Chapter 6

A study on people's reaction during IGE 2019

6.1 Introduction

The word irony is defined as the trope in which the literal meaning is reverse of the intended meaning, [198]. The study of irony on social media has gained a lot of attention in recent years. Social media is loaded with lots of ironic tweets. Understanding of irony requires the knowledge of the context in which a discussion is going on. Even though it is interesting and may be easy for the participants, it is often not so for strangers who do not have a complete knowledge of the scenario behind irony posts. Automatically identifying such posts is not trivial as they should not be taken for their literal meaning. During elections in India, ironic tweets go way high in number and this can be well used for sentiment analysis of people.

Automatic detection of irony is crucial for the development of irony-aware sentiment analysis systems. It is also an exciting conceptual challenge from a cognitive point of view and can help to shed some light on how people use irony as a communicative tool. Even

though irony has been a topic studied in various disciplines like linguistic, philosophy, and psychology, it is not very easy to define in formal terms. However, most theorists would agree that emotions play a role in the use of irony in different aspects. The important role of affected information for irony communication-comprehension has also been emphasized by recent psychological findings as [30, 31].

Although it may be easy for humans to understand a sentence as ironic, making a machine understand it, is non-trivial. For the classification of posts, we take into consideration only the text part of the posts and not its hashtags, emojis, timestamp, and location. We use different machine learning techniques on IGE 2019 data and SE-2018 T3 data-set to predict the ironic posts. We ensemble machine learning techniques using the voting classifier. We also train deep learning techniques on IGE 2019 data and SE-2018 T3 data and ensemble them using a weighted average method.

The task was introduced as binary classification of the tweets into irony or non-irony. The task focuses on identification of irony text during the election in India 2019, a sentiment-related classification task on domain-specific data. We explore different machine learning and deep learning models. Our approaches consist of ensemble of machine learning technique and ensemble of BERT and ELMo models (EBEM) for classification and domain adaptation on each ensemble setting (EMLT and EBEM). Specifically, we adopt voting-based ensembling in machine learning. With deep learning, we ensemble the BERT and ELMo models with the weighted average method. We perform domain adaptation on both the ensembled models for checking the results on domain-specific data with the SemEval-2018 Task 3 dataset.

6.2 Contribution

Researchers have been using a number of datasets for irony detection tasks in social media. Most of these datasets are taken from Twitter as this is publicly available without any privacy issues. We collected posts from Twitter and Facebook for our research. But unlike other datasets where the posts are mostly from general domain, ours is from a specific domain: general election in India.

6.2.1 Characteristics of posts

The dataset has a few special features some of which distinguish our research from previous ones.

- **Defining irony and non-irony:** Irony is defined as a figure of speech in which the conscious meaning is inconsistent with the words used [199]. People express irony on social media in various ways, some of them being the use of emojis or hashtags at the end or in between of their posts contradicting the meaning of their post.

- **Post Length:** Twitter is a free social networking and micro blogging service that allows users to post and read short messages (from a maximum of 140 characters earlier to 280 chars now) as tweets [42]. Facebook is a social networking site that allows users to connect with people worldwide and share materials with them. Facebook posts can have a maximum of 63206 characters [200].

- **Post availability:** For Twitter data, we use the REST API in our program to download tweets. REST API takes words as queries, and multiple queries can be combined as a comma-separated list. For Facebook data, we used the Facepager tool [186].

- **Post Language:** Social media users post messages from many different media, including their smartphones. The frequency of misspellings and slang are way higher in social media posts than formal text. We used English posts here.

- **Post Domain:** Social media users post short messages about a variety of topics. For example, they can be on specific topics like sports, movie reviews, trips, politics, celebrities, technology, etc. We focus on Indian election in particular.

We collected data from Facebook and Twitter during the Indian General Election between 11 April to 25 May, 2019 (IGE-2019) which focused on public opinions towards participating candidates. During the run-up of election campaign, vitriolic attacks and counter-attacks happen between candidates and political parties and their supporters. Some are straight-forward while some are ironic.

Table 6.1 Statistics of Datasets

No. of posts	Training	Testing	Total
IGE-2019	810	271	1081
SE-2018 T3	3834	784	4618

Our IGE - 2019 data is thus x domain-specific, focused to general election of India. We manually label the dataset binarily with '1' for irony tweets and '0' for non-irony tweets.

