ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Multi-Task Learning with Fusion: Framework for Handling Similar and Dissimilar Tasks

Pritam Pal[1], Shankha Shubhra Das[2], Dipankar Das[3] and Anup Kumar Kolya[4]

ABSTRACT

Multi-Task learning (MTL), which emerged as a powerful concept in the era of machine learning
deep learning, employs a shared model trained to handle multiple tasks at simultaneously. Numerous advantages of this novel approach inspire us to instigate the insights of various tasks with similar (Identification of Sentiment, Sarcasm, Hate speech, Offensive language, etc.) and dissimilar (Identification of Sentiment, Claim, Language) genres. This paper proposes two Multi-Task Learning (MTL) framework schemes based on Bidirectional LSTM (BiLSTM) to handle both similar and dissimilar tasks. The performance of these frameworks is evaluated and compared against standalone classifiers, demonstrating their effectiveness in improving classification accuracy. In order to train our proposed MTL frameworks, different task-related publicly available datasets were collected, and each sentence was annotated with all task labels with the help of publicly available pre-trained models. Along with a simple MTL framework, this paper presents an MTL framework with a fusion technique (MTL$_{fusion}$) that combines learning from task-specific layers to make predictions. Our proposed MTLfusion framework provides an F1 score of 0.76, 0.92, 0.809, 0.798, and 0.89 for sentiment, sarcasm, irony, hate speech, and offensive language classification tasks, respectively (similar tasks). It also provides an F1 score of 0.59, 0.586, and 0.707 for claim, sentiment, and language identification tasks, respectively. Our research also shows that MTL frameworks perform better than their corresponding standalone classifiers for similar tasks. On the other hand, for dissimilar tasks, the standalone classifiers perform better than MTL frameworks.

Article information:

Keywords: Multi-Task Learning, Deep Learning, Text Classification, Sentiment Analysis, Claim Detection

Article history:
Received: October 1, 2024
Revised: November 28, 2024
Accepted: February 20, 2025
Published: March 15, 2025
(Online)

DOI: 10.37936/ecti-cit.2025192.258581

1. INTRODUCTION

The popularity of the internet and social media platforms such as Facebook and Twitter allow people to express their feelings and opinions on various topics. Sometimes, these social media posts may contain negative or positive sentiments; they may be sarcastic or ironic. At the same time, they may contain some hateful speeches and offensive language.

The increasing advancement of Artificial Intelligence (AI) and Natural Language Processing (NLP) helps researchers to achieve state-of-the-art performance in different NLP tasks such as sentiment analysis, sarcasm detection, claim identification, detection of hate speech offensive languages in social media, etc. However, the majority of tasks focused on a single task i.e., either sentiment analysis or sarcasm detection. Furthermore, detecting both sentiment and sarcasm in the same text requires running two separate models, which may increase computational overhead. Although in the last couple of years, some researchers focused on combining two or three tasks, such as sentiment and sarcasm classification [1] [2], or hate-speech and offensive-language analysis [3] etc.

In this present article, two schemes of Multi-Task learning (MTL) frameworks are proposed: First, an

[1,4]The authors are with the RCC Institute of Information Technology, Kolkata, India , Email: pritampal522@gmail.com and anup.kolya@gmail.com
[2,3]The authors are with the Jadavpur University, Kolkata, India, Email: shankhasdas07@gmail.com and dipankar.dipnil2005@gmail.com

MTL framework that classifies five similar tasks of similar genre: sentiment, sarcasm, irony, hate speech, and offensive language, and second, an MTL framework working on relatively dissimilar tasks: claim detection, language identification, and sentiment analysis.

Multi-Task learning (MTL) as the name suggests, refers to a single shared machine-learning framework capable of learning and handling multiple different tasks simultaneously [4]. MTL provides several advantages over single-task learning: (1) it helps in achieving generalization for multiple tasks; (2) each task improves its performance in association with the other participating tasks; and (3) it offers reduced complexity because a single system can handle multiple tasks at the same time [5]. Additionally, in the context of sustainability issues, MTL frameworks are more sustainable than the standalone classifier or single-task learning (STL) frameworks.

The main objectives of this work are: (1) analyze whether adding different classification tasks (similar or dissimilar) into an MTL model can improve the overall performance of each classification over a single-task classifier or not; (2) identify how a task can gain knowledge in MTL environment with respect to the tasks of similar and dissimilar genres. Moreover, we explored various task combinations in MTL, such as sarcasm and hate speech classification, as well as claim and sentiment classification, to analyze performance in different MTL scenarios.

The contribution of this paper can be summarized as follows:

- We prepared a dataset of around 2K samples for similar tasks annotated with five labels in each sentence and 4K samples for dissimilar tasks to train the proposed MTL models.
- We proposed two schemes of MTL framework for both similar and dissimilar tasks: one is a simple MTL framework and another is a fusion-based MTL framework, and observed that the fusion-based MTL performs better than the simple MTL framework.

In Section 2, some previous NLP-related MTL works are discussed briefly. Section 3 is the data preparation section, where the data collection from different sources and preparation of the final dataset for MTL are discussed. In Section 4, the methodologies and MTL frameworks for similar and dissimilar tasks are proposed. Section 5 is the experiment and result section, where the results and performances in different MTL frameworks are discussed. In Section 6, a few error cases produced by our proposed frameworks are presented and analyzed. In Section 7, we discuss some insightful observations from our experiment. Section 8 concludes the paper and Section 9 discusses the limitations and future directions of this work.

2. RELATED WORK

In 1997, Caruana [6] proposed the concept of MTL which is a method of inductive transfer technique whose primary goal is to enhance the generalization performance by using domain knowledge from the training data of related tasks as an inductive bias [6]. Since then, MTL approaches have been extensively used across various fields of computer science and NLP to address complex problems. Furthermore, in 2017, Ruder [7] proposed two different schemes of MTL in their paper: the hard parameter-sharing approach and the soft parameter-sharing approach. In the hard parameter sharing approach, all tasks share the same hidden layers except the final output layers, and in the soft parameter sharing approach, each task has its own layers and parameters [7].

Liu *et al.* [8] proposed an LSTM-based MTL framework for text classification. The authors developed three MTL frameworks where two models were developed using unidirectional LSTM and the third model was developed using bidirectional LSTM (BiLSTM) with two classification heads in each model. The authors evaluated their proposed MTL models for some popular NLP benchmark datasets such as 'SST-1', 'SST-2', 'SUBJ' and 'IMBD', and their proposed MTL models perform better than the single-task learning models for those datasets.

Liu *et al.* [9] developed an adversarial MTL framework, introducing three MTL schemes primarily utilizing LSTMs. The authors evaluated their proposed models across 16 datasets related to Amazon and movie reviews. Among these, the adversarial network-based architecture demonstrated superior performance compared to all other proposed models.

Majumdar *et al.* [2], Tan et al. [1], and Savini et al. [10] proposed MTL models for sentiment and sarcasm analysis. The authors in [2] used a GRU-based architecture and attention mechanism to classify sentiment and sarcasm, whereas the authors in [1] and [10] used BiLSTM architecture in their study. In addition, [10] used a non-contextual pre-trained FastText [11] embedding, whereas [2] and [1] used pre-trained GloVe [12] and Word2Vec embeddings, respectively. Another sentiment and sarcasm analysis MTL framework was proposed by Mahdaouy et al. [13] for the Arabic language using the transformer-based pre-trained BERT [14] model.

Singh *et al.* [15] proposed an MTL architecture for sentiment, emotion and emoji analysis by using the pre-trained transformer-based model 'xlm-RoBERTa-base' [16]. In this study, the authors also showed that their MTL classifiers provide better performances with respect to standalone classifiers. Along with the classification tasks, the authors also analyze sentiment and emotion intensities in their study.

