Human-Centric Intelligent Systems
Vol. **1(3-4)**; December (2021), *pp*. 66–74
DOI: https://doi.org/10.2991/hcis.k.211109.001; eISSN 2667-1336
https://www.atlantis-press.com/journals/hcis

Research Article

Pretrained Natural Language Processing Model for Intent Recognition (BERT-IR)

Vasima Khan[1,*], Tariq Azfar Meenai[2]

[1]*Department of Computer Science & Engineering, Sagar Institute of Science & Technology (SISTec), Bhopal, Madhya Pradesh, India*
[2]*Department of Electronics & Communication, Smith Infotech Pvt. Ltd., Bhopal, Madhya Pradesh, India*

ARTICLE INFO

Article History

Received 15 July 2021
Accepted 25 October 2021

Keywords

Intent recognition
intent detection
natural language processing
BERT
deep learning
deep neural network

ABSTRACT

Intent Recognition (IR) is considered a key area in Natural Language Processing (NLP). It has crucial usage in various applications. One is the Search Engine- Interpreting the context of text searched by the user improves the response time and helps the search engines give appropriate outputs. Another can be Social Media Analytics-Analysing profiles of users on different social media platforms has become a necessity in today's applications like recommendation systems in the online world, digital marketing, and a lot more. Many researchers are using different techniques for achieving intent recognition but getting high accuracy in intent recognition is crucial. In this work, named BERT-IR, a pre-trained Natural Language Processing model called as BERT model, along with few add-ons, is applied for the task of Intent Recognition. We have achieved an accuracy of 97.67% on a widely used dataset which shows the capability and efficiency of our work. For comparison purposes, we have applied primarily used Machine Learning techniques, namely Naive Bayes, Logistic Regression, Decision Tree, Random Forest, and Gradient Boost as well as Deep Learning Techniques used for intent recognition like Recurrent Neural Network, Long Short Term Memory Network, and Bidirectional Long Short Term Memory Network on the same dataset and evaluated the accuracy. It is found out that BERT-IR's accuracy is far better than that of the other models implemented.

1. INTRODUCTION

1.1. Overview

In a recent scenario, research work corresponding to the domain of NLP and Computational Linguistics undergoes drastic improvements due to which a wide range of applications of this field has been evolved. Significant factors like the availability of enormous data for training Machine Learning Models, enhancement capable enough model-building techniques, and tremendous growth in computational power caused these improvements. Natural Language Processing is a study of human-computer interactions which comes under Artificial Intelligence [26]. It relates to the analysis of data generated by natural language. Since human language contains colloquialism, variability, ambiguity and is interpreted as per the context, it is associated with several issues in both spoken and written form. As the application areas in the domain of NLP are widely increased, numerous algorithms are getting invented.

The intent defines the context of the text, which is usually a combination of a verb and a noun. A few instances could be SearchRestaurant, OrderFood, and others [17]. Finding out the context corresponding to the user's text is known as Intent Recognition, also known as Intent Classification or Detection. When we deal with a Machine Learning scenario, Intent Recognition is a classification task in which there are predefined intents to which we classify user text [5]. As already specified that human language contains several constructs which are very complicated to handle, which makes this task of intent recognition a highly complex problem [19].

Numerous NLP applications, including Intent Recognition, use pre-trained models with self-attention encoder architectures [7,17]. Such models are self-supervised trained on a massive amount of text taken from Wikipedia [34]. After fine-tuning, these pre-trained models have been used in several downstream tasks, including NLP, which have given breakthrough results. However, it has been shown in previous work [23,31] that certain deficiencies are present in terms of accuracy if fine-tuning is done directly for finding the intent. One possible reason could be the length of the text message because sometimes only keywords are not enough to detect the intent [2]. Another reason could be a large number of intents. Many techniques are used to detect intent from text, from traditional machine learning techniques like SVM, AdaBoost, and Logistic Regression to deep learning techniques like RNN, CNN, and LSTM. However, as conveyed earlier, performing this task of intent recognition with high accuracy is difficult.

1.2. Author's Contribution

In our research work, we have used a recent model known as BERT (Bidirectional Encoder Representations from Transformers), which is a language representation model [7,21,22]. This model is developed to pre-trained deep bidirectional representations [7].

* *Corresponding author. Email: drvasimakhan88@gmail.com*
Peer review under responsibility of KEO (Henan) Education Technology Co. Ltd

We have applied this model with some add-ons while fine-tuning to the task of intent recognition. Although BERT is a simple and successful model, using this for the task of intent recognition has been proposed by our work for the first time. Hence it is a significant author's contribution with respect to research.

Also, there were many issues like data format compatibility, the difference in the kind of words used for previous training in BERT and those which were present in our application data, acquiring high accuracy, and a lot more. After dealing with all the issues and performing rigorous fine-tuning, we obtained very high accuracy and performance. This was the author's contribution in terms of implementation.

As we talk in terms of experimental view, our model's outstanding performance has been shown by the results we got. Our model has given excellent results compared to other previous approaches used in the task of intent Recognition. This work could be the baseline for a lot of recent key application areas like Human-Computer Interactions, Duplicate Question Problem, Stock Prediction and many more.