The IGE - 2019 data consists of total 1081 tweets, out of which 810 tweets are used for training and 271 tweets used for testing. For the SE-2018 T3 data, training set consists of 3,834 tweets and test set of 784 tweets. The highest classification score obtained of SE-2018 T3 data for binary classification is $F_1 =$ **0.71** at SE-2018 Task 3 [201]. Table 6.1 shows the data distribution of IGE-2019 and SE-2018 T3 dataset. One example each of irony or non-irony posts are given below.

FIGURE 6.1 An example of irony posts

FIGURE 6.2 An example of non-irony posts

6.2.2 Annotations

Each post is manually labelled by two independent annotators. They are under-graduate engineering students whose first language is Hindi but are educated in English medium with very good reading, writing and speaking skill in English. Inter-annotator agreement is found to be 0.9780 for English language [202].

6.2.3 Preprocessing

Preprocessing is a very important step and has a huge impact on overall performance. We first convert the tweets into lowercase. Then we replace all the links with the word “URL” and all the numbers with the word “number". We remove the leading and trailing white-spaces and replace white-spaces between words with a single space. We also remove the punctuation and stop words. We use Porter Stemmer to find word stems from the posts [203]. We use bag-of-words model to extract features from the posts [204].

6.3 Models

The task of irony post identification is considered as a binary classification problem. The classification done by eight different machine learning techniques and two deep learning techniques.

6.3.1 Machine Learning

The following state-of-the-art machine learning models are used for the binary classification of the posts.

- **k-Nearest Neighbors (kNN):** Due to the feature space and the closest training samples, this algorithm can predict and classify objects. A prediction depends on the nearest neighbor is given with percentage of confidently, this prediction result obtained firstly by checking the feature space; this is really how the KNN algorithm method work [205]. It assumes that similar data are proximal to each other. It works by finding the mode (for the tasks of classification) of 'k' most nearest data-points from the data-point in question (any suitable distance metric may be considered) [206].

- **Decision Tree (DT):** Decision Tree essentially classifies the training data into sets. These sets are formed by branching on the attribute values of the examples in the training data. A perfect DT is formed when all training examples at a node in the tree are under the same classification. However, this phenomenon rarely happens because of the presence of a few outliers due to noise [207]. It is like a tree with hierarchical decisions at each node and progressing from root node to leaf node to arrive at final classification prediction. We focus to get higher accuracy and build as small tree as we can [206].

- **Random Forest (RF):** Ensembling of multiple decision trees' work together yields a Random Forest. Each decision tree makes a prediction and the prediction with the most number of polls is our final result. When splitting a node, we consider only some features from the random subset of features [206].

- **Logistic Regression (LR):** Logistic regression (LR) gets its name from the logistic curve, or sigmoid. Because this curve approaches zero and one with a controllably quick transition, it is well-suited to binary classification [208]. Logistic regression is an apt choice when binary classification task is to be done [206].

- **Multinomial Naive Bayes (MNB):** It is a classification model based on the Bayes' theorem. It is a probabilistic machine learning model. It is based on an assumption that the features are independent of each other and presence of a feature does not affect as other feature [206].

- **Stochastic Gradient Descent (SGD):** Stochastic gradient descent is an iterative method for optimizing an objective function with suitable smoothness properties. It takes only few data points from the whole sample in each iteration. It appraises only 1 random data point when changing weight [206].

- **Support Vector Machine (SVM):** It classifies data points by finding a hyperplane in N-dimensional space where N is the number of features. For binary classification, hyperplane is that one which has maximum distance from boundary data points from both the classes [206].

- **XGBoost:**: XGBoost is short for the eXtreme Gradient Boosting, a scalable machine learning system for tree boosting. It makes use of gradient boosted decision trees. For tabular data, XGBoost outperforms most of the algorithms [209].

6.3.2 Deep Learning

It is a part of machine learning that brings in use of multiple layers to extract higher level features progressively from the input. We made use of the ELMo and BERT models to classify the posts as irony and non-irony.

