

# A joint learning classification for intent detection and slot filling with domain-adapted embeddings

Yusuf Idris Muhammad, Naomie Salim, Anazida Zainal

Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, Johor, Malaysia

## Article Info

### Article history:

Received Jun 22, 2024

Revised Sep 12, 2024

Accepted Sep 30, 2024

### Keywords:

Dialogue system

Intent detection

Joint learning

Natural language understanding

Slot filling

## ABSTRACT

For dialogue systems to function effectively, accurate natural language understanding is vital, relying on precise intent recognition and slot filling to ensure smooth and meaningful interactions. Previous studies have primarily focused on addressing each subtask individually. However, it has been discovered that these subtasks are interconnected and achieving better results requires solving them together. One drawback of the joint learning model is its inability to apply learned patterns to unseen data, which stems from a lack of large, annotated data. Recent approaches have shown that using pretrained embeddings for effective text representation can help address the issue of generalization. However, pretrained embeddings are merely trained on corpus that typically consist of commonly discussed matters, which might not necessarily contain domain specific vocabularies for the task at hand. To address this issue, the paper presents a joint model for intent detection and slot filling, harnessing pretrained embeddings and domain specific embeddings using canonical correlation analysis to enhance the model performance. The proposed model consists of convolutional neural network along with bidirectional long short-term memory (BiLSTM) for efficient joint learning classification. The results of the experiment show that the proposed model performs better than the baseline models.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Yusuf Idris Muhammad

Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia

Johor, Malaysia

Email: muhammadidris@graduate.utm.my

## 1. INTRODUCTION

Natural language understanding (NLU) plays a crucial role in enabling interactions between humans and dialogue systems [1]. Its primary goal is to interpret the meaning of a user's utterances and determine their intentions. NLU comprises two main tasks: intent detection (ID) and slot filling (SF). Intent detection is a classification task where features are extracted from an utterance and fed into a classification algorithm to predict the appropriate intent from a set of predefined classes. In contrast, slot filling is a sequence labeling task that identifies the semantic elements within an utterance, assigning a label to each word to represent the semantic information it carries, which further aids in intent identification. In previous studies, these tasks were traditionally handled separately. However, it is been acknowledged that they are interconnected, and addressing them jointly results in improved performance [2]. This simultaneous approach effectively handles both tasks, capturing intent and slot label distributions within an utterance while considering local and global contexts [3]. Unlike separate models, the joint model minimizes the error accumulation by employing a unified framework for both fine-tuning and training [4]. Training a joint model, however, necessitates a comprehensive annotated dataset that represents various user queries. Unfortunately, building this type of dataset is often costly, time-intensive, and requires considerable manual effort. To address this issue, transfer learning using pretrained word

embeddings were leveraged. Generic pretrained word embeddings like Glove [5], FastText [6] and Word2Vec [7], and which are trained on large corpus have shown a significant success as features for supervised learning in various tasks such as sentiment classification, slot filling, and intent detection, due to their ability to capture rich semantic information within the word vectors.

Many studies incorporate pretrained embeddings to reduce the reliance on large datasets and enhance model generalization. Some approaches employ pretrained GloVe embeddings to provide semantic information through different architectures [8]-[11]. Wang *et al.* [12] utilized convolutional neural network (CNN) to extract features from generic pretrained Word2Vec word embeddings, which were further encoded using bidirectional long short-term memory (BiLSTM) with attention mechanism. To capture richer semantic information, other studies combined Glove and Word2Vec embeddings [13], [14]. Kim *et al.* [15], augmented generic embeddings with to introduce additional semantic insights. However, many natural language processing (NLP) tasks involve specialized vocabularies and are constrained by relatively small datasets. In such cases, the performance of generic pretrained embeddings is limited because the embeddings obtained from general corpora may not capture domain specific semantics. Additionally, embeddings learned from small datasets tend to be low quality.

This paper presents a joint model for intent detection and slot filling with domain adapted word embeddings that capture both domain-specific semantics and generic semantics. The new joint model combines generic embeddings and domain-specific embeddings using canonical correlation analysis (CCA) [16] to align and project the new embeddings along directions of maximum correlation. The evaluation of the proposed method utilizes a CNN-BiLSTM architecture. To our knowledge, this approach has not yet been explored within the context of joint classification for intent detection and slot filling.

The structure of the paper is organized as follows: section 2 outlines the proposed model design for the joint learning applied to intent detection and slot filling. In section 3, the study's methodology is detailed. Section 4 presents the results and provides an analysis, while section 5 concludes the paper.

## 2. PROPOSED MODEL DESIGN

Figure 1 shows the architecture of the proposed CNN-BiLSTM joint learning model with domain-adapted embeddings. This model integrates CNN, as proposed by [17], with each component contributing to the enhancement of the model's overall effectiveness. This section provides an overview of the model structure.