An MTL model was developed by Plaza-del-Arco *et al.* [3] for analysing sentiment, emotion, target (focusing on a particular community such as black people, women, LGBT etc.), hate speech and offensive language utilizing the BERT [14] model.

A transformer-based MTL model for the classification of text was developed by Huang *et al.* [17], and their suggested architecture produced a state-of-the-art model for the 'GLUE' dataset.

In our present work, we focused on developing an MTL framework for five similar types of classification tasks and three comparatively dissimilar types of classification tasks utilizing BiLSTM and pre-trained GloVe embedding.

3. DATASET PREPARATION

3.1 Similar Task

To the best of our knowledge, no public datasets are available on the internet that are annotated with all the sentiment, sarcasm, irony, hate, and offensive labels. Therefore, different task-related publicly available datasets were collected from popular websites such as Kaggle[1] and HuggingFace[2] with at least one class label of sentiment, sarcasm, irony, hate-speech, or offensive language to prepare a dataset for learning five similar tasks. The majority of data were extracted from the 'TweetEval' [18] dataset for the sentiment, irony, hate speech, and offensive language task. Another sentiment dataset named 'Twitter US Airline Sentiment'[3] was also considered that contains the positive, negative and neutral tweets related to US airlines.

For sarcasm, the 'News Headlines Dataset For Sarcasm Detection' [19] [20], 'Multimodal Sarcasm Detection Dataset (MUStARD)' [21] and the 'Eye-tracking and Sentiment Analysis II' [22] datasets were collected. Although the 'Eye-tracking and Sentiment Analysis II' dataset was annotated with sentiment and sarcasm class labels, only the sarcasm class labels were considered as the sentiment class was annotated with two labels (positive and negative), and in our work, three sentiment class labels (positive, negative and neutral) were taken into account.

Besides that, some other datasets were collected from Kaggle, such as the 'Hate Speech and Offensive Language Dataset'[4] for hate speech and offensive language, and 'Tweets with Sarcasm and Irony' [23] for sarcasm and irony data. In the 'Hate Speech and Offensive Language Dataset' there were three classes: hate speech, offensive language and neither (no hate or offensive language). In our work, only the sentences with hate speech and offensive language class labels were considered, as those data were relevant to preparing the dataset. Similarly, in the "Tweets with Sarcasm and Irony" dataset, only the sarcasm, irony, and figurative (both irony and sarcasm) class labels were considered.

[1]https://www.kaggle.com/
[2]https://huggingface.co/
[3]https://bit.ly/twitter-airline-sentiment

3.2 Dissimilar Task

The datasets used by the authors of [24] in their paper were considered for the dissimilar tasks. These datasets contain sentences from LiveJournal weblogs and Wikipedia talk pages annotated for opinionated claims. In these datasets, there were 2190 instances, from LiveJournal and 2197 from Wikipedia. It contains the labels for claim detection (Yes or No) and sentiment annotation for all the texts.

Another dataset was collected for the dissimilar tasks, which was a pre-processed version of WiLI-2018[5], the Wikipedia language identification benchmark dataset. It contains 22 selective languages (English, Arabic, French, Hindi, Urdu, Portuguese, Persian, Pushto, Spanish, Korean, Tamil, Turkish, Estonian, Russian, Romanian, Chinese, Swedish, Latin, Indonesian, Dutch, Japanese, Thai) from the original dataset.

Upon collecting all these datasets, the following steps were performed: 1) Removal of duplicate and null entries., 2) Calculate other labels for each dataset using some publicly available pre-trained models[6,7,8]. For example, for sentiment datasets, sarcasm, irony, hate, and offensive labels were calculated. Similarly, for sarcasm datasets, sentiment, irony, hate, and offensive labels were calculated, and so on. 3) Along with labels, the confidence score for each determined label was calculated and stored. 4) Since no deep learning models are 100% accurate, there must be a high chance of false labeling while assigning labels to a dataset using pre-trained models. Therefore, after calculating the labels, each sentence is accepted for the final dataset along with its labels if the calculated label has a confidence score of ≥ 0.9; otherwise, the sentence is rejected.

A flow diagram for the dataset preparation strategy is provided in Figure 1.

After performing the above steps, 2109 and 4387 samples were included in the final dataset for similar and dissimilar tasks, respectively. Next, 10% of the data (212 samples for similar tasks, and 402 samples for dissimilar tasks) were split out and preserved for testing purposes and the remaining 90% of data (1897 samples for similar tasks and 3985 samples for dissimilar tasks) were used for training and development purposes respectively[9]. The distribution of labels for different tasks in our final datasets is provided in Figure 2 and Figure 3, respectively.

[4]https://bit.ly/hate-speech-offensive
[5]https://bit.ly/language-identification-datasst
[6]https://bit.ly/distilbert-multi-sentiment
[7]https://bit.ly/english-sarcasm-detector
[8]https://huggingface.co/cardiffnlp
[9]The full dataset is publicly available at: https://bit.ly/mtl-dataset

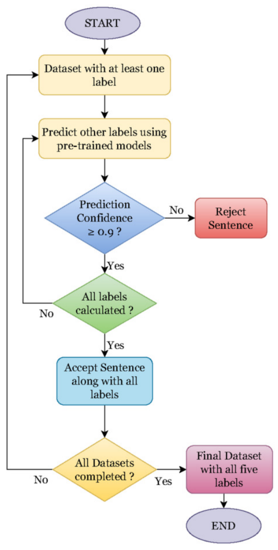
START
Dataset with at least one label
Predict other labels using pre-trained models
Prediction Confidence ≥ 0.9 ?
No
Reject Sentence
Yes
All labels calculated ?
No
Yes
Accept Sentence along with all labels
All Datasets completed ?
No
Yes
Final Dataset with all five labels
END

Fig.1: *Flow diagram of dataset preparation.*

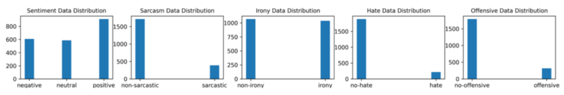

Fig.2: *Distribution of data for similar tasks.*

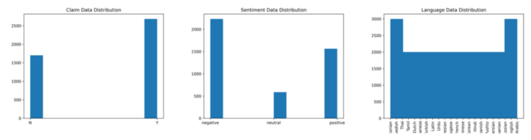

Fig.3: *Distribution of data for dissimilar tasks.*

4. METHODOLOGY

This section discusses the proposed methodologies for developing the MTL framework. Our main aim was to develop one neural network to classify five similar types of tasks and another to classify three dissimilar types of tasks (claim, language, and sentiment) and analyze their performance. The overall model frameworks for similar and dissimilar tasks are provided in Figure 4 and Figure 5, respectively.

Before diving into model training and evaluation, some basic pre-processing steps were performed in each sentence (S) in our final datasets, such as (1) Removal of HTML tags, (2) Convert S into a lowercase sentence, (3) Removal of punctuations and multiple spaces from S, (4) If S contains any username that starts with the character “@” then convert that into “@user,” (5) Convert all URL links to ‘HTTP’ in S, and (6) Convert any emoji in S to its corresponding text using the ‘emoji’ package[10] of Python.

After performing all the pre-processing steps in S, tokenize S into a sequence of tokens $[k_1, k_2, k_3, \ldots, k_n]$. Since every sentence generates a variable length of tokens, we converted every sentence to a fixed-sized sequence of tokens by padding zero at the end. Therefore, after padding zero, S becomes $[k_1, k_2, k_3, \ldots, k_L]$ where $L = 200$.

Task Definition: Given a tokenized sentence $X = [k_1, k_2, k_3, \ldots, k_L]$ where k_i’s are words (tokens) and L = 200. For similar-task MTL framework development, each tokenized sentence was annotated with five labels - sentiment (negative/ neutral/ positive), sarcasm (non-sarcastic/ sarcastic), irony (no-irony/ irony), hate (no-hate/ hate), and offensive (no-offensive/ offensive). For the dissimilar-task MTL framework development, each sentence was annotated with three labels - claim (Yes/ No), language (1 out of 22 different languages), and sentiment (positive/ negative/ neutral). Our main aim was to predict appropriate labels using a single neural network.