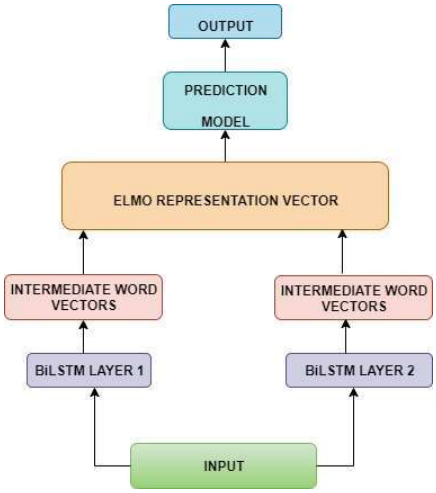

FIGURE 6.3 Working module of ELMo

- **ELMo:** It stands for Embeddings from Language Models [210]. It makes use of deep Bidirectional Long Short Term Memory (Bi-LSTM) to create word representation (Figure 6.3). It is a deep contextual character model, thus it can form representations for out-of-vocabulary words. We set the trainable values to false, and the activation functions as ReLU and sigmoid and use Adam algorithm as optimizer to our networks. We provide posts as input to the ELMo, where the words are first converted into raw word vectors using character-level convolutional neural network. The raw vectors are input to the bidirectional language model and undergo a forward pass as well as a backward pass. A forward pass stores the information of that word and the reference before that word and produces intermediate word vectors. The intermediate word vectors are used in next bidirectional language model. The backward pass contains information about that word and the subsequent meaning of

that word.

The final ELMo representation of the original input is the weighted sum of the raw word vectors and the intermediate word vectors from the two bidirectional language models. Now, this final representation is passed into our model, which makes predictions.

- **BERT:** BERT stands for Bi-directional Encoder Representations from Transformers [211]. It is based on a transformer and reads the input from both directions at once (Figure 6.4). It uses two training approaches, namely Masked Language Model and Next Sentence Prediction. In our approach, it is a pre-trained model, so fine-tuning is done to use it for a specific task. The BERT model was first used to classify sentiments [4] for the non-English dataset and the model did not perform well. But, the model gives a good result on the English dataset.

We use rectified linear units (ReLU) and sigmoid activation function, and Adam algorithm as an optimizer. Input data is converted in the form of input representation of BERT. In the training process, the model receives pair of sentences as input and learns to predict whether the second sentence in the pair is the subsequent sentence in the original document. The model differentiates between two sentences during training by adding CLS token at the beginning of the first sentence and a SEP token at the end of each sentence. Approximately 15 percent of the words in the input is masked. The input data is converted into a combination of token embedding plus sentence embedding, the transformer positional embedding. The transformer encoder reads the input embedding. The final embedding is fed into our model which makes the predictions.

