Chapter 5

A study on people's opinions during PEI 2019

5.1 Introduction

Social media has enabled large-scale text-based emotion sharing on many topics. People post on social media with multimedia content full of images, text messages, often sharing details of personal events, travel information, daily activities etc. Some people use the information with malicious intentions. Some use them to malign, offend or make propaganda, by twisting the content and context and/or doctoring them. While faking the content is a serious issue on social media, the other one is finding and tracking the emotion from the content. There have been quite a few works on mining user emotion over various topics from publicly available dataset on social media. The most important task in analyzing user's opinions is emotion categorization: determining whether a given text, such as a user's review, comment, or tweet, carries hate speech or offensive comment. During the election, people share many posts (as text and images) on political leaders. the posts often express their love or hate sentiment. People often make dirty comments on personal life and professional life of leaders with expletives like *pagal kutta* (mad dog), *chowkidar*

(watchman), *Pappu* (boy), *chaiwala* (teaman), *harami* (bastard), *gadha* (donkey) etc. and vent their anger on the leaders. People also praise the leaders whom they like. For example, they highlight achievements of a leader or the promises that the leader has kept etc.

Multi-task learning (MTL) is an effective method to improve the performance when there are a number of related tasks. In MTL, there are several learning tasks, each of which belong to a common learning task type such as supervised tasks (e.g., classification or regression problems), unsupervised tasks (e.g., clustering problems), semi-supervised tasks, reinforcement learning tasks, multi-view learning task or graphical model. In these learning tasks, all of them or at least one subset of them are thought to be related to each other. It is found that learning these tasks jointly can yield higher performance than single-task learning. This observation leads to the birth of MTL. Therefore MTL aims to improve the performance of the normalization of related tasks. Emotion plays a vital role in social media posts as it is a social platform where feelings and information are shared live with friends and family. People react to that post immediately or later with their opinion or emotion and re-post the post on his or her own timeline. Some people like the post, and some dislike, i.e., some posts are positive and some negative in sentiment. In this particular task, we attempt to automatically identify hate speech and offensive content. These posts can be considered as negative sentiments with finer details. Abusing a person or group based on religion, caste, color, sex, etc. is called **hate speech**. **Offensive speech** are those that cause somebody to feel angry, hurt, insulted and annoyed.

5.2 Contribution

Hate speech and offensive language (HOF) identification are usually modeled as a classification task that asks the model to decide whether a text contains HOF. This task is

challenging because of the many explicit and implicit methods for verbally attacking a target individual or group. We focused on hate and offensive content identification during the 2019 general election of India. Here, we show that multitask learning is a better way of combining multiple tasks in one model.

5.2.1 Multi-task learning with the convolution network (MTL-CN)

The multi-task learning model has been proposed to identify hate speech and offensive content of social media posts. The existing works show that multi-task learning has gained superior performance than single-task learning for various tasks, including hate speech and offensive media posts. While parsing the posts, emotion and semantic features must exist. The semantic information is affected by the initialization of the pre-trained embedding model, whereas pre-trained Deepmoji model leverages the emotion feature. As shown in Figure 5.1, the input is processed to feature extraction where contextual features and emotion features have been extracted. With the help of those features, the task projection layer is trained, which leverages the task-based information to predict the final output.

The proposed model (MTL-CN) illustrated in Figure 5.2, the words of the input posts are used to generate the word vector through a pre-trained distributional word embedding model. These generated word vectors have considered as input for the Bi-LSTM layer. It captures forward and backward relations among the words. These relation dependencies depend on contextual resemblance among the words, considered semantic features at here. The convolution layer with different filter sizes (treated as n-gram for text usually) has been deployed over Bi-LSTM's output, assuming that phrases are constructed through contextual words. The pooling operations have been performed over the obtained outputs and concatenated to generate the **C**ontextual **F**eature represented by CF. Along with CF, emotion features are also leveraged by the pretrained DeepMoji model. The pretrained

Deepmoji model takes posts as input and it has its own tokenizer and text encoder to generate **E**motion **F**eatures. Here, we have used this for leveraging features only, which EF represents.

