International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 09 Issue 11 ǁ 2021 ǁ PP. 34-41

Design and Analysis of Attention-Based Mechanisms for Intent Recognition and Classification

Sai Teja Bandela, Tharun Kumar Bandaru, Erra Sai Nitheesh
[1,2]*Department of Electrical and Electronics Engineering, CVR College of Engineering*
[3]*Department of Computer Science and Engineering, CVR College of Engineering*

Abstract:
Citizen groups that debate current events may flourish on social media platforms. From social good (such as volunteering to assist others) to economic interest, its material shows a wide range of motives (e.g., criticizing product features). To help organizations such as an emergency management unit organize their resources, mining internet intent may help sort through huge quantities of textual data. Due to uncertainty in interpretation and the scarcity of relevant social actions, effective intent mining is difficult to achieve. In this article, we demonstrate how to recognize and classify intent using attention processes. Instead of using traditional methods like Gated Recurrent Unit (GRU) based models, our new approach focuses on creating bidirectional encoder architectures that can be tested against each other. Our findings indicate a substantial increase in accuracy on testing sets from 89.56 percent to 97.85 percent of up to 8.29 percent. percent absolute. Intent mining can assist in the development of effective citizen-organization cooperative information systems to meet organizational information requirements.

Keywords: *Intent Recognition, Intent Classification, Gated Recurrent Units, Attention Models, Natural Language Processing*

Date of Submission: 24-10-2021 Date of acceptance: 07-11-2021

I. Introduction:

With the dawn of the artificial intelligence age, an increasing number of intelligent goods have found widespread use in our everyday lives, including emotional care robots, personal phone assistants Siri and Google Now, as well as Microsoft Research Asia's intelligent conversation robot Xiao Bing. These intelligent conversation systems significantly simplify users' lives. Automatic Speech Recognition (ASR), Spoken Language Understanding (SLU), Dialogue Management (DM), Dialogue Generation (DG), and Text to Speech (TTS) are the five major components of the dialogue system [1]. To better comprehend the user's expression and subsequently provide the right information to the user, spoken language comprehension is critical. As a sub-module of spoken language comprehension, intent detection (ID) is also critical to the human-machine conversation system.

Intent categorization is a procedure that entails associating text and determining the content's particular purpose (intent) or destination. This phenomenon is beneficial for deciphering the purpose of users' comments, emails, and chat discussions, and for giving findings in the form of similarity indices that indicate the text's resemblance to a certain subject. The intent classifier is capable of categorizing text based on purpose in the same manner that people do. The procedure is divided into two stages: the first stage is intent classification, during which the text is categorized into appropriate fields; the second stage is intent analysis, during which the text's resemblance to the classified fields is computed. The intentions are classified according to the following categories: opinion, news, complaint, suggestion, spam, and marketing. This phenomenon has a variety of uses, including spam detection, social media advertising, and customer service.

Analyzing what consumers think and feel about certain goods may provide businesses with the knowledge and advantage they need to continuously improve their products and outperform the competition. Simultaneously, any automated help system that intends to offer a variety of services must be capable of analyzing a person's intent in order to determine what they need assistance with. Additionally, review and opinion aggregator websites may use intent and/or sentiment analysis to determine the overall favorability or unfavorability of the reviews submitted on their site. These, and many other use cases, emphasize the need for effective intent analysis. There is a need for precise intent analysis systems that can determine what consumers want and deliver it to them. Such systems are critical, particularly in the business sector, where their implementation may result in increased efficiency and streamlining.

In traditionally spoken language comprehension, two subtasks are distinguished: intent identification and semantic slot filling. Due to the constraints imposed by early research on application scenarios, data, and computer capacity, the majority of spoken language comprehension was domain-specific. However, as a result of technological advancements and the development of multi-domain conversation systems, spoken language comprehension is often split into three tasks: domain identification, intent detection, and semantic slot filling [2].

Intent detection is critical in a conversation system. The purpose is the user's will, or what the user wishes to accomplish. Intents are sometimes referred to as "Dialog Acts" [3], which relate to the action of information that people exchange and continuously update throughout a conversation. The purpose is often indicated by the phrase "verb + noun," such as "query the weather" or "reserve a hotel." Intent detection also referred to as intent classification, classifies user utterances into previously established intent categories based on the domains and intentions involved [4]. Nowadays, throughout the application process of a human-machine dialogue system, users may have numerous intentions at various times, triggering multiple domains in the human-machine dialogue system, including task-oriented vertical domains and chats, among others. Only when the user's subject area is properly specified in the conversation system can the user's particular requirements be accurately evaluated; otherwise, incorrect intent detection will result.