Word embedding: The pre-trained “GloVe” [12] word embedding with dimension $D = 200$ was utilized to convert each token k_i of sentence X into a sequence of vector x_i of length D. Thus, from a tokenized sentence $X = [k_1, k_2, k_3, \ldots, k_L]$ we get $X_{L\times D} = [x_1, x_2, x_3, \ldots, x_L]$. Then, X was fed into a BiLSTM layer as depicted in Figure 4 and 5.

Model Selection: To develop the MTL framework, the Bidirectional LSTM (BiLSTM) model was chosen. In traditional neural networks, there are short-term memory problems. Also, the vanishing gradient problem is one of the major drawbacks of those models. The LSTMs effectively enhance performance by identifying patterns, retaining important information, and eliminating the vanishing gradient problem. The BiLSTMs are more powerful than normal unidirectional LSTMs by the capable of analysing the inputs from the beginning as well as from the end. Since it analyses inputs from both ends so, it has the capability of utilizing features from the past as well as from the future. This provides a better understanding of language compared to unidirectional LSTMs.

GlobalMaxPooling: After the BiLSTM layer, followed by a dropout layer, the output of the dropout layer was passed into a ‘GlobalMaxPooling’ layer. The ‘GlobalMaxPooling’ fetches the maximum value from the hidden output vectors. Therefore, if the output of the dropout layer is $[\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_n]_{L\times M}$ where $\hat{y}_i$’s are vectors of length L and M is the number of hidden units of a BiLSTM layer, then

$$Z_{GlobalMaxPooling1D} \\ = [Max(\hat{y}_1), Max(\hat{y}_2), Max(\hat{y}_3), \ldots, Max(\hat{y}_n)]_{[1\times M]}$$

MTL with shared dense layer: Figure 4(a) represents the flow diagram of the MTL with a shared dense layer, where the output of the ‘GlobalMaxPooling’ Layer was passed as an input to a dense layer of 300 neurons.

[10]https://pypi.org/project/emoji/

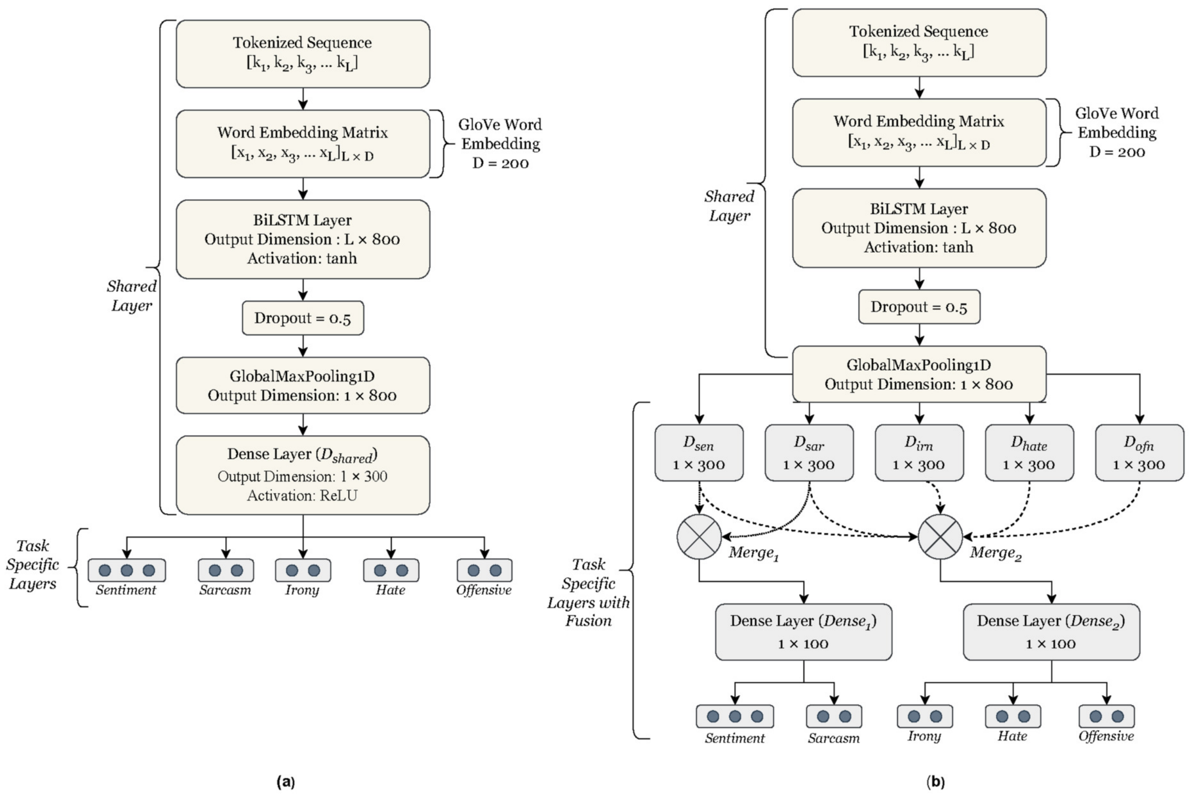

Fig.4: *Proposed Model framework for Similar Tasks: (a) MTL with a shared dense layer, (b) MTL with task-specific dense layer and fusion.*

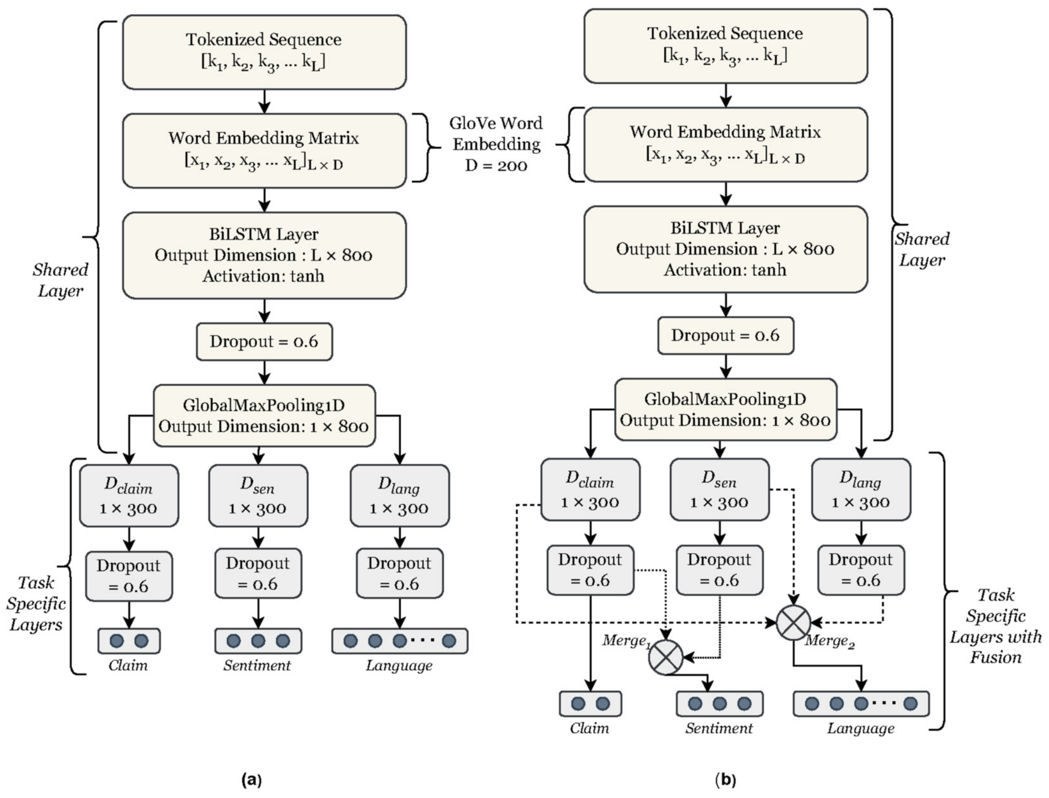

Fig.5: *Proposed Model framework for Dissimilar Tasks: (a) MTL with shared backbone and long task-specific heads, (b) MTL with shared backbone, long task-specific head, and fusion.*

$$D_{shared} = ReLU(Z_{GlobalMaxPooling1D})$$

MTL with long task-specific heads: Figure 5(a) provides the flow diagram of the MTL with long task-specific heads or layers, where the output of the 'GlobalMaxPooling' Layer was passed into individual dense layers corresponding to each task.

$$D_{tsp} = Dropout\left(ReLU\left(Z_{GlobalMaxPooling1D}\right)\right)$$

Where D_{tsp} represents the output of each task-specific dense layer with dropout in the dissimilar task.

