Attention-Based Joint Learning for Intent Detection and Slot Filling Using Bidirectional Long Short-Term Memory and Convolutional Neural Networks

(AJLISBC)

Yusuf Idris Muhammad, Naomie Salim, Sharin Hazlin Huspi, Anazida Zainal
Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81310, Malaysia

***Abstract*—Effective natural language understanding is crucial for dialogue systems, requiring precise intent detection and slot filling to facilitate interactions. Traditionally, these subtasks have been addressed separately, but their interconnection suggests that joint solutions yield better results. Recent neural network-based approaches have shown significant performance in joint intent detection and slot filling tasks. The two primary neural network structures used are recurrent neural networks (RNNs) and convolutional neural networks (CNNs). RNNs capture long-term dependencies and store previous information semantics in a fixed-size vector, but their ability to extract global semantics is limited. CNNs can capture n-gram features using convolutional filters, but their performance is constrained by filter width. To leverage the strengths and mitigate the weaknesses of both networks, this paper proposes an attention-based joint learning classification for intent detection and slot filling using BiLSTM and CNNs (AJLISBC). The BiLSTM encodes input sequences in both forward and backward directions, producing high-dimensional representations. It applies scalar and vectorial attention to obtain multichannel representations, with scalar attention calculating word-level importance and vectorial attention assessing feature-level importance. For classification, AJLISBC employs a CNN structure to capture word relations in the representations generated by the attention mechanism, effectively extracting n-gram features. Experimental results on the benchmark Airline Travel Information System (ATIS) dataset demonstrate that AJLISBC outperforms state-of-the-art methods.**

***Keywords*—Joint learning; intent detection; slot filling; multichannel**

I. Introduction

Owing to the integration of conversational agents into various applications, from virtual assistants to customer chatbots, the importance of accurately interpreting user inputs increases. In the field of Natural Language Understanding (NLU), two primary tasks are intent detection and slot filling [1, 2]. Intent detection is a classification problem involving the construction of features from a given utterance. These features are then subjected to a classification algorithm to predict the appropriate classes for utterances selected from the predefined classes [3].

Although intent detection is a classification problem, it differs from classical classification in that it addresses the spoken language. Therefore, engineered features must be oriented towards capturing the semantic meanings of these utterances [4]. This emphasis on semantics is crucial to understanding the underlying intent conveyed by the user. Recent approaches have expanded beyond the semantic content of individual words to internal aspects such as syntactic structures, word contextual relationships, and external information such as metadata [5].

Slot filling is a sequence-labeling problem that is used to identify the semantic constituents of a user's utterance and assign a semantic label to each word. The purpose of these labels is to describe the type of semantic information carried by the token, which can help identify the intent of the user [6, 7].

Traditionally, intent detection and slot filling tasks have been treated separately and assembled to form an entire system [8]. This type of methodology provides conceptual clarity, with each component independently addressing its specific challenges. However, there are some limitations in separating these models. It fails to leverage the interaction between the intent detection task and slot filling task, and this interaction plays a role in enhancing the overall system performance [9, 10]. Recent advances in Artificial Intelligence (AI), particularly deep learning, have opened the door to joint models. A joint model handles both intent detection and slot filling simultaneously by leveraging their interdependencies and shared representations to enhance overall performance and efficiency [11].

Encoder-decoder neural network architectures are generally used for the joint learning classification of intent detection and slot filling because of their powerful sequential processing capabilities. Early joint learning approaches were based on statistical models such as Support Vector Machines (SVMs) [12], maximum entropy models (MEM) [13], hidden Markov models (HMM) [14], and Conditional Random Fields (CRFs) [15], which require extensive feature engineering and struggle to capture the deep semantic nuances of language. The advent of deep learning has brought about a shift, enabling models to learn hierarchical representations from raw data. Convolutional Neural Networks (CNNs) [16], Recurrent Neural Networks (RNNs) [17], and transformer architecture [18] have been at the forefront of this revolution, offering powerful tools for sequence modeling.

A Recurrent Neural Network (RNN) is a widely used architecture for Natural Language Processing (NLP) tasks owing to its ability to maintain memory and capture

dependencies and patterns over time. This memory is implemented using recurrent connections within the network, allowing information to persist and update when new inputs are processed [19]. RNNs are particularly effective for intent detection and slot-filling tasks [20]. For example, in intent detection, understanding a user's intent requires considering the sequential nature of the dialogue. RNNs with recurrent connections can capture the context of previous words or phrases in a sentence, thereby enabling them to detect the intent of the user based on the entire input sequence. In slot-filling tasks, the presence and placement of entities within the input text are crucial for extracting slot values accurately [21]. RNNs can learn to recognize patterns in the sequential structure of sentences, allowing them to identify relevant slots based on their contextual relationships with other words or entities in a sentence [22]. Despite their capabilities, RNNs often face gradient vanishing or exploding issues. To address these challenges, Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) have been developed to improve memory handling. Another problem with RNNs is that long sentences tend to prioritize recent information over earlier information, which might be more significant.