1.3. Organization of Research Article

The organization of the remaining article is as follows. In the next section, previous research done has been explained. The baseline model used in our work has been explained briefly in Section 3. Section 4 discusses our approach for intent detection. Section 5 outlines the experimental details, including dataset used, data preprocessing, implementation setups, and model training. Experimental results containing the error and comparative analysis are reported in Section 6. At last, the paper is concluded in Section 7. Also, we have mentioned the scope for future work in the same section.

2. EXISTING APPROACHES

2.1. Traditional Approaches

During the last few years, various researchers have considered intent detection a Semantic Utterance Classification (SUC) task [6]. Initially, a semantic recognition approach based on rules and statistical features-based classification techniques was advised for intent detection [1]. Even though the rule-based technique is particular in terms of accuracy and needs comparatively much less data for training, it has many issues. The main problem with this method is that it has to be reconstructed from scratch [17].

2.2. ML Techniques

Statistical Feature classification technique requires that out of the huge text, key features need to be extracted [17]. If key features are extracted manually, it causes the cost to become very high, at the same time it cannot gives accurate results in terms of features selected which ends up in sparse data issues. Naive Bayes [20], Adaboost [27], Support Vector Machine (SVM) [10], and logistic regression [9] are few traditional technique examples. Traditional approaches to intent recognition are not capable of finding the actual context of raw text data taking into account the amateurishness of the text data [17].

2.3. DL Approaches

As the advancement in the deep learning field has gradually happened, Researchers applied many deep learning techniques in intent recognition problems, for instance, Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) Network, Gated Recurrent Unit (GRU), word embedding, Attention Mechanism and Capsule Neural Networks [33]. In contrast to traditional techniques, deep learning methods gave better results but still, there is excellent scope for improvement in terms of performance. Work done in these approaches are as follows:

2.3.1. CNN

Primitively, Image processing was the central area where CNN has been used [15]. Afterward, several researchers applied it to NLP tasks and got good results as well. [14] proposed improved performance model for text classification task using CNN. Further, CNN is used to recognize user queries by extracting features in the form of vectors [11]. This technique to extract features is better than traditional feature extraction methods in terms of both performance and effort. Still, there are numerous drawbacks of CNN with regards to representation due to which CNN is not considered a good choice for intent recognition.

2.3.2. RNN

In contrast to CNN, RNN can store a set of ordered words with the help of which it can learn relations among words corresponding to a context. Because of this ability of RNN, it is applied to solve the intent recognition problem, which gives good results in few cases [4]. RNN does not give good performance in few cases because it suffers from the problems of gradient vanishing or gradient explosion.

2.3.3. LSTM & GRU

[12] proposes an approach that uses LSTM, an extension to RNN, which contains storage to governs the data to be kept or deleted. Intent Recognition task is accomplished using LSTM over Air Travel Information System (ATIS) in a proposed work which shows betterment in error rate [24]. Furthermore, an extension of LSTM named GRU is applied to the intent recognition task, which can retain more no. of ordered words [8]. [25] shows the comparison of GRU and LSTM for this task trained on the ATIS dataset that ends up in the fact that both are the same from a performance perspective, but LSTM has a more complex model than GRU.

2.3.4. Word embeddings

Another technique has been continuously used in this task of intent recognition, known as word embedding, which works on the concept of gradual learning and is hence capable of solving sparse data problems [3]. [13] shown that when intent classification is done with word embedding has improved descriptive capability. It has been demonstrated that using rich word embedding for addressing intent detection tasks gives better results [13]. It is used jointly with other approaches for intent recognition.

2.4. Combined Approaches

As researchers are becoming aware that various machine and deep learning approaches are improvising in addressing intent recognition, numerous scholars worked in the direction of using a combination of these techniques for the same task. An approach is proposed that defines a self-attention mechanism that expresses word sequences using matrices [16].

An improvised version of LSTM is used, known as Bidirectional LSTM, which works by combining processing results in both left and right directions. In this approach, weighted summation of LSTM hidden layers is used to represent sentence relation [16]. This work can be applied for recognizing intent in case of multiple intents. [18] proposed another method for multi-intent detection in which word frequency-inverse document frequency (TF-IDF) is used along with word embedding to figure out the relationship among words in a sentence saved in a matrix.

Here we have explained the previous work done in the domain of intent recognition. Some methods give better performance than before, but there is still a need for improvement in terms of accuracy as far as intent recognition is concerned. In the next section, baseline model used is described.

3. BASELINE MODEL USED

A novel model is known as "Bidirectional Encoder Representations from Transformers (BERT)" has been used as a baseline model for our research work. In contrast to previous models, BERT Model has the unique feature of being bidirectional, due to which while understanding the text, it is considering context to its left and the right [7]. Since it is bidirectional, its performance on various language tasks is outstanding. BERT is ideationally simple and experimentally powerful. Numerous applications of NLP are applying the BERT model as it is very successful in terms of performance [7]. BERT model works in two stages called "pre-training" and "fine-tuning". In the first stage, training has been done on a vast amount of text which is not labeled. Further, during the second stage of processing, the model is initialized with the weights of the pre-trained model and further fine-tuned with task-specific data, which is labeled. Baseline model used is explained briefly. In the coming section, our approach is described.