The obtained feature representations (CF and EF) is considered as shared features among tasks. Task-specific information is projected to the task projection layer that transforms it into the same task-specific shared space. Now, the penultimate layer of the task-specific model can inherently share different task information since task-specific information is projected into the same space by the task projection layer. The proposed model is flexible for increasing the task by adding task projection and penultimate layers. Similarly, if there is only one task, the model reduces to a single task learning.

The working flow of the model is as follows:

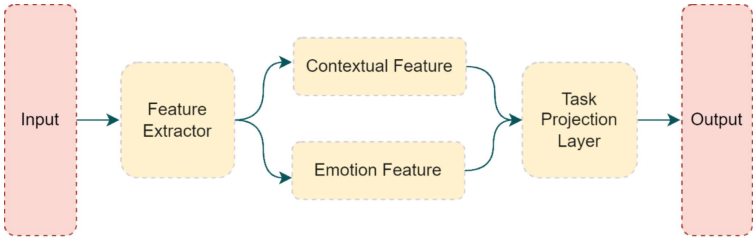

FIGURE 5.1 Overview of the MTL-CN architecture.

1. Let a post text has words $x_1, x_2, \ldots, x_n \in X$, where X is the entire vocabulary of words in our dataset. The obtained semantic feature, F^s will be:

$$F^s_{1,\ldots,n} = \overrightarrow{\text{LSTM}}(w_1, \ldots, w_n) \oplus \overleftarrow{\text{LSTM}}(w_1, \ldots, w_n) \tag{5.1}$$

Here w_i is the embedding vector of $i^{th} \in X$ word.

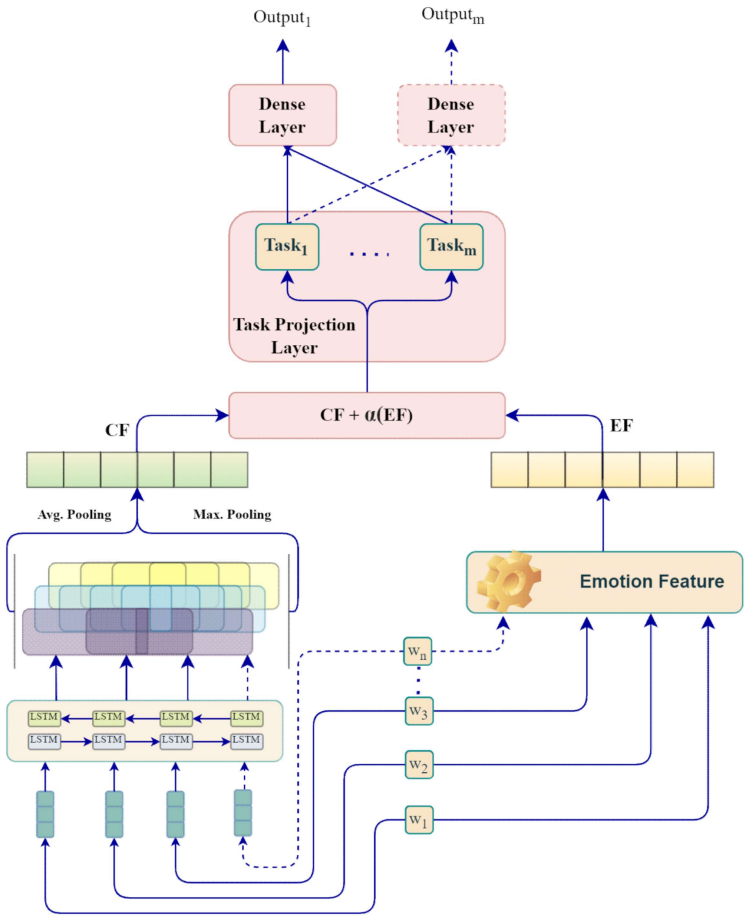

FIGURE 5.2 Emotion based MTL-CN architecture.