MTL$_{fusion}$: Figure 4(b) and Figure 5(b) provides the framework representation of the MTL with fusion technique; where instead of using a single dense layer of 300 neurons (in similar task framework), separate dense layers of 300 neurons were used for each task to perform the task-specific learning.

$$D_* = ReLU\left(Z_{GlobalMaxPooling1D}\right)$$

where D_* represents task-specific dense layer outputs for each task $(D_{se}, D_{sa}, D_{ir}, D_{ht}, D_{of})$[11] for similar tasks and $(D_{claim}, D_{sen}, D_{lang})$[12] for dissimilar tasks.

Now, for similar tasks, the task-specific learnings were concatenated or fused and then fed into another dense layer of 100 neurons as follows:

$$\begin{aligned} Merge_1 &= D_{se} \otimes D_{sa} \\ Dense_1 &= ReLU(Merge_1) \end{aligned}$$

and,

$$\begin{aligned} Merge_2 &= D_{se} \otimes D_{sa} \otimes D_{ir} \otimes D_{ht} \otimes D_{of} \\ Dense_2 &= ReLU(Merge_2) \end{aligned}$$

Where $\otimes$ represents the concatenation of the outputs of the dense layers.

On the other hand, for the dissimilar tasks, the outputs of the task-specific dense layers were passed to dropout layers, and then the outputs from the previous layers were concatenated and fed into the final task-specific dense layers as follows:

Classification: Separate dense layers for each task were used to classify the outputs. For MTL with a shared dense layer, D_{shared} was passed as an input to each of the five dense layers. All these five dense layers used SoftMax as their activation function.

For MTL with long task-specific heads, the outputs of the individual dropout layers were fed into task-specific dense layers, which use SoftMax as their activation function.

$$\begin{aligned} P_* &= softmax(D_*) \\ \hat{Y}_* &= argmax_j(P_*) \end{aligned}$$

Where P_* represents the probability value of each class of each classification task, $\hat{Y}_*$ represents the predicted class value and, j represents the number of classes in the classification task.

For MTL with task-specific dense layers and fusion (MTL$_{fusion}$), in case of similar tasks, the output of $Dense_1$ was fed as an input to the sentiment and sarcasm classification layer, and the output of $Dense_2$ was fed as an input to each irony, hate, and offensive classification task layers.

$$\begin{aligned} P_{se}, P_{sa} &= softmax(Dense_1) \\ P_{ir}, P_{ht}, P_{of} &= softmax(Dense_2) \end{aligned}$$

and,

$$\hat{Y}_* = argmax_j(P_*)$$

Where $P_{se}, P_{sa}, P_{ir}, P_{ht}, P_{of}$ represents the probability value of each class of each in the sentiment, sarcasm, irony, hate, and offensive classification tasks respectively. $\hat{Y}_*$ represents the predicted class value for each classification task and $P_{similar} \in \{P_{se}, P_{sa}, P_{ir}, P_{ht}, P_{of}\}$.

In case of dissimilar tasks, the output of $Dropout_{claim}$ was passed as an input to the claim detection layer, fed the output of $Dense_1$ as an input to the sentiment classification layer, fed the output of $Dense_2$ as an input to language identification task layer.

$$\begin{aligned} P_{claim} &= sigmoid(Dropout_{claim}) \\ P_{sen} &= softmax(Dense_1) \\ P_{lang} &= softmax(Dense_2) \end{aligned}$$

and,

$$\begin{aligned} \hat{Y}_{claim} &= \begin{cases} 0, if\ P_{claim} \leq 0.5 \\ 1, \text{otherwise} \end{cases} \\ \hat{Y}_{sen} &= argmax_j(P_{sen}) \\ \hat{Y}_{lang} &= argmax_j(P_{lang}) \end{aligned}$$

Where P_{claim}, P_{sen}, and P_{lang} represents the probability value of each class of claim, sentiment, and language classification tasks respectively and $\hat{Y}_{claim}, \hat{Y}_{sen}, \hat{Y}_{lang}$ represents the predicted class value for claim, sentiment and language classification tasks respectively.

[11] *'se', 'sa', 'ir', 'ht', 'of'* are the abbreviations of $sentiment, sarcasm, irony, hate$ and $offensive$, respectively.

[12] *'claim', 'sen', 'lang'* are the abbreviations of *claim, sentiment* and *language*, respectively.

4.1 Training

To train our proposed frameworks, the final dataset was split out in an 80-20 ratio, where 80% of the data was used to train the model (train split) and 20% of the data was used as a development set for fine-tuning the models.

For the Multi-Task loss function, the SparseCategoricalCrossEntropy and BinaryCrossEntropy (only for claim classification) loss functions were used and monitored loss for the development split of the dataset.

$$L_{total} = \sum_{i=1}^{K} L_i$$

Where L_i is the loss for different tasks and K is the number of tasks in the MTL framework.

To accomplish the training process, the Adam optimizer [25] was used with a learning rate of 0.0005. The initial number of epochs was chosen to be 25 to train the proposed frameworks, however, we used the 'EarlyStopping' method provided by TensorFlow's 'Keras' module to prevent overfitting during the training process and the batch size was chosen as 8 and 64 for similar and dissimilar tasks respectively. The learning curves for MTL in similar and dissimilar task frameworks are provided in Figure 6.

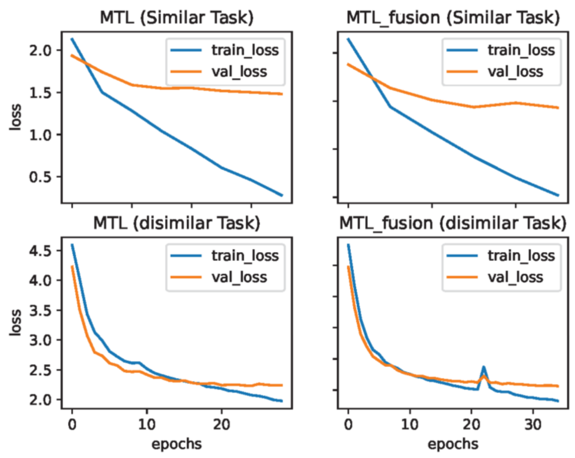

Fig. 6: Learning curve for MTL for similar and dissimilar tasks.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Postitive+False Positive}}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Postitive+False Negative}}$$