To address the problem of input element selection, an attention mechanism was introduced to assign different weights to the output of the RNNs. The essence of this mechanism is to allow RNNs to combine outputs according to their assigned importance and retain variable-length memory. Attention mechanisms have proven to be effective in joint learning tasks, where different parts of the input may be relevant to intent detection and slot filling. By assigning different weights to different parts of the input, attention mechanisms help models prioritize the information that is crucial for each task. However, it has limitations in terms of capturing the order of the input sequence, which is crucial for NLP tasks. For instance, the sentences "I want flight from Baltimore to Dallas" and "I want flight from Dallas to Baltimore" will have identical weighted sums despite having opposite meanings.

Convolutional Neural Networks (CNNs) are another architecture for NLP tasks, known for their ability to learn spatial hierarchies and local correlations of features from input data using convolutional filters. For instance, 2-gram features can effectively be extracted from the given example such as "from Baltimore" and "from Dallas" likewise "to Dallas" and "to Baltimore" using CNNs. This type of representation provides better information than the sum of the RNN hidden states in the input sequence. Several studies have demonstrated the importance of CNNs for NLP tasks. In [23], it was demonstrated that a simple CNN consisting of a single convolutional layer applied to word vectors derived from an unsupervised neural language model achieved good performance in text classification. In addition, [24] illustrated that CNNs can be effectively utilized to extract morphological details such as word suffixes or prefixes and encode them into neural representations. However, CNNs are limited in preserving the sequential order [25].

Some researchers have developed hybrid frameworks that combine CNNs and RNNs to exploit their respective strengths. One such framework, the Recurrent Convolutional Neural Network (RCNN), captures contextual information using a recurrent convolutional structure [26]. Another framework, the Convolutional Recurrent Neural Network (CRNN), combines the benefits of both CNNs and RNNs to extract diverse linguistic features [27]. However, these hybrid models often fail to account for the varying semantic contributions of different words when treating all words with equal importance.

Motivated by the abovementioned issues, this study proposes a joint learning classification model that leverages the strengths of RNN and CNN architectures, enhanced with scalar and vectorial attention mechanisms. The major contributions of the proposed model are summarized as follows:

1) The model employs BiLSTM to encode the input sequence in both forward and backward directions, ensuring the retention of chronological features within sequences.

2) An attention mechanism is introduced to generate multiple channels, simulating the diversity of the input information. Scalar attention assesses word-level importance, whereas vectorial attention evaluates feature level significance. This representation allows the model to learn multiple representations of the semantics of an input sequence.

3) CNN is utilized to identify word relations using attention mechanisms rather than relying on weighted sum calculations. This approach enhances the ability of CNN to extract n-gram features.

4) A series of experiments conducted on the Airline Travel Information System (ATIS) dataset demonstrated that the proposed approach outperformed baseline methods.

The remainder of this paper is organized as follows: Section II reviews the related work, Section III outlines the methodology, Section IV describes the experimental setup, Section V discusses the experimental results, Section VI provides an in-depth analysis of the findings, and Section VII concludes the paper.

II. Related Work

Joint learning for intent detection and slot filling has evolved from classical models, such as triangular-chain CRFs [28] and Maximum Entropy Models (MEM) combined with CRFs [29], to capture the dependencies between the intent and slots of an utterance. However, they face scalability and manual feature engineering challenges [5].

Deep learning approaches have emerged as more scalable alternatives. Recently, attention-based joint learning for intent detection and slot filling has gained popularity owing to its ability to enhance task performance by improving feature extraction and the flow of information between these two interdependent tasks. By focusing on the most relevant parts of the input sequences, attention mechanisms allow models to capture fine-grained contextual relationships, making them highly effective for natural language understanding.

The research in [30] introduced an asynchronous joint extraction algorithm that combines a GRU network with a TextCNN-based feature representation layer. Their model incorporated a keyword attention mechanism to capture contextual semantics precisely, enhancing both intent detection and slot filling. Adding adversarial training further strengthens

the robustness of the model against adversarial attacks, thereby improving its reliability in real-world scenarios.

The study in [31] proposed a joint model leveraging graph neural networks (GNNs) fused with external knowledge and a graph attention mechanism. This model significantly enhances the semantic representation by facilitating the exchange of information between slots and intents, resulting in superior task performance. Similarly, [32] emphasized bidirectional information flow within GNN-based models, improving information exchange and interaction between intent recognition and slot labeling processes.

