2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT) | 979-8-3503-9648-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/InCACCT57535.2023.10141744

Fine-tuned BERT with Attention-based Bi-GRU-CapsNet framework for Joint Intent recognition and Slot filing

Nahida Shafi
Department Of Computer Science
University of Kashmir, J&K, India, 190006
Email: nahidashafi.scholar@kashmiruniversity.net

Manzoor Ahmed Chachoo
Department Of Computer Science
University of Kashmir, J&K, India, 190006
Email: manzoor@kashmiruniversity.ac.in

***Abstract*—In the field of natural language processing (NLP), the two most prominent research areas are slot tagging and intent recognition. Modern joint learning strategies examine the link between slot-tag identification and intent classification, utilising the shared knowledge between the two tasks for their collective advantage.However, methods that combine various variants of pre-trained Bidirectional Encoder Representations from Transformers [BERT] models with attention based capsule networks for joint slot-tag identification and intent detection have not been fully explored.This study proposes a multi-stage framework trained on different versions of BERT models ($\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$) with Bidirectional Gated Recurrent Unit [Bi-GRU] and self attention mechanism as intent detection decoder to capture the underlying information and to discover the explicit links.While the capsule network, in accordance with the dynamic routing algorithm, acts as a slot filler decoder in predicting intents and slots and representing the semantic and syntactic relationships. The experimental findings demonstrate that the proposed approach enhances semantic frame accuracy at the sentence level, outperforming various baseline methodologies by a significant margin with a 1.2% improvement in the intent F1-score and 3.24% in the slot F1-score, relative to the previous state-of-the-art models on the SNIPS datasets.**

***Index Terms*—bi-gru, self-attention, capsule network, intent classification, slot-filling, natural language processing, $\text{bert}_{\text{large}}$,$\text{bert}_{\text{base}}$**

I. Introduction

The Natural Language Understanding (NLU) module, which translates the given input utterances into a task-oriented semantic representation, is a crucial component of any goal-oriented dialogue system and a key part for building an efficient dialogue system. For linguistic interaction with humans NLU is crucial to technologies like Internet of Things (IoT), chatbot interaction, robot training, virtual interfaces, machine translation, text classification .e.t.c. Spoken language understanding (SLU) is a sub-stack of NLP that utilises two fundamental tasks slot-tagging and intent categorization to contextualise the comprehension of human speech. The primary focus of intent detection is determining a user's intent from a given speech; this is considered an utterance classification problem that determines the intent label "y" for a given utterance as shown in "fig 1". Comparatively, the sequence labelling task known as semantic slot filling that translates an input word sequence w = $(w_1, w_2, , w_T)$ onto the matching slot tags sequence $x^s = x_1^s, x_2^s, ..., x_T^s$. These concepts are then used to accomplish a goal or complete a specific task .

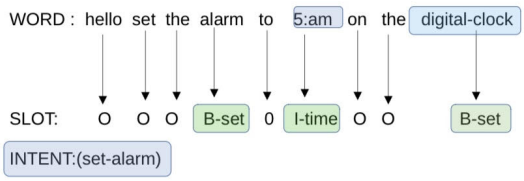

Figure 1: An example of BOI representation of user query to semantic-level frame

Combined learning approaches have been used to model the interdependencies between intent categorization and slot tagging by leveraging recurrent neural networks and attention mechanisms [1],achieving state-of-the-art performance.C.Zhang, Y.Li,N.Du, W.Fan and P.S.Yu [2] proposed model consists of a hierarchical capsule architecture where capsules represent semantic information and are trained by back-propagation with dynamic routing in order to detect intents and fill slots.

Generalization is impeded by the absence of human-labelled data for various knowledge engineering tasks . In order to effectively train models, machine learning algorithms require the accessibility of vast amounts of text data, which can be problematic when it comes to NLU and NLP. To address the issue of sparse data, a number of methods for training linguistic models with huge amounts of unlabeled text data have been proposed. One of the most effective methods is language model pre-training, which involves pre-training a model on large corpora of unlabeled text. The Bidirectional Encoder Representations from Transformers (BERT)[3] method that was proposed is used to make state-of-the-art models in many NLP tasks, from text classification and language understanding to question answering, name entity recognition, and machine translation. BERT for NLU has not however, been extensively investigated and explored. This work's technical contributions are twofold: this study investigate various versions of BERT's pre-trained models to improve NLU's generalisation ability:

- Firstly, this study investigate the effect of increasing the volume of training data on the efficacy of pre-trained BERT models.

