Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

A Slot-Aware Semantic Signaling Framework for Multi-intent Classification

Juhi Shah[1], Priyank Thakkar[2]

Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad, India.

Email: [1]18ftphde25@nirmauni.ac.in, [2]priyank.thakkar@nirmauni.ac.in

Abstract

Multi-Intent Detection (MID) and Slot Filling (SF) are fundamental tasks in Natural Language Understanding (NLU) for goal-oriented dialogue systems. Despite advances in various joint models, which improve performance by learning interactions between intents and slots, existing models might not adequately capture intricate and complex relationships between intents in multi-intent utterances, especially if slot-level semantic information is not adequately utilized during intent detection (ID). To this end, this paper proposes a novel framework, namely Slot-Aware Semantic Signaling for Multi-Intent Classification (S3MIC), which is a joint framework that leverages slot-level semantic information to improve intent prediction in multi-intent dialogue systems. We conduct experiments with the proposed framework on benchmark datasets, namely MixATIS and MixSNIPS, and report consistent improvements over existing benchmark models in terms of Slot F1, Intent Accuracy, and Semantic Frame Accuracy (SeFr Acc). Specifically, our proposed framework achieves 52.5% SeFr Acc on MixATIS and 86.28% SeFr Acc on MixSNIPS, thereby validating the efficacy of slot-aware semantic signaling in improving joint MID and SF performance in goal-oriented dialogue systems.

Keywords: Slot-Aware Semantic Signaling, Multi-Intent Detection, Slot Filling, Joint Learning.

1. Introduction

Goal-oriented conversation systems are based on ID and SF. In accordance with Figure 1, these skills are crucial for interpreting natural language. In order to cater to a vast number of users, it is necessary to understand the input query first before generating any response. ID helps to understand what the user wants and SF helps to understand which entities should be used to process the action. Together, this information helps the dialog manager to select appropriate policies and further serve the natural language generator to produce a meaningful response. For instance in Figure 1, consider the user query ”What will be the weather in Ahmedabad tomorrow at 11 am?”. It is intuitive for humans to understand this and answer the predicted temperature by fetching the data. For a dialogue system to be cognizant, it needs to decide its action based on two things. First, the intention of the user and details given by the user to fulfil the action. In this example the intention of the user is ”Get Weather” and the values needed to fetch the required details are Location and Time.

DOI: https://doi.org/10.36548/jtcsst.2026.1.009
Received: 04.02.2026, received in revised form: 02.03.2026, accepted: 16.03.2026, published: 25.03.2026

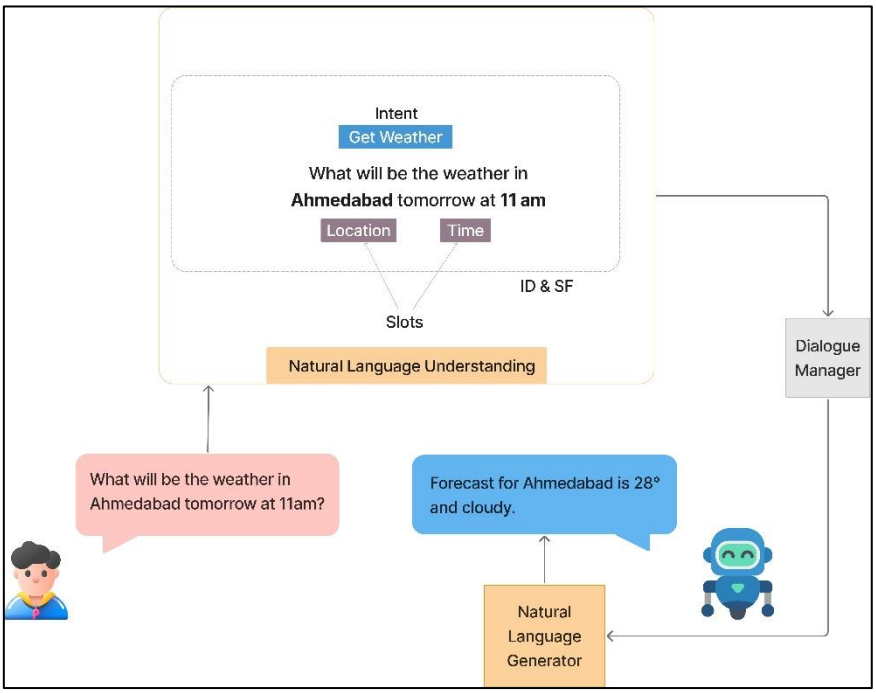


Figure 1. Significance of Intent Recognition and Slot Labeling Tasks in Dialogue Understanding