II. Literature Review:

In recent years, the majority of academics have seen intent detection as a problem of Semantic Utterance Classification (SUC) [5]. Traditional intent detection methods include rule-based template semantic recognition [6] (1993) and statistical feature-based classification algorithms [7,8]. (2002-2014). While the rule-based template matching approach does not need a large amount of training data, it does ensure detection accuracy. However, it does not address the issue of high template reconstruction costs associated with altering the intent category. However, the statistical feature classification technique requires the extraction of important characteristics from corpus text. However, manually extracting features is not only time-consuming but also uncertain in terms of quality, which contributes to data sparsity issues. Naive Bayes [9] (1998), Adaboost [10] (2000), Support Vector Machine (SVM) [11] (2003), and logistic regression [12] are all popular techniques (2007). Given the non-standard and ambiguous information included in user text, the conventional technique of intent detection is incapable of properly comprehending the deep semantics contained in user text. Accurately determining the user's true purpose is a difficult job.

With the advancement of deep learning, an increasing number of researchers are applying word embedding, convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory (LSTM) networks, gated recurrent units (GRU), attention mechanisms, and capsule networks to the task of intent detection. In comparison to conventional machine learning techniques, the deep learning model significantly improves detection performance.

Word embedding has been increasingly utilized in semantic analysis tasks in recent years, owing to the usage of unique lexical characteristics, which results in data-sparse issues, and continuous representation learning may address data-sparse problems. [13] (2003). Kim et al. [14] (2015) classified intent using word embeddings as lexical characteristics. In comparison to the conventional word bag model, the intent classification technique based on word embedding offers a higher capacity for representing and domain extensibility for a variety of classification contents. Given the dearth of semantic information in word embeddings, Kim et al. [15] (2016) utilized semantic vocabulary dictionary information to enrich word embeddings and therefore enhance the semantic representation of intent text. This model performed well, demonstrating that rich word embedding may assist enhance the performance of intent detection.

Initially, CNNs were employed for image processing [16]. With the advent of word embedding technology, CNN has been extensively used in the area of natural language processing and has produced some impressive research findings. Kim et al. [17] (2014) used CNN to perform text categorization tasks and obtained near-optimal results. Hashemi et al. [18] (2016) utilized CNN to extract text vector representations as a query classification feature in order to determine the user's search intentions. In comparison to conventional artificial feature extraction techniques, this method not only eliminates a significant amount of feature engineering but also provides a more detailed representation of the features. CNN, on the other hand, has representational constraints.

RNN is distinct from CNN in that it reflects a word sequence and may be trained to understand the semantics of word order based on the context. Bhargava A[19] (2013) demonstrated that integrating context information into intent detection tasks decreased the error rate of intent detection, suggesting that context information is helpful for intent detection. A basic RNN suffers from gradient expansion and gradient disappearance, making it unsuitable for simulating long-term dependency. LSTM [20] (1997) may address this issue by including a memory unit into the RNN structure that is capable of controlling the amount of information kept and forgotten. This approach is often used to address the issue of intent detection as well. Ravuri et al. [21] (2015) suggested that intent classification be solved using RNN and LSTM. Experiments on the Air Travel Information System (ATIS) dataset demonstrate that LSTM has a 1.48 percent lower error rate for intent recognition than RNN. The primary reason is that LSTM is effective at modeling the temporal connection between words and has a strong memory function for lengthy text input. GRU is an extension of the LSTM model [22], which is capable of retaining information across extended sequences and learning contextual semantic information. Ravuri et al. [23] (2016) compared the performance of intent detection on the ATIS and Cortana datasets using GRU and LSTM. Experiments demonstrate that GRU and LSTM perform almost identically in the intent detection task, although GRU has fewer parameters and a simpler model.

