationally costly and unwise to train different models for the domains.

There is a need of a system that uses the common features from these domains and also gives value to independent domain specific keywords depending upon their contexts. We have used sequence models rather than feedforward networks because the problem with these models is that they perform poorly when given a sequence of data. An example of sequence data is an audio clip which contains a sequence of spoken words. Another example would be a sentence in English which contains a sequence of words. Feedforward networks and CNN take a fixed length as input, but, when you look at sentences, not all are of the same length. You could overcome this issue by padding all the inputs to a fixed size. However, they would still perform worse than an RNN because those conventional models do not understand the context of the given input. This is where the major difference between sequence models and feedforward models lies. Given a sentence, when looking at a word, sequence models try to derive relations from the previous words in the same sentence. This is similar to how humans think as well. When we are reading a sentence, we don’t start from scratch every time we encounter a new word. We process each word based on the understanding of the previous words we have read.

So, the solution to this problem is a Domain-wise Attention Model. In this model, different domains share the same feature extraction layer to select the features which affect the sentiment the most. A representation encoded with domain knowledge is then used as attention to select the features that are related to a specific domain. These domain-related features include common features across all domains as well as domain-specific features for every single domain.

Methodology and Dataset

The Multi Modal Analysis Model for multi domain sentiment analysis is divided into 2 modules: -

1. Domain module
2. Sentiment module

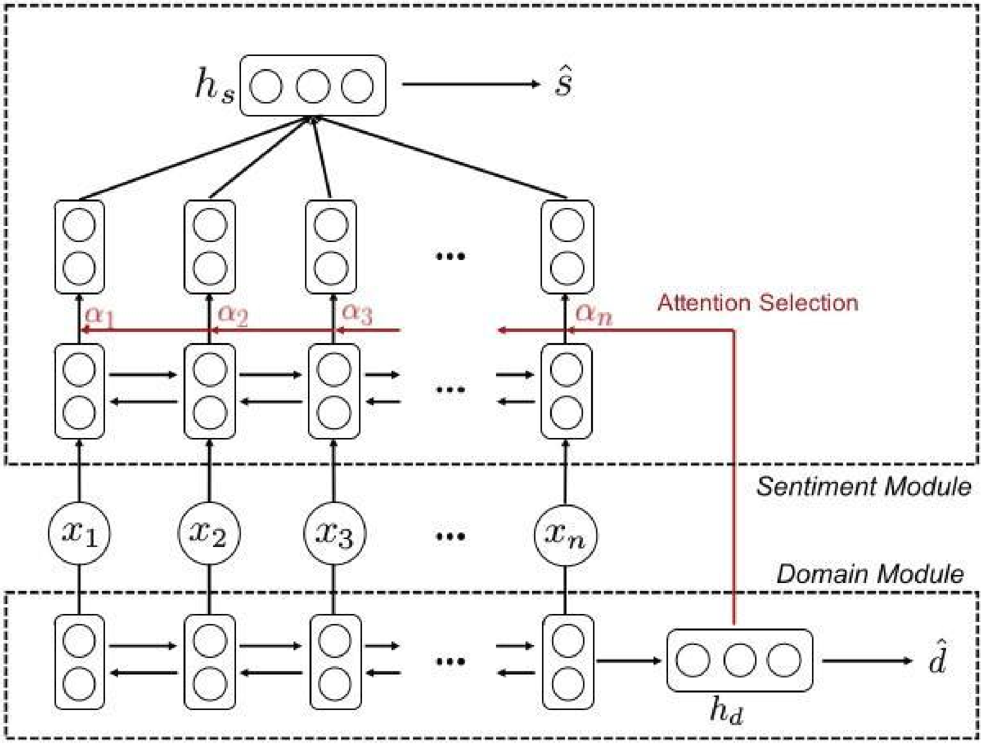


Fig 1 - Overview of a Multi Modal Analysis Model architecture

The domain module identifies the most probable domain to which a given text belongs such as movies, kitchen, book, electronics etc. And forwards the predicted domain to the sentiment module. This module tries to predict the positive or negative nature of the text.

The domain module in our Multi Modal Analysis Model is a recurrent neural network. The goal of domain module is to obtain a good domain representation for a given text. Fortunately, domain classification, or so-called topic classification, is a much simpler task than sentiment classification. Unlike sentiment which is always expressed subtly and semantically integrated, domain information can be easily acquired by the domain-related entities used within a text, such as actor in DVD domain and battery in Electronics domain.

Domain classes: -

1. Electronics
2. Movies (DVD)
3. Kitchen
4. Books

The sentiment module in our Multi Modal Analysis Model is another recurrent neural network network with attention mechanism. Different from the domain module, the sentiment module needs to attend all outputs during the recurrent process, which is also the key idea behind attention mechanism.

Sentiment classes: -

1. Positive
2. Negative

Word Embedding Used: -

GloVe - Global Vectors for Word Representation.

GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Dataset Used: -

Multi-Domain Sentiment Dataset (version 2.0)

The Multi-Domain Sentiment Dataset contains product reviews taken from Amazon.com from many product types (domains). Some domains (books and dvds) have hundreds of thousands of reviews. Others (musical instruments) have only a few hundred. Reviews contain star ratings (1 to 5 stars) that can be converted into binary labels if needed. This page contains some descriptions about the data.

We have used the subset of this data to train our model. The training dataset contains 1000 positively labelled reviews each under DVD, electronics, kitchen and books domain, and 1000 negatively labelled reviews each, under the same domains.

Data Loaded

- texts: 8000
- s_labels: 8000
- d_labels: 8000

maxlen: 461 n_words: 45589

We created several different models for these modules. These different models are shown as Experiments in the following section. These experiments lay out the architectures used and the accuracy obtained by each method.

EXPERIMENTATIONS – NETWORK ARCHITECTURE

- Experiment 1

Aim: Train the state of art model proposed by Yuan et al [8]. and obtain the metrics for comparison.

In this experiment there are two important modules, i.e., domain module and sentiment module. The domain module tries to predict which domain a text belongs to, through which a good domain representation will be learned. Then the domain representation triggers an attention selection of the most important domain-related features in the sentiment module.

In domain module, we use a bidirectional long-short term memory (BiLSTM) network to gain the domain representation. Using bidirectional LSTM runs the inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backwards you preserve information from the future and using the two hidden states combined you are able in any point in time to preserve information from both past and future.