4. OUR APPROACH: BERT-IR

We have used the BERT Model in combination with the DNN layer for intent recognition [7]. In our approach BERT-IR, that is, BERT for intent recognition, we use a pre-trained BERT model to be applied for intent recognition by fine-tuning it with some add-on layering. The distinctive feature of BERT-IR is its usage of the BERT model for intent recognition and its way of fine-tuning, which is giving outstanding results.

4.1. Pre-Training

As already mentioned, our approach is using the BERT model for pre-training. BERT model has done pre-training in two phases:

Task 1: In the first phase, "Masked Language Modelling (MLM)" is used for training [29]. In this method, some tokens are masked out of the total, and a prediction of these tokens is made, which is initially taken from the Cloze task defined in ancient work [28]. Finally, the softmax function is applied over the predicted value to convert it into probabilities for further processing.

Task 2: Next Sentence Prediction (NSP): Our downstream task is intent recognition, for which the connection between two sentences and between words of each sentence needs to be taken care of. That is why this task cannot be performed only via language modeling. For training purposes in this scenario, when the connection between sentences needs to be kept in mind, BERT pre-trained the model, predicting the following sentence of a given sentence. Further, while choosing the following sentence during training, for half of the samples, we choose a random sentence for a given sentence, and for the rest half, we choose a sentence that follows the given sentence.

4.2. Mechanism for Fine-tuning

Fine-tuning is performed by BERT-IR, which makes this work successful in addressing intent recognition because of the Transformer's "self-attention mechanism" along with the usage of correct data. We have used dense layers with dropout to make our model for the task of intent recognition. Experiments have been performed to find out the values of hyperparameters to fine-tune our model. These two stages are clubbed together, keeping in mind bidirectional processing within sequences for encoding. Model details are shown in the implementation section.

4.3. Model Architecture

The architecture of BERT is based on a bidirectional version of the Transformer described in previous work [30]. Nevertheless, in BERT, the transformer encoder has multiple layers. Since in our method BERT-IR, BERT and Transformer are implemented in the same way as described previously [7,30], we are not describing the whole structure and processing of these approaches in detail.

There are two versions of the BERT model: Base Version and Large Version. In the base version, twelve layers are implemented with hidden sizes like 768 and twelve attention heads leading to the 110M model's parameter. In contrast to the base version, twenty-four layers and the hidden size as 1024 and sixteen attention heads are used in a large size model ends up in 340M parameters. The base version is chosen for our work to justify the comparisons.

The overall architecture of our approach is shown in Figure 1. As input, we have used sentence pair representation in which we provide the input text and output label sequentially. A "sentence" could be any random sequence of text despite a meaningful one during our consideration. The text containing the ordered collection of words along with the label is known as "sequence". The pre-trained model is trained on word-piece embedding containing 30,522 token vocabularies [32]. Standard notations are used in which unique token [CLS] is for specifying starting of sequence, and as we have clubbed two sentences in a sequence, [SEP] is used to separate these two sentences. The critical value of word embedding, namely w1 to w4, is calculated by adding a token, segment, and position

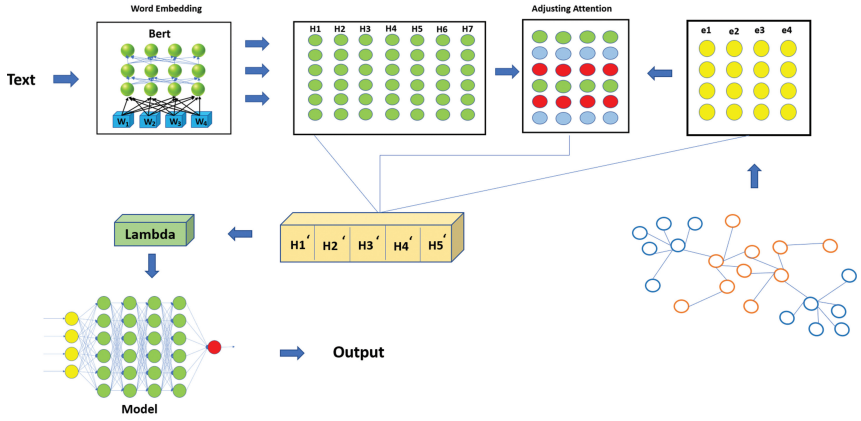


Figure 1 | Model architecture.

embedding, taken as input from the text. These are pre-trained using BERT which outputs a bunch of word vectors (specified as H1 to H7) for masked language modelling. This part shows the pre-training phase. Fine-tuning phase is shown as the right part of Figure 1. Here, we form the embedding from the actual input and output of our task that is sentence along with the actual intent, namely e1 to e4. The combination of these two phases forms the final model which performs the required task as shown in Figure 1.

5. EXPERIMENTAL DETAILS

5.1. Research Data

Our research presented the performance of our proposed approach by comparing our results with few previous ML and DL models named Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Gradient Boost, CNN, RNN, LSTM, GRU, and a combined approach. We have applied these models and our model on a well-known dataset called "SNIPS Natural Language Understanding bench-mark1 (SNIPS-NLU)". Table 1 carries the dataset details.