Taking into account the benefits and drawbacks of various deep learning models, the majority of researchers mix deep learning models with distinct advantages to identify user intentions. Qian et al. [24] (2017) presented a travel consumption intent detection model based on Convolutional-LSTM, which used CNN to extract deeper-level intent text characteristics and LSTM to construct text's temporal connection, resulting in a high-performing model. Yu et al. [25] (2018) addressed the issue of sparse data resulting from the brief text by proposing a multi-turn conversation intent recognition model based on the Biterm Topic Model (BTM) and Bidirectional Gated Recurrent Unit (BGRU). This integrated model achieves excellent results in detecting users' medical intent and outperforms the performance of the literature [26]. Huang (2018) presented a hybrid model of Character-CNN-BGRU deep learning. The combined model not only uses the character-based method to reduce the list of words but also solves the problem of unknown words. When combined with CNN, the combined model extracts local features from the intent text and BGRU ensures the text's temporal relationship, highlighting the combined model's advantages in the intent detection task. However, since the combined model's structure is complicated and the training time is lengthy, the issue of simplifying the combined model is worth addressing.

III. Methodology:

GRU Architecture:

GRU networks are classified as RNNs, which are neural networks with an underlying architecture of at least one cycle in their inter-neuronal connections. They were launched in 1997 and improved significantly during the subsequent years. GRUs are a subclass of gated RNNs that are used to address the issue of disappearing and inflating gradients in conventional RNNs while learning long-term dependencies.

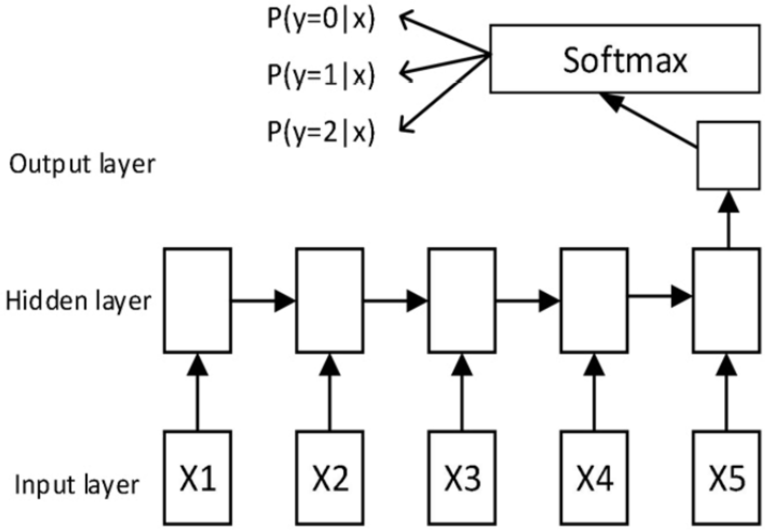


Figure. 1. GRU Model Architecture

As shown in Figure 1, the input layer is composed of many neurons, the number of which is controlled by the size of the feature space. Similarly, the output layer's number of neurons correlates to the output space. The GRU networks' primary tasks are covered by the hidden layer(s) including memory cells. The cell's state is changed and maintained through two gates: a reset gate r_t and an update gate t_u. The construction of a memory cell is shown in Fig. 2 as a circuit diagram.

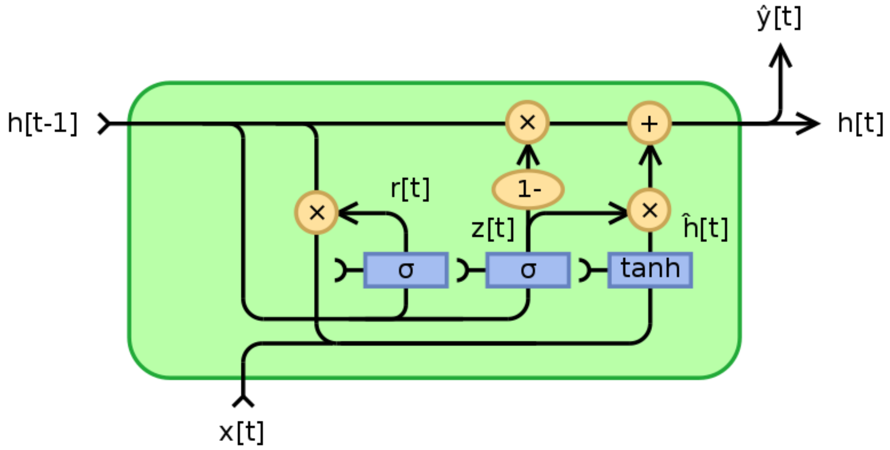


Figure. 2. GRU Memory Cell

Both gates have two components in common: x_t and h_{t-1}. The former is derived from the input sequence, whereas the latter is derived from the preceding time point's memory cell output value. As a result, each gate performs a unique function in order to accomplish the filtering objective:

The reset gate h_{t-1} is used to limit the effect of h_t(the unit's prior timestep information) on the present information x_t. If h_{t-1} is irrelevant to x_t, then r_t may be opened, thus ensuring that h_{t-1}has no effect on x_t. The update gate indicates whether the current information x_t should be ignored. When t_u is activated on the under branch, we will disregard the present value of x_t and create a "short-circuit connection" between h_{t-1} and h_t. This causes the gradient to propagate in the opposite direction, thus resolving the gradient vanishing issue. The equations below explain how memory cells in the GRU networks' hidden layers are refreshed at each time step.

The reset gate's equation is as follows: $u_t = s(W_u.[h_{t-1}, x_t])$
The update gate's equation is as follows:$r_t = s(W_r.[h_{t-1}, x_t])$

Output: $\widehat{h_t} = tan \quad h(W.[r_t * h_{t-1}, x_t])$

When GRU networks are dealing with an input sequence, the eigenvalues must be entered in chronological order. Thus, the GRU networks process the relevant inputs in the order specified above. The network will provide the final result only after the last piece in the sequence has been processed. The parameters (weights and bias) are adjusted in a manner similar to that of conventional feedforward neural networks. The loss of the goal function from the training sets is reduced throughout training. Due to the classification job, cross-entropy is selected as the objective function. We use three widely used methods to successfully train the GRU networks. To begin, the optimizer is configured to use RMSprop. It was developed from the prop and optimized using a mini-batch. This is a common technique for RNNs. Second, dropout regularisation is performed on the hidden layers to improve generalization and prevent overfitting. Thus, throughout the training phase, the neural network unit will be removed from the network with a given frequency. Thirdly, we use early halting as a secondary strategy to accomplish the same goal. Thus, training data is partitioned into a training set and a validation set based on a predefined ratio. The former is utilized for training, whereas the latter is used to acquire test results (for example, every 5- epoch makes a test). If the loss on the validation set rises as time increases, cease training. Following the halt, the weights are restored as the network's final parameters.

BERT Architecture:

The BERT architecture is a cutting-edge design that is based on the transfer learning technique. This method is defined as a breakthrough solution that is gradually becoming the standard in natural language processing; it is the solution that supplants previous methods for performing common NLP tasks such as sentiment analysis, Q&A (questions and answers), machine translation, classification, summarization, and named entity recognition. The BERT method's primary differentiating characteristic is its ability to detect and record the contextual meaning included in a phrase or text. This contextuality implies that the numerical representation of a word or token during the word embedding process is context-dependent. This implies that each time we consider the context of a word, the numerical value of the word changes. This is a different interpretation of the term "embedded procedure" then is used for static methods.

Static word embeddings are used to code unique words independently of their context, i.e. in NLP techniques that use this kind of word embeddings, a unique word will always be coded the same manner. While the GloVe technique is likewise a static method, it makes use of global statistical data to determine word co-occurrences. Co-occurrences in the Application of the BERT-Based Architecture in the Detection of Fake News 241 the linguistic sense may be regarded as an indication of semantic closeness or an idiomatic phrase. In contrast to the GloVe, the BERT is a dynamic word embedding technique. Dynamic techniques are used to build language models that take into consideration, not just the context stated before, but also the syntax and semantics of a text. The numerical representation of a unique word is not the same as it is in static techniques; it is determined by the word's neighborhood and the total number of words (tokens) in the sentence (segment) of the text.

A significant benefit and feature of the BERT is the use of the TL (Transfer Learning) concept, comparable to the techniques employed in CV (Computer Vision). The TL is built on the usage of pre-trained models that were developed using huge datasets and then fine-tuned. The fine-tuning process results in the adaptation of the architecture's parameters, which were previously trained on particular NLP tasks, such as Q&A. There is little variation between the pre-trained and final downstream architectures in the BERT.

The acronym BERT stands for Bidirectional Encoder Representations from Transformers; the word Transformers refers to the network architecture built on Transformer blocks. The Transformer idea is based on the substitution of self-attention blocks for RNN (Recurrent Neural Network) blocks in the neural network design.