$$\text{Recall} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$$

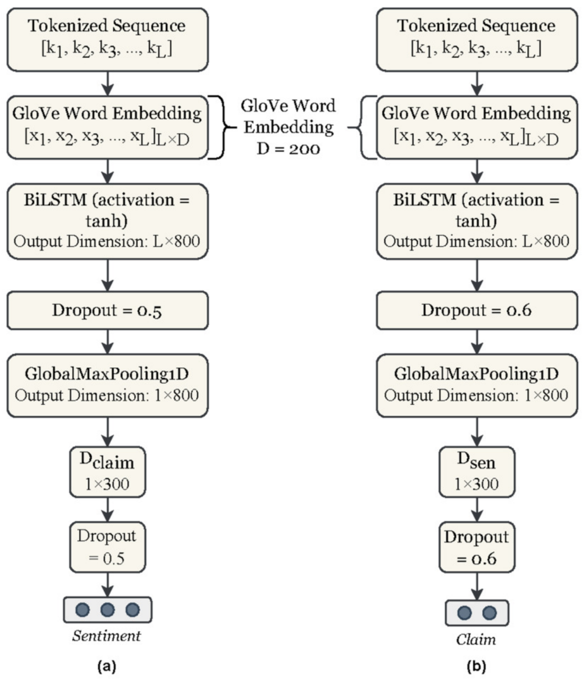
Tokenized Sequence [k1, k2, k3, ..., kL]
GloVe Word Embedding [x1, x2, x3, ..., xL]L×D
GloVe Word Embedding D = 200
BiLSTM (activation = tanh) Output Dimension: L×800
Dropout = 0.5
GlobalMaxPooling1D Output Dimension: 1×800
Dclaim 1×300
Dropout = 0.5
Sentiment
(a)
Tokenized Sequence [k1, k2, k3, ..., kL]
GloVe Word Embedding [x1, x2, x3, ..., xL]L×D
BiLSTM (activation = tanh) Output Dimension: L×800
Dropout = 0.6
GlobalMaxPooling1D Output Dimension: 1×800
Dsen 1×300
Dropout = 0.6
Claim
(b)

Fig. 7: *Diagrammatic representation of single task learning (STL) framework with one classification head. Here, we showed two STL model examples: (a) Sentiment classification, (b) Claim Classification.*

Table 1: *Performance comparison of 1-TL vs 5-TL vs 5-TL$_{fusion}$ for similar tasks.*

	Task	Precision	Recall	F1-score
1-TL	Sentiment	0.736	0.721	0.719
	Sarcasm	0.896	0.912	0.904
	Irony	0.811	0.811	0.81
	Hate	0.794	0.712	0.744
	Offensive	**0.927**	**0.899**	**0.912**
5-TL	Sentiment	0.75	0.74	0.739
	Sarcasm	0.896	0.912	0.904
	Irony	**0.852**	**0.848**	**0.848**
	Hate	0.8	0.759	0.778
	Offensive	0.904	0.878	0.89
5-TL$_{fusion}$	Sentiment	**0.764**	**0.768**	**0.76**
	Sarcasm	**0.911**	**0.929**	**0.92**
	Irony	0.82	0.81	0.809
	Hate	**0.813**	**0.784**	**0.798**
	Offensive	0.904	0.878	0.89

5. EXPERIMENT AND RESULT

5.1 Experimental Setup

All the experiments were performed using the modules from 'TensorFlow' and 'Keras' and the source code was executed in the Google Collaboratory environment with NVIDIA Tesla T4 GPU. For evaluation, the Precision, Recall and macro F1 scores were calculated for the test split of our final dataset.

Table 2: *Performances of all combinations of MTLs for similar tasks.*

	Sentiment			Sarcasm			Irony			Hate			Offensive		
Task	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
se	0.736	0.721	0.719	-	-	-	-	-	-	-	-	-	-	-	-
sa	-	-	-	0.896	0.912	0.904	-	-	-	-	-	-	-	-	-
ir	-	-	-	-	-	-	0.811	0.811	0.81	-	-	-	-	-	-
ht	-	-	-	-	-	-	-	-	-	0.794	0.712	0.744	-	-	-
of	-	-	-	-	-	-	-	-	-	-	-	-	0.927	**0.899**	0.912
se+sa	0.762	0.746	0.746	0.907	0.915	0.911	-	-	-	-	-	-	-	-	-
se+ir	0.729	0.721	0.723	-	-	-	0.807	0.806	0.805	-	-	-	-	-	-
se+ht	0.73	0.725	0.724	-	-	-	-	-	-	0.803	0.69	0.729	-	-	-
sa+ir	-	-	-	0.901	0.926	0.913	0.845	0.843	0.843	-	-	-	-	-	-
sa+ht	-	-	-	0.896	0.912	0.904	-	-	-	0.794	0.712	0.744	-	-	-
sa+of	-	-	-	0.896	0.912	0.904	-	-	-	-	-	-	0.921	0.881	0.9
ir+ht	-	-	-	-	-	-	0.817	0.815	0.815	0.808	0.737	0.766	-	-	-
ir+of	-	-	-	-	-	-	0.836	0.834	0.834	-	-	-	0.921	0.881	0.9
ht+of	-	-	-	-	-	-	-	-	-	0.834	0.692	0.739	0.853	0.886	0.868
se+of	0.752	0.747	0.747	-	-	-	-	-	-	-	-	-	**0.982**	0.87	0.916
se+sa+ir	0.751	0.752	0.744	0.902	0.902	0.902	0.814	0.805	0.804	-	-	-	-	-	-
se+sa+ht	0.727	0.73	0.727	0.907	0.915	0.911	-	-	-	0.752	**0.796**	0.772	-	-	-
sa+ir+ht	-	-	-	0.907	0.915	0.911	0.82	0.811	0.809	0.803	0.69	0.729	-	-	-
se+sa+of	0.741	0.727	0.719	0.907	0.915	0.911	-	-	-	-	-	-	0.904	0.878	0.89
sa+ir+of	-	-	-	0.902	0.902	0.902	0.817	0.815	0.815	-	-	-	0.888	0.875	0.881
sa+ht+of	-	-	-	0.907	0.915	0.911	-	-	-	0.821	0.762	0.787	0.936	0.865	0.896
se+ir+ht	0.732	0.667	0.676	-	-	-	0.82	0.82	0.82	0.8	0.759	0.778	-	-	-
se+ht+of	0.727	0.729	0.724	-	-	-	-	-	-	0.736	0.682	0.704	0.941	0.883	0.909
se+ir+of	0.726	0.724	0.724	-	-	-	**0.861**	**0.858**	**0.858**	-	-	-	0.961	0.886	**0.919**
ir+ht+of	-	-	-	-	-	-	0.825	0.825	0.825	0.795	0.782	0.788	**0.982**	0.87	0.916
se+sa+ir+ht	0.748	0.751	0.749	0.907	0.915	0.911	0.817	0.815	0.815	0.808	0.737	0.766	-	-	-
se+sa+ir+of	0.763	0.738	0.746	0.896	0.912	0.904	0.816	0.806	0.804	-	-	-	0.888	0.875	0.881
se+sa+ht+of	0.734	0.738	0.735	0.896	0.912	0.904	-	-	-	0.808	0.737	0.766	0.958	0.868	0.906
se+ir+ht+of	0.73	0.722	0.72	-	-	-	0.835	0.829	0.829	0.813	0.784	**0.798**	0.936	0.865	0.896
sa+ir+ht+of	-	-	-	0.891	0.923	0.906	0.844	0.844	0.844	**0.847**	0.717	0.763	0.866	0.854	0.86
5-TL	0.75	0.74	0.739	0.896	0.912	0.904	0.852	0.848	0.848	0.8	0.759	0.778	0.904	0.878	0.89
5-TL_fusion	**0.764**	**0.768**	**0.76**	**0.911**	**0.929**	**0.92**	0.82	0.81	0.809	0.813	0.784	**0.798**	0.904	0.878	0.89
5-TL (BiGRU)	0.749	0.725	0.73	0.907	0.915	0.911	0.828	0.825	0.824	0.777	0.687	0.72	0.904	0.878	0.89

At an experimental level, we experimented with both BiGRU (Bidirectional Gated Recurrent Unit) and BiLSTM for similar tasks. However, the BiLSTM provides a better result than BiGRU, therefore we selected the BiLSTM for developing STL and all MTL frameworks. We have added the result for BiGRU in Table 2.