Our dataset consists of a bulk of spoken language text collected from several sources to make it closer to actual lingual text. The training of the acoustic model takes a considerable amount of lingual data, which is a transcript of an audio clip of extremely long duration. To make training data versatile, it is assembled from numerous random sources. To remove errors from the data, correspondence of audio and transcript is checked. After all this processing has been done, the dataset becomes precise in a form compatible with the training process. Further, three sets have been formed from the overall dataset that is training set, testing set, and validation set.

Seven intents have different complexity in this dataset [33]:

I. SearchCreativeWork (For instance: Can you show me some fine artwork)
II. GetWeather (For instance: It seems Bhopal will have heavy rain today)
III. BookRestaurant (For instance: I am in a mood to have Chinese food)
IV. PlayMusic (For instance: Let us listen to Bollywood songs)
V. AddToPlaylist (For instance: This track should be present while you are traveling)
VI. RateBook (For instance: In contrast to the previous one, I like Sydney Sheldon's recent addition very much)
VII. SearchScreeningEvent (For instance: I want you to see the timings of Sharukh Khan's show in Canada next month)

Table 1 | Dataset detail

Dataset	"SNIPS-NLU"
Vocab size	30,522
Maximum position embeddings	512
Intents	7
Training samples	13,084
Validation samples	700
Test samples	700

5.2. Text Preprocessing

In context to the lingual text data collected from random sources, preprocessing is necessary [29]. Preprocessing is needed since this kind of data is usually not in a form to which we can directly apply the training process to make the model. Nevertheless, as far as our data is concerned, it is already defined in terms of informal texts and impurities. So, we needed to perform few basic steps for preprocessing, like removing special characters. we have done three important steps in preprocessing:

1. Tokenize the data
2. Converting tokens into numbers
3. Padding is added

Besides these essential steps, since we were dealing with text in only English, text in other languages has been removed from the dataset using the available tools. In addition to this, messages containing less than three words have been eliminated from the dataset. Furthermore, the same messages present in the data are taken only once. So, we have assigned an index to every message to avoid repetition.

5.3. Model's Summary

While constructing our model, Bert layer is used as the base layer. For taking input data, we have used an input layer. After the BERT layer, we have included two dense layers along with dropout. For clubbing the Bert layer with the layers added afterwards, a lambda layer included in between.

Table 2 | Our model's summary

Layer type	Output shape	Param
Input layer	[(None, 38)]	0
Bert model layer	(None, 38, 768)	108890112
Lambda layer	(None, 768)	0
Dropout layer	(None, 768)	0
Dense layer	(None, 768)	590592
Dropout layer	(None, 768)	0
Dense layer	(None, 7)	5383

Total params: 109,486,087; Trainable params: 109,486,087 height; Non-trainable params: 0.

Table 3 | Hyperparameters used in other ML models

Model name	Hyper parameter	Values taken
Naive bayes	Alpha	**1.0**
	Algorithm	**Gaussian**
Logistic regression	C	**1.0**
	Max. iteration	50, **100**, 150
	Penalty	**Euclidean distance (L2)**
	Tolerance	0.01, 0.001, **0.0001**
Decision tree	Max. depth	10, 20, **50**, 80
Random forest	Max. depth	10, 20, **50**, 80
	No. of trees	3, 5, **10**, 15
Gradient boost	Max. depth	10, 20, **50**, 80
	No. of trees	3, 5, **10**, 15
	Learning rate	**0.1**, 0.01, 0.001, 0.05, 0.005
	Tolerance	0.01, 0.001, **0.0001**

Several combinations of different kind of layers are tried out but the demonstrated model is found out to be the best as far as performance is concerned. Our model's summary is shown in Table 2.

5.4. Hyperparameters Used for Implementation

As mentioned above, we have implemented five Machine learning and three Deep Learning techniques and our approach on the same dataset. Here, we have specified the values of the hyperparameters corresponding to Machine Learning models and Deep Learning models used, which are considered for experimental work in Tables 3 & 4. Also, values of hyperparameters taken for tuning used in our model are shown in Table 5. The best value has been shown in bold text in the following Tables.

5.5. Training of Model

Models based on the previously specified techniques and our approach have been trained using predefined libraries, namely Scikit-learn, Tensorflow, and Keras. For training in BERT-IR, we have used Adam optimizer with a learning rate of 0.00005. A validation split of 0.1 has been used for choosing the best value of various hyperparameters. To detect intents, we set no. of epochs as 50 and a batch size as 16. Intending to eliminate the problem of overfitting, we have done dropout in the Dense layer with a dropout rate of 0.1 and early stopping with two epochs as patience.

While fine-tuning the model, we have checked the performance measures of the model by putting different values to the hyperparameters shown above in Table 4. Table 6 demonstrates this process of fine-tuning concerning the dropout rate as an illustration. It is found out that we have got the best results for a dropout rate of 10%. Similarly, we have fine-tuned our model concerning every hyperparameter.