Early NLU systems assumed that one user utterance corresponds to a single goal or only a single intent. During this phase, ID models were primarily optimized as single-task classifiers to map an utterance to a categorical label, such as ”GetWeather” or ”BookFlight”, and simultaneously extracted entities through SF. As dialogue systems moved into the real-world applications, it became clear that users frequently express multiple intentions within a single turn. This realization forced a paradigm shift toward MID. Unlike the single-intent paradigm, MID requires the system to identify a set of co-occurring intents (e.g., FindRestaurant, BookTable, and SendInvitation) within a single utterance. A single utterance may be connected to multiple intents simultaneously. For instance, the request ”Find a table for two at 8 PM and invite Kavya,” consists of two intents: the BookTable intent and the SendInvitation intent. However, the ”8 PM” slot belongs to the BookTable intent. Ultimately, transitioning to a multi-intent framework enables a system to evolve from machine-like command processing to understanding conversation more naturally. Without the ability to parse multi-part instructions, goal-oriented systems remain fragile and fail to meet the expectations of modern human-computer interaction.

ID involves classifying an utterance at the sentence level to determine its overall purpose. Whereas SF operates at the word level, identifying and labeling specific tokens to extract meaningful information. Researchers in NLP tackle the ID and SF complexities using a variety of approaches, from the traditional rule-based systems depending on pre-defined patterns to complex machine learning algorithms. Each of these approaches has its own strengths and weaknesses. Recent joint models and slot–intent mapping mechanisms have improved the interaction between ID and SF. However, many existing architectures rely primarily on explicit slot–intent associations and do not fully utilize the contextual slot representations produced by the encoder. As a result, the semantic information captured in slot representations may not be effectively propagated to the sequence-level intent classifier. This can limit the model’s ability to leverage entity-level information during intent prediction. S3MIC addresses this by using slot propagation. This research introduces the following noteworthy contributions:

1. For the MID challenge, a novel hybrid architecture (S3MIC) is proposed, which leverages the token level slot representation. Our research is the first to examine the integration of slot-aware semantic signaling with sequence-level intent prediction.

2. The proposed model is comprehensively tested on two publicly available corpora, and the experimental findings highlight that the method has good generalization ability for multi-intent accuracy and semantic frame accuracy, which effectively improves multi-intent detection.

This paper begins with a brief summary of the research presented in Section 2. In Section 3, we further analyze the distinct characteristics of the datasets utilized. Section 4 & 4.3 explain the proposed approach in detail with considerations of the training environment. In Section 5.1, a comparison of results is performed on two datasets. We summarize the findings of this study in Section 6.

2. Related Work

Early studies conducted by Haffner et al. [12], Raymond and Riccardi [13], notably modeled ID and SF individually as two separate problems. Most neural models of SF and ID contain more than one layer: the input layer, the latent representation layer, and an inference Layer. The specifics of these layers largely differentiate between models. The input layer of these neural models is used to map tokens in a sequence to word embeddings [25]. Various studies have compared different types of embeddings, such as pre-trained SENNA embeddings [26], RNN Language Model (RNNLM) embeddings [28], and random embeddings. SENNA embeddings usually yield a better result, and fine-tuning word embeddings in general improves performance. However, task-specific embeddings learned directly from ATIS data that include both words and named entities, along with syntactic features, have outperformed SENNA embeddings [27]. Ravuri and Stolcke [29] emphasized character representation to address out-of-vocabulary (OOV) issues. A variety of RNN-based models [30], [31] have been explored for encoding both SF and IC tasks separately. Long Short-Term Memory Network (LSTM) encoders generally outperform Jordan [33] and Elman [32] encoders [25]. Additionally, they experimented with a bi-directional version of the Jordan RNN, achieving a noteworthy SF score of 93.89 F1, outperforming the conventional CRF method by approximately 1 absolute F1 improvement. Meanwhile, Xu and Sarikaya [34] chose for a CNN-based method by leveraging it to capture 5-gram features. This method uses max-pooling to extract representations of words before feeding them into the output layer. Unlike RNNs, CNNs achieved inferior results for the task of SF on the ATIS dataset. Other researchers [36], [37] chose to adapt LSTMs for SF, resulting in a superior performance compared to the traditional methods like CNN, RNN, and CRF. Researchers in [29] delved into the comparison between vanilla RNNs and LSTMs for IC. They found that vanilla RNNs performed better for shorter utterances, while LSTMs excelled with longer ones. A popular method for the output layer is to use a softmax function to predict at particular time steps. The R-CRF model, first presented by Yao et al. [35] , successfully blends the sequence-level optimization of CRFs with the feature learning powers of RNNs. On the ATIS and Bing query understanding datasets, the hybrid model RNN and CRF outperformed both CRF and vanilla RNNs.