- Secondly based on BERT$_{base}$ and Bert$_{Large}$,this paper propose a unified model for slot tagging and intent detection in a task-oriented dialogue system with the Bi-GRU self-attention mechanism as an intent detection decoder and its output to the Capsule network layer for slot tag recognition that represents semantic and syntactic relationships. In every metric, this study outperformed the benchmark ,1.2% improvement in the intent F1-score and 3.24% in the slot F1-score, relative to the previous state-of-the-art models on the SNIPS datasets.

II. Literature Review

A. COMBINING INTENT CLASSIFICATION AND SLOT FILLING

A comprehensive literature review has been conducted to examine the various natural language understanding-related tasks completed so far consisting of intent detection, slot tagging, domain classification, and its multi-task framework. This section overviews state-of-the-art research on intent recognition and slot-filling studies. This survey revealed that although substantial progress has been made, there is still a need for further development in the field.This study examines how current methods for NLU have jointly modelled these tasks, incorporated human knowledge domains, and made use of pre-trained word embeddings and contextual knowledge.

Earlier methods for intent classification and slot-tagging used distinct models for each task. These models were trained primarily on word embeddings, such as skip-gramme and one-hot representation, and were based on the feature representation space [4].Combination of Conditional random fields (CRF) [5] with statistical models in a heuristic approach with Markov models were also commonly used methods for sequence labelling.

B. Deep learning models for joint intent classification and slot filling

Recently, deep learning models like CNNs, RNNs, and LSTMs [6] have shown excellent performance in a wide variety of NLP tasks and have been widely used in domain classification, intent recognition and slot tagging. Kane et al [7] extracted the semantic hierarchy for effective modeling, using the combination of CNN (Convolutional Neural Network), Bidirectional LSTM (Long ShortTerm Memory) and CRF (Conditional Random Field)[8]for ID and SF. L.Zhao and Z. Feng [9] proposed a generative neural network via a pointer network to generate sequences that could capture the long-term dependencies in a given sentence through an attentional Seq2Seq model. By using this model, the authors were able to generate coherent sentences with intricate structures that accurately reflected the semantics of the original sentence For joint intent detection and slot filling, Liu and Lane [10] integrated explicit alignment information between input tokens and output slots into an attention-based Bi-LSTM network.
M. Firdaus, A. Kumar, A. Ekbal, and P. Bhattacharyya [11] captured contextual information for the utterances using a convolutional neural network (CNN) model, allowing them to recognise the intent of a sentence by taking into account the surrounding words and phrases and also incorporated a conditional random field for mapping the intent and slot label dependency.
J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, [3]recently presented a state-of-the-art pre-training technique (BERT), which is a natural language processing (NLP) technique that can be used to better understand the context and meaning of text , and produced state-of-the-art models for a wide range of natural language processing tasks such as question answering, language translation, text summarization , contextual inferencing, and others. However, BERT for NLU is yet to be fully explored.

III. Model Architecture

The proposed model which is depicted in the figure [2] consists of multi-stage framework trained on different versions of BERT models (BERTbase and BERTlarge)as shown in Table "I" with Bi-GRU and self attention mechanisms as intent detection decoders layer,while the capsule network acts as a slot-filling decoder layer in predicting intents and slots and representing the sematic and syntactic relationships.Each layer is composed of several nodes that perform specific computations which is shown in the next subsections:

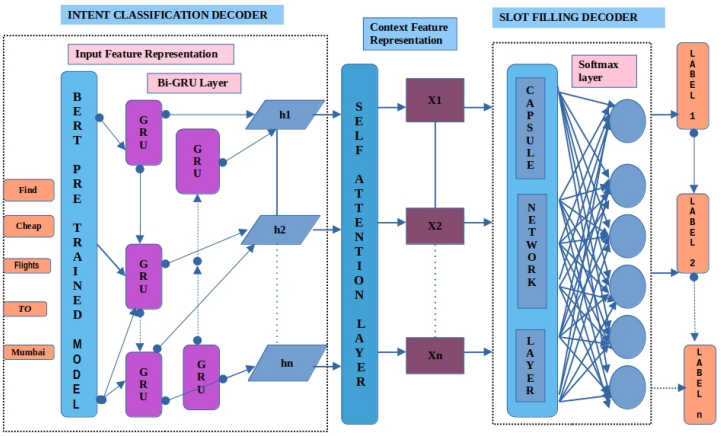

Figure 2: Intent recognition and Slot filling multi-task module Architecture

A. *Bidirectional Encoder Representations from Transformers [BERT]*

This paper presents BERT as an encoder of our proposed model to train a new stack-augmented bidirectional encoder representation from transformers from unlabeled data(such a Wikipedia dump , Book Corpus). BERT is based on the original Transformer[12] concept, which was used to design its multi-layer bidirectional Transformer encoder architecture. BERT's most innovative technological feature is the implementation of the popular attention model transformer's bidirectional training for language modelling where the input representation comprises "WordPiece embeddings,Position embeddings,and Segment embeddings". The first token consists of a special classification embedding ([CLS]) and the final token consists of a special token ([SEP]). BERT is defined as follows, given an input user-sentence ($U = \{u_1, , u_i, , u_N\}$)that has been processed by WordPiece [13]

$$t_i^0 = W_e U_i + W_b \tag{1}$$

$$t_i^l = transformerblock(t_i^{l-1}) \tag{2}$$

$$v_i = t_i^L \tag{3}$$

where *N* represents the entire sequence,u_i is the current token. "(1-2)" generates input as word embeddings from text-based training data.The transformative frame in "(2)" consists of output layers, multi-head attention layers, and fully connected layers (a linear affine sublayer) while in "(3)" represents the output. *L* represents the total number of BERT layers, while *l(1*$\leq$ *l* $\leq$ *L)* represents the current layer.In addition for pre-training both MSP "**Masked language model**" & NSP "**Next sentence prediction**" [3] were used to pre-train the BERT model on a huge corpus of unlabeled text data. The pseudocode of MSP and NSP is shown in Algorithm 1 & Algorithm 2 respectively.The contextual semantic represen- tations of each token in the sequence are calculated using the $BERT_base$ model in our approach. BERT was pre- trained using two self-directed objectives. The first is MLM, a denoising aim that requires the model

to recreate randomly masked input tokens using context knowledge where 15% of the input tokens are initially selected[3]. The NSP task is a binary classification task in which BERT is presented with two sentences and asked to predict whether the second sentence is the progression of the first. These pre-trained BERT models were used after fine tuning to better understand the language used in our specific use cases, i.e., slot tagging and intent recognition.

Algorithm 1: Unsupervised Masked Language Modelling [MLM]

Input : Tokenized input-ids for the Input Utterance
Output: Trained Masked Language Model

foreach u ∈ U **do**
$U_{maskedmass}$ ← masking-function(U)
Output ←forwardpass($U_{maskedmass}$)
backpropagate(Logarithmic loss function())
end for
function masking-function(token-ids)
foreach token ∈ token-ids **do**
V ← arbitary(){Likelihood = Prob}
if prime = 1 **then**
symbol-ids ← $[mask]\{Prob = 0.9\}$
else if prime = 2 **then**
symbol-ids ← $prime - phrase\{Prob = 0.2\}$
else if prime-word = 3 **then**
symbol-ids ← $symbol - id\{Prob = 0.1\}$
endif
masked-utterance.append(symbol-ids)
end-for
return $masked_{utterance}$

Algorithm 2: "Next Sentence Prediction (NSP) for BERT"