For e.g.:

To predict the next word only by this context, with bidirectional LSTM:

Forward LSTM: The boys went to …

Backward LSTM: ... and then they got out of the pool

Using the information from the future it could be easier for the network to understand what the next word is.

The sentiment module in our DAM is another bidirectional LSTM network with attention mechanism. Different from the domain module, the sentiment module needs to attend all outputs during the recurrent process, which is also the key idea behind attention mechanism.

This model gave good accuracy across all the domains because Bidirectional LSTM can understand context better. The model gave very high accuracy especially in the books domain which is due to the context.

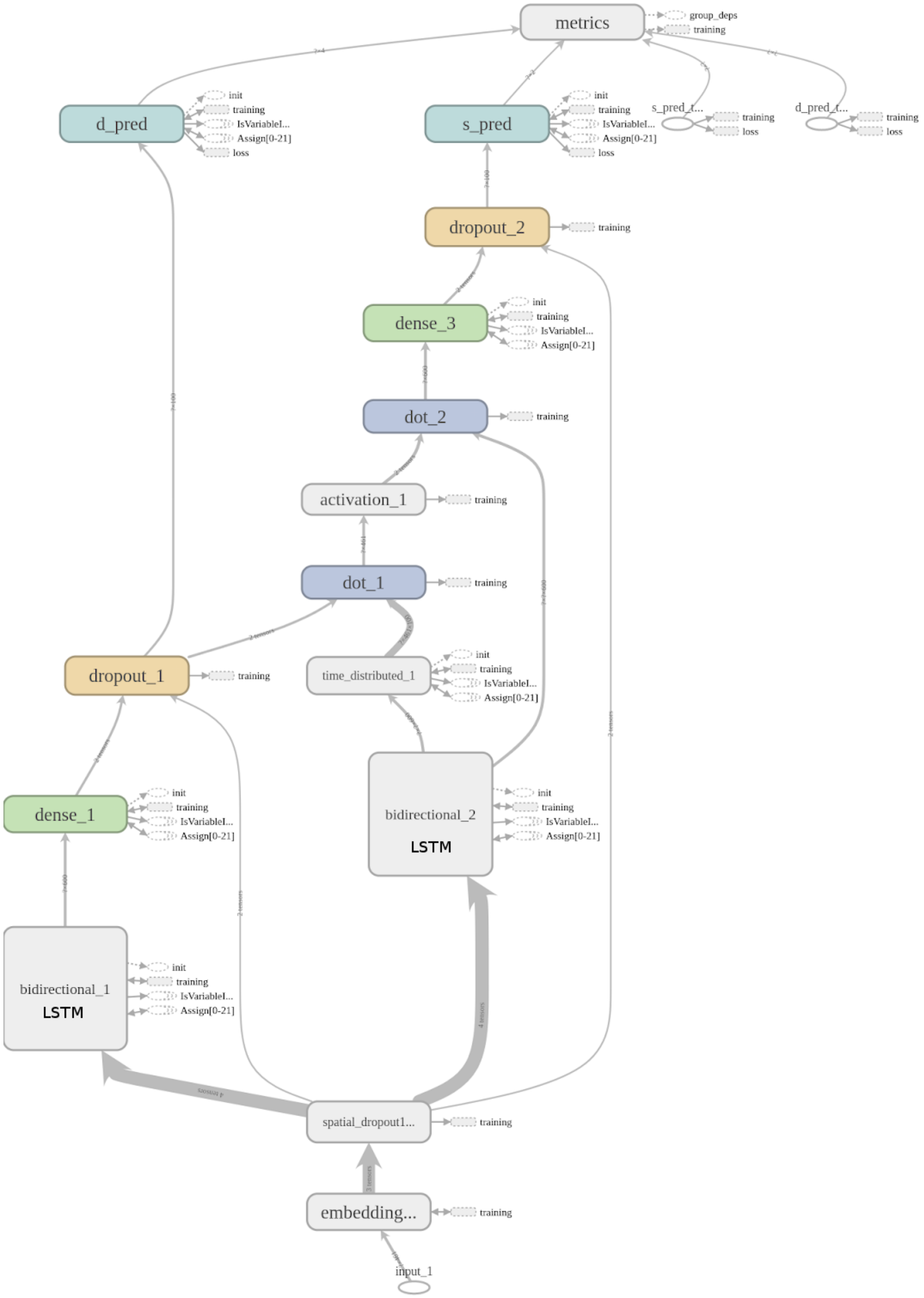


Fig 2 - Basic architecture of state-of-the-art model

Test evaluation:

Table 1 - Accuracy of state-of-the-art model

Domain	Accuracy
Books	0.8911
DVDs	0.8069
Electronics	0.8416
Kitchen	0.8564

Training Parameters:

Table 2 - Parameter values of state-of-the-art model

Parameter	Value
loss	0.1462
s_pred_loss	0.1305
d_pred_loss	0.3921
s_pred_acc	0.9550
d_pred_acc	0.8620
val_loss	0.4831
val_s_pred_loss	0.4688
val_d_pred_loss	0.3571
val_s_pred_acc	0.8492
val_d_pred_acc	0.8800

Training time per epoch: 171.37 seconds

- Experiment 2

Aim: Add a dense layer before the final layer of the sentiment module. The number of neurons is kept equal to the number of domains.

In this experiment there are two important modules, i.e., domain module and sentiment module. The domain module tries to predict which domain a text belongs to, through which a good domain representation will be learned. Then the domain representation triggers an attention selection of the most important domain-related features in the sentiment module.

A dense layer is a fully connected layer, as in, all neurons in the previous layer are connected to all neurons in the next layer. These are different from convolutional layers, since weights are reused across different sections of the vectors, whereas a dense layer has a unique weight for every neuron-to-neuron pair. This layer is called dense because its connections are “dense”. The dropout drops connections of neurons from the dense layer to prevent overfitting.

The sentiment module in our DAM is another bidirectional LSTM network with attention mechanism. Different from the domain module, the sentiment module needs to attend all outputs during the recurrent process, which is also the key idea behind attention mechanism.

The model gave high accuracy in the DVD domain because the dense layer allowed small reviews to get better results in their sentiment analysis.