2. The CF has calculated over the semantic features. We have calculated the CF in two ways: Maximum pooling and Average pooling on the convoluted output, which keeps the different size of semantic features.

$$CF^{max}_{1,\ldots,n} = \text{MaxPooling}(\text{Conv}(F^{s}_{1,\ldots,n})) \tag{5.2}$$

$$CF^{avg}_{1,\ldots,n} = \text{AvgPooling}(\text{Conv}(F^{s}_{1,\ldots,n})) \tag{5.3}$$

$$CF = CF^{max}_{1,\ldots,n} \oplus CF^{avg}_{1,\ldots,n} \tag{5.4}$$

3. Used the pretrained Deepmoji model to extract the EF from the input that generates the a feature vector.

$$EF = \text{Deepmoji}(x_1, \ldots, x_n) \tag{5.5}$$

4. Although the parameter values of Deepmoji did not update at the time of model training, both these features have been concatenated with weightage or threshold values (α) to prevent model overfitting.

$$F = CF + \alpha(EF) \tag{5.6}$$

5. The final obtained feature, F of a particular input post text is passed to the task projection layer. This layer follows the linear transformation into a fixed dimensional vector, which projects the F to a common space, $\hat{F}$ that is considered by a task-specific penultimate layer.

$$\hat{F}_{1:m} = W.F \tag{5.7}$$

6. Before predicting the output for the particular task, it has a classification layer. The feed-forward layer is considered a classification layer with the number of units. The number of units depends upon the number of classes (c) in the tasks (m).

$$\hat{P}^c_i = \text{Softmax}(W.\hat{F}_{1:m} + b) \tag{5.8}$$

Where, $i \in [task_1, task_2, \ldots, task_m]$. The overall goal of learning is to reduce the cross-entropy loss $\mathscr{L}$ for each instance with respect to the target task through the following equation:

$$\mathscr{L}_i = \sum_{c=1}^{M} y_i^c \log(\hat{P}_i^c) \tag{5.9}$$

The pseudo-code of emotion based MTL-CN model learning algorithm is described in algorithm 1.

Algorithm 1: Emotion based MTL-CN for Hate and Offensive content classification

Input: N number of training instances, α as weightage factor, $task_i$
Output: Label prediction
while i = *1,2,...,N* **do**
  `/* For` i^{th} `training example */`
  Obtain word vectors $w_{1:n}$
  for $j = 1$ *to* n **do**
    `/* With each word vectors */`
    Obtain semantic features $F^s_{1:n}$ by **Eq. (5.1)**
  end
  `/* With each semantic features */`
  Obtain contextual features, CF_i by concatenation of max and avg-pooling features according to **Eq. (5.4)** `/* For each training example */`
  Obtain emotion features, EF_i by **Eq. (5.5)**
  Obtain final features, F_i by **Eq. (5.6)**
  Apply task projection layer on F_i by **Eq. (5.7)**
  Calculate label prediction and loss according to **Eq. (5.8) and (5.9)**
  $i \leftarrow i+1$
end

Multi-task learning considers several task specific nodes, adapting since related tasks share common information via robust feature representation. A different task needs a separate layer. For compiling multi-task learning, we have finetuned our model for the task, i.e., $task_i$ on the model trained to all tasks, $task_{1:m}$.

5.3 Experimental Setup

This section presents the experimental settings of our proposed method. First, we introduce the used datasets, which correspond to Twitter and Facebook posts. Then, to facilitate

the replicability of our results, it shows the implementation details along with the post pre-processing to the proposed emotion-based MTL-CN.