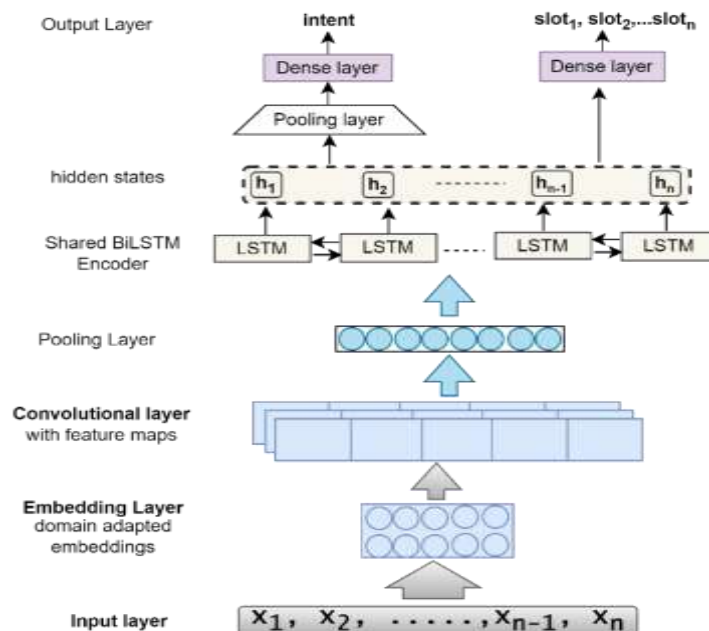


Figure 1. Proposed CNN-BiLSTM architecture with domain-adapted embeddings

## 2.1. Embedding layer

At this layer, the input sequence is converted into a matrix of representations. This conversion uses embedding techniques to translate each word into its corresponding vector form, which is then utilized by the model for further processing. Thus, in the proposed work, a domain-adapted embeddings was obtained by combining both generic pretrained embeddings and domain-specific embeddings. A CCA was used to project word vectors along the direction of maximum correlation. To create the generic embeddings, a pretrained word embedding derived from a corpus of 100 billion words from Google's dataset was employed [7]. Meanwhile, the domain-specific embeddings were produced using the continuous bag of words (CBOW) approach from the Word2Vec model, trained on the SNIPS and air travel information system (ATIS) datasets. Each word in the model is represented by a vector with 300 dimensions. To ensure sentence length uniformity, word padding is employed to set the length equal to the maximum sequence in our datasets.

Let  $W_{DE} \in \mathbb{R}^{|V_{DE}| \times d}$  and  $W_{GE} \in \mathbb{R}^{|V_{GE}| \times d}$  represent a matrices of domain-specific and generic pretrained embeddings respectively, where  $V_{DE}$  and  $V_{GE}$  are the vocabularies for domain-specific and generic pretrained embeddings respectively and  $d$  is the dimension. Let  $V_{DA} = V_{GE} \cap V_{DE}$  and  $w_{i,DE}$  and  $w_{i,GE}$  be the embedding of word  $i$  in domain specific and generic embeddings respectively. For 1-dimensional CCA, let  $\phi_{DE}$  and  $\phi_{GE}$  be the projection directions of  $w_{i,DE}$  and  $w_{i,GE}$  respectively. The projected values can be given as:

$$\bar{w}_{i,DE} = w_{i,DE} \phi_{DE} \quad (1)$$

$$\bar{w}_{i,GE} = w_{i,GE} \phi_{GE} \quad (2)$$

CCA maximizes the correlation between  $\bar{w}_{i,DE}$  and  $\bar{w}_{i,GE}$  to obtain  $\phi_{DE}$  and  $\phi_{GE}$  such that,

$$\rho(\phi_{DE}, \phi_{GE}) = \max_{\phi_{DE}, \phi_{GE}} \frac{\mathbb{E}[\langle \bar{w}_{i,DE}, \bar{w}_{i,GE} \rangle]}{\sqrt{\mathbb{E}[\bar{w}_{i,DE}^2]} \sqrt{\mathbb{E}[\bar{w}_{i,GE}^2]}} \quad (3)$$

where  $\rho$  is the correlation between the embeddings and  $\mathbb{E}$  is the expectation over all words  $i \in V_{DA}$ . The domain adapted embeddings for word  $i$  is given by,

$$\hat{w}_{i,DA} = \alpha \bar{w}_{i,DE} + \beta \bar{w}_{i,GE} \quad (4)$$

where the parameters  $\alpha$  and  $\beta$  are obtained by solving the following optimization,

$$\min_{\alpha, \beta} \|\bar{w}_{i,DE} - (\alpha \bar{w}_{i,DE} + \beta \bar{w}_{i,GE})\|_2^2 + \|\bar{w}_{i,GE} - (\alpha \bar{w}_{i,DE} + \beta \bar{w}_{i,GE})\|_2^2 \quad (5)$$

In (5) gives a weighted combination with  $\alpha = \beta = \frac{1}{2}$ , and the domain adopted vector is equal to the average of the two projections:

$$\hat{w}_{i,DA} = \frac{1}{2} \bar{w}_{i,DE} + \frac{1}{2} \bar{w}_{i,GE} \quad (6)$$