As the name implies, the BERT architecture is composed entirely of Transformer encoders. For instance, the 12-layer BERT design includes 12 encoder blocks. The BERT is built out of identical layers. The number of layers in the network is the hyperparameter. Each encoder block is divided into two sublayers. The first is a multi-head self-attention mechanism, whereas the second is a basic, fully linked feed-forward network that is position-dependent. Self-attention is a method for connecting various locations in a single sequence in order to calculate its representation. The multi-head attention function is responsible for the computations associated with a variety of self-attention functions. Multi-head was described as a technique that enables the model to concurrently attend to input from several representation subspaces at multiple locations. A critical feature of the BERT is its bidirectionality, which enables the examination of tokens in both directions of the analyzed text fragment. This model guarantees that the neural network is trained with the context of tokens in mind. The BERT is intended to pre-train deep bidirectional representations from the unlabeled text by conditioning each layer on both left and right context.

IV. Results & Discussion:

The section discusses the implementation of the designs described before. For the purpose of evaluating the performance of the machine learning algorithms to be used, we settle on the SNIPS dataset, which contains data from the personal voice assistant Snips. The training, development, and test sets include 13,084,700 and 700 utterances, respectively. The training set has 72 slot labels and seven different types of intent. As shown in Fig. 3, the dataset visualization demonstrates the class distribution of the texts per purpose that was considered.

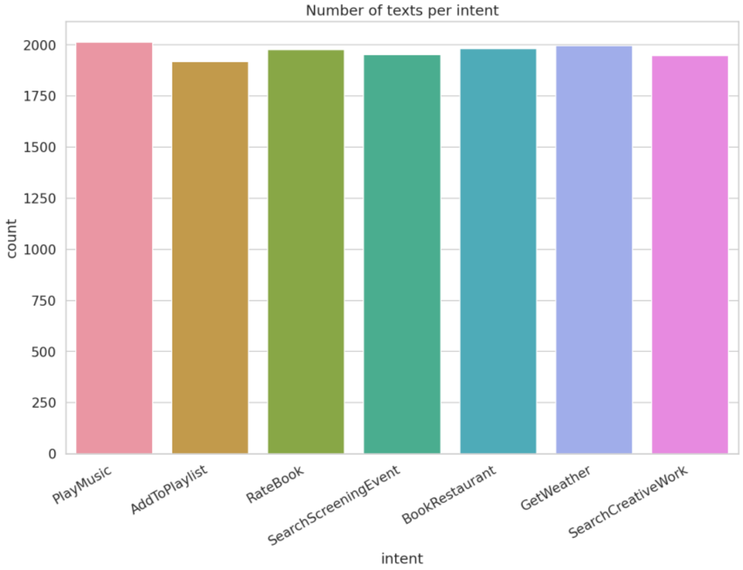
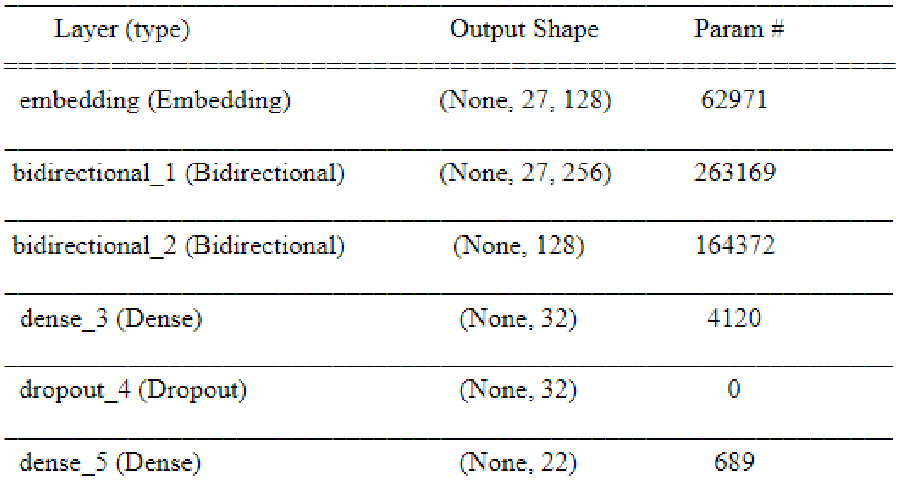


Fig. 3. Dataset Distribution

A train-test split of 70:30 has been taken into consideration for the analysis. We start with the performance analysis of the gated recurrent unit network. A novel architecture was designed for intent recognition and classification. The hyperparameter tuning was conducted on the basis of the trial and error method which led to a highly accurate approach towards the performance analysis and inference of each tuning parameter on the overall results. The following observations also helped in the designing of the final architecture as given in Table I. The total parameters taken into consideration were 495, 321 with 432,336 trainable parameters and 62,976 non-trainable parameters.