Because ID and SF models were trained separately, they did not interact, which limited the exchange of knowledge between them. Researchers in [17], [18] have shown considerable improvement with joint learning or multi-task learning, where both tasks use a joint encoder to

derive overlapping features without engaging in direct interaction. Joint models are often distinguished as either implicit or explicit. Joint learning with explicit interaction enhances performance as it controls the knowledge transfer process. Using a shared encoder, implicit joint modeling captures similar features without explicit interaction. A shared RNN was presented by Zhang and Wang [17] for Joint ID and SF tasks in order to understand the connection between intent and slots. Liu and Lane [38] created a framework called Attention BiRNN that uses a shared encoder-decoder and an attention system for ID and SF. Liu and Lane [39] conducted research with the same goal of improving online prediction capabilities by using a shared RNN to execute Joint SLU-LM, which combines SF, ID, and language modeling. A shared RNN-LSTM architecture was also introduced by Hakkani-Tu¨r et al. [18] for joint modeling usage. Explicit joint modeling can be further divided into two categories: single flow interaction and bidirectional flow interaction. Other studies [21], [22], [7] have modelled multi-task framework with one-way interaction, focusing primarily on exploring the intent guidance for extracting slots. Many [9], [19], [20] incorporate joint learning with explicit two-way interaction, considering the synergy between the tasks. These explicit modeling approaches offer the advantage of directly controlling the interaction process. They brings forth notable benefits like enhanced performance and improved interpretability, which are critical factors in advancing the efficacy of NLU systems.

Pre-trained neural models utilizing transformer architecture like BERT, GPT, T5, and their variants, have revolutionized recent trends in NLP tasks. Some work has investigated models pre-trained with BERT [11] for slot labeling tasks, using a common BERT architecture as the encoder to generate contextual representations. While the representation of the whole sequence is used for SF, ID uses the representation obtained from the special token. For instance, Chen et al. [10] investigated the application of BERT for SLU, utilizing BERT to capture joint contextual embeddings for both ID and SF. This method produced a clear performance gain compared with models that were not pretrained. For the concurrent modeling of ID and SF, Castellucci et al. [16] applied a comparable architecture (BERT-Joint). Qin et al. [7] decided to improve the performance of the model by using a pre-trained embedding encoder instead of the attention encoder (Stack-Propagation and BERT). Qin et al. [40] also looked at using BERT for Spoken Language Understanding (Co-Interactive Transformer and BERT), obtaining state-of-the-art results.

However, these studies did not consider the presence of different intents that can exist for a single statement, which is unavoidable in real-world situations. According to researchers in [8], multi-intent features are present in more than half of the instances in the Amazon internal dataset, demonstrating the applicability of multi-intent in practical scenarios. A collaborative learning framework was proposed for detecting multiple intents and filling slots. Qin et al. [6] explored a flexible graph-interactive framework designed to illustrate the interactions at the token level between various intents and slots. Cai et al. [1] included an explicit slot-intent classifier to learn the connection between slots and intents, resulting in commendable performance. To address the limitations of slow sequential models, Qin et al. [46] use a global-local graph that represents dependencies among different intents and slot labels using a parallel computation approach. Wan et al. [2] proposed a unified strategy to address the challenge of slot-nesting. Chen et al. [41] introduced a Prediction-Aware Contrastive Learning (PACL) framework with a two-stage approach to enhance multi-intent NLU by employing a word-level data augmentation method to generate a pre-training dataset. In contrastive fine-tuning, the framework assigns roles to instances in a flexible manner by using a prediction-aware contrastive loss, which improves the effectiveness of contrastive learning. He et al. [42] incorporate conceptual knowledge to augment the understanding of intent-slot relationships,