## 2.2. Convolution layer

Features were extracted from the embedding layer by performing convolution on the encoded input sentences (vectors) to generate local features. A zero-padding token was added prior to the convolution to maintain consistent output sizes across various filters. To extract the local features of the domain adapted features  $\hat{w}_{i,DA}$  different filters and kernel sizes were applied. Conventionally, the convolution operation involves computing the inner product between filter weights  $W \in R^{h \times ws}$  and a local region of the input data, followed by an activation function, where  $h$  denotes the embedding dimension for each word, and  $ws$  specifies the window size, which controls the number of adjacent words considered. Let  $w_{DA} \in R^{h \times ws}$  be filters employed on window size  $ws$  for the domain adapted embeddings. The generated features can be expressed as (7).

$$Z_i = f(w \cdot x_{i:i+ws-1} + b) \quad (7)$$

Where  $(\cdot)$  is the convolution operator,  $f$  is a non-linear activation function such as  $\tanh$  or  $ReLU$ ,  $b$  is the bias for weight  $w$ . This function is applied to each window and is denoted by  $[x_{1:ws}, x_{2:ws+1}, \dots, x_{n-ws:n}]$ . In this

study, employed a rectified linear unit (ReLU) activation function was employed because of its efficiency compared to tanh [18]. Where  $w_{da} \in R^{n-ws+1}$ . The feature map generated for the domain adapted embedding matrix can be expressed as (8).

$$z_{DA} = [z_1^{DA}, z_2^{DA}, \dots, z_{n-ws+1}^{DA}] \tag{8}$$

**2.3. Maxpooling**

The maxpooling layer is employed to reduce the size of the feature maps produced by the convolutional layers, helping to lower computational demands and mitigate overfitting. It helps to achieve translation invariance by preserving the important features, regardless of their exact spatial location in the data. Additionally, it aids in extracting higher-level representations by consolidating information from neighboring tokens or words. There are different types of pooling operations, with maxpooling being the most common. Maxpooling operates by selecting the highest value from each localized segment of the feature map, thereby emphasizing the most prominent features [18]. In general, the maxpooling layer selects the maximum value from the feature map generated by the convolutional layer. Specifically, this operation can be expressed as (9).

$$\tilde{Z}_{DA} = \max[z_1^{DA}, z_2^{DA}, \dots, z_{n-ws+1}^{DA}] \tag{9}$$

After applying max pooling, the final output was derived by concatenating the extracted features, as represented by (10).

$$Z = \tilde{Z}_{AD}^1 \oplus \dots \oplus \tilde{Z}_{AD}^m \tag{10}$$

Where  $\oplus$  is the concatenator operator and m is the filter for the features.

**2.4. Shared BiLSTM layer**

BiLSTMs enhance traditional recurrent neural networks (RNNs) by effectively capturing long-term dependencies in sequential data. Unlike basic LSTM models, BiLSTMs incorporate both forward and backward LSTM components, allowing them to obtain a richer understanding of contextual information [19]. BiLSTM networks utilize two distinct LSTM structures to analyze sequential data. One LSTM processes the sequence from the beginning, and the other LSTM processes it from the end. The results from these two networks are combined at each point in the sequence. This configuration allows the model to integrate context from both earlier and later parts of the sequence, enhancing its ability to capture long-term dependencies. The output Z from the convolutional layer was combined and then fed into the BiLSTM layer to produce the final output  $h_t$ .

$$\vec{h}_t = LSTM(z_t, \vec{h}_{t-1}) \tag{11}$$

$$\overleftarrow{h}_t = LSTM(z_t, \overleftarrow{h}_{t-1}) \tag{12}$$

$$h_t = \vec{W}_t \cdot \vec{h}_t + \overleftarrow{W}_t \cdot \overleftarrow{h}_t + b \tag{13}$$

The output of the BiLSTM will be denoted as (14).

$$H = [h_1, h_2, \dots, h_n] \tag{14}$$

**2.5. Output layer**

The model generates distinct outputs for intent detection and slot filling. For intent detection, a maxpooling layer is applied to extract the most important representation of the whole sequence from the BiLSTM, similar to the method employed in [20]. In this research, the output from the maxpooling layer was additionally forwarded to a global max pooling layer, producing a summary of the entire sequence. Following this, a fully connected layer with Softmax activation was utilized to identify the intent. Consequently, the intent output vector is determined using (15).

$$h_{maxpool} = \{\max(h_1, h_2), \dots, \max(h_{n-1}, h_n)\} \tag{15}$$

In the global maxpooling layer, the highest value from the entire the entire sequence  $h$ , as illustrated in (16) is used to obtain the intent label  $y^i$  as in (17), where  $b$  is a bais vector and  $W$  is the transformation matrix.

$$h_{global} = \max(h_1, h_2, \dots, h_n) \quad (16)$$

$$y^i = \text{Softmax}(W \cdot h_{global} + b) \quad (17)$$

For the slot filling output, the BiLSTM's output is fed into a fully connected layer with Softmax activation. The resulting output vector is computed as in (18). Where  $b, W$  are the bias vector and transformation matrix respectively,  $y^s$  is the slot label.

$$y_i^s = \text{softmax}(W \cdot h_i + b) \quad (18)$$

### 3. METHOD

This section offers a summary of the datasets used in the experiments, followed by the experimental methodology used to assess the effectiveness of the proposed approach. Additionally, a comparative analysis of the baseline methods is included. The model's performance was evaluated using accuracy for intent detection and the F1-score for slot filling.