Using a different approach, [11] developed the JPIS model, which integrates user-specific profile information along with a slot-to-intent attention mechanism. This approach proved highly effective in scenarios where profile-based customization was required, substantially improving accuracy.

The study in [8] also explored the efficiency of attention mechanisms by developing a Fast Attention Network tailored to edge devices. This model balances accuracy with latency by utilizing a refined attention module to enhance semantic accuracy, while maintaining fast response times in real-time applications.

Although these models have demonstrated significant advancements in intent detection and slot filling, they highlight the need to balance model complexity with computational efficiency, especially for real-time applications. Enhanced scalar and vectorial attention mechanisms offer a potential solution by allowing the model to capture both the word- and feature-level importance in a more structured manner. Scalar attention assesses the significance of individual words, whereas vectorial attention evaluates feature-level relevance, enabling the model to generate multiple representations of input sequences that can be processed concurrently. This approach enhances the richness of the information captured, thereby improving the overall performance while maintaining the computational efficiency. Thus, the proposed enhanced scalar and vectorial attention mechanisms are justified by their ability to address these existing challenges while optimizing the interaction between intent detection and slot-filling tasks.

III. METHODOLOGY

The architecture of the proposed model is shown in Fig. 1. The proposed model comprises an input layer, a BiLSTM layer, a convolutional layer with subsequent max pooling, and two dense layers that implement softmax functions. These components jointly detect the intent of the user's input utterance and the associated slots by assigning them with multiclass labels (B, I, O), where "I," "O," and "B" signify Inside, Outside, and Beginning of slots, respectively. Details of the model are described in the following subsections.

A. Embedding Layer

First, the dialog must be transformed into a feature vector matrix to serve as the input layer of the model. In the proposed model, Google's word2vec [33] embedding technique is employed to translate each word feature into a word-embedding vector. As a result, dialog vectors are obtained as inputs $X = (x_1, x_2, \ldots, x_n)$.

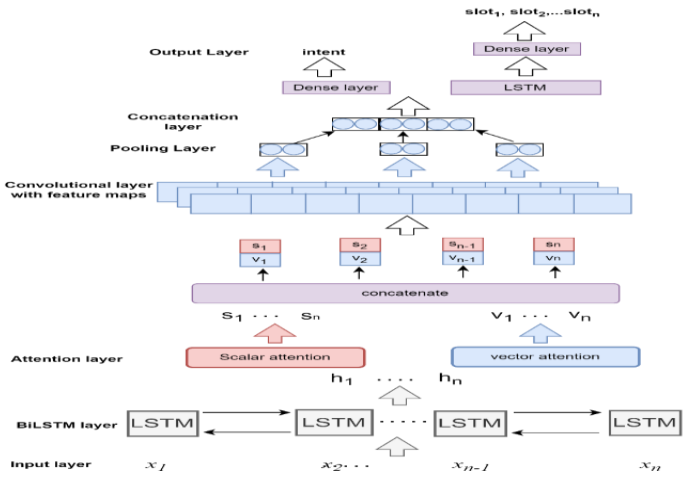

Fig. 1. The architecture of the proposed AJLISBC model.

B. Long Short-Term Memory Network

RNNs are widely used in NLP owing to their ability to handle sequential data and capture temporal dependencies. RNNs process a variable-length sequence at each time step t and updates its hidden state ht based on the current input x_t and previous hidden state h_{t-1}:

$$h_t = f(W[h_{t-1}, x_t] + b) \quad (1)$$

where, W is the weight matrix that combines the hidden states, and the current input vector and b is the bias vector.

However, basic RNNs have been avoided by researchers owing to issues such as the vanishing gradient problem. To address these problems, LSTM networks have been developed and have demonstrated good performance.

To convert a sentence consisting of n words, into a dense vector $x_{i,}$ an embedding matrix is used first. BiLSTM is then applied to generate word annotations by processing the sentence in both forward and backward directions. The forward LSTM process the sequence from x_i to x_n and produce $\overrightarrow{h_i}$,

whereas backward LSTM processes the sequence from x_n to x_i and produce $\overleftarrow{h_i}$

$$\overrightarrow{h_i} = LSTM\left(x_i, \overrightarrow{h_{i-1}}\right) \quad (2)$$

$$\overleftarrow{h_i} = LSTM\left(x_i, \overleftarrow{h_{i-1}}\right) \quad (3)$$

$$h_i = \overrightarrow{W_i}.\overrightarrow{h_i} + \overleftarrow{W_i}.\overleftarrow{h_i} + b \quad (4)$$

The hidden states from the forward $\overrightarrow{h_i}_i$ and backward $\overleftarrow{h_i}$ LSTMs are concatenated at each time step to provide a summary of the input sequence h_i.