Table I. GRU Architecture



Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 27, 128)	62971
bidirectional_1 (Bidirectional)	(None, 27, 256)	263169
bidirectional_2 (Bidirectional)	(None, 128)	164372
dense_3 (Dense)	(None, 32)	4120
dropout_4 (Dropout)	(None, 32)	0
dense_5 (Dense)	(None, 22)	689

The accuracy (loss) vs validation set accuracy (AUC) has been displayed in Fig. 4. given below. The GRU architecture with the optimum parameters gives an accuracy of 89.56% on the testing data.

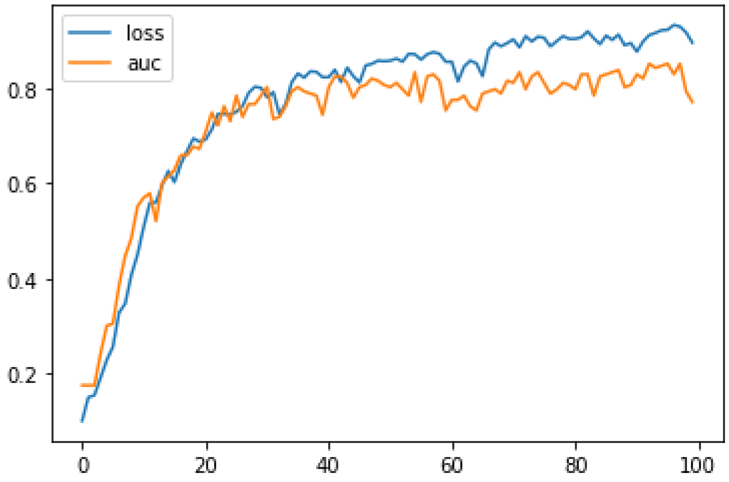
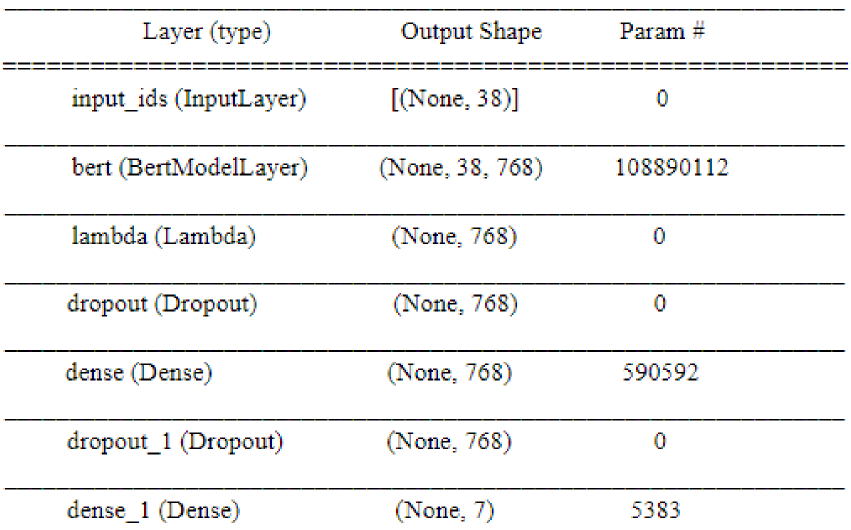


Fig. 4. GRU model accuracy curve

The BERT model (Devlin et al., 2019) has the same structure as a Transformer encoder, with the input being a word sequence and the output being representations of the input words. Multi-head self-attention systems encode the input contexts. Pre-training BERT models are provided. The distributed models are 12 or 24 layers deep, which is more than the depth of the Transformer base model (six layers). Users (or system developers) build a variety of systems by adding a tiny network to adapt BERT to their own activities and fine-tuning the system using task-specific data. When the BERT model is used for document classification, for example, a classifier is constructed by adding a classification generation layer to the BERT model (which consists of linear and softmax sublayers). Similarly, while constructing a named entity recognizer, generating layers are added that transform word representations to named entity tags, and the whole model is fine-tuned. These models have a significantly lower number of extra parameters than the BERT model. Pre-trained on two tasks, the BERT model performs intent categorization and recognition. Both challenges teach the machine to increase its skill at language prediction.