#### 3.1. Dataset

To verify the proposed model, experiments were carried out with two of the most commonly utilized datasets in NLU research: SNIPS [21] and ATIS [22]. SNIPS is a dataset that contains 15,884 utterances, 7 intents, and 72 slots. It covers various domains such as weather, restaurants, and entertainment. The ATIS dataset consists 4,978 training samples, 893 test samples, 21 intents, and 128 slots [23]. The dataset statistics are summarized in Table 1.

Table 1. Statistics of SNIPS and ATIS data sets

| Dataset | Size   | Intent | Slot | Training data | Test data | Validation data |
|---------|--------|--------|------|---------------|-----------|-----------------|
| SNIPS   | 15,884 | 7      | 72   | 13,084        | 700       | 700             |
| ATIS    | 4,978  | 21     | 128  | 4,478         | 893       | 500             |

An example of a semantic frame for an utterance from the SNIPS dataset, “find fish story,” is shown in Table 2. The intent for this utterance is labeled as “SearchScreeningEvent,” and the slots are annotated using the IOB (in-out-begin) tagging scheme. In this scheme, “O” indicates that the word does not belong to any named entity, “B-movie\_name” marks the beginning of a named entity, and “I-movie\_name” represents a word that continues within the same entity

Table 2. Illustration of semantic frame and IOB tagging for SNIPS dataset

| Entity | slots        | Intent               |
|--------|--------------|----------------------|
| Find   | O            |                      |
| Fish   | B-movie_name | SearchScreeningEvent |
| Story  | I-movie_name |                      |

#### 3.2. Experimental set-up

A grid search was performed to fine-tune the hyperparameters of our model. Three filter sizes (2, 3, and 5) were tested, along with 128 feature maps. To address overfitting, a dropout rate of 0.5 was applied to the feature maps. The shared encoder, which included 200 hidden units and used the ReLU activation function, was followed by an additional dropout rate of 0.5 to randomly drop some units. For the intent detection and slot filling tasks, L2 regularization with a coefficient of 0.001 was applied to the dense layer weights, which were activated by the SoftMax function. The model was trained using the Adam optimizer and categorical cross-entropy loss, with performance evaluated through accuracy for intent detection and the F1-score for slot filling. Training was conducted with a batch size of 32.

#### 3.3. Comparative methods

The effectiveness of the proposed model was evaluated by comparing it with the following baseline models:

- RNN-LSTM [24]: this model uses a BiRNN with LSTM to perform a joint learning classification using lexical features represented by 1-hot encoding.

- BiRNN-attention [25]: this model employs an encoder-decoder architecture with an attention mechanism with random embedding initialization.
- CNN-BiLSTM [12]: employs CNN to extract features from generic pretrained Word2Vec word embeddings and the BiLSTM layer as an encoder and uses an attention-based RNN as decoder.
- BiGRU/BiLSTM-MLP [14]: a multi-task ensemble method with features obtained from glove, Word2Vec, and part-of-speech (POS) tags.
- BiLSTM+attention [26]: a joint model with POS scaling attention to help the model focus on verbs and nouns that are important in representing user behavior and object operations, respectively.
- SASGBC(BERT only) [27]: this model uses bidirectional encoder representations from transformers (BERT) to encode the input sequence, integrate intent information with slot gates, and establish a contextual semantic relationship with self-attention.
- BiLSTM+attention [28]: this model exploits the pretrained BERT model together with BiLSTM and co-interactive attention, and initializes the embedding layer with glove embeddings.

It should be noted that following the common practice in the literature [14], [29], this study directly use the findings of the aforementioned baseline methods presented in their original publications and compares them with our proposed model.

#### 4. RESULTS AND DISCUSSION

Table 3 shows the performance of the proposed model across various settings. Model-1, which used randomly initialized embeddings, performed worse than the other models that used pretrained embeddings. This demonstrates the advantage of pretrained word embeddings over randomly initialized ones in the joint learning classification of intent detection and slot filling, consistent with previous research findings [9], [11], [30]. Pretrained embeddings, enriched with semantic information from large corpora, provide a stronger foundation for model training, leading to more accurate predictions.

The performance difference between model-2, which used generic embeddings, and model-3, which utilized domain-specific embeddings, further illustrates the importance of embeddings relevance to the task at hand. Model-3 outperformed model-2, likely due to the richer, domain-specific vocabulary present in the embeddings, which better captured the nuances of the dataset. This finding indicates that embeddings trained on domain-specific corpora can greatly improve the model’s capacity to comprehend and handle domain-specific language, leading to better performance.