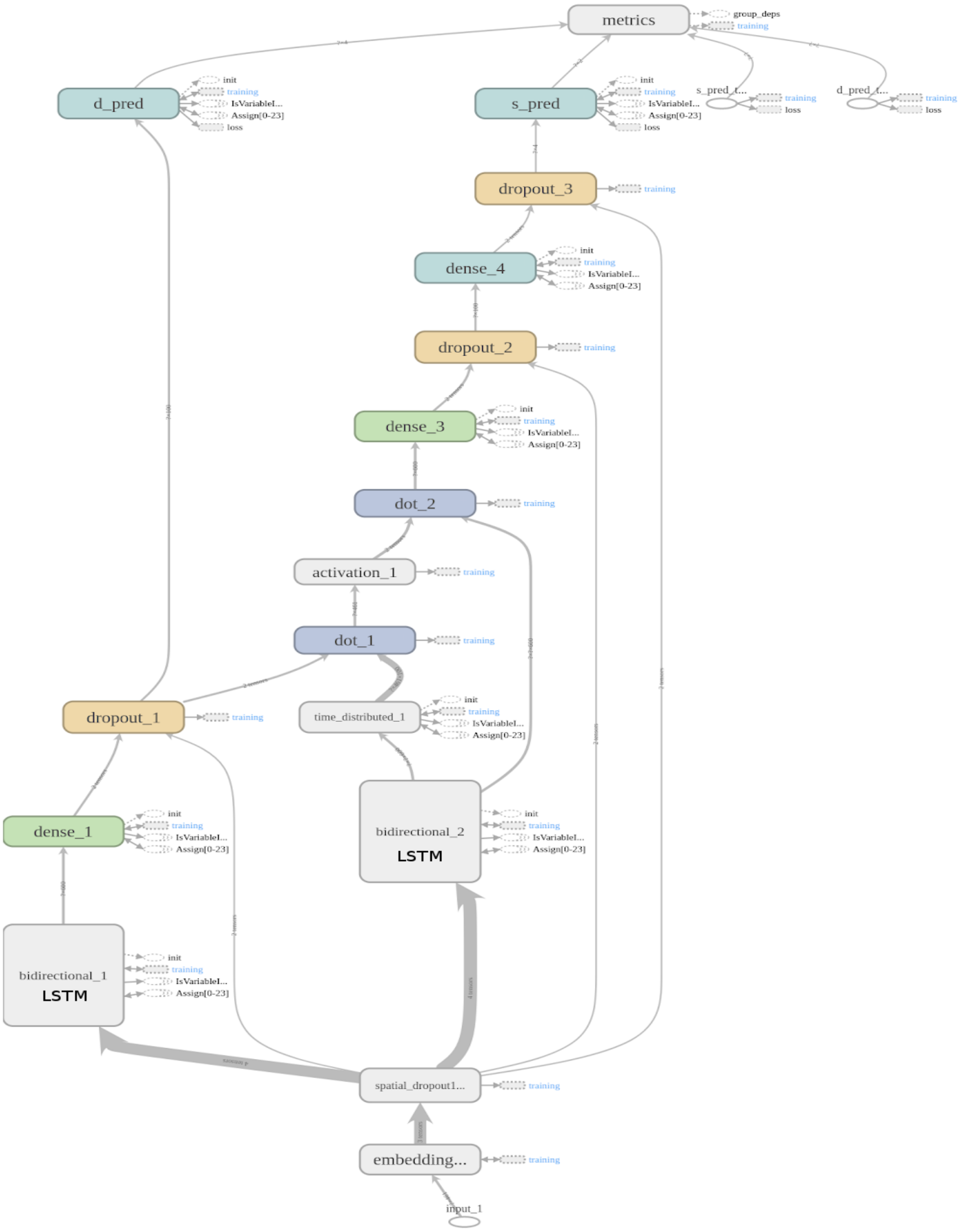


Fig 3 - DAM model with 1 dense layer added before the final layer of sentiment module

Test evaluation:

Table 3 - Accuracy of DAM model with 1 dense layer added

Domain	Accuracy
Books	0.8911
Dvds	0.8317
Electronics	0.8465
Kitchen	0.8762

Training Parameters:

Table 4 - Parameter values of DAM model with 1 dense layer added

Parameter	Value
loss	0.2057
s_pred_loss	0.1948
d_pred_loss	0.2736
s_pred_acc	0.9432
d_pred_acc	0.9100
val_loss	0.4294
val_s_pred_loss	0.4184
val_d_pred_loss	0.2740
val_s_pred_acc	0.8405
val_d_pred_acc	0.9014

Training time per epoch: 157.90 seconds

- Experiment 3

Aim: Use a combination of LSTM and GRU in the hidden layers. Use LSTM layer as the bidirectional RNN in the domain module and GRU in the sentiment module.

In this experiment there are two important modules, i.e., domain module and sentiment module. The domain module tries to predict which domain a text belongs to, through which a good domain representation will be learned. Then the domain representation triggers an attention selection of the most important domain-related features in the sentiment module.

In domain module, we use a LSTM network to gain the domain representation. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

The sentiment module in our DAM is bidirectional Gated recurrent unit (GRU) network with attention mechanism. To solve the vanishing gradient problem of a standard RNN, GRU uses, so called, update gate and reset gate. Basically, these are two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.Different from the domain module, the sentiment module needs to attend all outputs during the recurrent process, which is also the key idea behind attention mechanism.

This model gave good accuracy across all the domains because the modules worked well together. The model was very quick to train as well.