5.3.1 Datasets

The MTL-CN model has experimented on different datasets for hate speech and offensive content detection. Here, we have considered six datasets where four datasets are based on English posts that are FIRE-2019, PEI-2019, SemEval-2019 and FIRE-2020 and two datasets, FIRE-2019 and PEI-2019 also contain Hindi posts. Each dataset (except FIRE-2020) has comprised three tasks that belong to either the binary or multi-class category. Statistics of the datasets is summarized in Table 5.1 and Table 5.2, indicating the categories assigned to each task along with their frequency.

Table 5.1 Label distribution in each task of PEI and FIRE datasets

Dataset (Language)	Task-A		Task-B				Task-C			Total
	NOT	HOF	NONE	HATE	PRFN	OFFN	NONE	TIN	UNT	
PEI-2019 (English)	1702	1814	1702	1454	-	360	1702	1619	195	3516
PEI-2019 (Hindi)	1240	801	1244	484	-	313	1242	739	60	2041
FIRE-2019 (English)	4456	2549	4456	1267	760	522	4456	2286	263	7005
FIRE-2019 (Hindi)	2909	3074	2909	746	1455	873	2909	2087	987	5983
FIRE-2020 (English)	2253	2355	2368	179	1668	393	-	-	-	4608

PEI-2019, FIRE-2019 and FIRE-2020 datasets followed the same guidelines to annotate the labels irrespective of the language. Moreover, SemEval-2019 has distinct labels for Task B and Task C. For these datasets, the description of labels according to tasks are mentioned below:

Table 5.2 Label distribution in each task of SemEval dataset

Dataset (Language)	Task-A		Task-B		Task-C		
	NOT	HOF	TIN	UNT	IND	GRP	OTH
SemEval-2019 (English)	8840	4400	3876	524	2407	1074	395
Total	13240		4400		3876		

1. **Task A** is a coarse-grained classification of posts into hate speech (HOF) and non-offensive content (NOT). The posts that do not contain offence or profanity or hate have categorized as NOT. In PEI-2019, a post that contains only hate or offensive content has been referred to as HOF, whereas in the remaining datasets, FIRE-2019 and FIRE-2020 have extended the HOF category with profane content.

2. **Task B** represents a fine-grained classification. Hate-speech and offensive posts from Task A are further classified into three categories. The hate (HATE) category has assigned to posts with negative attributes, hateful comments relevant to race, political opinion, sexual orientation, gender, social status, health condition, or similar. Posts that are deteriorating, dehumanizing, insulting an individual, threatening with violent acts are categorized into Offensive (OFFN). Similarly, posts contain unacceptable language in the absence of hate and offensive content. Typically, concerns the usage of obscenity, swearwords, and cursing are classified as Profanity (PRFN). The rest of the posts comes under NONE. The PEI-2019 dataset does not consist of PRFN category.

   Task B Semeval-2019 has two categories, Targeted Insult and Untargeted, the same as the Task C categories of the rest of the datasets (described in the following item).

3. **Task C** further classified the HOF category of Task A into Targeted Insult (TIN) and Untargeted (UNT). As categories' names indicate that posts containing an insult/threat to an individual, group, or others are categorized as TIN and posts containing non-acceptable language or non-targeted offensive and hates speech come under UNT. Similar to Task B, the rest of the posts are comes under the NONE.

   SemEval-2019 has different categories, Individual (IND), Group (GRP) and Other (OTH) for Task C. These focus on who is the target of offences in insults or threats post. If posts target an individual such as a famous person, a named individual, or

an unnamed participant in the conversation, it is categorized in IND. The target of offensive posts is a group of people considered as unity due to the same ethnicity, gender or sexual orientation, political affiliation, religious belief, or other common characteristics, which are labelled as GRP. Apart from the last two categories, The target of offensive posts come under the category of OTH, e.g., an organization, a situation, an event, or an issue.

5.3.2 Pre-processing

Twitter posts contain a lot of noise texts, which will not help in obtaining useful classification features. We perform the following steps to remove the noise before model training:

- We removed all mentions of user tokens "username", retweet mentions "RT username:", URL tokens, "https" tokens, diacritics and emojis.