Model-4, which combined generic and domain-specific embeddings through a simple concatenation operation, outperformed both model-2 and model-3. This improvement can be attributed to the hybrid representation that leverages the strengths of both embedding types, generic embeddings provide broad coverage of language, while domain-specific embeddings offer depth in the relevant domain. This combination allows the model to benefit from both general linguistic patterns and specific domain knowledge, leading to better overall performance.

In Figure 2, it is evident that both model-2 and model-3 exhibited higher accuracy on the SNIPS dataset compared to the ATIS dataset, despite the use of pretrained embeddings. This discrepancy can be attributed to the ATIS dataset's smaller size and its significant class imbalance [1]. Such imbalance poses challenges for the model in learning to distinguish between less frequent classes, which likely contributed to the reduced accuracy.

Model-5, a variant of model-4, outperformed all other models by aligning generic and domain-specific embeddings using CCA. This method allows for better integration of the two types of embeddings, capturing the underlying structure of the data more effectively. The superior performance of model-5 aligns with previous studies demonstrating that the combination of different embeddings can lead to substantial improvements in model accuracy [14], [31].

Table 3. Performance of different versions of the proposed model

| Model   | Embeddings   | ATIS         |              | SNIPS        |              |
|---------|--|--------------|--------------|--------------|--------------|
|         |  | Accuracy (%) | F1-score (%) | Accuracy (%) | F1-score (%) |
| Model-1 | Random   | 79.00        | 96.04        | 86.79        | 94.01        |
| Model-2 | Generic word2vec                                     | 81.34        | 95.54        | 97.30        | 98.28        |
| Model-3 | Domain-specific word2vec                             | 88.76        | 96.88        | 97.99        | 98.24        |
| Model-4 | Generic word2vec +<br>Domain-specific word2vec       | 93.88        | 97.54        | 97.79        | 98.48        |
| Model-5 | CCA (Generic word2vec +<br>Domain-specific word2vec) | 97.87        | 98.16        | 98.10        | 98.43        |

Figure 3 further demonstrates the efficacy of the proposed model variants, especially in the slot filling task for both datasets. The high F1-scores, especially those achieved by model-5, highlight the superiority of non-contextual embeddings such as Word2Vec, GloVe, and FastText for tasks requiring fine-grained word-level representations. This finding highlights that natural language processing models can be significantly enhanced by word-level representations, especially in entity recognition and classification tasks [32]-[34].

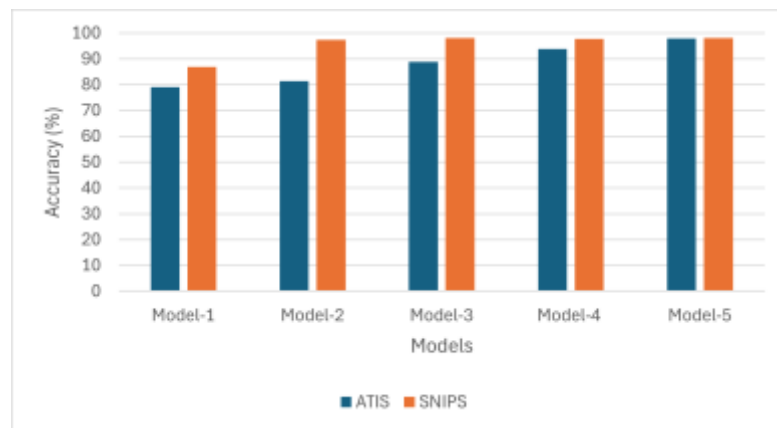


Figure 2. Graphical representation of the performances of the different versions of the proposed model based on accuracy

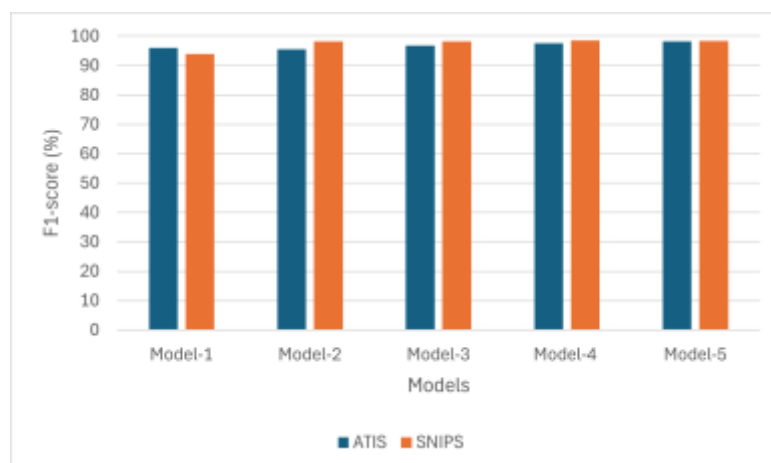


Figure 3. Graphical representation of the performances of the different versions of the proposed model based on F1-score

#### 4.1. Comparison with baseline models

Table 4 shows that utilizing domain-adapted embeddings outperformed several baseline models across both the ATIS and SNIPS datasets. Compared to the RNN-LSTM model with classical one-hot encoding, the proposed model achieved a significant gain of 3.27% in accuracy and 8.76% in F1-score on the ATIS dataset. These results highlight the superiority of utilizing word embeddings over traditional methods like one-hot encoding.