enabling better generalization and handling of complex interactions. Wu et al. [43] introduce a dual-level interaction mechanism for multi-intent SLU, which explicitly captures both local (token-level) and global (sentence-level) intent-slot interactions, improving the understanding of multi-intent utterances. Qin et al. [44] introduce the Divide-Solve-Combine (DSC) framework, an interpretable zero-shot prompting strategy that decomposes complex utterances into single-intent sub-clauses to improve the accuracy of MID in LLMs. Researchers in [45] developed a joint multi-intent SLU model that utilizes a Bidirectional Graph Attention Network (BiGAT) enhanced by LLM-generated features to capture intricate word-level dependencies for simultaneous ID and SF.

Table 1 summarizes a comparative analysis of various studies in ID and SF tasks, which have been experimented on two benchmark corpora, namely ATIS and SNIPs (versions with single and multiple intent). Many studies ([5], [4]) have experimented with non-autoregressive models which need less computation time. Optimizing inference speed typically comes at the cost of accuracy, making it challenging to achieve both speed and precision. It is evident that joint learning strategies are powerful in achieving higher performance. We extend the work of Cai et al. [1] by using slot propagation to enhance multi-intent classification. We discuss the limitations of strict slot–intent mapping approaches and motivate the use of slot-aware semantic signaling to support multi-intent detection.

3. Dataset

We use two multi-intent datasets, Mix-ATIS [6] and Mix-SNIPS [6], which are derived from ATIS [14] and SNIPS [15] respectively to form a multi-intent dataset. We have adopted the version of these two datasets annotated by Cai et al. [1], which is labeled with token-level slot-intent. As seen in Table 2, Mix-ATIS consists of 14748 utterances, whereas Mix-SNIPS consists of 44173 utterances.

IOB (Inside, Outside, Beginning) formatting has been used in both datasets for tagging each token with a slot. In this scheme, ”B” and ”I” are used to denote the beginning and continuation of entities or slots, respectively. They are particularly useful when dealing with slots that span across multiple words within a sentence. ”I”: Signifies that the term is located within or at the end of a slot. ”O”: Does not correspond to any designated slot. ”B”: Signifies the starting point of a slot. For single-word slots, only ”B” is necessary as there is no continuation of the slot covering the next word in the same slot. The following characteristics were observed in both datasets:

- Intent detection in a single intent dataset is approached as a multi-class sequence classification problem where the entire utterance is assigned a single intent label. Figure 2 illustrates that the utterance ”show me all flights from philadelphia to baltimore” is assigned to the intent class ”atis flight”.
- Intent detection within a multi-intent dataset that reflects real-world scenarios poses the challenge of multi-label sequence classification. In this context, each utterance may belong to one or more intent classes simultaneously, as seen in Figure 3a. Since utterances may contain more than one intent label, intent annotations are represented using multi-hot vectors during training.
- Slot-filling poses a multi-class token classification problem where each token withinan utterance is categorized into a specific slot class. For instance, in Figure

3a, the utterance ”what does ff mean and also what are the cities that American Airlines serves” slot-filling would involve classifying each word into slots such as ’O’ , ’B-airline code’, ’B-airline name’ and ’I-airline name’.

- Slot-Intent classification is also a multi-class token classification problem where each token is classified into a slot-intent. In this problem, tokens that do not belong to slot-O are tagged as one of the intent labels.
- Case 1: Figure 3 illustrates how different slots in multi-intent utterances correspond to distinct intentions, while in single-intent utterances, all the slots belong to a single intent. Case 2: Not all utterances necessarily have tokens with slots, but they have intents. For instance in Figure 4a, the utterance ”names of airport” has only one intent i.e., atis airport. But none of the tokens belong to any slot class, and hence all tokens are tagged with ”O”. In a complex utterance with multiple intents, such as in Figure 4b, it is possible that there are one or more intent classes for which the slot-intent mapping may not exist.