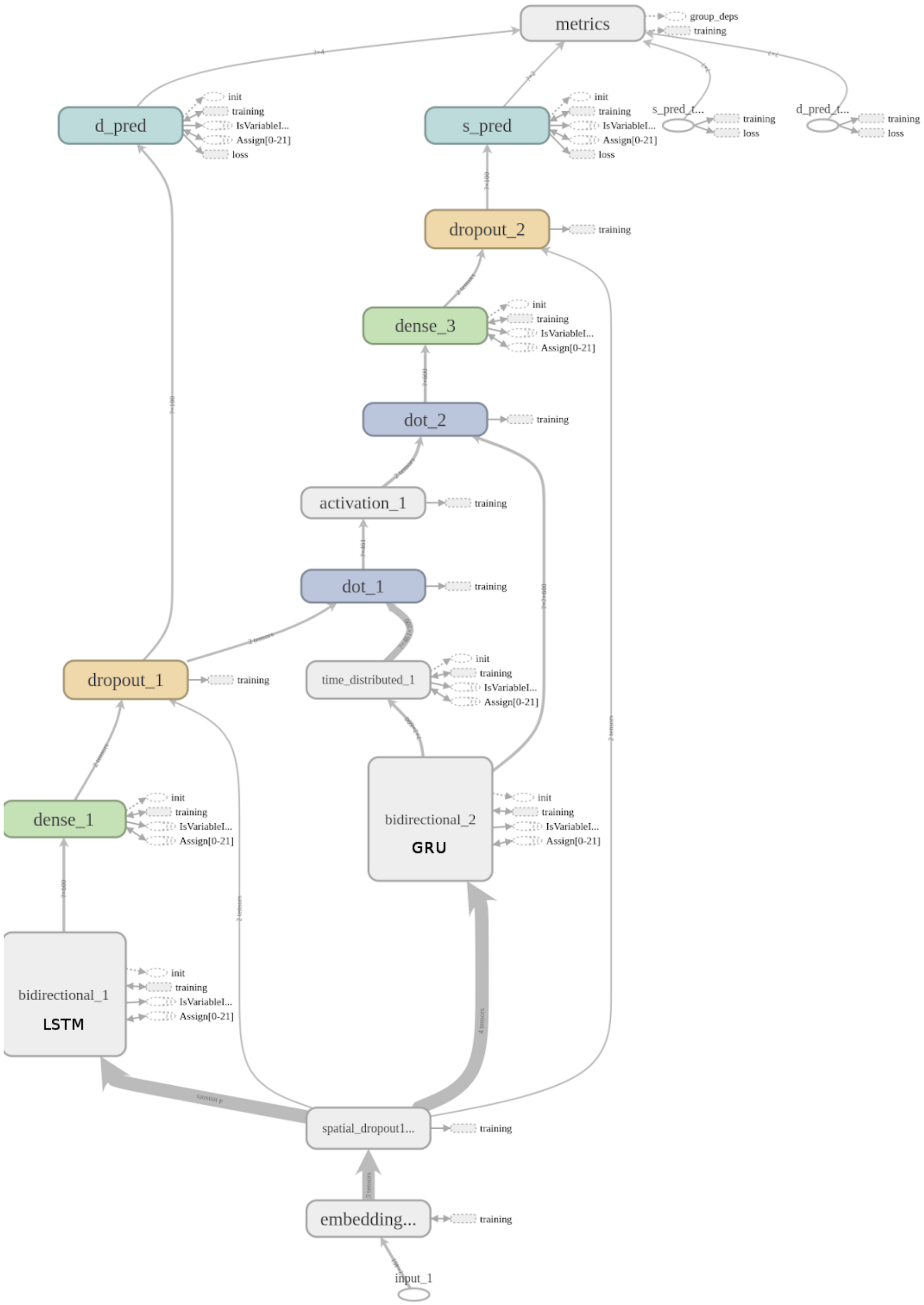


Fig 4 - DAM model with GRU layer in sentiment module

Test evaluation:

Table 5 - Accuracy of DAM model with GRU layer in sentiment module

Domain	Accuracy
Books	0.8911
Dvds	0.8020
Electronics	0.8218
Kitchen	0.8515

Training Parameters:

Table 6 - Parameter values of DAM model with GRU layer in sentiment module

Parameter	Value
loss	0.2092
s_pred_loss	0.1785
d_pred_loss	0.7671
s_pred_acc	0.9316
d_pred_acc	0.6791
val_loss	0.4722
val_s_pred_loss	0.4428
val_d_pred_loss	0.7346
val_s_pred_acc	0.8442
val_d_pred_acc	0.6979

Training time per epoch: 131.87 seconds

- Experiment 4

Aim: Use a combination of LSTM and GRU in the hidden layers. Use GRU layer as the bidirectional RNN in the domain module and LSTM in the sentiment module.

In this experiment there are two important modules, i.e., domain module and sentiment module. The domain module tries to predict which domain a text belongs to, through which a good domain representation will be learned. Then the domain representation triggers an attention selection of the most important domain-related features in the sentiment module.

In domain module, we use a GRU network to gain the domain representation. The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future. That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem.

The sentiment module in our DAM is LSTM network with attention mechanism. Different from the domain module, the sentiment module needs to attend all outputs during the recurrent process, which is also the key idea behind attention mechanism.

This model gave good accuracy across all the domains because the modules worked well together. The model was not as good when compared to other models.