Table. II. Pre-trained BERT architecture



Layer (type)	Output Shape	Param #
input_ids (InputLayer)	[(None, 38)]	0
bert (BertModelLayer)	(None, 38, 768)	108890112
lambda (Lambda)	(None, 768)	0
dropout (Dropout)	(None, 768)	0
dense (Dense)	(None, 768)	590592
dropout_1 (Dropout)	(None, 768)	0
dense_1 (Dense)	(None, 7)	5383

Following the performance of the GRU architecture, certain provisions were made in order to optimize the performance of the intent recognition modeling. As shown in the Table. II, the architecture was used. A total of 109,486,087 trainable parameters were utilized for the model. The configuration was successful in resulting in an accuracy of 99.28% on the training dataset and 97.85% on the testing data which outperformed the previously cited architectures to the best of our knowledge. Furthermore, the confusion matrix has been shown in Figure. 5. Which gives a distribution of the predicted labels to the actual labels.

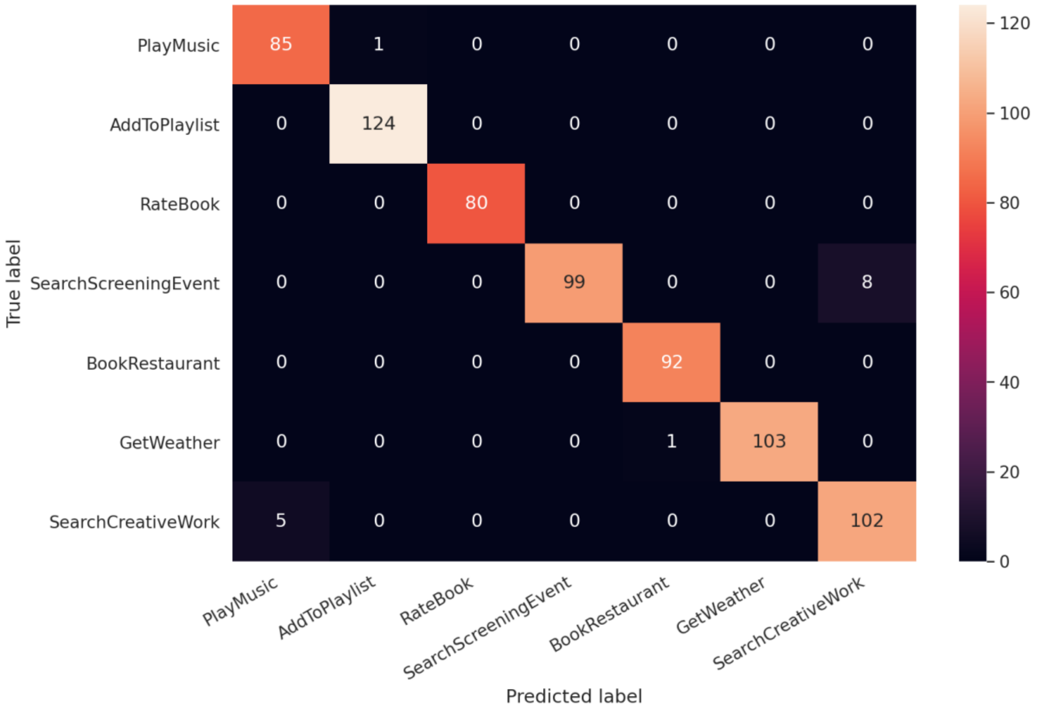


Fig. 5. Predicted labels vs. True labels

V. Conclusion:

The purpose of this article is to discuss the difficulties and techniques of detecting intent in a human-machine conversation system. It describes and contrasts the deep learning model's intent detection techniques. While conventional techniques of intent detection are unable to comprehend the user's purpose in detail, the deep learning model demonstrates its benefits. The GRU and BERT model performs well in the task of intent detection and also performs well in multi-label classification. The self-attention model is capable of extracting a variety of semantic characteristics from sentences during the intent detection process, thus adding to the field of multi-intent detection research. At the moment, intent detection is used in a variety of areas, including e-commerce, travel consumption, medical care, and chat, as well as network intrusion, network fraud, and air target warfare, to ensure network security. Traditionally, conversation systems have been primarily focused on the identification of single intents in particular areas. With the growing frequency of human-machine contact, users' discourse expression is no longer confined to a single purpose.