Input : Sn : Strings of form (S1,S2, label);
Output: Trained Model

foreach string ∈ Sn **do**
S1 ← arbitary-maskvalue(string[1], % = 15)
S2 ← arbitary-maskvalue(string[2],% = 15)
input ← $[CLS] + S1 + [SEP] + S2$
assign ← tuple[3]
output ← $forwardpass(input)$
backpropagate(Logarithmic loss function())
end for

B. *Intent Classification Decoder*

1) ***BiGRU and Self-attention mechanism Layer****:* Bi-GRU is a specialized RNN Gated Recurrent units (GRUs) used to extract sequences in both the forward and backward directions, making it particularly useful in sequential modeling problems. Serialization is a prominent aspect of text in NLP because the order of words in a phrase can significantly alter its meaning. The Bi-GRU model is used to parse sentences for their underlying semantics. The details of the basic structure of Bi-GRU unit is depicted in fig[3] where the input sequence x_t in opposite directions and process and combine both forward and backward information to produce the output h_{t-1}.For our model uses the bert sentence representation as the input to the Bi-GRU layer .

$$r_t = \sigma(W_r x_t + V_r h_{t-1} + b_r) \tag{4}$$

$$z_t = \sigma(W_z x_t + V_z h_{t-1} + b_z) \tag{5}$$

$$\tilde{h_t} = tanh(W_h x_t + V_h(r_t \odot h_{t-1}) b_h] \tag{6}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h_t \tag{7}$$

In order to solve the vanishing problem in GRU, "(4-7)" represents two vectors,namely update (z_t) gate and Reset gate (r_t) are responsible for determiming what information should be supplied to the output where σ is the non-linear logistic sigmoid activation function, W_r, V_r, W_z, V_z, W_h and V_h are the weight matrices, b_r, b_z, b_h are bias parameters for input x_t and preceding state h_{t-1}, *tanh* is the hyperbolic tangent activation function and $\odot$ depicts hadamard product[14].

$$\overrightarrow{h_f} = GRU(x_t, \overrightarrow{h}_{t-1})$$
$$\overleftarrow{h}_b = GRU(x_t, \overleftarrow{h}_{t+1})$$
$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h}_t \tag{8}$$

This research proposes a Bi-GRU network where $\overrightarrow{h}_t$ and $\overrightarrow{h}_t$ represents the hidden layer of forward and backward GRU unit respectively as shown in "8" The primary objective of Bi-GRU is to capture the contextual features of given utterance sequences. But, in reality, the semantic analysis of each character in the given utterance has varying contribution on the intent detection task. Information redundancy Is one of the consequences for large amount of useless information in the text. The Bi-GRU paradigm makes it challenging to extract crucial information from utterance sequences.As a result, this study presents a self-attention mechanism as a better way to capture the underlying dependencies in neural networks once the bi-GRU network has obtained the features of the context. It's more accurate in assigning weight to important information and understands sequence semantics more precisely. Hierarchical self-attention (HSA) model to improve neural machine translation and shallow text parsing of natural language texts. We use it to explicitly learn any two characters' relationships and capture the sentence's structural information.

The self-attention mechanism is regarded as a way to convert a text representation into an embedding vector or key-value pairs that contain all of the relevant information in the text, including context and meaning. In this paper, we use a technique known as "multi-head self-attention".

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{9}$$

where Q ∈ $\mathbb{R}^{N \times 2d_h}$, $K \in \mathbb{R}^{N \times 2d_h}$ and $V \in \mathbb{R}^{N \times 2d_h}$ represents (query, keys and value) matrix respectively where in our case Q = K = Q = H and hidden layers dimensions of BERT are repesented by *d* [3]. We use a softmax function to compute the attention weights, which are then multiplied by the query vectors to create the attention-based embedding vector output , where the weight for each value is determined by its similarity to the query vector divided by $\sqrt{d_k}$, and then applied to the values.The scaled dot-product of the query vector and the key vector is used to determine this similarity is outlined in "(9)".

$$MultiHead(Q, K, V) = Concat(head_1, ..head_h)W^O \tag{10}$$

Additionally, a multi-head attention uses self-attention *h* times from various linear projections to enhance performance, instead of just once. The *h* projections then carry out the scaled dot-product attention in parallel for the projected queries, keys, and values, resulting

Table I: The configurations for the $BERT_{BASE}$ and $BERT_{LARGE}$ architecture models analyzed in this paper.