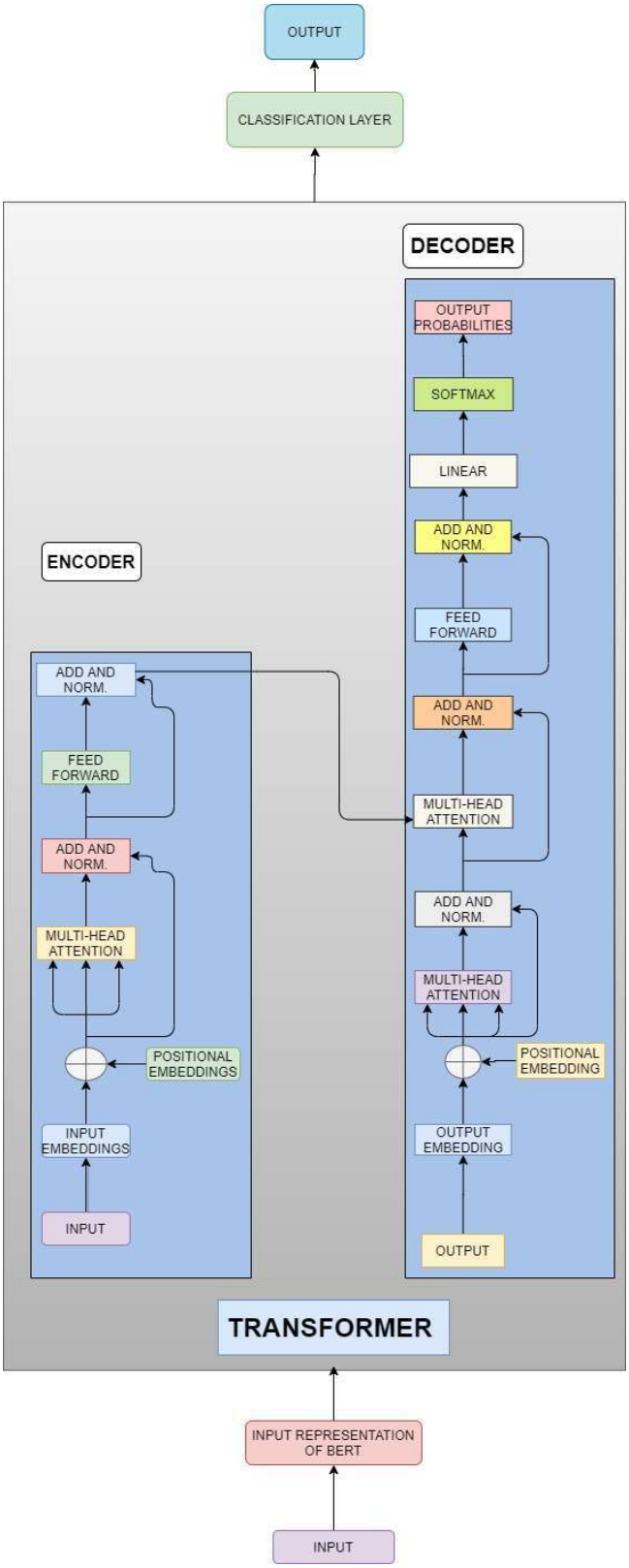

FIGURE 6.4 Working module of BERT [4]

- **NIHRIO System:** NIHRIO system [212] was used for SemEval-2018 Task 3 Irony detection in English tweets. It is one of the best reported systems comparable to our task of binary classification. The authors propose a simple NN model of MLP with different type of input features like: lexical, syntactic, semantic and polarity features. Model parameters are learned to minimize the cross-entropy loss with L2 regularization. They used 10-fold cross-validation based voting strategy to split the training set. Their system achieved third rank using the accuracy metric and fifth using F_1 metric. We use the system for comparison against our proposed ensemble-based system.

6.4 Our Approaches

The main target of the proposed work is to identify whether a given sentence is ironic or non-ironic. We adopt a principled approach of exploring a set of classification techniques. First, we consider a set of 8 different ML techniques on both the datasets (IGE-2019 and SE-2018). We then consider two deep learning based techniques (BERT and ELMo) on both the datasets. Later, we consider ensembling-based techniques.

Ensemble learning is a learning algorithm that combines multiple machine learning techniques or deep learning techniques into one predictive model to reduce variance, bias, or improves predictions. Ensemble is a technique which combines the output of a number of classifiers resulting into a composite classifier that performs better than the individual classifiers [213].

Our ensembling-based experiments can be classified into three sub-sets as given below.

- Ensemble of Machine Learning techniques (EMLT)

- Ensemble of BERT and ELMo models (EBEM)

- Domain adaptation on each ensemble setting (EMLT and EBEM)

We detail each of them below.

6.4.1 Ensemble of Machine Learning techniques (EMLT)

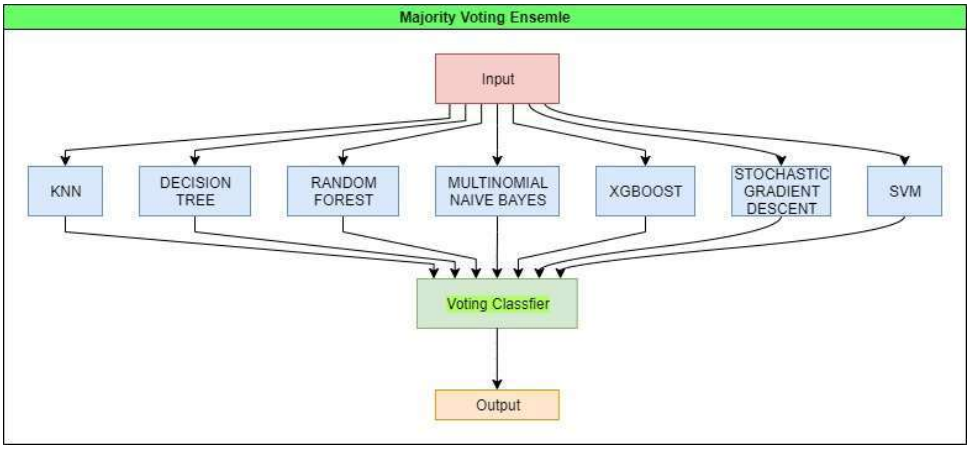

FIGURE 6.5 Machine learning techniques based on Majority Voting Ensemble

In ensemble learning, many learners are called 'base learners' that are trained to make classifications on the same problem. The essence of ensemble-based decision making is actually group-based decision, and it is a natural decision-making process.