- For hashtags, we followed the underscore within a hashtag "_ " with white space to regain separate understandable tokens, and we pad the hashtag with white-space as well.

- All the characters, meant for punctuation like |,:,?,+,! were removed along with the numbers.

- All the emoticons were categorized into 4 categories, namely love, sad, happy, and neutral.

- The words are converted to the lower case so that words such as "good", "Good", and "GOOD" have the same syntax and use the same pre-trained embedding values.

- The maximum sequence length is set to 100. Post padding is done if any sentence is less than 100, and pruning is performed from the last if the sentence is greater than 100.

5.3.3 Implementation details

The proposed architecture evaluated on the PEI 2019 dataset, covering the tasks of Hate speech and offensive content identification. According to the analysis of sentence length in the dataset, we set the max_length of the models to be 100. The word embeddings in the model initialized with publicly available pre-trained vectors, created using the glove embedding. For the domain-specific dataset, we used 300-dimensional embeddings for word representation and the same dimensional vector as hidden unit for LSTM.

Sentences are grouped into batches of size 64 and parameters are optimized using Adam with learning rate, $1e^{-3}$ and epochs to be 10 for the model. Training ceased when performance on the development set was not improved after 9 epochs. Performance on the development set was also used to select the best model, which was then evaluated on the test set.

5.4 Results and Analysis

The proposed model, MTL-CN is based on the hard-parameter sharing mechanism while tuning the task-specific outputs. The MTL-CN is compared to the current baseline with respect to language for each dataset in Table 5.3. It provides the F_1-score obtained on PEI-2019, SemEval and FIRE datasets. Figures 5.3 and 5.4 show the accuracy of MTL-CN for all three datasets.

- **PEI-2019 (Hindi)**: For task A, the SGD classifier was used as a baseline technique which achieved a 0.76 F_1-score. The highest F_1-score 0.40 and 0.51 have been

Table 5.3 Comparison of the MTL-CN to baseline on different datasets

Data	Task	F_1-score			
		Hindi		English	
		MTL-CN	Baseline	MTL-CN	Baseline
PEI-2019	Task A	**0.80**	0.76	**0.74**	0.71
	Task B	**0.66**	0.40	**0.63**	0.59
	Task C	**0.59**	0.51	**0.64**	0.59
FIRE-2019	Task A	**0.84**	0.81	0.78	**0.78**
	Task B	**0.66**	0.58	**0.56**	0.54
	Task C	**0.72**	0.57	0.51	**0.51**
FIRE-2020	Task A	NA	NA	**0.90**	0.51
	Task B	NA	NA	**0.54**	0.26
SemEval-2019	Task A	NA	NA	**0.89**	0.82
	Task B	NA	NA	0.75	**0.75**
	Task C	NA	NA	**0.72**	0.66

obtained for tasks B and C, respectively, by using linear SVM. The proposed architecture model, MTL-CN, outperforms both SGD and linear SVM with +0.0474, +0.2693, and +0.0803 for tasks A, B and C, respectively.

- **PEI-2019 (English)**: SGD conferred better performance in English compared to Hindi for all the tasks. However, in comparison to MTL-CN, SGD performance has been inferior for these tasks with 0.0379, 0.0423, and 0.0514, respectively.

- **FIRE-2019 (Hindi)**: The **QutNocturnal** [191] used Convolutional Neural Network for task A, where convolution was applied on top of word embeddings of tweet's word. For task B **3Idiots** [192] considered the BERT model. As compare to these existing models, The MTL-CN outperforms with +0.0337, 0.0883, and 0.1485 for tasks A, B and C, respectively.

- **FIRE-2019 (English)**: The best performing system for task A is **YNU_Wb** [193], which used the attention-based LSTM model with ordered neurons and for task B and task C is **3Idiots** [192], which used the BERT model to this dataset. We have compared all these models with the MTL-CN, where it outperformed with +0.0007, 0.0215, and 0.0063 for tasks A, B, and C, respectively.