Table 4 | Hyperparameters used in other DL models

Model name	Hyper parameter	Values taken
RNN	Learning rate	**0.00001**, 0.0001, 0.0005
	Optimizer	SGD, **Adam**, RMS
	Activatin function	ReLu, SeLu, Sigmoid, **Tanh**
	Validation split	**10%**, 20%, 30%
	Batch size	8, **16**, 24
	No. of epochs	10, 20, **30**, 50
	Loss optimizer	**Sparse categorical crossentropy**
LSTM	Learning rate	**0.00001**, 0.0001, 0.0005
	Optimizer	SGD, **Adam**, RMS
	Activatin function	ReLu, SeLu, Sigmoid, **Tanh**
	Stopping condition	**Early stopping with patience = 2**
	Validation split	**10%**, 20%, 30%
	Batch size	8, **16**, 24
	No. of epochs	10, 20, **30**, 50
	Loss optimizer	**Sparse categorical crossentropy**
Bidirectional LSTM	Learning rate	**0.00001**, 0.0001, 0.0005
	Optimizer	SGD, **Adam**, RMS
	Activatin function	ReLu, SeLu, Sigmoid, **Tanh**
	Stopping condition	**Early stopping with patience = 2**
	Validation split	**10%**, 20%, 30%
	Batch size	8, **16**, 24
	No. of epochs	10, 20, 30, **50**
	Loss optimizer	**Sparse categorical crossentropy**

Table 5 | Hyperparameters used in our approach, BERT-IR

Hyper parameter	Values
Learning rate	**0.0005**, 0.005, 0.001, 0.0001
Optimizer	**Adam**
Activation function	ReLU, SeLU, Sigmoid, **Tanh**
Stopping condition while training	Early stopping with monitoring parameter = **Validation accuracy** and patience = **2**
Validation split	**10%**, 20%, 30%
Batch size	8, **16**, 24
No. of epochs	5, 10, 20, **50**
Loss optimization	**Sparse categorical crossentropy**

Table 6 | BERT-IR's fine tuning based on dropout rate

Dropout rate	Accuracy	Average precision	Average recall	Average F1 score
10%	**97.67**%	**97.85**%	**97.72**%	**97.77**%
20%	97.57%	97.78%	97.64%	97.64%
30%	97.00%	97.18%	97.15%	97.08%
50%	97.00%	97.09%	97.21%	97.08%

6. EXPERIMENTAL RESULTS

6.1. Metrics Used

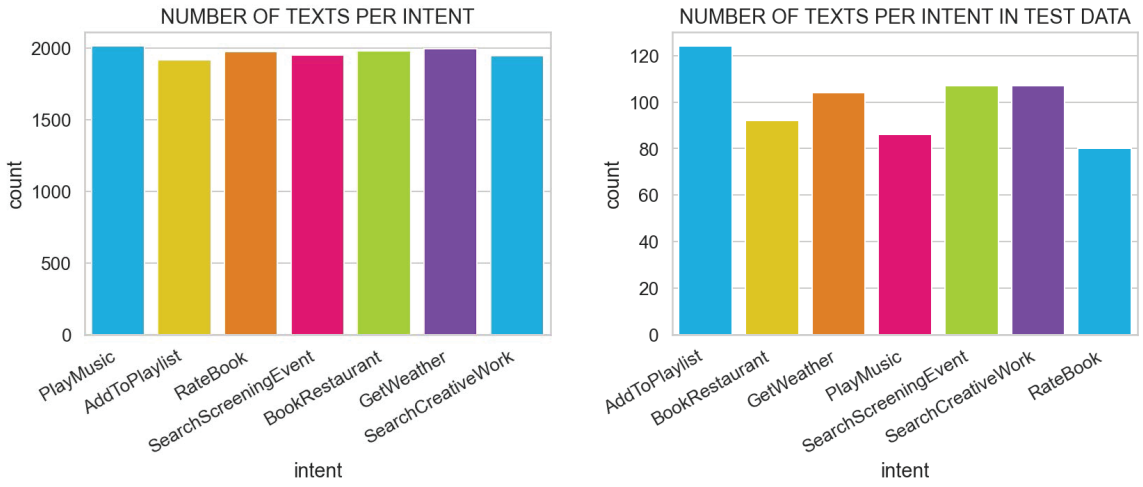
Since the task IR can be considered as a classification task, we have used the following metrics to access our approach:

Accuracy: Accuracy is the percentage of times we are predicting correctly. In terms of a mathematical equation, we can define it as:

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

Precision (P): Precision is the percentage of correct predictions out of total positive predictions. In terms of a mathematical equation, we can define it as:

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (2)$$



(a) Train Data (b) Test Data

Figure 2 | Number of texts for each intent.