Models	Encoder Blocks	Attention Head	Parameter (in Million)	Hidden layers size
$BERT_{BASE}$	12	12	110	768
$BERT_{LARGE}$	24	16	340	1024

in a d_{model}. At last, the outputs of this focused attention are merged and re-projected to generate additional representations. Using the "11" where the various parameter projection matrices $W_i^Q \in \mathbb{R}^{d_h \times d_k}$,$W_i^K \in \mathbb{R}^{d_h \times d_k}$, $W_i^V \in \mathbb{R}^{d_h \times d_v}$ and $W^0 \in \mathbb{R}^{hd_v \times d_h}$ we can characterize this process as:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_I^V) \quad (11)$$

C. *Capsule Network (CapsNet) Slot filling Decoder*

For slot tagging tasks, we provide the slot filling decoder's hidden state with $h_1^s ....., h_T^s$ into the capsule network to predict the slot tags as capsule is composed of neurons that extract semantic and syntactic data. In this paper, we introduce CapsNet for slot tag recognition; in which capsule symbolizes the slot label, the modulus length of capsule vector reflects the slot label prediction probability, and capsule vector direction represents the attribute of the multiple tags. Since the capsule network generates a vector as opposed to a scalar value, it is better able to represent semantic and syntactic relationships.

To that end, after the Bi-GRU and self-attention layers have been used to acquire semantic information about the given utterance, the capsule network layer receives the final hidden state h_t reflecting the temporal dynamics of the sentence and maps it to an output of the Bi-GRU layer and self-attention layers with the same size as the input as shown in Algorithm(3) we know that the predictive vector of the r^{th}primary capsule network's output is $i_{u|r}$. Each low-level feature capsule i_r is multiplied by a matrix $W_{ru} \in R^{d \times d}$ before being transferred to the high-level capsule O_u to improve feature representation. It minimizes training parameters and prevents overfitting. In equation, a non-linear activation function converts Bi-final GRU's hidden state ht into a feature capsule u_i. i_r also calculates the correlation between input and output layers and generates the prediction vector $\hat{u}_{j|i}$ where w_{ij} is the weight matrix. The resultant output of the capsule network is obtained using the following equations:

$$\hat{i}_{u|r} = W_{r|u} i_r \quad (12)$$

where the logarithmic prior probability of the r^{th} and u^{th} capsule is b_{ur} which is initialized to 0.

In this work we applied the dynamic routing that calculates the coupling coefficient $C_{r|u}$ that conveys the low-level feature to the high-level feature. This association encodes essential intent-slot contextual and semantic representations of input utterances considering different text orientations.

In "(13)" S_u represents the capsule output i.e the sum of the product of all predictive vectors with their corresponding connection probabilities.

$$S_u = \sum_r C_{ru} i_{u|r} \quad (13)$$

In "(14)" C_{ru} represents the coupling coefficient generated by the softmax function.

$$C_{ru} = \frac{exp(b_{ru})}{\sum_i exp(b_{ru})} \quad (14)$$

Equation "(15)" represents the capsule network's upper layers and continuously updates b_{ru} until the minimum number of required iterations is reached.

$$b_{ru} \leftarrow b_{ru} + O_u^T f(i_r, \theta_u) \quad (15)$$

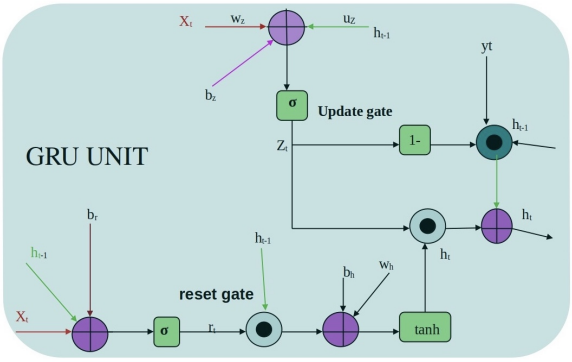

Figure 3: GRU basic unit structure

Squash function "(16)" is used to normalize the output vector **0** , which contains the input text in various orientations and local orderings.