C. Attention Mechanism

In NLP tasks such as intent detection and slot filling, not all words have the same significance in representing the input sequence. To address this, an attention mechanism was introduced to highlight the importance of each word by assigning greater weights to the crucial elements in the final output. However, the traditional attention mechanism struggles to preserve temporal order information. To resolve this, attention mechanisms are incorporated into the hidden states of BiLSTM, and these states are combined into a matrix that maintains the order information instead of relying on the weighted sum of vectors. In addition, by employing scalar and vectorial attention, multiple matrices were created, serving as multichannel inputs to the CNN.

1) Scalar attention mechanism: To determine the importance weights of all input sequences, scalar attention was employed. This attention is represented by a matrix M which captures the relationships between words in the sequence. The value in the ith row and the jth column of M indicates the level of association between the word in ith and jth column. In each channel L, a mask matrix V is applied, and the masked association matrix Mli, is calculated.

$$M_{l_{i,j}} = \tanh\left(\left[h_i, W_l.h_j\right] + b_l\right) \quad (5)$$

The i^{th} channel mask matrix $V_{l_{i,j}}$ obeys binomial distribution and is defined as:

$$V_{l_{i,j}} \sim B(1,p), i \in [1,n], j \in [1,n] \quad (6)$$

Given association matrix $M_{l_{i,j}}$ and mask matrix $V_{l_{i,j}}$, the channel is calculated as follows:

$$A_l = M_l \otimes V_l \quad (7)$$

$$s_{l_k} = \sum_x A_{l_{xk}} \quad (8)$$

$$p_k = \{^{-99999,}_{0,} \quad ^{if\ x_k\ is\ from\ pad}_{otherwise} \quad (9)$$

$$score_{l_k} = p_k + s_{l_k} \quad (10)$$

$$a_{l_k} = \frac{exp(score_k)}{\sum_i^n exp(score_i)} \quad (11)$$

$$c_{l_i} = a_{l_i}.h_i \quad (12)$$

where, $\otimes, \boldsymbol{c_{l_i}}, \boldsymbol{h_i}, \boldsymbol{l^{th}}, \boldsymbol{p_k}$ denotes element wise multiplication, updated hidden state, channel and padded mask respectively. To ensure that the padding symbol contains nearly zero attention $\boldsymbol{a_{l_k}}$, scalar attention $\boldsymbol{s_{l_k}}$ is subtracted from 99999 before applying the softmax function.

2) Vectorial attention mechanism: In NLP, words and sentences are transformed into *n*-dimensional vector to capture their meanings in a format suitable for computational models. Each dimension within this vector encodes a different aspect of a word or sentence's meaning, allowing for a rich and multifaceted representation of linguistic data. For example, consider the sentence "I want to book a flight from New York to Boston on July 20th". In intent detection task, the model can easily identify the intent as "book a flight" by focusing on dimensions related to booking and travel. In slot-filling tasks, the model can accurately fill slots using dimensions related to locations and dates.

A vectorial attention mechanism was proposed for the joint model based on the above assumptions.

$$score_{l_i} = W_a^T \sigma(W_b.h_i + b) \quad (13)$$

$$a_{vl_i} = \frac{exp\left(score_{l_i}\right)}{\Sigma_i\left(score_{l_i}\right)} \quad (14)$$

$$c_{vl_i} = a_{vl_i} \odot h_i \quad (15)$$

$$C_l = \left[c_{vl_1}, c_{vl_2}, c_{vl_3} \ldots., c_{vl_n}\right] \quad (16)$$

where W_a, W_b are weight matrices in the vectorial attention and b is the bias vector, $\odot$ is the element wise multiplication c_{l_i} denotes the output of h_i in l^{th} channel. Multichannel attention is generated by concatenating vector and scalar attention as follows:

$$c_{l_i} = a_{l_i}\left(a_{vl_i} \odot h_i\right) \quad (17)$$

Therefore, multichannel attention has the strengths of both vectorial attention and scalar attention.

