Deciphering Multi-task Learning: Comparative Insights for Similar and Dissimilar Tasks

Anonymous ACL submission

Abstract

Multi-Task Learning (MTL), emerged as a powerful concept in the era of machine learning, employs a shared model trained to handle multiple tasks at the same time. Numerous advantages of this novel approach inspire us to instigate the insights of various tasks with similar (Identification of Sentiment, Emotion, Sarcasm, Irony, Hate and Offensive) and dissimilar (Identification of Sentiment, Claim, Language) genres and to analyze the change in their performances with respect to long and short head approaches. We shed light on the methods employed and critical observations to promote more efficient learning paradigm across similar and dissimilar tasks.

1 Introduction

The popularity of internet and social media not only allows users to express their opinions, sentiments, emotions or sarcasm but at the same time, such social media posts can also contain hateful and offensive contents that are vulnerable for teenagers. In the past decades, most of the researchers have worked on single tasks such as classification of sentiment, sarcasm, emotion, hateful sentences etc. while a few researchers have emphasized on two or multiple classification tasks e.g., sentiment and sarcasm (Majumder et al., 2019; El Mahdaouy et al., 2021; Tan et al., 2023), sentiment and emotion (Akhtar et al., 2019; Singh et al., 2022) etc.

Multi-task learning (MTL) as the name suggests, refers to a single shared machine-learning model that can perform multiple different tasks simultaneously (Kundu, 2023). The MTL provides three advantages over single-task learning - i) it helps in achieving generalization for multiple tasks; ii) each task improves its performance in association with the other participating tasks; and iii) it offers reduced complexity (Akhtar et al., 2019).

In the present article, we proposed two schemes of multi-task learning: First, a MTL model that classifies six related tasks of similar genre: sentiment, sarcasm, emotion, irony, hate speech and offensive and Second, a similar multi-task learning model working on relatively dissimilar tasks: claim detection, language identification, and sentiment analysis. The main objectives of our work is 1) to analyze whether adding different classification tasks (similar or dissimilar) into a MTL model can improve the overall performance of each classification over single-task or not; 2) to identify whether and how a task can gain out of MTL with respect to the tasks of similar and different flavours. Besides, we performed various combinations of tasks in MTL such as emotion and sarcasm classification, sentiment and hate speech classification, etc. to analyze the performance in a different scenario.

2 Dataset Preparation

In order to accomplish our first task, to the best of our knowledge, no publicly available dataset includes all the class labels together. Thus, we collected different task datasets from various sources with single labels and identify other labels using some pre-trained models[1][2]. For example, in case of sentiment dataset, the sarcasm, emotion, irony, hate, and offensive labels were identified; for the sarcasm dataset, the sentiment, emotion, irony, hate, and offensive labels were calculated, and so on. For the sentiment, irony, emotion, hate, and offensive sentences, we use the Tweet_Eval[3] dataset (Barbieri et al., 2020). In order to develop MTL model for dissimilar tasks, we collect another sentiment dataset from Kaggle known as the airline_tweet_sentiment[4] dataset. For sarcasm, the Sarcasm_News_Headline[5] dataset and the MUS-

[1] https://huggingface.co/cardiffnlp
[2] https://bit.ly/english-sarcasm-detector
[3] https://huggingface.co/datasets/tweet_eval
[4] https://bit.ly/twitter-airline-sentiment
[5] https://bit.ly/sarcasm_news_headline

	Dataset	#Texts
Similar Tasks	Sentiment	59899^a + 14640^a
	Sarcasm	55328^c + 690^d
	Emotion	5052^a
	Irony	4601^a
	Hate	12962^a
	Offensive	14100^a
Dissimilar Tasks	Claim	2190^e
	Claim	2197^f
	Language	21859^g

Table 1: Datasets and number of texts in those datasets (^{a}:Tweet_Eval; ^{b}:twitter-airline-sentiment; ^{c}:Sarcasm_News_Headline; ^{d}:MUStARD; ^{e}:LiveJournal; ^{f}:Wikipedia; ^{g}:WiLI-2018)