Figure 6.5 shows the machine learning ensemble with majority voting to get the best result out of eight machine learning techniques. The majority voting approach counts the votes of all the participating models and selects the class with the maximum votes. The voting uses bootstrap samples to obtain data subsets to train base learners. We use voting for classification to collect the output of learners.

6.4.2 Ensemble of BERT and ELMo models (EBEM)

We ensemble BERT and ELMo embedding using Multi-Layer Perceptron and weighted-average ensemble technique. We separately pass the BERT and ELMo embeddings into their MLP models, each having four hidden layers. Prediction from their MLP is used to ensemble the models using the weighted average technique. Figure 6.6 shows the working of the ensembling of BERT and ELMo.

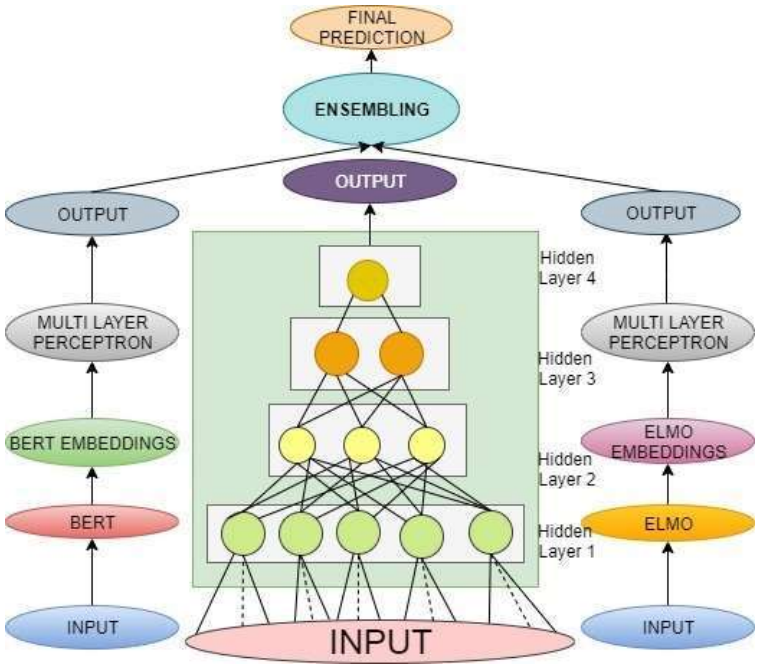

FIGURE 6.6 Ensemble of BERT and ELMo model

6.4.3 Domain adaptation on each ensemble setting (EMLT and EBEM)

Domain adaptation arises when we build a well-performing model learning from a source data distribution but apply on a different (but related) data distribution [214].

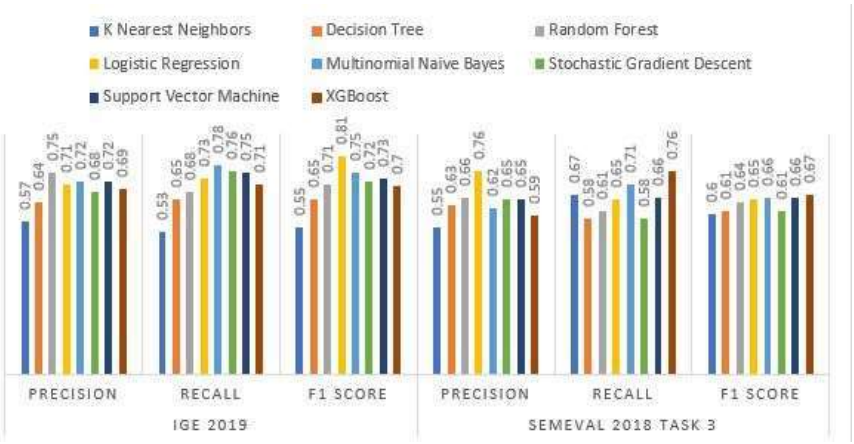

FIGURE 6.7 Performance on IGE-2019 and SemEval-2018 in terms of Precision, Recall, F_1-scores