Additionally, the proposed model outperformed the BiRNN-attention model, which used random embeddings, with a notable gain of 3.47% in accuracy and 2.38% in F1-score. The improvement here can be attributed to the use of semantically meaningful embeddings, as opposed to randomly initialized ones. Even when compared to the CNN-BiLSTM model, which shares a similar architecture but uses generic Word2Vec embeddings, the proposed showed improved on the ATIS dataset, with gains of 0.7% in accuracy and 0.4% in F1-score. This improvement is probably attributed to the domain-specific characteristics of the embeddings

This finding demonstrates that leveraging domain-adapted embeddings, particularly when aligned with generic embeddings through CCA, significantly boost the performance of intent detection and slot filling tasks by capturing both broad linguistic patterns and domain-specific nuances. The result is a marked improvement in performance for these tasks. This approach offers a straightforward yet powerful method for enhancing model performance. It stands out for its efficiency, as it avoids the need for extensive feature engineering or overly complex architectures. By focusing on optimizing embedding quality, this method provides a practical solution for improving natural language processing models.

Given its effectiveness, the proposed method presents a promising avenue for future research in NLP. Further exploration and refinement of domain-adapted embeddings, especially in combination with other techniques, could lead to even greater advancements. This approach holds potential to expand the capabilities of NLU tasks.

The findings of this study align with prior research emphasizing the benefits of pretrained embeddings and their domain-specific adaptations in natural language understanding tasks. The superior performance of the model, particularly when utilizing domain-adapted embeddings obtained through CCA, reinforces the idea that embedding relevance and alignment play roles in enhancing model accuracy. This align with previous studies that have demonstrated improvements in various NLP tasks through the careful selection and adaptation of embeddings [9], [11], [30], [14], [31]-[34].

While the proposed model has shown strong performance on the ATIS and SNIPS datasets, further research is needed to validate its generalizability across more challenging and diverse NLU datasets. Exploring different architectures and combining additional types of embeddings, such as contextual embeddings with domain-adapted embeddings, could yield further improvements. Additionally, investigating the impact of the proposed approach on other NLP tasks, such as entity recognition or sentiment analysis, would be valuable.

Table 4. Evaluation of published metrics on SNIPS and ATIS datasets

| Models                      | Features                            | ATIS         |              | SNIPS        |              |
|-----------------------------|-------------------------------------|--------------|--------------|--------------|--------------|
|                             |                                     | Accuracy (%) | F1_score (%) | Accuracy (%) | F1_score (%) |
| RNN-LSTM [24]               | Lexical features (one-hot encoding) | 94.6         | 89.40        | -            | -            |
| BiRNN+attention [25]        | Random embeddings                   | 94.4         | 95.78        | -            | -            |
| CNN-BiLSTM [12]             | Word embeddings(word2vec)           | 97.17        | 97.76        | -            | -            |
| Bi-GRU+feature [14]         | Glove + Word2vec + POS              | 97.76        | 97.93        | -            | -            |
| BiLSTM+attention [26]       | POS                                 | 95.70        | 95.60        | 97.70        | 89.2         |
| BC [27]                     | BERT embeddings                     | 97.20        | 96.34        | 98.0         | 95.68        |
| <b>CNN-BiLSTM (model-5)</b> | <b>Domain adapted embeddings</b>    | <b>97.87</b> | <b>98.16</b> | <b>98.10</b> | <b>98.43</b> |

#### 4.2. Comparison of separate and joint model of the proposed model

The experimental results comparing the joint model with the separate model of the proposed work are presented in Table 5. It can be observed that in both tasks across all datasets, the joint model surpasses the separate models, showcasing the effectiveness of joint training. The joint model’s loss function combines the loss functions of both tasks, allowing it to learn the tasks’ interdependence and enhance performance through shared representation. Due to the information sharing across both tasks, the joint model outperforms the separate model. Furthermore, the joint model balances the loss function of both tasks and shares parameters, which accelerates the training process.

Table 5. Comparison of separate and joint model of the proposed work

| Models                | ATIS         |              | SNIPS        |              |
|-----------------------|--------------|--------------|--------------|--------------|
|                       | Accuracy (%) | F1-score (%) | Accuracy (%) | F1-score (%) |
| Intent detection only | 94.38        | -            | 97.72        | -            |
| Slot filling only     | -            | 98.0         | -            | 98.37        |
| Joint model           | <b>97.87</b> | <b>98.16</b> | <b>98.10</b> | <b>98.43</b> |

#### 4.3. Filter size settings

In the convolutional layer, filters extract features similar to n-gram features. The different sizes of these filters lead to different mappings of local features, which significantly impact the results of the convolution. Using multiple filter sizes generally improves performance compared to using a single filter size. Based on the findings in Table 6, a filter size of [2,3,5] outperformed other setups in the experiments.



This emphasizes the crucial role of tri-gram features in capturing local features. It is hypothesized that sequences of tri-gram features can enable BiLSTM to achieve a more effective semantic representation of the input utterances.