D. Convolutional Neural Network Layer

To extract local features from the attention layer, convolution operations are employed on the combined attention layer. Typically, a zero-padding token is introduced before convolution to ensure uniform output sizes across different filters. Different filters and kernel sizes were applied to the multichannel attention c_{li}, to extract local features.

$$C = [c_1, c_2, \ldots c_n] \quad (18)$$

In the convolution operation, a filter $m \in \mathbb{R}^{l \times k}$ is applied to l consecutive words to generate a new feature. Here, $C \in \mathbb{R}^n$, where *k* and *n* are the embedded dimensions and input sequence length, respectively.

$$x_i = f(m.c_{i:i+l-1} + b) \quad (19)$$

where, $c_{i:i+l-1}$ is the concatenation of $c_i \ldots c_{i+l-1}$, f is a nonlinear activation function such as RELU, and $b \in \mathbb{R}$ is the bias term. After the filter *m* slides across, a feature map can be obtained as,

$$z = [z_1, z_2, \ldots, z_{n-l+1}] \quad (20)$$

Maxpooling is then applied to the feature map *z* to extract the most significant features for each filter *m*. To capture the different features of the input sequence, filters of various sizes are applied, resulting in a vector that is used at the output layer.

E. Output Layer

The proposed model consists of intent detection and slot filling outputs. The intent detection output is obtained using a fully connected layer with a softmax function to output the probability distribution over the intents. Therefore, the intent output vector is computed as follows:

$$y^i = Softmax(W.q + b) \quad (21)$$

For the slot-filling output, the feature vectors are passed to an LSTM decoder to capture the sequential nature of the slots and use the softmax function for the output.

$$d_i = LSTM(q_i, d_{i-1}) \quad (22)$$

$$y_i^s = softmax(W.d_i + b) \quad (23)$$

where, y^s is the slot label and W, b are the transformation matrix and bias vectors, respectively.

IV. Experimental Study

This section describes the datasets used in the experiments followed by a detailed experimental methodology to assess the effectiveness of the proposed approach. A comparative analysis of the baseline methods is presented. The performance of the model was evaluated using the widely adopted metrics of accuracy for the intent detection task and the F1-score for the slot-filling task.

A. Dataset

To validate the proposed model, experiments were conducted using the Airline Travel Information System (ATIS) dataset, which is one of the most widely recognized and historically significant datasets in Natural Language Understanding (NLU) research. The ATIS dataset has been a benchmark for Spoken Language Understanding (SLU) tasks for over three decades, making it an ideal choice for evaluating advancements in the field. The dataset focuses specifically on air travel-related queries and provides information on flights, fares, airlines, airports, cities, and ground services. It features 21 different intents and 128 slots, with a training set of 4478 samples, test set of 893 samples, and validation set of 500 samples [5].

The ATIS dataset presents several unique characteristics that support its use as a standard for model comparisons. One notable feature is the imbalanced distribution of intent types, with approximately 75% of intents belonging to a single class (atis_flight). This imbalance poses a challenge for intent detection models, making the dataset a rigorous test for the proposed approach. Moreover, the well-defined structure of the dataset allows for clear benchmarking and facilitates a direct comparison with existing models in the NLU domain. Its long-standing use in research ensures that the performance of the proposed model can be contextualized within the vast body of prior work, further validating its efficacy.

Table I gives an example of a semantic frame for an utterance from an ATIS dataset "I want fly from Baltimore to Dallas round trip." The slots adhere to the widely used IOB (in-out-begin) format for representing slot tags. This sentence pertains to airline travel with the intent of finding a flight. Notably, 'Baltimore' is tagged as departure city, 'Dallas' as arrival city and 'round trip' as round trip.

TABLE I. An Example of Frame

Entity	slots	Intent
I	O	atis_flight
want	O	
to	O	
fly	O	
from	O	
Baltimore	B-fromloc.city_name	
to	O	
Dallas	B-toloc.city_name	
round	B-round_trip	
trip	I-round_trip	

B. Experimental Settings

A grid search is employed to determine the optimal hyperparameters for the model. Specifically, three different filter sizes (2, 3, and 5) were tested and 128 feature maps were used. To prevent overfitting, a rate of 0.5 was applied to the feature maps. The shared encoder was configured with 200 hidden units and a rectified linear unit activation function was used. Additionally, a dropout rate of 0.5 was applied after the shared encoder, randomly dropping units to improve training was applied after the shared encoder. For intent detection classification and slot filling outputs, L2 regularization with a value of 0.001 was applied to the weights of the dense layers using a softmax activation function. The Adam optimizer and categorical cross-entropy loss functions were employed during training. Accuracy metrics were used to evaluate intent detection, and the F1-score was used for slot filling. A batch size of 32 was selected for the study. The input sequence was padded to a fixed length to fit the convolutional layer with a maximum length of 45 for the ATIS dataset. In the proposed model, the weights of the embedding layer were initialized with publicly available word2vec vectors, whereas words not included in the pretrained set were initialized with values from a uniform distribution to maintain consistent variance across all word vectors.