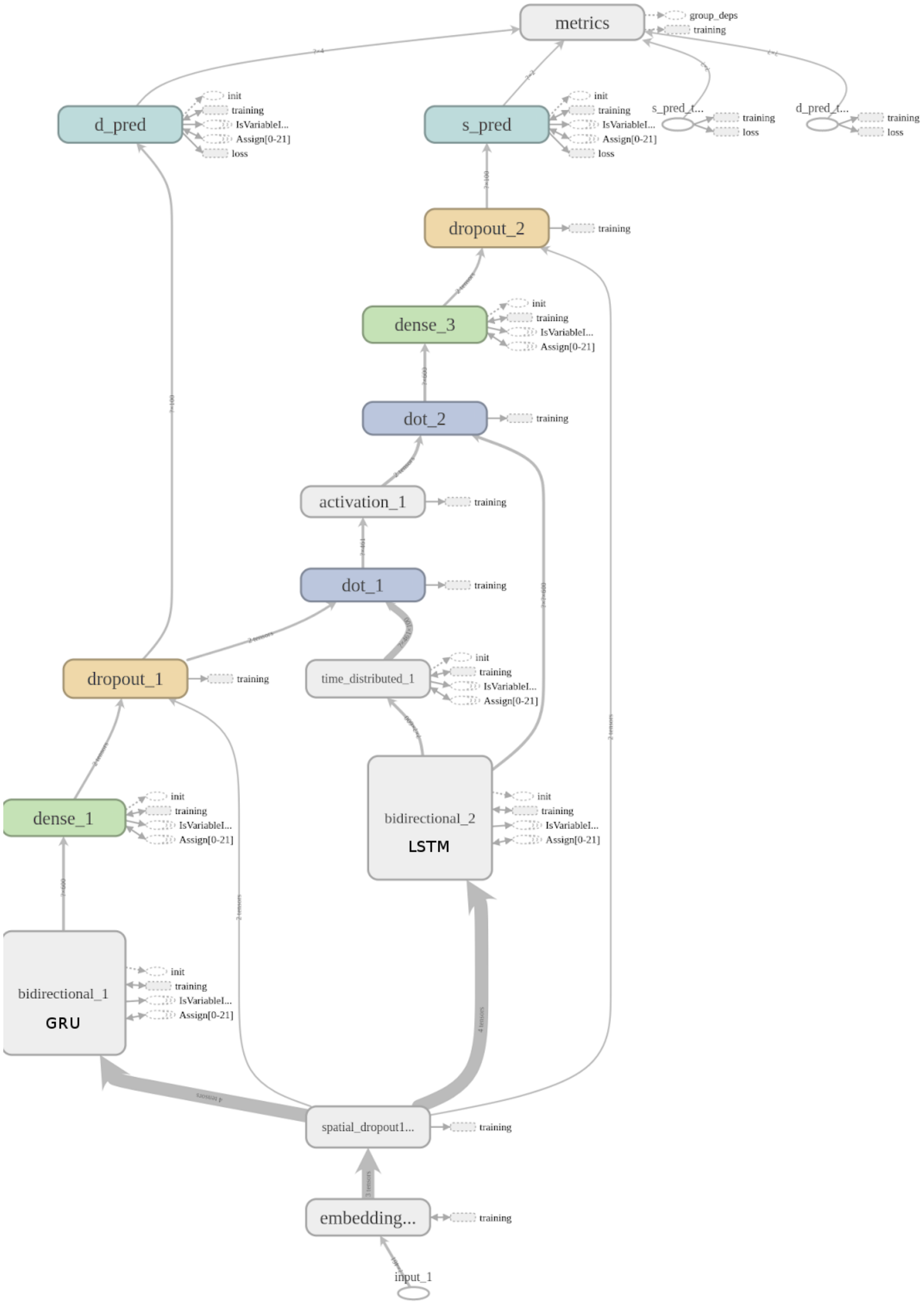


Fig 5 - DAM model with GRU layer in domain module

Test evaluation:

Table 7 - Accuracy of DAM model with GRU layer in domain module

Domain	Accuracy
Books	0.8861
Dvds	0.7921
Electronics	0.8267
Kitchen	0.8713

Training Parameters:

Table 8

Parameter	Value
loss	0.1383
s_pred_loss	0.1163
d_pred_loss	0.5494
s_pred_acc	0.9564
d_pred_acc	0.7932
val_loss	0.5500
val_s_pred_loss	0.5298
val_d_pred_loss	0.5058
val_s_pred_acc	0.8430
val_d_pred_acc	0.8059

Training time per epoch: 145.78 seconds

- Experiment 5

Aim: Use GRU layer as the bidirectional RNN in the domain module as well as in the sentiment module.

In this experiment there are two important modules, i.e., domain module and sentiment module. The domain module tries to predict which domain a text belongs to, through which a good domain representation will be learned. Then the domain representation triggers an attention selection of the most important domain-related features in the sentiment module.

In domain module, we use a GRU network to gain the domain representation. The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future. That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem.

The sentiment module in our DAM is again GRU network with attention mechanism. Different from the domain module, the sentiment module needs to attend all outputs during the recurrent process, which is also the key idea behind attention mechanism. GRUs have been shown to exhibit better performance on smaller datasets [9] and so was used.

This model gave good accuracy across all the domains because the modules worked well together. The model gave highest accuracy in most of the domains and had less training time as well.

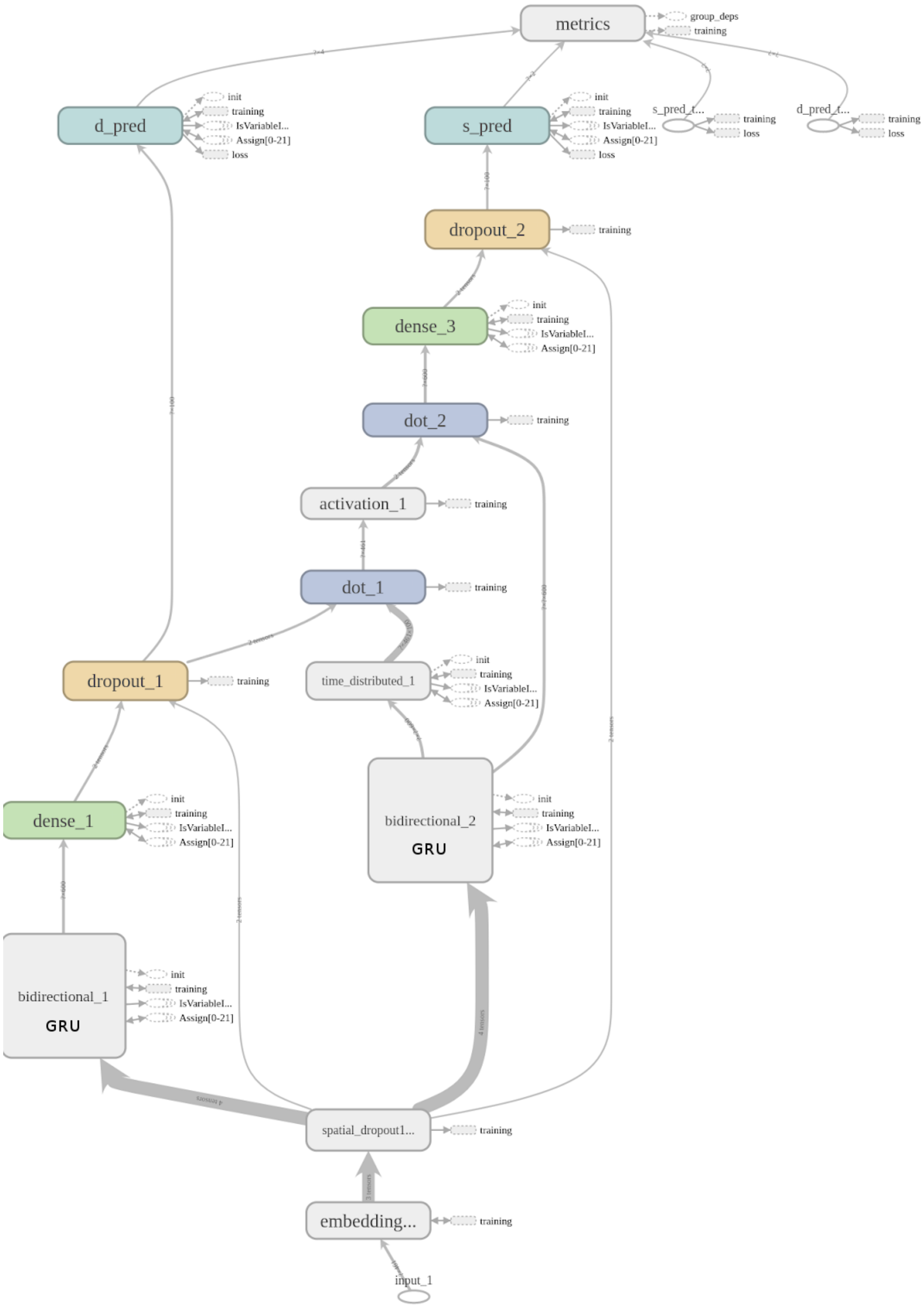


Fig 6 - DAM model with GRU layer in both modules

Test evaluation:

Table 9 - Accuracy of DAM model with GRU layer in both modules

Domain	Accuracy
Books	0.8812
Dvds	0.8416
Electronics	0.8515
Kitchen	0.8762

Training Parameters:

Table 10 - Parameter values of DAM model with GRU layer in both modules

Parameter	Value
loss	0.2028
s_pred_loss	0.1781
d_pred_loss	0.6180
s_pred_acc	0.9343
d_pred_acc	0.7475
val_loss	0.4485
val_s_pred_loss	0.4271
val_d_pred_loss	0.5358
val_s_pred_acc	0.8480
val_d_pred_acc	0.7990

Training time per epoch: 145.41 seconds

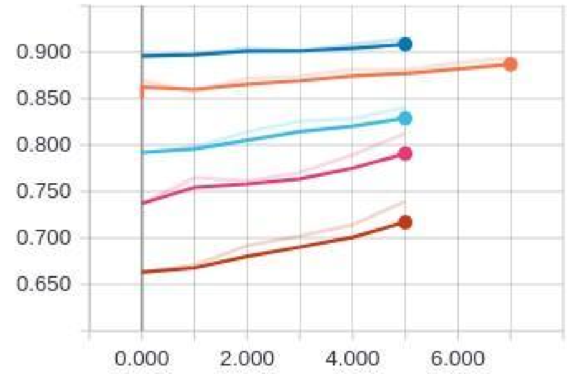
RESULTS

These graphs (Fig 7) show the various parameter values of all the experiments and compares them with each other.

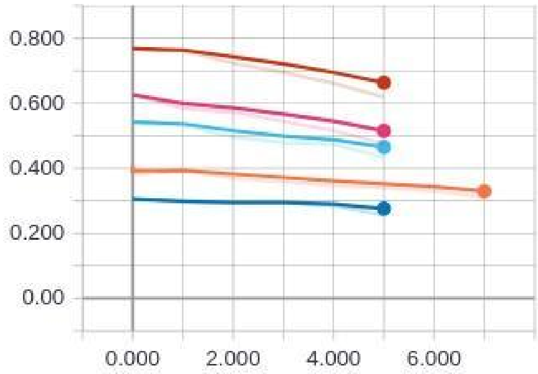
The parameters are listed below:

- **d_pred_acc :** The accuracy of the model to predict the domain. Accuracy is the fraction of predictions our model got right.
- **d_pred_loss:** The loss in the model to predict the domain. Loss is a number indicating how bad the model's prediction was on a single example.
- **loss:**The complete loss from both the domains. We use mean squared error for finding the loss in our models.
- **s_pred_acc :** The accuracy of the model to predict the sentiment.
- **s_pred_loss:** The loss in the model to predict the sentiment.
- **val_d_pred_acc :** The accuracy of the model to predict the domain during validation. Accuracy is the fraction of predictions our model got right.
- **val_d_pred_loss:** The loss in the model to predict the domain during validation. Loss is a number indicating how bad the model's prediction was on a single example.
- **val_loss:**The complete loss from both the domains during validation. We use mean squared error for finding the loss in our models.
- **val_s_pred_acc :** The accuracy of the model to predict the sentiment during validation.
- **val_s_pred_loss:** The loss in the model to predict the sentiment during validation.