Table 6. Performance based on convolutional filters

| Filter size | ATIS         |              | SNIPS        |              |
|-------------|--------------|--------------|--------------|--------------|
|             | Accuracy (%) | F1-score (%) | Accuracy (%) | F1-score (%) |
| 2           | 92.27        | 84.55        | 93.37        | 97.63        |
| 3           | 89.96        | 97.47        | 97.79        | 98.27        |
| 5           | 87.86        | 97.46        | 98.03        | <b>98.49</b> |
| [2,3,5]     | <b>97.87</b> | <b>98.16</b> | <b>98.10</b> | 98.43        |

## 5. CONCLUSIONS AND FUTURE WORK

This study has demonstrated the effectiveness of using domain-adapted embeddings, particularly when aligned with generic embeddings through CCA, in enhancing the performance of deep learning models for intent detection and slot filling. By combining both generic and domain-specific features, the proposed approach has shown superior accuracy, and F1-scores compared to baseline models across the SNIPS and ATIS datasets. The joint learning model, supported by these enriched embeddings, effectively captures the interdependence of intent detection and slot filling tasks, leading to significant performance gains. The experimental results emphasize the critical role of embedding relevance and alignment in improving model accuracy, as well as the benefits of joint training over separate models. The findings align with prior research, reinforcing the importance of embedding quality and task interdependence in natural language understanding tasks. The use of multiple convolutional filter sizes in the model further contributed to its success by capturing essential n-gram features, particularly trigrams, which are crucial for effective semantic representation. Given the promising results, future research should focus on further validating this approach on more challenging and diverse NLU datasets. Additionally, exploring the integration of other embedding types, such as contextual embeddings, and applying this methodology to other NLP tasks, like entity recognition or sentiment analysis, could lead to even greater advancements in the field. The efficiency of the proposed method make it a valuable direction for ongoing research in NLP.

## ACKNOWLEDGEMENTS

Authors thanks the Ministry of Higher Education Malaysia for partially funding this research under Fundamental Research Grant scheme (FRGS/1/2022/ICT06/UTM/01/1) with grant vote No. R.J130000.7851.5F568. Furthermore, appreciation is extended to Universiti Teknologi Malaysia (UTM) for providing the resources used in this research work.





## REFERENCES

- [1] 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*, vol. 55, no. 8, 2022, doi: 10.1145/3547138.
- [2] S. M. Suhaili, N. Salim, and M. N. Jambli, "Service chatbots: a systematic review," *Expert Systems with Applications*, vol. 184, p. 115461, 2021, doi: 10.1016/j.eswa.2021.115461.
- [3] S. Louvan and B. Magnini, "Recent neural methods on slot filling and intent classification for task-oriented dialogue systems: a survey," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 480-496, doi: 10.48550/arXiv.2011.00564.
- [4] M. Firdaus, A. Ekbal, and E. Cambria, "Multitask learning for multilingual intent detection and slot filling in dialogue systems," *Information Fusion*, vol. 91, pp. 299-315, 2023, doi: 10.1016/j.inffus.2022.09.029
- [5] J. Pennington, R. Socher, and C. D. Manning, "Glove: global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543, doi: 10.3115/v1/D14-1162.
- [6] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135-146, 2017, doi: 10.1162/tacl\_a\_00051.
- [7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013, doi: 10.48550/arXiv.1301.3781.
- [8] S. Dadas, J. Protasiewicz, and W. Pedrycz, "A deep learning model with data enrichment for intent detection and slot filling," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019, pp. 3012-3018, doi: 10.1109/SMC.2019.8914542.
- [9] Q. N. T. Do and J. Gaspers, "Cross-lingual transfer learning for spoken language understanding," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5956-5960, doi: 10.18653/v1/D19-1153.
- [10] E. Okur, S. H. Kumar, S. Sahay, A. Arslan Esmé, and L. Nachman, "Natural language interactions in autonomous vehicles: Intent detection and slot filling from passenger utterances," *International Conference on Computational Linguistics and Intelligent Text Processing*, 2019: Springer, pp. 334-350, doi: 10.1007/978-3-031-24340-0\_25.
- [11] A. Bhasin, B. Natarajan, G. Mathur, and H. Mangla, "Parallel intent and slot prediction using mlb fusion," in *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, 2020, pp. 217-220, doi: 10.1109/ICSC.2020.00045.