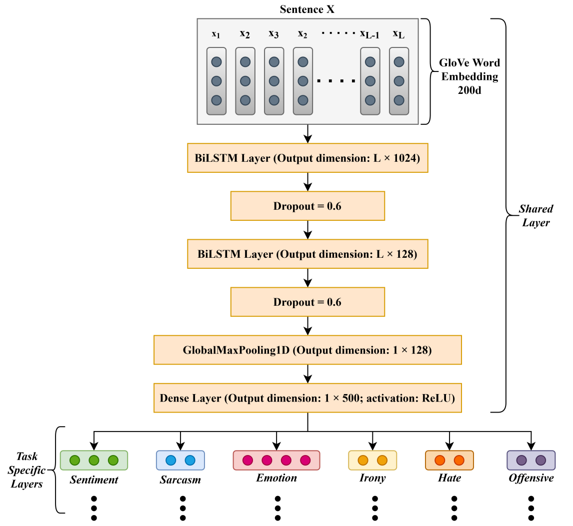

Figure 1: Proposed model architecture

tARD[6] (Castro et al., 2019) dataset have been used. The number of texts in each dataset is given in Table 1 (Similar Tasks).

For validation, 10% of the data was preserved while the remaining data was used for training and testing purposes. After that, we merged all the datasets into a single dataset. For the second task, we have used the datasets used by (Rosenthal and McKeown, 2012) in their paper. These datasets contain sentences from LiveJournal weblogs and Wikipedia talk pages annotated for opinionated claims. In these datasets, we have 2190 instances, from LiveJournal and 2197 from Wikipedia. We have collected another dataset which is a preprocessed version of WiLI-2018[7], the Wikipedia language identification benchmark dataset. The number of texts in each dataset is given in Table 1 (Dissimilar Task).

After collecting this dataset, the sentiment labels for claim datasets were identified using some pre-trained model[8].

3 Methods

In this section, we describe our proposed methodology. We aim to develop a single multi-task learning model that can classify six similar types of tasks (sentiment, sarcasm, emotion, irony, hate, and offensive) in the first case and three dissimilar types of tasks (claim, language, and sentiment) in the second case. The overall model architecture is depicted in Figure 1.

For each sentence S, first, we conducted some basic preprocessing in S such as — i) Removal of HTML tags, ii) Convert S into a lowercase sentence, iii) Removal of punctuations and multiple spaces, iv) If S has any username that starts with the character '@' then convert that into '@user', v) If S has any website links, then convert that link into 'http'.

Then we converted S into a sequence of tokens $[k_1, k_2, k_3, ..., k_n]$. Since every sentence gives a variable length token, we convert every sentence into a fixed-sized sequence of tokens by padding 0 at the end. So, after padding 0, S now becomes in the form of $[k_1, k_2, k_3, ..., k_L]$ 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 the first task, each tokenized sentence has six labels - sentiment (negative/ neutral/ positive), sarcasm (non-sarcastic/ sarcastic), emotion (anger/ joy/ optimism/ sadness), irony (no-irony/ irony), hate (no-hate/ hate) and offensive (no-offensive/ offensive). For the second task, each sentence has three labels - claim label (Yes/ No), language label (1 out of 22 different languages), and sentiment label (positive/ negative/ neutral). Our main task is to predict appropriate label using a single neural network.

Word Embedding: For word embedding, we use pre-trained "GloVe"(Pennington et al., 2014) word embedding with dimension D = 200, to convert each token k_i of sentence X into a sequence of vector x_i of length D. Thus, from a tokenized sentence $X = [k_1, k_2, k_3, ..., k_L]$ we get $X_{L \times D} = [x_1, x_2, x_3, ..., x_L]$. Then, X is fed into a BiLSTM layer as depicted in Figure 1.

[6] https://github.com/soujanyaporia/MUStARD
[7] https://bit.ly/language-identification-datasst
[8] https://bit.ly/multilingual-cased-sentiments-student

GlobalMaxPooling Layer: We integrated two BiLSTM layers followed by a dropout layer of 0.6 (Figure 1) and used a "GlobalMaxPooling" layer. The "GlobalMaxPooling1D" gives the maximum value from the hidden output vectors. So, if the output of the 2nd dropout layer is $[\hat{y_1}, \hat{y_2}, \hat{y_3}, ..., \hat{y_M}]_{L\times M}$ where $\hat{y_i}$'s are vectors of length L and M is the number of hidden units of a BiLSTM layer then,

$$Z_{GlobalMaxPooling1D} = [Max(\hat{y_1}), Max(\hat{y_2}), Max(\hat{y_3}), ..., Max(\hat{y_M})]_{[1\times M]}$$