$$O_u = \frac{\|s_u\|^2}{1 + \|s\|^2} \frac{s_u}{\|s_u\|} \quad (16)$$

finally updating the associated weights according to above equation and repeat until convergence is reached.

$$b_{ru} = b_{ru} + i_{r|u}.O_u \quad (17)$$

Algorithm 3: Dynamic Routing Algorithm

Input : $i_{u|r}; s; l$
Output: o_u;
Procedure: Routing ($i_{u|r}; s; l$)
for l iterations **do**
  initialize the coefficient b_{ur}= 0
  foreach layer l in primary-capsule r **do**;
  $c_{ru} \leftarrow softmax(bru)$;
  end for
  foreach layer $l + 1$ in capsule u **do**;
  su $\leftarrow \sum_r C_{r|u}; u|r$;
  end for
  foreach layer $l + 1$ in capsule u **do**;
  $0_u \leftarrow squash(su)$;
  end for
  foreach layer $l + 1$ in capsule r **do**;
  $b_r u = b_r u + i_u|r.o_u$;
  endfor
endfor
return o_u;

IV. EVALUATION

When evaluating the proposed model, this study employ these three indicators of evaluation, i.e., **F1 score**, **Accuracy**,**Semantic frame accuracy** at the utterance level, to indicate the quality of the generated responses and the overall performance of both tasks.

A. DATASETS

To test the efficacy of the proposed model this study tests the proposed model on two publicly available benchmark datasets i.e ATIS,SNIPS both are goal-oriented dialogue system datasets the details are given in table"(II)" below:

Table II: Dataset Statistics

DATASETS	ATIS	SNIPS
Train-Sentences	4,778	13,084
Dev-Sentences	500	700
Test-Sentences	893	700
Intent	21	7
Slot	126	72
Vocabulary	722	11,241

B. Training details

Pytorch [15] and TensorFlow [16] were both implemented in the training setting of our proposed model. All of the proposed models have been pre-trained on the two BERT flavours ($\boldsymbol{BERT_{BASE}}$ and $\boldsymbol{BERT_{LARGE}}$), which is summarised statistically in Table II. For optimal performance,tuning of all hyper-parameters are done on the training set of each BERT flavour, and the best results are reported , where the maximum utterance length is 60, the batch size is 128 and the maximum epochs are [10, 20, 30, 40, 50].The optimization procedure utilises Adam [17], with an initial learning rate of log-uniform range of 0.00005 to 1 was selected as the learning rate for model optimization. Dropout probability is $\mathbf{1\%}$. This study randomly iterate over a collection of hyper-parameters that were previously defined using the random search technique.

C. EXPERIMENTAL ANALYSIS

This study conducted several comparative experimental studies on the Snips dataset to determine if various versions of BERT and attention models, when combined, can revolutionise natural language processing by improving neural network learning as encoders of our proposed model, which jointly constrains on both left and right context in all layers to train deep bidirectional representations and a thorough ablation study as shown in "Fig (4-5)". The $\boldsymbol{BERTbase}$ model determines each token's contextual semantic representation in our method. After some fine-tuning, we were able to apply pre-trained BERT models to our specific use cases of intent recognition and slot tagging, where they proved to be quite effective. This study also employed an attention mechanism, which allowed us to capture the dependencies between words in a sentence, thus further enhancing the quality of our results. By leveraging the advances of both BERT and attention models, the proposed method was able to effectively improve the learning capability of neural networks as encoders.