V. Experimental Results

The performance of the proposed model, AJLISBC-x, on the ATIS dataset is presented in Table I, where x represents the number of channels used during training. These channels enable the model to capture different representations of the input sequence, and the effectiveness of this multichannel representation is evident in the results.

As illustrated in Table II, all proposed AJLISBC models outperformed the baseline models. Specifically, AJLISBC-2, which utilizes two channels, demonstrated the highest accuracy

and F1-score. Among the configurations tested, AJLISBC-1, which operates with a single channel, showed inferior results compared with the multichannel models. This indicates that increasing the number of channels positively influences the model performance, although there may be diminishing returns as the number of channels increases beyond two.

TABLE II. COMPARISON OF AJLISBC WITH BASELINE RESULTS

Model	ATIS Dataset	
	Accuracy	F1-score
Bi-GRU + feature [34]	97.76	97.93
BiLSTM+Attention [35]	95.70	95.60
BC [36]	97.20	96.34
AJLISBC -1	97.89	98.32
AJLISBC -2	**98.19**	**98.61**
AJLISBC -3	97.99	98.38
AJLISBC -4	98.09	98.45
AJLISBC -5	97.89	98.56

In addition to channel variations, the effects of the attention mechanisms were evaluated. Table III presents the performance of AJLISBC-2 when using scalar attention, vectorial attention, or a combination of both. Scalar attention, which assigns importance weights to all elements of the input sequence, yields a slightly better accuracy than vectorial attention. However, both types of attention achieve the same F1-score, demonstrating that either mechanism is effective at improving the performance for this task. When scalar and vectorial attention are combined, the model achieves its highest F1-score and improved accuracy compared with either mechanism alone.

TABLE III. PROPOSED MODEL PERFORMANCE BASE ON ATTENTION MECHANISM

Model	Attention	ATIS Dataset	
		Accuracy	F1-score
AJLISBC -2	Scalar	97.09	98.42
AJLISBC -2	Vector	96.99	98.42
AJLISBC -2	Scalar + vector	97.89	98.56

VI. DISCUSSION

The results underscore the effectiveness of multichannel representation in enhancing the model performance. AJLISBC-2's superior performance compared to AJLISBC-1 suggests that using multiple channels helps the model capture diverse patterns within the input sequence. This is particularly relevant for complex tasks such as intent detection and slot filling, where different dimensions of the input can provide complementary information. The use of a single channel in AJLISBC-1 limits the ability of the model to process and leverage multiple facets of the input, leading to inferior results. Therefore, multichannel representation appears to be an effective strategy for improving the model performance.

However, it is worth noting that while increasing the number of channels generally improves the performance, the results show that the performance does not increase indefinitely. For example, AJLISBC-5 showed a slightly lower accuracy than AJLISBC-2, indicating that simply adding more channels may not necessarily result in better performance beyond a certain point. This may be due to the model encountering diminishing returns from the additional channels or because the increased complexity of the model requires more sophisticated optimization strategies. It is hypothesized that selecting the number of channels based on the number of informative words in a sentence can yield even better results, allowing the model to tailor the complexity of its representation to the specific needs of each input.

In examining attention mechanisms, scalar attention proves to be particularly effective for accuracy because of its ability to calculate the importance of all elements in the input sequence. This helps to identify the most relevant parts of the sequence for intent detection, which may explain its superior performance in this regard. Scalar attention is particularly useful in scenarios where the relationship between different elements in a sequence plays a crucial role, such as in slot-filling tasks. By contrast, vectorial attention selectively emphasizes features that are more relevant for specific tasks, thereby enhancing the robustness of the model. This mechanism introduces controlled perturbations in the hidden state, which allows the model to generalize more effectively to new inputs.

The combination of scalar and vectorial attention mechanisms leads to the best performance because it capitalizes on the strengths of both methods. Scalar attention helps to compute the overall importance of elements in the input, whereas vectorial attention fine-tunes the focus to specific dimensions of the input. This dual approach results in better performance in both intent detection and slot-filling tasks. The synergy between these two mechanisms also enables the model to indirectly assign varying learning rates to different dimensions of the hidden state, allowing more informative dimensions to be updated more rapidly than less informative ones do. This dynamic adjustment contributes to the observed performance improvement when both types of attention are used together.

VII. CONCLUSION

This paper presents an attention-based joint learning classification model for intent detection and slot-filling that combines BiLSTM and CNN (AJLISBC). The BiLSTM architecture captures contextual information, whereas scalar and vectorial attention mechanisms generate multichannel representations of the input sequence semantics. CNNs are applied to these multichannel representations to extract n-gram features and enhance performance in both intent detection and slot-filling tasks. Experimental results on the ATIS dataset show that the model outperforms baseline models, demonstrating the effectiveness of combining BiLSTM, CNN, and attention mechanisms for natural language understanding tasks. Despite the promising results, several limitations of this study should be acknowledged, as they may impact the validity, reliability, and generalizability of the findings. One limitation is the exclusive use of the ATIS dataset, which is relatively small, domain-specific, and focuses on flight-related queries.