- **FIRE-2020 (English)**: The LSTM model with pretrained GloVe embedding has been explored for task A by [194], whereas Chrestotes [195] used BERT and its variants for task B. The MTL-CN outperforms with +0.3886, 0.2753 for tasks A, task B, respectively.

- **SemEval-2019 (English)**: The NULI [76] obtained 0.829 F_1-score for task A using BERT-base-uncased with default-parameters. The **jhan014** [196] model is used the Rule-based approach with a keyword filter based on a Twitter language behavior list, which included strings such as hashtags, signs, etc., achieving a F_1-score of 0.755 for task B. For task C, the **vradivchev_anikolov** [197] has leveraged the BERT model along with pre-trained word embedding, GloVe. The MTL-CN outperforms with +0.0702 and 0.0627 for tasks A and task C, respectively. For task B, the MTL-CN is a modest difference of 0.0034.

5.4.1 Effect of α-value

The proposed model, MTL-CN leverages contextual features and emotion features, which are associated with the α-factor. α regulates the weighting of emotion features. Here, the value of α was kept between 0.01 to 0.05 to show the effect on the model's performance. Figures 5.5, 5.6, 5.7 indicate that the performance consistently improved on all datasets to

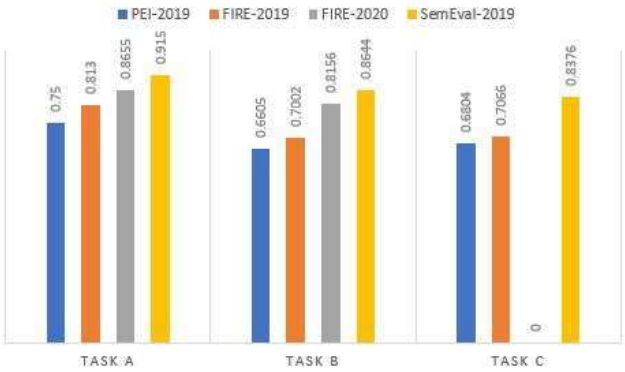

FIGURE 5.3 Accuracy score of MTL-CN for English

FIGURE 5.4 Accuracy score of MTL-CN for Hindi

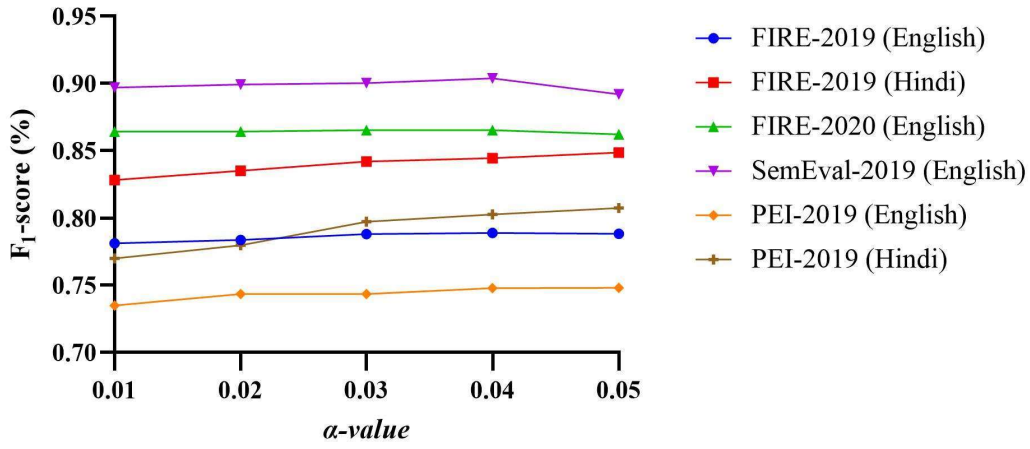

FIGURE 5.5 $F_1 - score$ **of Task A**

0.04 for all three tasks, Task A, Task B and Task C, respectively. With a value of α as 0.05, the model's performance has been stepped on a very small number of datasets while the rest have admitted languish. This implies that the high weighting on emotion features, on making 0.05, the model leads to overfit.