Table 1. Comparison of Various Models Used in Intent Classification and Slot Filling. Parameters: 1. Multi-Intent, 2. Joint Learning, 3. Few-Shot Learning, 4. Non-Autoregressive, Transformer-Based, 6. Explicit Interaction, 7. Graph-Based

Model	1	2	3	4	5	6	7
Bi-Model		✓					
SF-ID		✓					
Joint Multiple ID-SF	✓	✓					
Joint BERT		✓			✓		
Stack Prop		✓			✓	✓	
AGIF	✓	✓					✓
SlotRefine		✓		✓	✓		
ConProm		✓	✓		✓		
Retriever		✓	✓	✓			
multi-grained label refinement		✓			✓		✓
SDJN	✓	✓					
Wheel-GAT		✓			✓		✓
SLIM	✓	✓			✓		
GLGIN	✓	✓		✓	✓	✓	✓
MTLN-GP	✓	✓			✓		
DGIF	✓	✓			✓	✓	✓
TKDF	✓	✓		✓	✓	✓	
DSCP	✓		✓		✓	✓	
Joint Bi-GAT	✓	✓			✓	✓	✓
S3MIC (Proposed Model)	✓	✓			✓	✓	

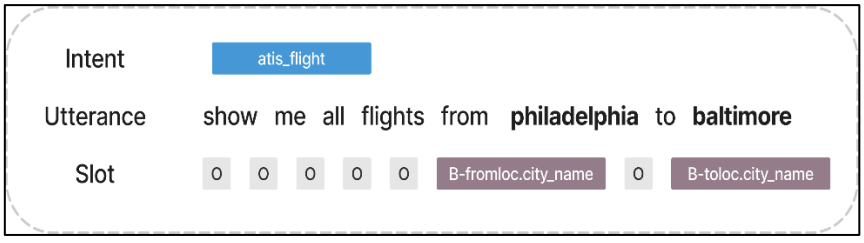


Figure 2. Sample of Single Intent Utterance from MixATIS Dataset

Table 2. Dataset Statistics

MixATIS	
intents	24
slots	121
train	13161
valid	759
test	828

MixSNIPS	
intents	10
slots	74
train	39776
valid	2198
test	2199

4. Proposed Approach

We present a detailed study of the suggested S3MIC framework in this section. First, we state the problem and define it clearly to set the right foundation for the recommended solution. Then, we describe the model architecture with some explanations about the motivation behind critical design choices. The training procedure is elaborated further in Section 4.2 and Algorithm 1. discuss about the limitations we met during training and discuss the factors that have influenced the model's behavior and development. This section presents an overview of the approaches used in our work.

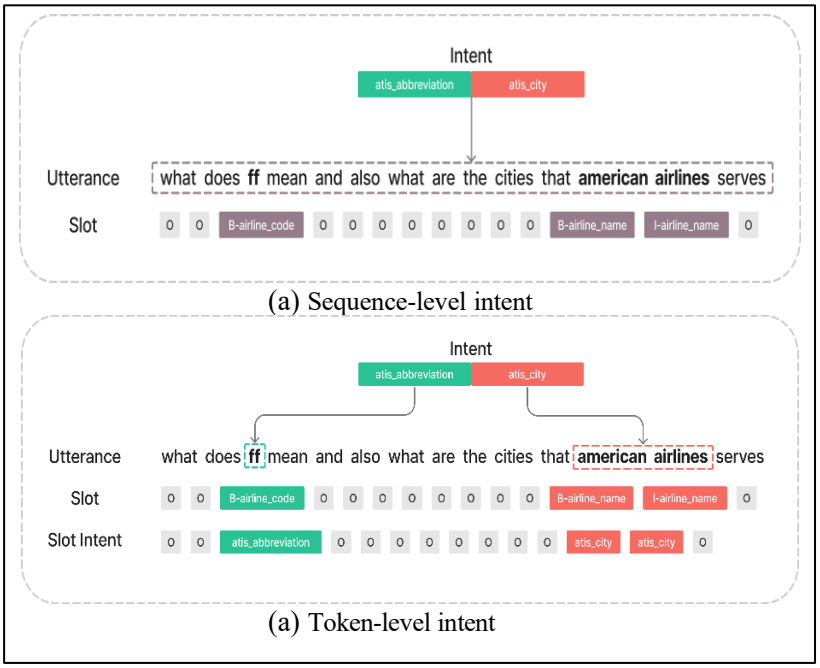


Figure 3. Sequence-Level vs Token-Level Intent Example from MixATIS Dataset

4.1 Problem Statement

In the context of a given input utterance u consisting of a sequence of tokens $u = (u_1, u_2, \ldots, u_n)$, the integrated approach to MID and SF involves two main components. Utterance-level intent detection involves predicting intents I_u associated with the input utterance, in which I represents the collection of intents. Word-level slot-recognition predicts the slot category for every word in an input utterance from a collection of potential slots (T). When compared to the standard single-intent SLU task, three important premises are adopted:

- Utterance-Level Intent: Each utterance u is assumed to have at least one sequence-level intent, denoted as $|I_u| \geq 1$

- Token-Level Slot: Each token t is assumed to be categorized into exactly one slot class, denoted as s_t. If the token does not belong to any slot u_t will be marked as O. Whenever there is an utterance u such that it does not consist of any entity, will have no slots and hence every token of that utterance will be classified as a non-slot word as "O"
- Slot-Intent Mapping: Every slot $s_m = \{u_{m1}, \ldots, u_{mj}\}$ is composed of j tokens and is
- explicitly linked to a corresponding intent at the utterance level, expressed as $i_m \in I_u$.

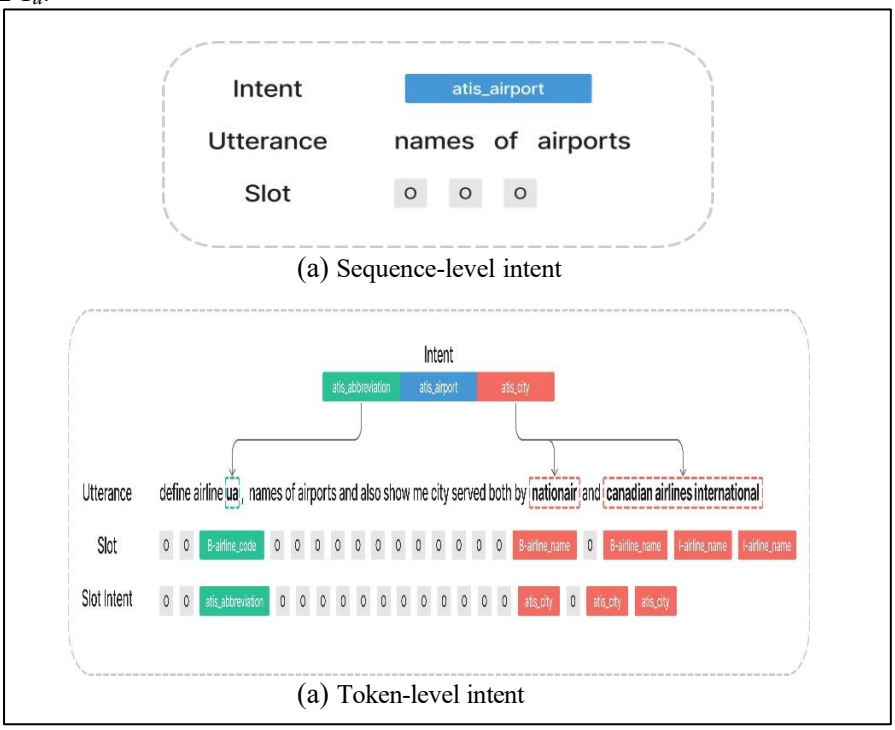


Figure 4. Case 2: Sequence-Level vs Token-Level Intent Example from MixATIS Dataset

4.2 Model Structure

Figure 5 illustrates the overall structure of the S3MIC model. The proposed model (S3MIC) comprises a shared encoder followed by three separate decoder modules designed for each classification task. This inherent sharing of parameters enables the model to draw more intrinsic dependency between the features of two tasks, whereas individual decoders craft more task specific independent features. To guide multi-intent classification, we incorporate slot logits into the intent prediction module. This allows the model to utilize slot-level semantic signals while jointly optimizing slot filling and intent detection.

4.2.1 Shared Encoder

Our model leverages BERT Large as a unified encoder to simultaneously learn three tasks: sequence-level MID, token-level SF, and token-level slot-intent classification. The entire input sequence is tokenized using BERT Tokenizer, ensuring that a special token [CLS] is affixed at the beginning and [SEP] is affixed at the end. Both the pooled output i.e., h_{cls} and the

sequential output of BERT i.e., $h = (h_{cls}, h_1, ..., h_n, h_{sep})$ have been employed for the three classifiers.