References:

[1]. Hongshen Chen, Xiaorui Liu, Dawei Yin, et al. A survey on dialogue systems: recent advances and new frontiers. Acm Sigkdd Explorations Newsletter, 2017, 19(2):25-35.

[2]. Tur G. Spoken Language Understanding: Systems for extracting semantic information from speech. NewYork, NY: John Wiley and Sons, 2011.

[3]. Austin J A. How to do things with words Harvard University Press. Cambridge: 1962.

[4]. Celikyilmaz A, Hakkani-Tur D, Tur G, et al. Exploiting distance based similarity in topic models for user intent detection Automatic Speech Recognition & Understanding. IEEE, 2011:425-430.

[5]. auphin Y N, Tur G, Hakkani-Tur D, et al. “Zero-shot learning for semantic utterance classification,” arXiv preprint arXiv:1401.0509, 2013.

[6]. Appelt D, Bear J, Cherny L, et al. GEMINI: A natural language system for spoken-language understanding // Proc. Meeting of the Association for Computational Linguistics. 1993:54-61.

[7]. Yan Pengju. Research on natural language understanding in conversational systems . Beijing: Tsinghua University, 2002.

[8]. Ahmad A S, Hassan M Y, Abdullah M P, et al. A review on applications of ANN and SVM for building electrical energy consumption forecasting . Renewable & Sustainable Energy Reviews, 2014, 33(2):102 -109.

[9]. Andrew McCallum, Kamal Nigam, et al.A comparison of event models for naive bayes text classification // In AAAI-98 workshop on learning for text categorization. 1998:41–48.

[10]. Schapire R E, Singer Y. BoosTexter: a boosting-based system for text categorization . Machine Learning, 2000, 39(2-3):135-168.

[11]. Haffner P, Tur G, Wright J H. Optimizing SVMs for complex call classification // IEEE International Conference on Acoustics. IEEE, 2003:632-635.

[12]. Genkin A, Lewis D D, Madigan D. Large-Scale Bayesian Logistic Regression for Text Categorization. Technometrics, 2007, 49(3):291-304.

[13]. Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model .Journal of Machine Learning Research, 2003, (3): 1137-1155.

[14]. Kim D, Lee Y, Zhang J, et al. Lexical feature embedding for classifying dialogue acts on Korean conversations //Proc. of 42th Winter Conference on Korean Institute of Information Scientists and Engineers, 2015: 575–577.

[15]. Kim JK, Tur G, Celikyilmaz A, et al. Intent detection using semantically enriched word embeddings // Spoken Language Technology Workshop. IEEE, 2016:414-419. AIACT 2019 IOP Conf. Series: Journal of Physics: Conf. Series 1267 (2019) 012059 IOP Publishing doi:10.1088/1742-6596/1267/1/012059 10

[16]. Lecun Y L, Bottou L, Bengio Y, et al. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, 1998, 86(11):2278-2324.

[17]. Kim Y. Convolutional Neural Networks for Sentence Classification // Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014:1746–1751.

[18]. Hashemi HB., Asiaee A, Kraft R, Query intent detection using convolutional neural networks // International Conference on Web Search and Data Mining, Workshop on Query Understanding , 2016.

[19]. Bhargava A, Celikyilmaz A, Hakkanitur D, et al. Easy Contextual Intent Prediction and Slot Detection // IEEE International Conference on Acoustics. IEEE, 2013: 8337-8341.

[20]. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Computation, 1997, 9(8):1735-1780.

[21]. Ravuri S V, Stolcke A. Recurrent neural network and LSTM models for lexical utterance classification // 16th Annual Conference of the International Speech Communication Association. 2015:135-139.

[22]. Dey R, Salemt F M. Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks[C]//IEEE 60th International Midwest Symposium on Circuits and Systems.IEEE, 2017:1597-1600.

[23]. Ravuri S, Stolcke A. A comparative study of recurrent neural network models for lexical domain classification // Proc. of the 41th IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 2016: 6075-6079.

[24]. Hui Yu, Xupeng Feng, Lijun Liu, et al. Identification method of user’s medical intent in chatting robot.Journal of Computer Applications, 2018, 38(8): 2170-2174.

[25]. Jiawei Huang. Research on the classification method of user intent in the human-machine dialogue system. Wuhan: Central China Normal University, 2018.

[26]. Yue Qian. Research on the identification method of users’ travel consumption intent in chat robot . Harbin: Harbin Institute of Technology, 2017.