Table 7 | BERT-IR's intent wise performance

Intents	Precision	Recall	F1 score
SearchCreativeWork	94.50%	100.00%	97.17%
GetWeather	100.00%	99.99%	100.00%
BookRestaurant	99.98%	100.00%	99.98%
PlayMusic	100.00%	89.71%	94.58%
AddToPlaylist	98.92%	100.00%	99.45%
RateBook	99.97%	99.03%	99.51%
SearchScreeningEvent	90.26%	95.32%	92.72%

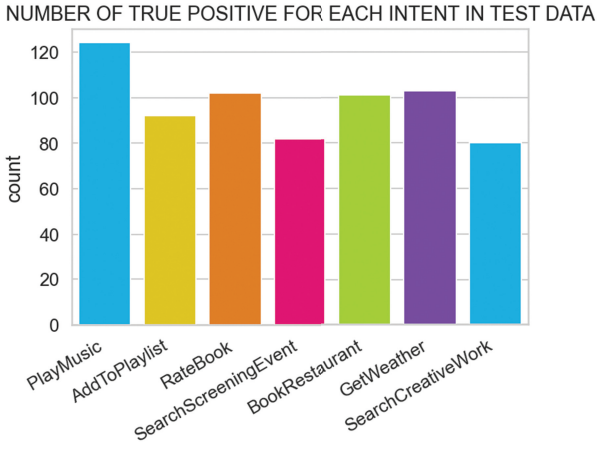


Figure 3 | Number of true positive for each intent in test data.

Recall (R): Recall is the percentage of correct predictions out of total actual positive predictions. In terms of a mathematical equation, we can define it as:

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \tag{3}$$

F1 Score (F1): F1 score is the measure of the balance between precision and recall. In terms of a mathematical equation, we can define it as:

$$\text{F1 Score} = 2 * (\text{Precision} * \text{Recall})/(\text{Precision} + \text{Recall}) \tag{4}$$

where, TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

6.2. BERT-IR's Performance

The above-shown results show that using BERT-IR is superior to the previous work. To present our work clearly, it is analyzed from another perspective. For that, metrics of our model for individual intents in the dataset are demonstrated. Figure 2a and 2b show the total number of text corresponding to each intent present in the train and test dataset, respectively.

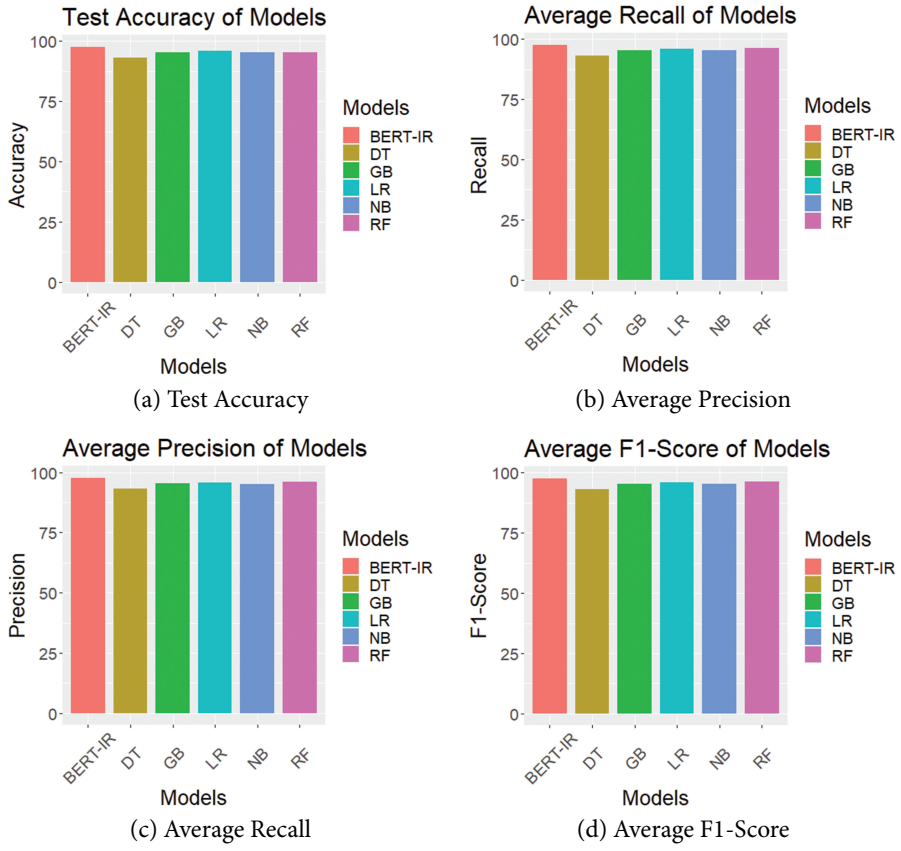
The Table 7 shows the precision, recall, and F1 score of individual intents of our model. Also, from Table 7, we can say that few labels are hard to predict compared to the others. Some text examples are classified wrongly as some intent instead of the actual one due to similarity between the intents. For example, many times, the intent type "PlayMusic" is predicted as "AddToPlaylist". It is demonstrated in Figure 3, which tells the number of true positives for each intent while testing.

6.3. Comparative Analysis

In this research, we experimented with various machine learning and deep learning models mentioned above and our approach BERT-IR on the SNIPS English dataset. The comparison of our model with other models from a performance's point of view is represented in the tables below.

Based on the comparison demonstrated in the Tables 8 and 9, our model BERT-IR is far more potent in terms of performance than the other models we have implemented. BERT-IR has got a test accuracy of 97.67%, which is the highest among all other models.