After that, $Z_{GlobalMaxPooling1D}$ fed into a dense layer with 500 neurons:

$$Z_* = ReLU(Z_{GlobalMaxPooling1D})$$

Classification:

Short-Head Approach: In the case of similar tasks, for six classification tasks, we use six different dense layers. We fed Z_* as an input in each of six dense layers.

$$P_* = softmax(Z_*)$$

where P_* means probability values for either sentiment, sarcasm, emotion, irony, hate, or offensive classes, respectively.

Long-Head Approach: In the case of dissimilar tasks, for each of the three classification tasks, we have a series of task-specific layers consisting of dense and dropout layers.

$$O_1 = dense(Z_*); O_2 = tanh(O_1);$$
$$O_3 = dropout(O_2); O_4 = dense(O_3);$$
$$P_* = softmax(O_4)$$

where, O_1 to O_4 are intermediate output values from corresponding layers, and P_* means probability values for either sentiment, claim, or language classes.

Training: For the multi-task loss function, we use CrossEntropy loss for each of the tasks and monitor the loss for the test 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.

To train our proposed model, we took 50 epochs, but we used the "early stopping"[9] method to eliminate overfitting in our model. The parameters that are used to train the model are given in Table 2.

[9] https://keras.io/api/callbacks/early_stopping/

Parameter	Value
Embedding	GloVe 200d
1st BiLSTM hidden units	$2\times512 = 1024$
2nd BiLSTM hidden units	$2\times64 = 128$
Dropout	0.6
Loss function	CrossEntropy
Optimizer	Adam
Learning rate	0.0005
Epoch	50
Batch size	32

Table 2: Parameters used to train the model

4 Experiment and Result

4.1 Experimental Setup

We use TensorFlow[10] and Keras[11] to implement our proposed model and use the Collaboratory[12] environment to execute the code and calculate the F1-Score to evaluate the performance.

4.2 Result

Similar Task Comparison: Here, we will compare and contrast how these similar tasks have performed in our MTL framework. We perform all the combinations of MTLs such as 2-TL (combination of 2 tasks), 3-TL (combination of 3 tasks), 4-TL (combination of 4 tasks), 5-TL and 6-TL. A performance comparison of different tasks in 6-TL (all similar classification tasks) vs the best MTL combination score vs each of the standalone classifiers (1-TL) is illustrated in Table 3.

For similar tasks as shown in Table 3, in sentiment classification, the best performance is given by the combination of all tasks (sentiment + sarcasm + emotion + irony + hate + offensive). For sarcasm classification, the MTLs failed to give the best performance. The best result is provided by the sarcasm standalone classifier. For emotion classification, we can see that the 6-TL shows an improvement over standalone emotion classification, but the best result is given by the sarcasm + emotion combination of MTL. Similarly, for irony, hate and offensive classification, 6-TL shows an improvement over standalone classifiers, but the best result is provided by sarcasm + irony, sarcasm + emotion + hate and sarcasm + hate + offensive for irony, hate and offensive classification, respectively.

[10] https://www.tensorflow.org/

[11] https://keras.io/

[12] https://colab.research.google.com/

	Task	1-TL	K#-TL	Best Score	σ	ψ
Similar Task	Sentiment (se)	0.687	0.767	0.767 (all task)	11.645%	0%
	Sarcasm (sa)	0.957	0.909	0.957 (sa)	0%	5.28%
	Emotion (em)	0.682	0.742	0.848 (sa+em)	24.340%	14.286%
	Irony (ir)	0.649	0.819	0.875 (sa+ir)	32.823%	6.838%
	Hate (ht)	0.718	0.793	0.83 (sa+em+ht)	15.599%	4.666%
	Offensive (of)	0.722	0.874	0.884 (sa+ht+of)	22.438%	1.144%
Dissimilar Task	Sentiment (se)	0.668	0.539	0.682 (se+cl)	2.095%	26.53%
	Claim (cl)	0.706	0.623	0.706 (cl)	0%	13.322%
	Language (la)	0.953	0.690	0.953 (li)	0%	38.116%

Table 3: F1-Score comparison of 1-TL vs K-TL vs best MTL combination (#: K = 6 for similar tasks and K = 3 for dissimilar tasks; σ : Performance improvement in best MTL combination w.r.t. 1-TL; ψ : Performance improvement in best MTL combination w.r.t. K-TL)

Dissimilar Task Comparison: For dissimilar tasks, it can be seen from Table 3 that the performance in 3-TL degrades over 1-TL, but only the sentiment classification gives an improvement in the sentiment + claim combination of MTL. The claim and language classification gives the best performance in the standalone classifier.