IGE - 2019 data is from election domain, whereas SE-2018 Task-3 (sub-task A) data is from mixed domain. However, the task is the same: classification between irony or non-irony posts. To find the effectiveness of domain adaptation, training and testing are changed, i.e. train on IGE-2019 data (SE 2018 Task 3 (sub-task A)) and test on SemEval 2018 Task 3 (sub-task A) (IGE-2019) for all the techniques used in our experiments. The SE - 2018 Task 3 (sub-task A) dataset includes various domains, while IGE - 2019 only falls under the election domain.

For all the experiments, training and testing are done according to section 6.2.1.

6.5 Results and Analysis

We use binary classification. Our goal is to classify a given post from social media into irony or non-irony class. We measure the performance of a model using accuracy, precision, recall, and F_1-score.

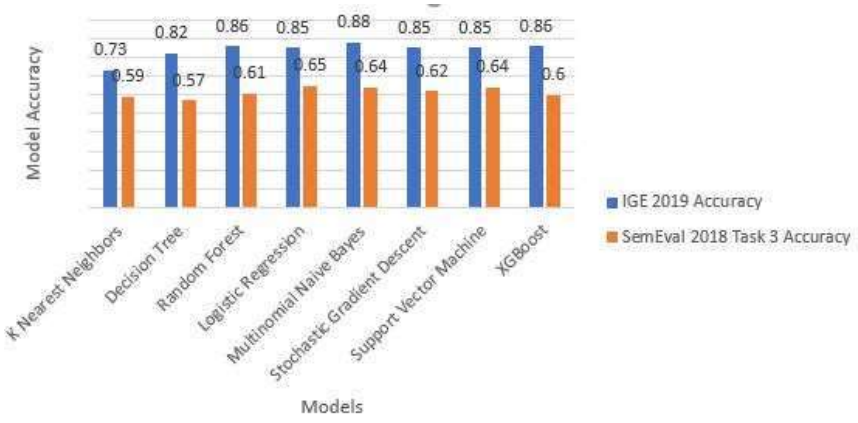

FIGURE 6.8 Accuracy of machine learning models on IGE 2019 and SemEval 2018 datasets

6.5.1 Performance of ML techniques

Figure 6.7 shows the results of different machine learning classifiers on IGE - 2019 and SemEval 2018 datasets. The F_1 scores obtained for IGE - 2019 are quite comparable to that obtained with SemEval 2018 using four classifiers RF, LR, SGD, and SVM (the best F_1 score at SemEval 2018 Task 3 of Sub-task A was 0.705). In our experiments, the SemEval 2018 dataset consistently yields lower F_1 scores compared to IGE - 2019. This is possibly due to the fact that IGE - 2019 is a domain-specific data vis-a-vis general nature of SemEval - 2018. Irony in a given domain has some stereotypes and therefore possibly easy to detect once trained with good amount of data. On the other hand, in a general domain irony is more diverse and multifaceted, and therefore, computationally more challenging to capture.

Figure 6.8 shows the accuracy figures of machine learning models on both IGE 2019 and SemEval 2018 Task 3. We train models on IGE 2019 and do not use any pre-trained features as ready information for models. We achieve maximum accuracy of **0.88** for IGE 2019 whereas it is much lower for SemEval 2018 (maximum 0.65). The difference can be attributed to the phenomena of specificity of IGE 2019 vs generality of SemEval 2018.