5.2 Result

The proposed MTL frameworks were evaluated for all possible combinations of tasks starting from 1-TL or standalone classifier or STL, which are nothing but the same framework as described in Figure 4(a) and Figure 5(a) with one classification head only (a diagrammatic representation of STL framework for sentiment classification and claim classification are provided in Figure 7), 2-TL (all possible combinations of two tasks) to the highest possible combination of MTL (5-TL for similar task and 3-TL for dissimilar tasks).

5.2.1 Similar Task Performance

The performance results for similar tasks are provided in Tables 1 and 2. In Table 1, we compared the performances between STL (1-TL), MTL (5-TL) and MTL with fusion (5-TL$_{fusion}$) and in Table 2, we present the results for all comparisons of MTLs in similar tasks.

Sentiment classification: For sentiment classification, it can be seen that the 5-TLfusion model outperforms 1-TL, 5-TL, and other combinations of MTL (Table 2). Moreover, the 5-TLfusion gives a performance improvement by 5.39% over the sentiment classification STL framework (1-TL).

Sarcasm classification: Like sentiment classification, the sarcasm classification task also provides the best performance in the 5-TL$_{fusion}$ model. Also, the sarcasm classification performance for 5-TL$_{fusion}$ provides a performance improvement (F1-score) by 1.74% for both 1-TL and 5-TL.

Irony classification: Among the 1-TL, 5-TL and 5-TL$_{fusion}$, the simple 5-TL framework provides the best performance for the irony classification tasks with precision, recall, and f1-score of 0.852, 0.848 and 0.848 respectively. However, for all other task combinations perspectives, the performance is slightly better in sentiment + irony + offensive combination of MTL with precision, recall, and F1-Score of 0.861, 0.858, and 0.858 respectively.

Hate speech classification: For the hate speech classification, it can be seen from Table 2 that the best precision score of 0.847 is achieved by sarcasm + irony + hate + offensive combination, the best recall of 0.796 is achieved by sentiment + sarcasm + hate combination and best F1-score of 0.798 is achieved by both 5-TL$_{fusion}$ and sentiment + irony + hate + offensive combination of MTL.

Offensive language classification: In the offensive language classification, the performances in 5-TL and 5-TL$_{fusion}$ models degrade compared to other combinations of MTLs. The best precision of 0.982 is achieved by sentiment + offensive and irony + hate + offensive combination of MTL. The best recall of 0.899 and F1-Score of 0.919 is achieved by the standalone offensive classifier (1-TL) and sentiment + irony + offensive combination of MTL.

5.2.2 Dissimilar Task Performance

The performances for dissimilar tasks are provided in Table 3 and 4 where in Table 3 we compared the performances between STL (1-TL), MTL (3-TL) and MTL with fusion (3-TL$_{fusion}$). In Table 4 we provide the results for all comparisons of MTLs in dissimilar tasks.

Table 3: *Performance comparison of 1-TL vs 3-TL vs 3-TL$_{fusion}$ for dissimilar tasks.*

	Task	Precision	Recall	F1-score
1-TL	Claim	**0.711**	**0.711**	**0.711**
	Sentiment	**0.66**	**0.669**	**0.664**
	Language	**0.788**	**0.738**	**0.723**
3-TL	Claim	0.589	0.564	0.534
	Sentiment	0.597	0.528	0.55
	Language	0.762	0.705	0.715
3-TL$_{fusion}$	Claim	0.61	0.599	0.59
	Sentiment	0.63	0.566	0.586
	Language	0.722	0.706	0.707

Claim detection: In the case of claim detection, the 3-TL$_{fusion}$ framework performed better compared to simple 3-TL framework. However, all MTL frameworks perform worse than the 1-TL framework. The best precision, recall and F1-score are 0.711, 0.711, and 0.711 respectively which all were achieved by the 1-TL model. Although it is noticeable that the claim detection task benefits when combined with the sentiment classification task as we can see in Table 4.

Sentiment classification: Like claim identification, in sentiment classification task, the 3-TL$_{fusion}$ framework performs better compared to simple 3-TL model. The best F1-score of 0.664 was achieved by the single task (1-TL) model and sentiment + claim combination of MTL. The best precision of 0.709 and recall of 0.642 was achieved by the sentiment + claim combination of MTL (2-TL).

Language identification: For the language identification task, the 1-TL framework overperforms all other combinations of MTL. In this scenario, the performance in simple 3-TL is better than the performance of 3-TL$_{fusion}$ framework. The best precision, recall, and f1-score are 0.788, 0.738, and 0.723 which are achieved by the 1-TL framework.

6. ERROR ANALYSIS

This section briefly discusses the error cases for similar and dissimilar tasks in MTL and STL scenarios. The error analysis was conducted only on the test split of the dataset, and a few examples were presented in tabular format in Table 5 and Table 6. For simplicity, the error analysis was conducted only between the baseline classifier (STL), M-TL, and M-TL$_{fusion}$. (M-TL represents the highest combination of tasks: for similar tasks, M-TL corresponds to 5-TL, and for dissimilar tasks, M-TL corresponds to 3-TL).

6.1 Analysis of Similar Tasks

The error analysis for similar tasks is provided in Table 4. From Table 5, for the example S_1, the 5-TL$_{fusion}$ framework correctly detects the neutral sentiment in the sentence, but simple 5-TL and 1-TL failed to predict the proper class label. Therefore, it can be said that the fusion of sentiment and sarcasm learning helps to predict 'neutral' sentiments in a sentence better way. Considering another example S_2, from Table 5, it can be observed that our 1-TL and 5-TL frameworks predict the sentence "*Good luck out there! I hope you find what you're looking for.*" as sarcastic (actual: non-sarcastic), whereas the 5-TL$_{fusion}$ framework correctly predicts the sentence as non-sarcastic. In this case, the learning from the sentiment task fused with the sarcasm task helps the model to predict the sentence as non-sarcastic.

However, in some cases, the 5-TL$_{fusion}$ framework failed to identify proper class labels. For example, in Sentence S_3, the 5-TL$_{fusion}$ was unable to predict the true class label for irony but correctly predicted the sentiment class. One possible reason is the fus-

Table 4: *Performances of all combinations of MTLs for dissimilar tasks.*

	Claim			Sentiment			Language		
Task	P	R	F	P	R	F	P	R	F
claim	**0.711**	**0.711**	**0.711**	-	-	-	-	-	-
sen	-	-	-	0.66	**0.669**	0.664	-	-	-
lang	-	-	-	-	-	-	**0.788**	**0.738**	**0.723**
claim+sen	0.709	0.709	**0.709**	**0.709**	0.642	**0.664**	-	-	-
sen+lang	-	-	-	0.571	0.589	0.576	0.746	0.697	0.695
lang+claim	0.588	0.567	0.542	-	-	- 0	.687	0.692	0.681
3-TL	0.589	0.564	0.534	0.597	0.528	0.55	0.762	0.705	0.715
3-TL$_{\text{fusion}}$	0.61	0.599	0.59	0.63	0.566	0.586	0.722	0.706	0.707

Table 5: *Error analysis between 1-TL, 5-TL, and 5-TL$_{fusion}$ for similar tasks.*