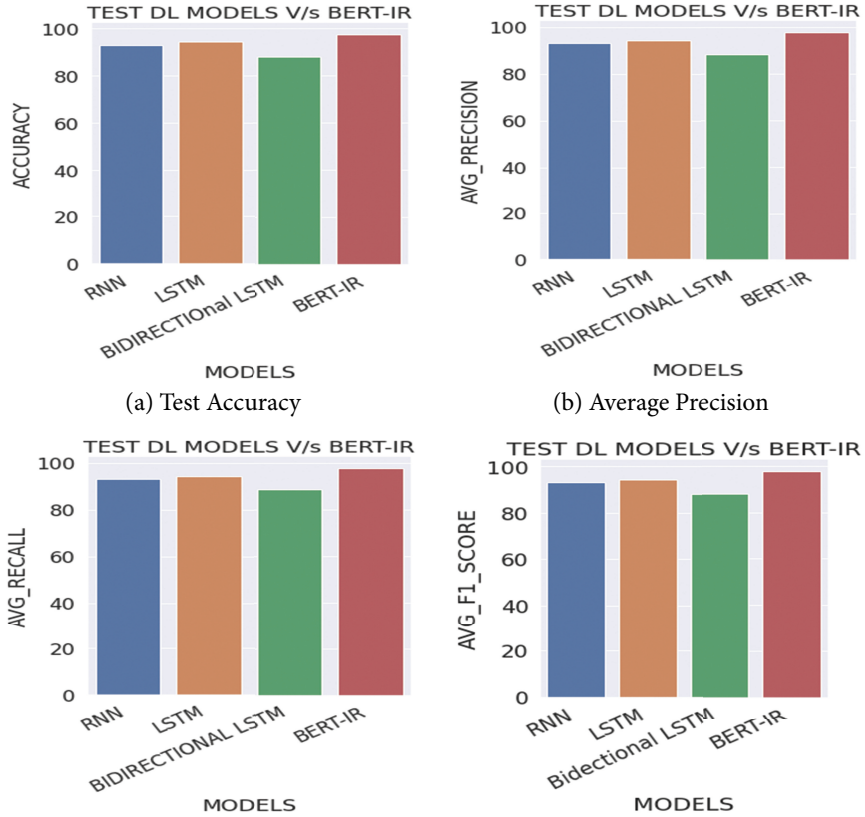
Also, Figures 4a–4d and 5a–5d shows the graphs for comparing the Accuracy, Recall, Precision, and F1-Score of various ML and DL models implemented in our work with our model respectively. It can be seen from the graphs that our model is having the best performance with respect to all the metrics in comparison with the other ML and DL models implemented.



(a) Test Accuracy (b) Average Precision

(c) Average Recall (d) Average F1-Score

Figure 4 | Comparison of metrics of ML models and BERT-IR.



(a) Test Accuracy (b) Average Precision

(c) Average Recall (d) Average F1-Score

Figure 5 | Comparison of metrics of DL models and BERT-IR.

Table 8 | Comparison of our model with other DL models

Model name	Test accuracy	Average precision	Average recall	Average F1 score
RNN	92.83%	93.18%	93.25%	93.08%
LSTM	94.32%	94.40%	94.37%	94.34%
Bidirectional LSTM	88.15%	88.44%	88.56%	87.97%
BERT-IR	**97.67%**	**97.85%**	**97.72%**	**97.77%**

Table 9 | Comparison of our model with other ML models

Model name	Test accuracy	Average precision	Average recall	Average F1 score
Naive Bayes	95.28%	95.30%	95.28%	95.24%
Logistic regression	95.85%	95.90%	96.01%	95.89%
Decision tree	93.0%	93.24%	93.01%	93.11%
Random forest	95.42%	96.23%	96.43%	96.28%
Gradient boost	95.28%	95.48%	95.28%	95.36%
BERT-IR	**97.67%**	**97.85%**	**97.72%**	**97.77%**

7. CONCLUSION

In our approach, namely BERT-IR, we have applied a pre-trained natural language processing model for intent recognition. As per our knowledge and understanding, this is the first time the BERT model is used for intent recognition with the add-ons we have done in this work. We have achieved an accuracy of 97.67%, which is very high compared to the previous work done for this task. For comparing the performance of our approach, we have implemented many other machine learning models. It is found out from the results that our approach has given high performance than other models in terms of various performance metrics used. Furthermore, our model is easy to use and also helpful when the data is not very huge as the dataset used for training in our work is medium-sized. In the future, we would like to extend our work for implementing chatbots which is a handy and vital task in the field of Natural Language Processing in the current scenario.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

AUTHORS' CONTRIBUTION

Vasima Khan and Tariq Azfar Meenai both come up with the idea of using BERT for Intent Recognition. Vasima Khan developed the theory and the algorithm that can be implemented. Tariq Azfar Meenai implemented the work using the algorithm. Both the authors analysed the work done. Then they discussed the results and contributed to the final manuscript.

REFERENCES

[1] A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah, et al., A review on applications of ANN and SVM for building electrical energy consumption forecasting, Renewable and Sustainable Energy Reviews 33 (2014), 102–109.

[2] S. Bao, H. He, F. Wang, H. Wu, H. Wang, PLATO: Pre-trained dialogue generation model with discrete latent variable, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 85–96.

[3] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, The journal of machine learning research 3 (2003), 1137–1155.