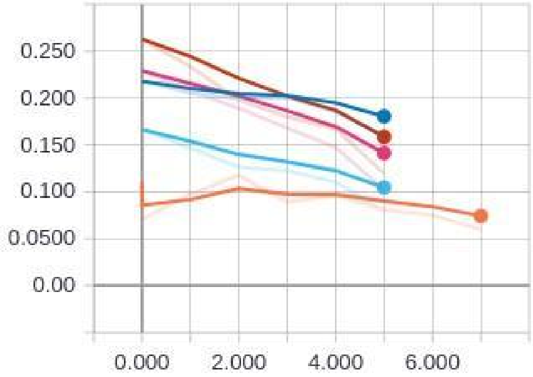
d_pred_acc



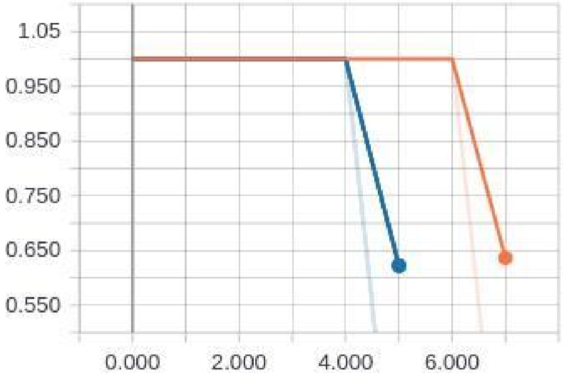
d_pred_loss



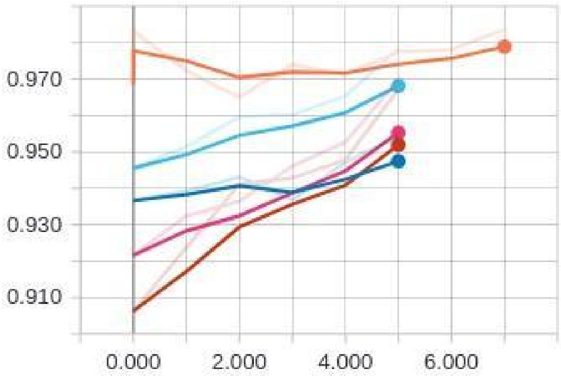
loss



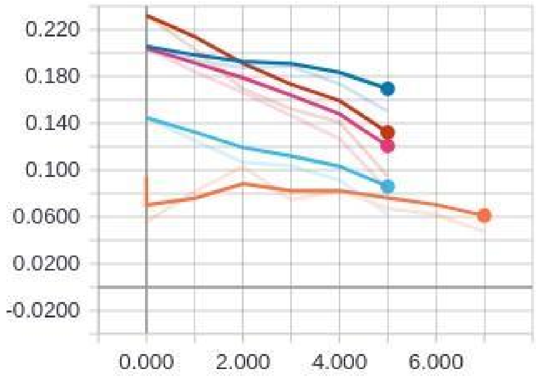
lr



s_pred_acc



s_pred_loss



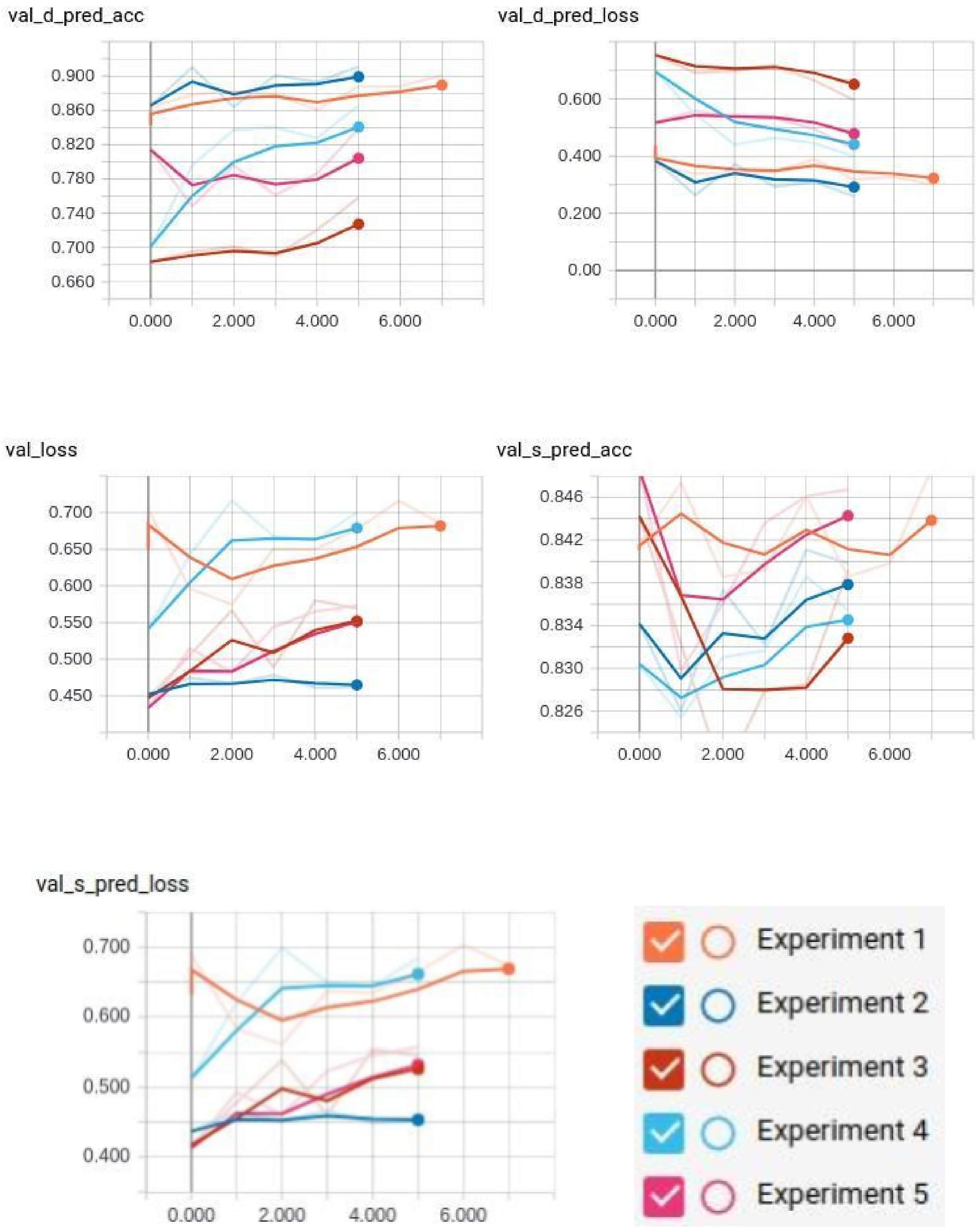


Fig 7 - Comparison of parameters after training

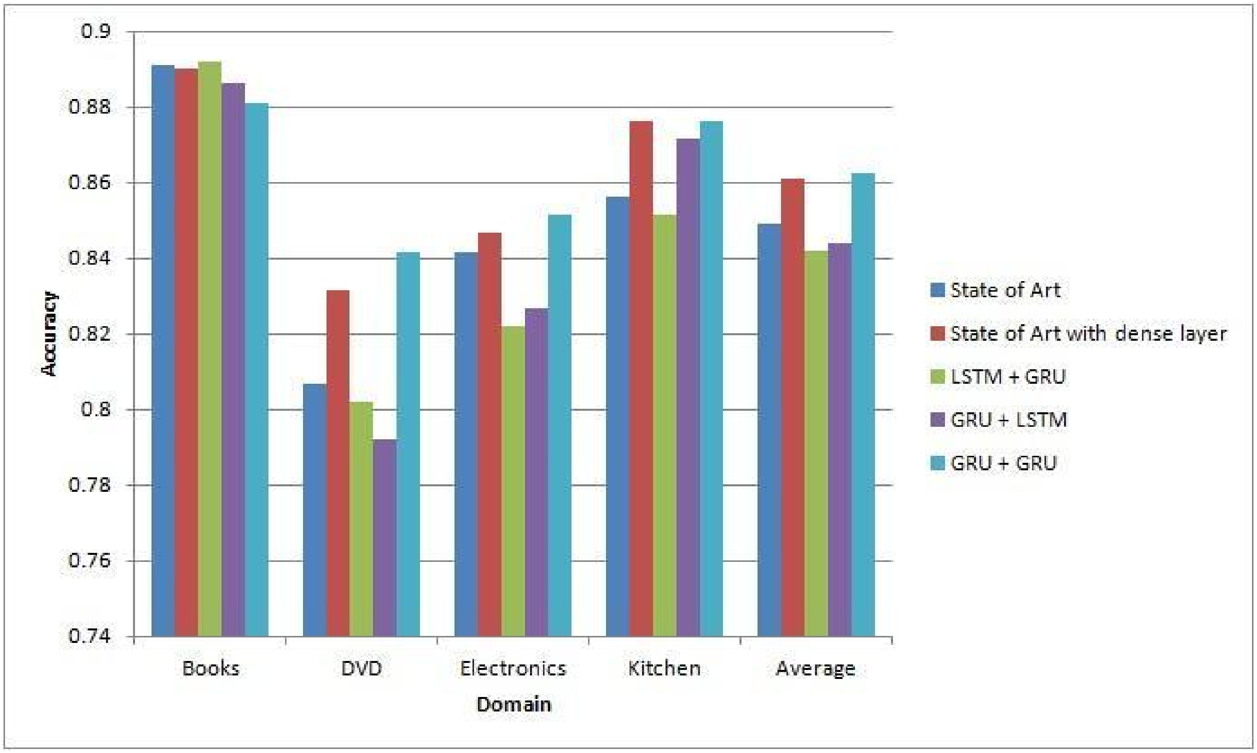


Fig 8 - Accuracy measures of each model

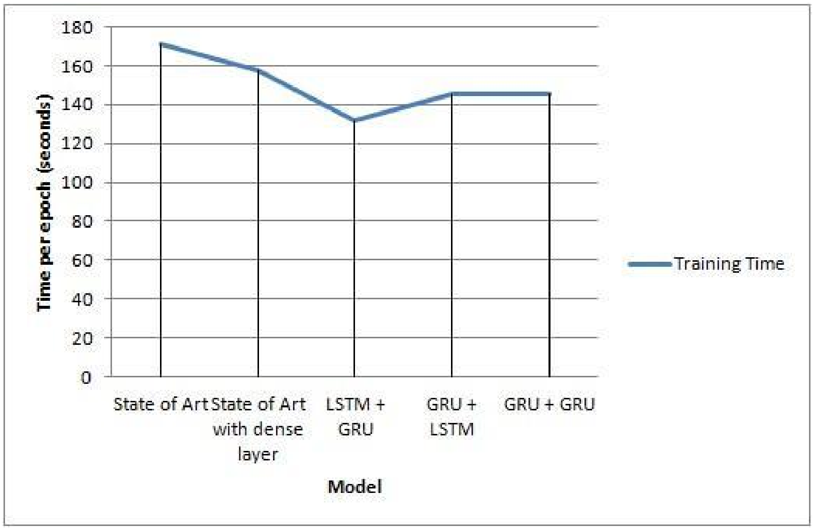


Fig 9 - Training time of each model

Table 11 - Accuracy Values and Training time for all models

Model Summary	Training time(seconds/epoch)	Accuracy
State of Art	171.3	**boo acc: 0.8911** dvd acc: 0.8069 ele acc: 0.8416 kit acc: 0.8564
State of Art with additional dense layer before final layer of sentiment module	157.9	**boo acc: 0.8911** dvd acc: 0.8317 ele acc: 0.8465 **kit acc: 0.8762**
LSTM + GRU	**131.8**	**boo acc: 0.8911** dvd acc: 0.8020 ele acc: 0.8218 kit acc: 0.8515
GRU + LSTM	145.7	boo acc: 0.8861 dvd acc: 0.7921 ele acc: 0.8267 kit acc: 0.8713
GRU + GRU	145.4	boo acc: 0.8812 **dvd acc: 0.8416** **ele acc: 0.8515** **kit acc: 0.8762**

The graphs show that the model with both the modules being GRU has good accuracy and also has less loss. The other models also perform well, especially the one proposed by Yuan et al[8] but the new model that we propose with the modules being GRU increases the accuracy of the model and it also takes less training time to train the dataset. The average accuracy (Fig. 8) shows the very good performance of our model.

FUTURE SCOPE

Thus, we found that the GRU layer as the bidirectional RNN in the Multi Modal Analysis Model gives efficient performance. It not only takes less training time than the state of the art, but gives higher validation accuracy in most of the domains. On the other hand, if we consider the validation accuracy in both the modules separately then state of art gives a higher accuracy in the domain module whereas our model in experiment 5 gives a higher validation accuracy in the sentiment module. Overall, we require our model to predict the sentiment accurately. Thus we can, to some extent, use the Multi Modal Analysis Model with GRU instead of LSTM as a bidirectional RNN layer gives a better performance. This can be attributed to the nature of GRU being more efficient for small datasets.