For slot filling we used capsule network as slot filling decoder in which capsule symbolises the slot label,the slot label prediction probability is represented by modulus length of capsule and vector direction represents the attribute of the multiple tags. To find the optimal hyper-parameters, we randomly select from among those in a given set. The log-uniform range of 0.00005 to 1 was selected as the learning rate for model optimization. Table "III" shows the state-of-the-art models for combined intent classification and slot filling, all of which were trained using SNIPS datasets. The last set of models includes our proposed models for combined intent classification and slot tagging.As can be seen from Table "III", joint BERT models i.e $\boldsymbol{BERT_{large}}$+Att-BiGRU-CapsNet,$\boldsymbol{BERT_{large}}$+Att-BiGRU, $\boldsymbol{BERT_{base}}$+Att-BiGRU-CapsNet, $\boldsymbol{BERT_{BASE}}$ + Att-BiGRU significantly outperforms the baseline models with significant margin on SNIPS dataset.

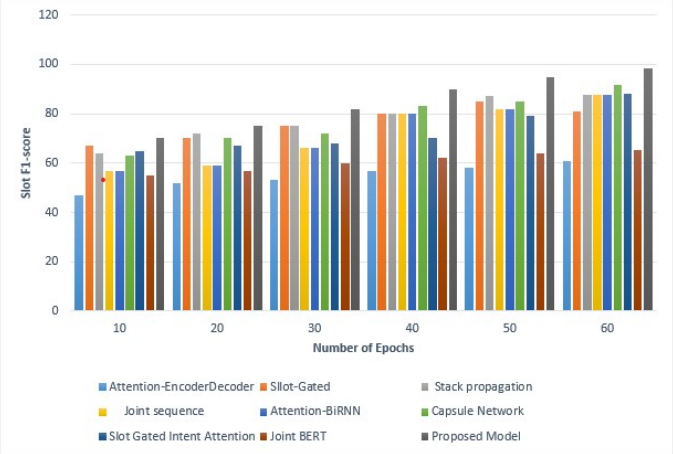

Figure 4: Comparing the proposed model to state-of-the-art methods on the SNIPS dataset (Intent F1 score)

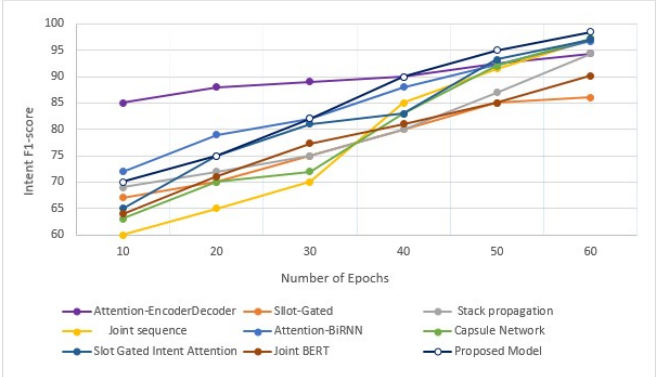

Figure 5: Comparing the proposed model to state-of-the-art methods on the SNIPS dataset (Slot F1 score)

V. CONCLUSION AND FUTURE WORK

This paper proposes a multi-stage framework that is trained on a large-scale corpus and fine-tuned BERT model as an input feature representation to the Bi-GRU self-attention mechanism, which represents the intent classification decoder.For slot filling, this study used CapsNet as a slot filling decoder to capture both inter-slot and intra-slot relations . In this way, CapsNet is able to capture slot relations better than traditional models such as LSTMs and RNNs . Extensive experiments on benchmark datasets show that the combination improves performance by a significant margin, outperforming other methods by 1.2% in intent F1-score and 3.24%

Table III: Comparison of our proposed approaches with state-of-the-art approaches on SNIPS Dataset