[4] A. Bhargava, A. Celikyilmaz, D. Hakkani-Tür, R. Sarikaya, Easy contextual intent prediction and slot detection, 2013 IEEE international conference on acoustics, speech and signal processing, IEEE, Vancouver, BC, Canada, 2013, pp. 8337–8341.

[5] A. Celikyilmaz, D. Hakkani-Tur, G. Tur, A. Fidler, D. Hillard, Exploiting distance based similarity in topic models for user intent detection, 2011 IEEE Workshop on Automatic Speech Recognition & Understanding, IEEE, Waikoloa, HI, USA, 2011, pp. 425–430.

[6] Y.N. Dauphin, G. Tur, D. Hakkani-Tur, L. Heck, Zero-shot learning for semantic utterance classification, arXiv preprint arXiv:1401.0509, 2013.

[7] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805, 2018.

[8] R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS), IEEE, Boston, MA, USA, 2017, pp. 1597–1600.

[9] A. Genkin, D.D. Lewis, D. Madigan, Large-scale bayesian logistic regression for text categorization, technometrics 49 (2007), 291–304.

[10] P. Haffner, G. Tur, J.H. Wright, Optimizing SVMs for complex call classification, 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., IEEE, Hong Kong, China, 2003, pp. I–I.

[11] H.B. Hashemi, A. Asiaee, R. Kraft, Query intent detection using convolutional neural networks, International Conference on Web Search and Data Mining, Workshop on Query Understanding, ACM, 2016, pp. 1–5.

[12] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997), 1735–1780.

[13] J.K. Kim, G. Tur, A. Celikyilmaz, B. Cao, Y.Y. Wang, Intent detection using semantically enriched word embeddings, 2016 IEEE Spoken Language Technology Workshop (SLT), IEEE, San Diego, CA, USA, 2016, pp. 414–419.

[14] Y. Kim, Convolutional neural networks for sentence classification, arXiv, 2014.

[15] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998), 2278–2324.

[16] Z. Lin, M. Feng, C.N. dos Santos, M. Yu, B. Xiang, B. Zhou, et al., A structured self-attentive sentence embedding, 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net. Available from: https://openreview.net/forum?id=BJC_jUqxe.

[17] J. Liu, Y. Li, M. Lin, Review of intent detection methods in the human–machine dialogue system, Journal of Physics: Conference Series, IOP Publishing, 1267 (2019), 012059.

[18] Q. Liu, J. Wang, D. Zhang, Y. Yang, N. Wang, Text features extraction based on TF-IDF associating semantic, 2018, pp. 2338–2343.

[19] T.L. Luong, M.S. Cao, D.T. Le, X.H. Phan, Intent extraction from social media texts using sequential segmentation and deep learning models, 2017 9th International Conference on Knowledge and Systems Engineering (KSE), IEEE, Hue, Vietnam, 2017, pp. 215–220.

[20] A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, AAAI-98 workshop on learning for text categorization, AAAI Press, Madison, Wisconsin, 1998, pp. 41–48.

[21] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 2227–2237.

[22] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding with unsupervised learning, Technical report, OpenAI, 2018.
[23] H. Rashkin, E.M. Smith, M. Li, Y.L. Boureau, Towards empathetic open-domain conversation models: a new benchmark and dataset, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5370–5381.
[24] S. Ravuri, A. Stolcke, Recurrent neural network and LSTM models for lexical utterance classification, Sixteenth Annual Conference of the International Speech Communication Association, Dresden, Germany, 2015, pp. 135–139.
[25] S. Ravuri, A. Stolcke, A comparative study of recurrent neural network models for lexical domain classification, 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, Shanghai, China, 2016, pp. 6075–6079.
[26] F. Ren, Y. Bao, A review on human-computer interaction and intelligent robots, International Journal of Information Technology & Decision Making 19 (2020), 5–47.
[27] R.E. Schapire, Y. Singer, BoosTexter: a boosting-based system for text categorization, Machine learning 39 (2000), 135–168.
[28] W.L. Taylor, "cloze procedure": a new tool for measuring readability, Journalism quarterly 30 (1953), 415–433.
[29] O.T. Tran, T.C. Luong, Understanding what the users say in chatbots: a case study for the Vietnamese language, Engineering Applications of Artificial Intelligence 87 (2020), 103322.
[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, et al., Attention is all you need, Advances in neural information processing systems, Curran Associates, Inc., Long Beach, CA, USA, 2017, pp. 5998–6008.
[31] T. Wolf, V. Sanh, J. Chaumond, C. Delangue, Transfertransfo: a transfer learning approach for neural network based conversational agents, arXiv preprint arXiv:1901.08149, 2019.
[32] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, et al., Google's neural machine translation system: bridging the gap between human and machine translation, arXiv preprint arXiv:1609.08144, 2016.
[33] C. Zhang, Y. Li, N. Du, W. Fan, P. Yu, Joint slot filling and intent detection via capsule neural networks, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, 5259–5267.
[34] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, et al., Aligning books and movies: towards story-like visual explanations by watching movies and reading books, 2015 IEEE International Conference on Computer Vision (ICCV), IEEE, Santiago, Chile, 2015, pp. 19–27.