This limited scope raises concerns about the generalizability of the findings to other domains or to larger, more diverse datasets. The model's performance might have been overestimated owing to the homogeneity of the dataset. Future work will involve testing the model on diverse datasets from various domains to better assess its generalizability and robustness across different natural language processing tasks. Another limitation is the manual selection of the number of channels used for multichannel representations. Although multichannel representations have shown effectiveness, the process of determining the optimal number of channels is empirical and not rigorously optimized. This could affect the reliability and consistency of the performance of the model across different datasets or tasks, as it may not generalize well to varying sentence lengths or input complexities. Future work will explore automated methods for determining the optimal number of channels, such as incorporating adaptive mechanisms based on input-data characteristics. This approach ensures that the model adapts more flexibly and consistently to diverse input scenarios. Future studies will address these limitations to further validate the effectiveness of the model and enhance its adaptability and applicability to a broader range of natural language understanding tasks.

ACKNOWLEDGMENT

This research was partly funded by the Ministry of Higher Education Malaysia under grant R.J130000.7851.5F568. The authors would also like to thank Universiti Teknologi Malaysia (UTM) for providing the resources used in this study.

REFERENCES

[1] P. Ni, Y. Li, G. Li, and V. Chang, "Natural language understanding approaches based on joint task of intent detection and slot filling for IoT voice interaction," Neural Computing and Applications, vol. 32, pp. 16149-16166, 2020.

[2] A. Algherairy and M. Ahmed, "A review of dialogue systems: current trends and future directions," Neural Computing and Applications, vol. 36, no. 12, pp. 6325-6351, 2024.

[3] S. Huang, P. Huang, Y. Xu, J. Liang, and J. Niu, "Exploring Label Hierarchy in Dialogue Intent Classification," in ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024: IEEE, pp. 11511-11515.

[4] T. Wu, M. Wang, Y. Xi and Z. Zhao, "Intent recognition model based on sequential information and sentence features," Neurocomputing, vol. 566, p. 127054, 2024.

[5] H. Weld, X. Huang, S. Long, J. Poon, and S. C. Han, "A survey of joint intent detection and slot filling models in natural language understanding," ACM Computing Surveys (CSUR), 2021, doi: https://doi.org/10.1145/3547138.

[6] 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, doi: https://doi.org/10.1016/j.knosys.2019.07.017.

[7] A. S. M. Zailan, N. H. I. Teo, N. A. S. Abdullah, and M. Joy, "State of the Art in Intent Detection and Slot Filling for Question Answering System: A Systematic Literature Review," International Journal of Advanced Computer Science & Applications, vol. 14, no. 11, 2023.

[8] L. Huang, S. Liang, F. Ye, and N. Gao, "A fast attention network for joint intent detection and slot filling on edge devices," IEEE Transactions on Artificial Intelligence, 2023.

[9] J. Wu, I. G. Harris, H. Zhao, and G. Ling, "A graph-to-sequence model for joint intent detection and slot filling," in 2023 IEEE 17th International Conference on Semantic Computing (ICSC), 2023: IEEE, pp. 131-138.

[10] M. Firdaus, H. Golchha, A. Ekbal, and P. Bhattacharyya, "A deep multi-task model for dialogue act classification, intent detection and slot filling," Cognitive Computation, vol. 13, no. 3, pp. 626-645, 2021, doi: https://doi.org/10.1007/s12559-020-09718-4.

[11] T. Pham and D. Q. Nguyen, "JPIS: A Joint Model for Profile-Based Intent Detection and Slot Filling with Slot-to-Intent Attention," in ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2024: IEEE, pp. 10446-10450.

[12] F. Mairesse et al., "Spoken language understanding from unaligned data using discriminative classification models," in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009: IEEE, pp. 4749-4752.

[13] Y.-Y. Wang, "Strategies for statistical spoken language understanding with small amount of data-an empirical study," in Eleventh Annual Conference of the International Speech Communication Association, 2010.

[14] A. Celikyilmaz and D. Hakkani-Tur, "A joint model for discovery of aspects in utterances," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2012, pp. 330-338.

[15] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," in 2013 ieee workshop on automatic speech recognition and understanding, 2013: IEEE, pp. 78-83.