5.5 Summary

Using six different data sets—four for English and two for Hindi—we have discussed multitask learning for categorizing posts. For the classification task, we propose emotion-based Multi-Task Learning with the Convolution Network (MTL-CN) method for the multi-label classification of hate speech and offensive content in Hindi and English languages. We utilized the Deepmoji library to determine emotions. We discovered that a hate speech detection model's capacity to generalize to new datasets and distributions is greatly enhanced by the use of an MTL framework. For all datasets, regardless of the languages, the MTL-CN performs better than the baseline model.

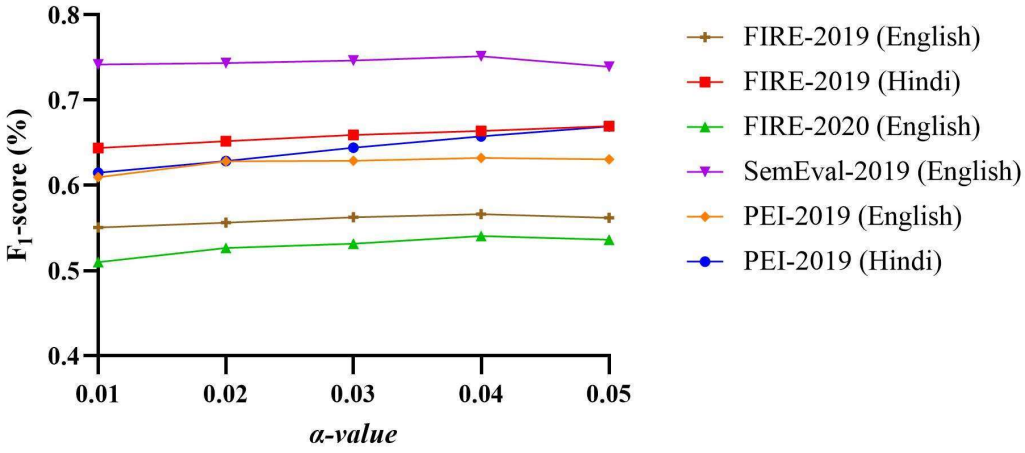

FIGURE 5.6 $F_1 - score$ **of Task B**

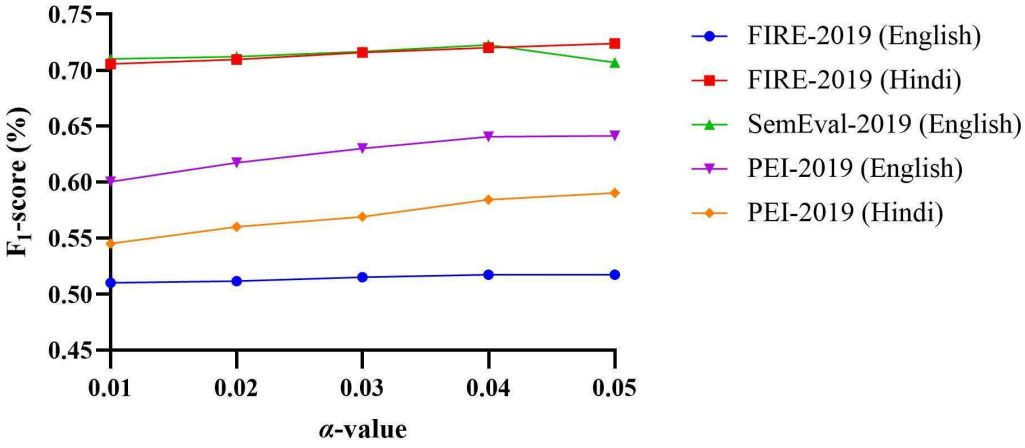

FIGURE 5.7 $F_1 - score$ **of Task C**