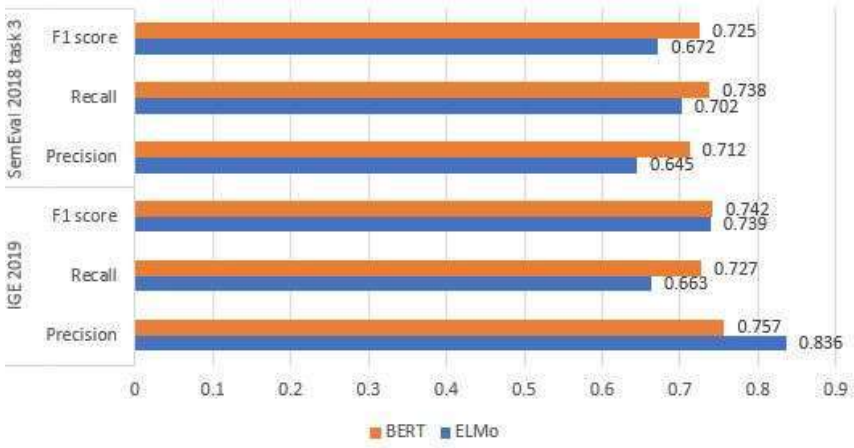

FIGURE 6.9 Performance of deep learning models on IGE 2019 with SemEval 2018 dataset

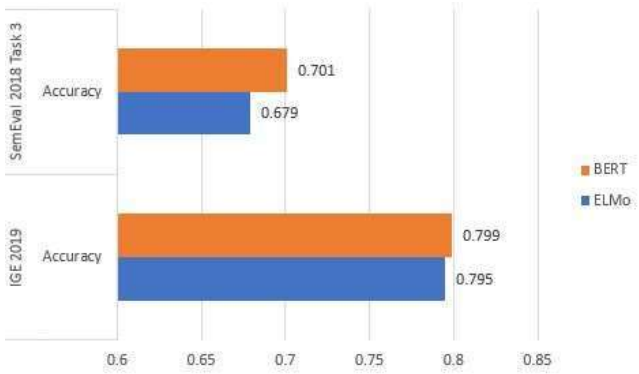

FIGURE 6.10 Accuracy of deep learning models on IGE 2019 and SemEval 2018 datasets

6.5.2 Performance of deep learning techniques

Figure 6.9 shows the precision, recall and F_1-scores of BERT and ELMo on IGE 2019 and SemEval 2018 data. The F_1-scores on IGE 2019 are higher than those with SemEval 2018 data. The F_1 scores of BERT and ELMo are comparable on IGE 2019 data but BERT performs better on SemEval 2018. In terms of accuracy scores (Figure 6.10), the observations are the same.

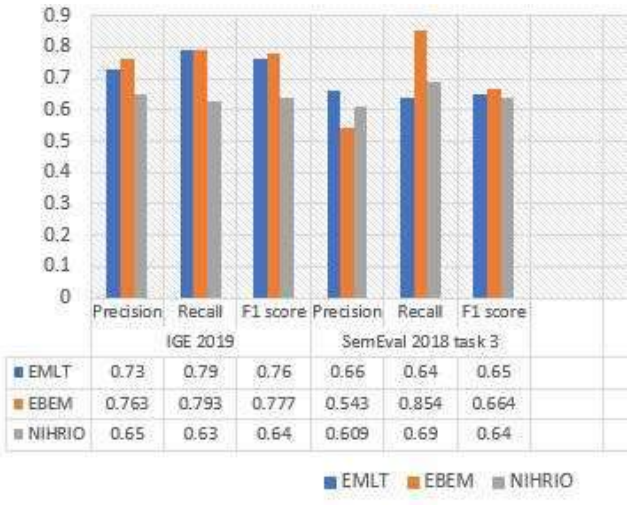

FIGURE 6.11 Ensemble results of IGE 2019 with SemEval 2018 Task 3 data-set

6.5.3 Performance of ensembling techniques

Figure 6.11 shows the results of our proposed ensembling models. The ensemble of machine learning techniques (EMLT) on IGE 2019 data outperforms that on SemEval 2018 data as far as F_1 scores are concerned (0.76 and 0.65 respectively).

The ensemble of BERT and ELMo (EBEM) on IGE data vs that on SemEval 2018 data also show the similar pattern in terms of F_1 scores (0.77 vs 0.66).

We also perform the classification with one of the best reported models (NIHRIO at SemEval-2018 Task 3) which is MLP-based neural network model [212]. All the models perform better in term of accuracy and F_1-scores on IGE 2019 data compared to their reported results in SemEval-2018 conference. Also, our ensemble-based techniques (EMLT and EBEM) are comparable and/or even better compared to the state-of-the-art NIHRIO technique.