[16] M. Giménez, A. Fabregat-Hernández, R. Fabra-Boluda, J. Palanca, and V. Botti, "A detailed analysis of the interpretability of Convolutional Neural Networks for text classification," Logic Journal of the IGPL, p. jzae057, 2024, doi: https://doi.org/10.1093/jigpal/jzae057.

[17] A. Orvieto et al., "Resurrecting recurrent neural networks for long sequences," in International Conference on Machine Learning, 2023: PMLR, pp. 26670-26698.

[18] K. K. Jayanth, G. B. Mohan, R. P. Kumar, and M. Rithani, "Intent Recognition Leveraging XLM-RoBERTa for Effective NLU," in 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2024: IEEE, pp. 877-882.

[19] U. Farooq, M. S. Mohd Rahim, and A. Abid, "A multi-stack RNN-based neural machine translation model for English to Pakistan sign language translation," Neural Computing and Applications, vol. 35, no. 18, pp. 13225-13238, 2023.

[20] W. A. Abro, G. Qi, Z. Ali, Y. Feng, and M. Aamir, "Multi-turn intent determination and slot filling with neural networks and regular expressions," Knowledge-Based Systems, vol. 208, p. 106428, 2020.

[21] M. Jbene, S. Tigani, R. Saadane, and A. Chehri, "A robust slot filling model based on lstm and crf for iot voice interaction," in 2022 IEEE Globecom Workshops (GC Wkshps), 2022: IEEE, pp. 922-926.

[22] S. Das, A. Tariq, T. Santos, S. S. Kantareddy, and I. Banerjee, "Recurrent neural networks (RNNs): architectures, training tricks, and introduction to influential research," Machine Learning for Brain Disorders, pp. 117-138, 2023.

[23] 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), 2021: IEEE, pp. 342-347.

[24] S. Cong and Y. Zhou, "A review of convolutional neural network architectures and their optimizations," Artificial Intelligence Review, vol. 56, no. 3, pp. 1905-1969, 2023.

[25] R. Patel and S. Patel, "Deep learning for natural language processing," in Information and Communication Technology for Competitive Strategies (ICTCS 2020) Intelligent Strategies for ICT, 2021: Springer, pp. 523-533.

[26] Y. Hui, J. Wang, N. Cheng, F. Yu, T. Wu, and J. Xiao, "Joint intent detection and slot filling based on continual learning model," in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021: IEEE, pp. 7643-7647.

[27] S. Yu, D. Liu, W. Zhu, Y. Zhang, and S. Zhao, "Attention-based LSTM, GRU and CNN for short text classification," Journal of Intelligent & Fuzzy Systems, vol. 39, no. 1, pp. 333-340, 2020.

[28] M. Jeong and G. G. Lee, "Triangular-chain conditional random fields," IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 7, pp. 1287-1302, 2008.

[29] D. Yu, S. Wang, and L. Deng, "Sequential labeling using deep-structured conditional random fields," IEEE Journal of Selected Topics in Signal Processing, vol. 4, no. 6, pp. 965-973, 2010.

[30] L. Zhang and M. Yang, "Asynchronous joint extraction algorithm based on intent-slot attention mechanism," in International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024), 2024, vol. 13180: SPIE, pp. 824-831.

[31] H. Huang, X. Feng, and Z. Wan, "Joint Model of Intent Recognition and Slot Filling Based on Graph Neural Network fusion of external knowledge base," in 2024 36th Chinese Control and Decision Conference (CCDC), 2024: IEEE, pp. 323-329.

[32] J. Huang and H. Tang, "A Joint Model of Multiple Intent Recognition and Slot Filling Based on Graph Neural Network," 2024.

[33] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013. [Online]. Available: https://arxiv.org/abs/1301.3781.

[34] M. Firdaus, S. Bhatnagar, A. Ekbal, and P. Bhattacharyya, "A deep learning based multi-task ensemble model for intent detection and slot filling in spoken language understanding," in Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13-16, 2018, Proceedings, Part IV 25, 2018: Springer, pp. 647-658, doi: https://doi.org/10.1007/978-3-030-04212-7_57.

[35] W. Chao, Y. Ke, and W. Xiaofei, "POS Scaling Attention Model for Joint Slot Filling and Intent Classification," in 2020 IEEE 20th International Conference on Communication Technology (ICCT), 2020: IEEE, pp. 1483-1487, doi: 10.1109/ICCT50939.2020.9295901.

[36] C. Wang, Z. Huang, and M. Hu, "SASGBC: Improving sequence labeling performance for joint learning of slot filling and intent detection," in Proceedings of 2020 the 6th International Conference on Computing and Data Engineering, 2020, pp. 29-33, doi: https://doi.org/10.1109/CCDC62350.2024.10587455.