5 Observation

In this study, our main motive was to study the performance of our model for different similar and dissimilar tasks and draw some insights from that. After all the experiments, there were a few noticeable points we delved deep into —

Firstly, we have observed that the performances of similar tasks as a whole are far better than dissimilar tasks in our MTL setting. One of the reasons can be the size of the dataset used for similar and dissimilar tasks or similar tasks help one another to perform better than dissimilar tasks do.

Secondly, as already discussed in Section 3, for similar tasks we have used the Short-Head approach, and for dissimilar tasks we have used the Long-Head approach. The reason behind this is the simple fact that similar tasks have many attributes in common among them. So, their common or shared layers are more in number rather than the individual task-specific layers. Whereas, the dissimilar tasks have very few things in common among them and each task needs extra standalone attention. For this reason, for dissimilar tasks, we have used more layers in the individual task-specific layers.

Thirdly, we already discussed in Section 2 that to prepare our dataset, we have used some open-source models to produce the missing labels needed for our experiments. Hence, 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.

And lastly, for similar tasks, it can be seen from Table 1 that the number of texts in sentiment and sarcasm datasets are much larger than the emotion, irony, hate and offensive dataset's number of texts. So, there may be a performance bias in our overall classification.

6 Conclusion

In this paper, we proposed a multi-task learning approach using deep learning that can classify sentences into similar classes like sentiment, sarcasm, emotion, irony, hate, and offensive. We also proposed a multi-task architecture that is used to handle dissimilar tasks like claim detection, sentiment analysis, language identification, etc. Our main motive for these experiments was to study the performances of different tasks whether similar or dissimilar, and analyze how the multi-task learning framework helps or affects the performances. From our study, we can see that in the case of similar tasks, the performance of all classification tasks has improved in the multi-task learning framework except the sarcasm classification. However, the same cannot be said in the case of dissimilar tasks, where we can see the trend of single tasks outperforming most of the multi-combinations of tasks.

In future, our main aim will be to reduce false labelling in datasets as much as possible. Also, we will explore different state-of-the-art models such as "GPT" or "BERT" and try to add other classification tasks to our MTL model like emoji classification.

References

Md Shad Akhtar, Dushyant Chauhan, Deepanway Ghosal, Soujanya Poria, Asif Ekbal, and Pushpak Bhattacharyya. 2019. 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, Volume 1 (Long and Short Papers)*, pages 370–379, Minneapolis, Minnesota. Association for Computational Linguistics.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy. Association for Computational Linguistics.

Abdelkader El Mahdaouy, Abdellah El Mekki, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. Deep multi-task model for sarcasm detection and sentiment analysis in Arabic language. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 334–339, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Rohit Kundu. 2023. Multi-Task Learning in ML: Optimization amp; Use Cases [Overview].

N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, and A. Gelbukh. 2019. Sentiment and sarcasm classification with multitask learning. *IEEE Intelligent Systems*, 34(03):38–43.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Sara Rosenthal and Kathleen McKeown. 2012. Detecting opinionated claims in online discussions. In *Sixth IEEE International Conference on Semantic Computing, ICSC 2012, Palermo, Italy, September 19-21, 2012*, pages 30–37.

Gopendra Vikram Singh, Dushyant Singh Chauhan, Mauajama Firdaus, Asif Ekbal, and Pushpak Bhattacharyya. 2022. 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*, pages 166–174, Manila, Philippines. Association for Computational Linguistics.

Yik Yang Tan, Chee-Onn Chow, Jeevan Kanesan, Joon Huang Chuah, and YongLiang Lim. 2023. Sentiment Analysis and Sarcasm Detection using Deep Multi-Task Learning. *Wireless Personal Communications*, 129(3):2213–2237.