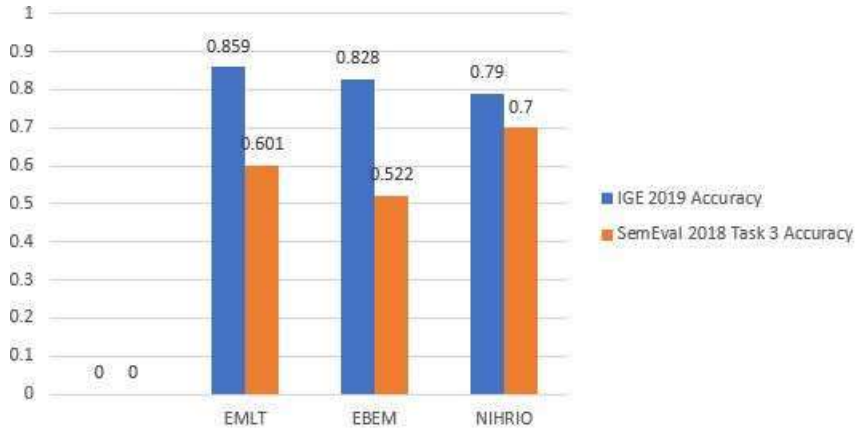

FIGURE 6.12 Accuracy of ensemble of deep learning and machine learning models on IGE 2019 and SE 2018

Figure 6.12 shows the accuracy of both the ensemble models along with NIHRIO. The accuracy on IGE 2019 data is also better compared to that on SemEval 2018 Task 3 data.

6.5.4 Results of Domain Adaptation

Table 6.2 Results of our proposed models with comparison of SemEval (SE) 2018 Task 3 dataset

Models	Train	Testing	Accuracy
ELMo	IGE 2019	SE 2018 Task 3	0.451
ELMo	SE 2018 Task 3	IGE 2019	0.656
BERT	IGE 2019	SE 2018 Task 3	0.530
BERT	SE 2018 Task 3	IGE 2019	0.718
EBEM	IGE 2019	SE 2018 Task 3	0.502
EBEM	SE 2018 Task 3	IGE 2019	0.55
EMLT	IGE 2019	SE 2018 Task 3	0.498
EMLT	SE 2018 Task 3	IGE 2019	0.552

In domain adaptation, training and testing data are different. We train on IGE 2019 data (SemEval 2018) and test on SemEval 2018 (IGE 2019) for all the techniques that are used in our experiments.

Table 6.2 summarizes the performance of our experiments on domain adaptation in terms of accuracy. The accuracy figures are much lower here compared to their non-adaptation counterparts (when the same data is used for both training and testing).

When SE-2018 is used to train, and the techniques are tested on IGE 2019 dataset, accuracy improves but not vice-versa. The scores on IGE 2019 are much higher than those with SemEval 2018 tasks for all 4 techniques such as ELMo, BERT, EMLT and EBEM. The gain in accuracy over SemEval 2018 data are +20.51, +18.85, +2.84 and +5.37 in percentage points respectively for the above techniques.

Training on a dataset of general domain (SE 2018) can work (although with reduced accuracy) on a dataset of specific domain (IGE 2019), but the reverse is not true.

6.6 Summary

Several ML techniques are tried, along with two deep learning techniques for irony detection in our study. LR appears to be the most effective and reliable ML method for both datasets in terms of F_1 scores. K-NN does not seem to work well for irony detection. However, as far as accuracy is concerned, NB performs the best. The deep learning techniques are not found to be better than ML ones for irony detection tasks on the IGE-2019 dataset, however comparable (F_1 scores) or better (accuracy) on SE-2018 data (See Figure 6.7, 6.8, 6.9, 6.10). Deep learning models are pre-trained on data of a general domain, so they work well for SE-2018 data, but they can not perform the same on domain-specific data. Because of its inherent capability of capturing context through deep bi-directional nature at multiple layers, BERT exhibits better performance than ELMo.

In the ensembling experiments, EMLT performs better than the average of the individuals in comparison for both the datasets but can not beat the best performer (see Figure 6.7, 6.11). On the contrary, ensembling betters individual performances of deep learning techniques. The improvement is particularly prominent in the SE-2018 dataset since deep learning models are pre-trained in the general domain.

In the domain adaptation experiments also, BERT overshadows ELMo individually and provides the best accuracy among all. Here, ensembles do not work well in general. Nevertheless, it is useful and can offer decent results in a resource-constrained environment when training data is insufficient.