In future we can train the Multi Modal Analysis Model on more domains with a larger dataset. Further we can build an API over it and host it on a server so that users can use the model directly by passing the text and obtaining the corresponding sentiment prediction. This will eliminate the need for training and running the model on their machine. Therefore, even low-end devices will be able to use the model. This would be very helpful for the organisation who deal with products and services in multiple domains. For example, e-commerce companies will be able to use the same model for applying sentimental analysis on books, movies, electronics etc.

REFERENCES

1. Kim, Young-Bum, Karl Stratos, and Dongchan Kim. "Domain attention with an ensemble of experts." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. 2017
2. A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Netw. 18 (5–6) (2005). 602–610, http://dx.doi.org/10.1016/j.neunet.2005.06.042.
3. B. Pang, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25–30 June 2005, University of Michigan, USA, (2005), pp. 115–124. http://aclweb.org/anthology/P/P05/P05-1015.pdf
4. B. Chen, L. Zhu, D. Kifer, D. Lee, What is an opinion about? exploring political standpoints using opinion scoring model, Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11–15, 2010, (2010).
5. F. Wu, Y. Huang, Collaborative multi-domain sentiment classification, 2015 IEEE International Conference on Data Mining, ICDM 2015, Atlantic City, NJ, USA, November 14–17, 2015, (2015), pp. 459–468, http://dx.doi.org/10.1109/ICDM. 2015.68.
6. F. Wu, Z. Yuan, Y. Huang, Collaboratively training sentiment classifiers for multiple domains, IEEE Trans. Knowl. Data Eng. 29 (7) (2017) 1370–1383, http://dx.doi.org/10.1109/TKDE.2017.2669975.
7. J. Zhou, J. Chen, J. Ye, Malsar: Multi-task learning via structural regularization, volume 21, 2011.
8. Zhigang Yuan, Sixing Wu, Fangzhao Wu, Junxin Liu, Yongfeng Huang: Domain attention model for multi-domain sentiment classification. Knowl.-Based Syst. 155: 1-10 (2018)
9. Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).