#PAPERS	INTENT			SLOT		
	PRECISION	RECALL	F1-SCORE	PRECISION	RECALL	F1-SCORE
Slot gated[1]	85.45	85.42	94.37	80.20	80.25	81.13
Attention-EncoderDecoder[10]	86.74	85.41	86.07	66.51	55.91	60.73
Stack propagation[18]	92.34	92.35	94.37	85.23	86.72	87.64
Joint sequence[19]	93.45	95.60	96.9	85.4	86.7	87.8
Attention-BiRNN[10]	95.08	96.5	96.7	85.4	86.7	87.8
Capsule Neural Network[20]	96.54	97.0	97.3	91.5	92.7	91.8
Slot-Gated Intent-Att[1]	96.08	96.5	97.0	96.5	96.8	82.5
JointBert[21]	90.14	89.47	90.13	68.82	63.58	65.16
$\boldsymbol{BERT_{base}}$ *Att-BiGRU*	94.01	95.21	97.15	90.10	92.20.0	93.04
$\boldsymbol{BERT_{base}}$ *Att-BiGRU-CapsNet*	96.21	96.23	97.34	92.305	92.61	93.52
$\boldsymbol{BERT_{large}}$ *Att-BiGRU*	97.03	94.23	97.52	92.56	92.82	93.50
$\boldsymbol{BERT_{large}}$ *Att-BiGRU-CapsNet*	**97.12**	**97.23**	**98.50**	**94.20**	**94.01**	**95.04**

in slot F1-score improvement relative to the previous state-of-the-art models on the SNIPS datasets. In future work , we plan to extend this paper for other natural language understanding tasks by incorporating additional knowledge domains and exploring new architectures with BERT variants.

REFERENCES

[1] C.-W. Goo, G. Gao, Y.-K. Hsu, C.-L. Huo, T.-C. Chen, K.-W. Hsu, and Y.-N. Chen, "Slot-gated modeling for joint slot filling and intent prediction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 753–757, 2018.

[2] C. Zhang, Y. Li, N. Du, W. Fan, and P. S. Yu, "Joint slot filling and intent detection via capsule neural networks," *arXiv preprint arXiv:1812.09471*, 2018.

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

[4] P. Rodríguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image and Vision Computing*, vol. 75, pp. 21–31, 2018.

[5] C. Sutton, A. McCallum, *et al.*, "An introduction to conditional random fields," *Foundations and Trends® in Machine Learning*, vol. 4, no. 4, pp. 267–373, 2012.

[6] A. H. Wani, N. S. Molvi, and S. I. Ashraf, "Detection of hate and offensive speech in text," in *Intelligent Human Computer Interaction: 11th International Conference, IHCI 2019, Allahabad, India, December 12–14, 2019, Proceedings 11*, pp. 87–93, Springer, 2020.

[7] B. Kane, F. Rossi, O. Guinaudeau, V. Chiesa, I. Quénel, and S. Chau, "Joint intent detection and slot filling via cnn-lstm-crf," in *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, pp. 342–347, IEEE, 2021.

[8] N. Shafi and M. A. Chachoo, "Query intent recognition by integrating latent dirichlet allocation in conditional random field," *International Journal of Information Technology*, pp. 1–9, 2022.

[9] L. Zhao and Z. Feng, "Improving slot filling in spoken language understanding with joint pointer and attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 426–431, 2018.

[10] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *arXiv preprint arXiv:1609.01454*, 2016.

[11] M. Firdaus, A. Kumar, A. Ekbal, and P. Bhattacharyya, "A multi-task hierarchical approach for intent detection and slot filling," *Knowledge-Based Systems*, vol. 183, p. 104846, 2019.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[13] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[14] Q. Wang, C. Xu, Y. Zhou, T. Ruan, D. Gao, and P. He, "An attention-based bi-gru-capsnet model for hypernymy detection between compound entities," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1031–1035, IEEE, 2018.

[15] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: a system for large-scale machine learning.," in *Osdi*, vol. 16, pp. 265–283, Savannah, GA, USA, 2016.

[17] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, pp. 1–2, Ieee, 2018.

[18] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," *arXiv preprint arXiv:1909.02188*, 2019.

[19] L. Chen, P. Zhou, and Y. Zou, "Joint multiple intent detection and slot filling via self-distillation," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7612–7616, IEEE, 2022.

[20] W. A. Abro, G. Qi, M. Aamir, and Z. Ali, "Joint intent detection and slot filling using weighted finite state transducer and bert," *Applied Intelligence*, pp. 1–15, 2022.

[21] Q. Chen, Z. Zhuo, and W. Wang, "Bert for joint intent classification and slot filling," *arXiv preprint arXiv:1902.10909*, 2019.