- [12] Y. Wang, L. Tang, and T. He, "Attention-based CNN-BLSTM networks for joint intent detection and slot filling," in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 17th China National Conference, CCL 2018, and 6th International Symposium, NLP-NABD 2018*, Changsha, China, 2018, pp. 250-261, doi: 10.1007/978-3-030-01716-3\_21.
- [13] 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: 10.1016/j.knosys.2019.07.017.
- [14] 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: 10.1007/978-3-030-04212-7\_57.
- [15] J. -K. Kim, G. Tur, A. Celikyilmaz, B. Cao and Y. -Y. Wang, "Intent detection using semantically enriched word embeddings," *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, USA, 2016, pp. 414-419, doi: 10.1109/SLT.2016.7846297.
- [16] X. Zhuang, Z. Yang, and D. Cordes, "A technical review of canonical correlation analysis for neuroscience applications," *Human brain mapping*, vol. 41, no. 13, pp. 3807-3833, 2020, doi: 10.1002/hbm.25090.
- [17] 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*, 2024, doi: 10.1093/jigpal/jzae057.
- [18] Y. Kim, "Convolutional neural networks for sentence classification," 2014, doi: 10.48550/arXiv.1408.5882
- [19] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, p. 420, 2021, doi: 10.1007/s42979-021-00815-1.
- [20] X. Zhang and H. Wang, "A joint model of intent determination and slot filling for spoken language understanding," in *IJCAI*, 2016, vol. 16, no. 2016, pp. 2993-2999.
- [21] A. Coucke *et al.*, "SNIPs voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018, doi: 10.48550/arXiv.1805.10190.
- [22] P. Price, "Evaluation of spoken language systems: the ATIS domain," in *HLT '90: Proceedings of the workshop on Speech and Natural Language*, 1990, doi: 10.3115/116580.116612.
- [23] 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: 10.1145/3547138.
- [24] D. Hakkani-Tür *et al.*, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Interspeech*, 2016, pp. 715-719, doi: 10.21437/Interspeech.2016-402.
- [25] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *Proceedings of the Annual Meeting of the Speech Communication Association (INTERSPEECH'16)*, 2016, pp. 685-689, doi: 10.21437/Interspeech.2016-1352.
- [26] W. Chao, Y. Ke and W. Xiaofei, "POS scaling attention model for joint slot filling and intent classification," *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, Nanning, China, 2020, pp. 1483-1487, doi: 10.1109/ICCT50939.2020.9295901.
- [27] 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: 10.1109/CCDC62350.2024.10587455.
- [28] L. Qin, T. Liu, W. Che, B. Kang, S. Zhao, and T. Liu, "A co-interactive transformer for joint slot filling and intent detection," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 8193-8197, doi: 10.1109/ICASSP39728.2021.9414110.
- [29] T. He, X. Xu, Y. Wu, H. Wang, and J. Chen, "Multitask learning with knowledge base for joint intent detection and slot filling," *Applied Sciences*, vol. 11, no. 11, p. 4887, 2021, doi: 10.3390/app11114887.
- [30] 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: 10.1007/s12559-020-09718-4.
- [31] A. Siddhant, A. Goyal, and A. Metallinou, "Unsupervised transfer learning for spoken language understanding in intelligent agents," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, vol. 33, no. 01, pp. 4959-4966, doi: 10.48550/arXiv.1811.05370.
- [32] K. Saifullah, M. I. Khan, S. Jamal, and I. H. Sarker, "Cyberbullying text identification based on deep learning and transformer-based language models," *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, vol. 11, no. 1, pp. e5-e5, 2024, doi: 10.4108/eetinis.v11i1.4703.
- [33] E. Sezerer and S. Tekir, "A survey on neural word embeddings," *arXiv preprint arXiv:2110.01804*, 2021. [Online]. Available: <https://arxiv.org/pdf/2110.01804>.
- [34] S. J. Johnson, M. R. Murty, and I. Navakanth, "A detailed review on word embedding techniques with emphasis on word2vec," *Multimedia Tools and Applications*, vol. 83, no. 13, pp. 37979-38007, 2024, doi: 10.1007/s11042-023-17007-z.

## BIOGRAPHIES OF AUTHORS







**Yusuf Idris Muhammad**     was born in Kano, Nigeria in 1984. He received his B.Eng. Computer Engineering from Bayero University Kano, Nigeria and M.Sc. from Mevlana University Konya, Turkey. He is currently pursuing his Ph.D. degree in computer science with Faculty of Computing, Universiti Teknologi Malaysia. He is also a lecturer at Saadatu Rimi University of Education Kumbotso Kano, Nigeria. His current research area includes NLP, cloud computing, and big data analytics. He can be contacted at email: muhammadidris@graduate.utm.my.



**Prof. Dr. Naomie Salim**     received her B.Sc. degree in computer science from Universiti Teknologi Malaysia, the M.Sc. degree in computer science from Western Michigan University, and Ph.D. degree in information studies from the University of Sheffield. She is currently a professor with Faculty of Computing, Universiti Teknologi Malaysia where she is the Director Big data centre. She has authored more than 100 journal articles and conferences papers since the inception of her research career. Her main research interests include text mining, machine learning, information retrieval, cheminformatics, and NLP. She can be contacted at email: [naomie@utm.edu.my](mailto:naomie@utm.edu.my).



**Assoc. Prof. Dr. Anazida Zainal**     received the B.Sc. degree in computer science from Rutgers University, NJ, USA, in 1990, and M.Sc. degree in computer science and the Ph.D. degree in computer science and the network security from Universiti Teknologi Malaysia. She can be contacted at email: [anazida@utm.my.edu](mailto:anazida@utm.my.edu).