			Sentiment		Sarcasm		Irony		Hate		Offensive	
			True	Pred	True	Pred	True	Pred	True	Pred	True	Pred
S_1	I think this was the first night of Louis with Briana? (That's not her tho) I saved it on April 17 but I DK	1-TL	neu	pos	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	n-of	n-of
		5-TL	neu	pos	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	n-of	n-of
		5-TL$_F$	neu	neu	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	n-of	n-of
S_2	Good luck out there! I hope you find what you're looking for.	1-TL	pos	pos	n-sr	sr	n-ir	ir	n-ht	n-ht	n-of	n-of
		5-TL	pos	pos	n-sr	sr	n-ir	ir	n-ht	n-ht	n-of	n-of
		5-TL$_F$	pos	pos	n-sr	n-sr	n-ir	ir	n-ht	n-ht	n-of	n-of
S_3	@JetBlue I would like to communicate directly with a "Customer Experience" executive. Does Joanna Geraghty have an email address?	1-TL	neg	neu	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	n-of	n-of
		5-TL	neg	neu	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	n-of	n-of
		5-TL$_F$	neg	neg	n-sr	n-sr	n-ir	ir	n-ht	n-ht	n-of	n-of
S_4	For the 1st time, women are elected into political office in Saudi Arabia. What does this mean? #womeninpolitics"	1-TL	neu	neu	n-sr	n-sr	ir	ir	n-ht	ht	n-of	n-of
		5-TL	neu	neu	n-sr	n-sr	ir	ir	n-ht	ht	n-of	n-of
		5-TL$_F$	neu	neu	n-sr	n-sr	ir	ir	n-ht	n-ht	n-of	n-of
S_5	@DaJetlyfe my nicca	1-TL	pos	neu	n-sr	n-sr	n-ir	n-ir	ht	n-ht	of	of
		5-TL	pos	neu	n-sr	n-sr	n-ir	n-ir	ht	n-ht	of	n-of
		5-TL$_F$	pos	neu	n-sr	n-sr	n-ir	n-ir	ht	n-ht	of	n-of
S_6	Hope you will beat this crap and achieve your full potential man. Best of luck for you :)	1-TL	pos	pos	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	of	n-of
		5-TL	pos	pos	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	of	n-of
		5-TL$_F$	pos	pos	n-sr	n-sr	n-ir	n-ir	n-ht	n-ht	of	n-of

Table 6: *Error analysis between 1-TL, 3-TL and, 3-TL$_{fusion}$ for dissimilar tasks.*

			Claim		Sentiment		Language	
id	Text	Task	True	Pred	True	Pred	True	Pred
S_1	I will admit it has less than the Sabbath albums before it, but it still very much holds onto the blues.	1-TL	yes	yes	pos	pos	eng	eng
		3-TL	yes	yes	pos	neg	eng	eng
		3-TL$_F$	yes	yes	pos	pos	eng	eng
S_2	Maybe I could do my own statistics.	1-TL	yes	no	neu	neu	eng	eng
		3-TL	yes	yes	neu	neu	eng	indo
		3-TL$_F$	yes	no	neu	neg	eng	eng
S_3	Have not got around to sorting out the history yet.	1-TL	no	no	neg	neg	eng	eng
		3-TL	no	yes	neg	neg	eng	dut
		3-TL$_F$	no	yes	neg	pos	eng	eng
S_4	müller mox figura centralis circulorum doctorum vindobonesium fiebat quibus interearant petrus	1-TL	no	no	neu	neg	lat	lat
		3-TL	no	yes	neu	neu	lat	por
		3-TL$_F$	no	no	neu	neg	lat	spa

ing of the learnings from sarcasm and irony in the Merge2 in Figure 4(b). The combination of sarcasm and irony misidentifies non-ironic sentences as ironic. For sentence S_5, the 1-TL predicts the proper class label, whereas the 5-TL and 5-TL$_{fusion}$ failed to predict. Also, in some cases, both the 1-TL, 5-TL, and 5-TL$_{fusion}$ was unable to predict the correct class label. For example, in S_5, all three models classify the sentence as 'neutral' whereas the original label is 'positive' and cannot find the hate in that sentence. For S_6, the models failed to predict the sentence as an offensive sentence.

6.2 Analysis of Dissimilar Tasks

The error analysis for dissimilar tasks is provided in Table 6. From Table 6, in the example S_1, it is seen that although claim and language labels were correctly assigned, the 3-TL framework failed to correctly predict the positive sentiment of the sentence. One possible reason for this is that the tasks were relatively dissimilar; therefore, the 3-TL framework failed to predict the correct class. However, the fusion of the Dropout layer of the D_{claim} and the D_{sen} overcomes this issue and predicts its proper sentiment class. In example S_2, for claim detection, 1-TL and 3-TL$_{fusion}$ frameworks failed to predict the claim correctly, but the 3-TL framework did. In this case, the fusion of learnings failed to predict the proper sentiment class. For sentence S_3, only the 1-TL framework correctly predicted the claim, but the MTL frameworks couldn't. The 3-TL framework also incorrectly predicts it as a sentence in 'Dutch' whereas it is an 'English' sentence. 1-TL and 3-TL$_{fusion}$ frameworks predict this sentence as a sentence with negative sentiment despite this being a neutral sentence.

7. OBSERVATIONS

One of our main motives in these experiments was to study the performance of our developed frameworks for different similar and dissimilar tasks and draw some insights from that. Upon performing all the experiments, there were a few noticeable points we delved deep into:

Firstly, it is observed that the performances of similar tasks as a whole were far better than dissimilar tasks in our MTL framework. This is because learning from similar tasks enhances performance, whereas in dissimilar tasks, learning from one task does not effectively benefit the others. As a result, we observed a performance drop in MTL for dissimilar tasks.

Secondly, as discussed in Section 4, the short-head approach was used for similar tasks, while the long-head approach was used for dissimilar tasks. The reason behind this is that similar tasks have many attributes in common among them. So, the number of shared layers is greater than the individual task-specific layers. Whereas dissimilar tasks have very little in common, each task requires additional independent attention. For this reason, for dissimilar tasks, we used more layers in the individual task-specific layers.

Thirdly, as already discussed in Section 3, to prepare our dataset, some open-source models were used to produce the missing labels needed for our experiments. Although we considered the labels which are having confidence scores of more than 0.9, still there might be some false labelling as those models are not 100% accurate. It must have a negative effect on the overall performance in the individual tasks. However, to overcome this issue the approach of MTL was performed.

Finally, it can be seen that when the experiments were carried out in the fusion frameworks as shown in Figure 4(b) and Figure 5(b), the performances were better in most of the cases. After seeing these statistics, it can be said that learnings of one task help the other tasks in an MTL framework.

8. CONCLUSION

In this paper, a deep learning-based MTL framework is proposed that can classify similar types of tasks such as sentiment, sarcasm, emotion, irony, hate, and offensive sentences at a time using a neural network. Besides that, another MTL framework is proposed that is capable of handling dissimilar tasks such as claim detection, sentiment analysis, and language identification. Our main motive for these experiments was to study the performances of different tasks, whether similar or dissimilar, and analyze how the MTL framework helps or affects the performances.

We observed that similar tasks performed better compared to their corresponding STL frameworks for similar tasks. On the other hand, the dissimilar tasks performed better in the STL frameworks. Also, the fusion technique of MTL in dissimilar tasks performs comparatively better than the simple MTL for dissimilar tasks. In future works, we will perform our experiments with a larger sample size of data to validate the robustness of the frameworks.

9. LIMITATION AND FUTURE WORK

Our proposed work also has some limitations. Firstly, the size of our final dataset was relatively small; therefore, a larger dataset is needed to validate the robustness of the MTL models. Secondly, there was a significant imbalance in the sarcasm, hate, and offensive classes in our final dataset (Figure 2); thus, the model may exhibit performance bias for those tasks. Lastly, the lack of exploration of transformer models, such as BERT [14], is one of the limitations of our work.

In the future, we will increase the sample size of our existing datasets by including more text to create a more robust model. We will also try to include texts from other languages, such as Bengali, Hindi, etc. In addition, we will explore transformer-based models in our future work.

AUTHOR CONTRIBUTIONS

Conceptualization, P.P., and S.S.D.; methodology, P.P., S.S.D., and D.D; validation, P.P., S.S.D, D.D., and A.K.K.; formal analysis, D.D., and A.K.K; investigation, D.D., and A.K.K.; data curation, P.P., and S.S.D.; writing—original draft preparation, P.P., and S.S.D.; writing—review and editing, P.P., S.S.D, D.D., and A.K.K.; visualization, P.P. and S.S.D.; supervision, D.D., and A.K.K. All authors have read and agreed to the published version of the manuscript.

References

[1] Y. Y. Tan, C.-O. Chow, J. Kanesan, J. H. Chuah and Y. Lim, "Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning," *Wireless Personal Communications*, vol. 129, pp. 2213–2237, March 2023.

[2] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria and A. Gelbukh, "Sentiment and Sarcasm Classification With Multitask Learning," in *IEEE Intelligent Systems*, vol. 34, no. 3, pp. 38-43, 1 May-June 2019.

[3] F. M. P. Del Arco, S. Halat, S. Padó and R. Klinger, "Multi-Task Learning with Sentiment, Emotion, and Target Detection to Recognize Hate Speech and Offensive Language.," [Online]. arXiv (Cornell University), 2021. Available from: `https://doi.org/10.48550/arXiv.2109.10255`

[4] R. Kundu, *Multi-Task Learning in ML: Optimization & Use Cases [Overview]*, 2023.

[5] M. S. Akhtar, D. Chauhan, D. Ghosal, S. Poria, A. Ekbal and P. Bhattacharyya, "Multi-task Learning for Multi-modal Emotion Recognition and Sentiment Analysis," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 370-379, Jun. 2019.

[6] R. Caruana, "Multitask Learning," *Machine learning*, vol. 28, pp. 41–75, Jul. 1997.

[7] S. Ruder, "An Overview of Multi-Task Learning in Deep Neural Networks," [Online]. arXiv [cs.CL]. 2017. Available from: `https://doi.org/10.48550/arXiv.1706.05098`

[8] P. Liu, X. Qiu and X. Huang, "Recurrent neural network for text classification with multi-task learning," [Online]. arXiv (Cornell University), 2016. Available from: `https://doi.org/10.48550/arXiv.1605.05101`

[9] P. Liu, X. Qiu and X. Huang, "Adversarial Multi-task Learning for Text Classification," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1-10, Jul. 2017.

[10] E. Savini and C. Caragea, "A Multi-Task Learning Approach to Sarcasm Detection (Student Abstract)," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 10, pp. 13907-13908, Apr. 2020.

[11] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, "Enriching Word Vectors with Subword Information," [Online]. arXiv preprint, 2016., Available from: `https://doi.org/10.48550/arXiv.1607.04606`

[12] J. Pennington, R. Socher and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532-1543, Oct. 2014.

[13] A. El Mahdaouy, A. El Mekki, K. Essefar, N. El Mamoun, I. Berrada and A. Khoumsi, "Deep Multi-Task Model for Sarcasm Detection and Sentiment Analysis in Arabic Language," in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine, pp. 334-339, Apr. 2021.

[14] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, vol. 1, pp. 4171-4186, 2019.

[15] G. V. Singh, D. S. Chauhan, M. Firdaus, A. Ekbal and P. Bhattacharyya, "Are Emoji, Sentiment, and Emotion Friends? A Multi-task Learning for Emoji, Sentiment, and Emotion Analysis," in *Proceedings of the 36th Pacific Asia Conference on Language, Information and Computation*, Manila, Philippines, pp. 166-174, 2022.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," [Online]. arXiv [cs.CL], 2019., Available from: `https://doi.org/10.48550/arXiv.1907.11692`

[17] J. Huang, T. Liu, J. Liu, Á. D. Lelkes, C. Yu and J. Han, "All Birds with One Stone: Multi-task Text Classification for Efficient Inference with One Forward Pass," [Online]. arXiv [cs.CL], 2022. Available from: `https://doi.org/10.48550/arXiv.2205.10744`

[18] F. Barbieri, J. Camacho-Collados, L. Espinosa Anke and L. Neves, "TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1644-1650, 2020.

[19] R. Misra and J. Grover, *Sculpting Data for ML: The first act of Machine Learning*, 2021.

[20] R. Misra and P. Arora, "Sarcasm Detection using News Headlines Dataset," *AI Open*, vol. 4, pp. 13-18, 2023.

[21] S. Castro, D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea and S. Poria, "Towards Multimodal Sarcasm Detection (An _Obviously_ Perfect Paper)," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, vol. 1, pp. 4619-4629, Jul. 2019.

[22] . A. Mishra, D. Kanojia and P. Bhattacharyya, "Predicting readers' sarcasm understandability by modeling gaze behavior," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[23] J. Ling and R. Klinger, "An Empirical, Quantitative Analysis of the Differences Between Sarcasm and Irony," *The Semantic Web*, pp. 203-216, 2016.

[24] S. Rosenthal and K. McKeown, "Detecting Opinionated Claims in Online Discussions," *2012 IEEE Sixth International Conference on Semantic Computing*, Palermo, Italy, pp. 30-37, 2012.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," [Online]. arXiv (Cornell University). 2014. Available from: `https://doi.org/10.48550/arXiv.1412.6980`

Pritam Pal completed his Master of Technology (M.Tech.) degree in Computer Science and Engineering from RCC Institute of Information Technology, Kolkata, India, in 2024. Before that, he completed his Master of Science (M.Sc.) and Bachelor of Science with Honors (B.Sc. Hons.) degree in Computer Science from Maulana Abul Kalam Azad University of Technology and University of Calcutta, respectively. His research interests include Natural Language Processing, Multi-Task Learning, Sentiment Analysis, Claim Identification and analysis, NLP applications in Bengali and Indian languages, etc.

Shankha Shubhra Das is a software engineer and Project Manager at Bajaj Auto Credit Limited. He completed his Master of Engineering (M.E.) in Computer Science and Engineering from Jadavpur University, Kolkata, India in, 2024. He was one of the best performers of his batch and was also the central placement coordinator of all post-graduation departments. His research interests are in the areas of Artificial Intelligence, Natural Language Processing, Big Data, Transformer Architecture, Deep Learning, etc. Besides this, he has several projects in Sentiment Analysis, Computer Vision, Claim Detection, etc.

Dipankar Das is an Assistant Professor (STAGE-III) at the Department of Computer Science and Engineering at Jadavpur University. He was a young faculty research fellow under the Visv esvaraya PhD Scheme for Electronics & IT, Media Lab Asia, MeitY, India. Before Jadavpur University, he served as an Assistant Professor at the Department of Computer Science and Engineering at NIT Meghalaya from 2012 – 2014. He is leading different research projects under DRDO and RUSA 2.0. He has over 25 journal publications and more than 180 publications in reputed conferences, workshops, and book chapters. His research interests include Natural Language Processing, Speech Processing, Claim Detection and Verification, Large and Small Language Models, Social Networks, Sentiment/ Emotion analysis, Ontology Engineering, Psycho-linguistics, Machine Learning, Code-Mixing, Dialogue Management, etc.

Anup Kumar Kolya is an Assistant Professor and Head of the Department of Computer Science & Engineering at RCC Institute of Information Technology, Kolkata, India. Before that, he worked as a Senior Research Engineer in the Department of Computer Science & Engineering, Jadavpur University, Kolkata, in the DIT, MCIT, Government of India Sponsored Project titled “Development of English to India Language Machine Translation System” (EILMT) from 2009 to 2014. He completed his Ph.D. (Engg.) and M.E. from the Department of Computer Science & Engineering, Jadavpur University, Kolkata. His research interests include Natural Language Processing, Event and Temporal Expression Recognition, Event and Time Relation Identification, Machine Learning, Rule Base Machine Translation, Sentiment Analysis, Event Tracking, and Information Retrieval. He has over 50 publications in top journals, conferences, and workshops and is the editor and author of several books and book chapters.