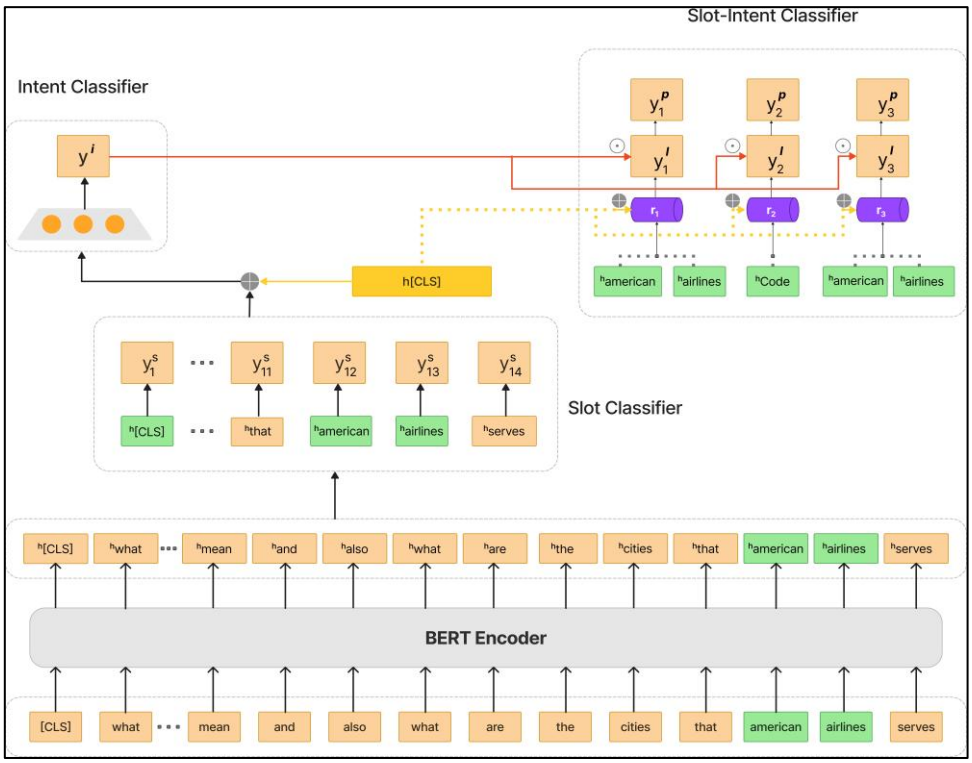


Figure 5. Architecture of S3MIC (Proposed Model)

4.2.2 Token-Level Slot Classifier

While performing the slot recognition, we aim to predict the slot category y_{sk} for the k-th position within a sequence. To achieve this, we feed the sequential output from the encoder to a separate slot classification network and apply $Softmax$ as :

$$y^s{}_k = Softmax(W^s{}_{hk} + b^s) \quad (1)$$

where W^s is a matrix in $\mathbb{R}^{j \times T \times d}$, and y_{sk} is a vector in $\mathbb{R}^{j \times T}$ representing the probability distribution of slots for the token x_k at position k.

4.2.3 Token-Level Slot-Intent Classifier

The slot-intent classifier is a token-level intent classifier which is devised similarly to Cai et al. [1]. Token embeddings within the same slot span are averaged using mean pooling to form a slot-level representation. Since slots may contain multiple tokens, averaging their contextual embeddings captures the overall meaning of the slot while producing a fixed-size vector. As the transformer encoder already generates context-aware token representations, this process preserves the semantic information needed for slot–intent interaction. This is represented as:

$$r_m = \frac{1}{j} \sum_{i=1}^{j} h_{mi} \quad (2)$$

, where h_{mi} denotes the representation of token x_{mi}. Subsequently, we concatenate the pooled output of BERT encoder h_{cls} with rm and calculate the slot-intent ylm as:

$$y_m^l = \text{Softmax}(Wl[h_{cls} \oplus r_m] + b^l) \quad (3)$$

Here, W_l is a matrix in $\mathsf{R}^{j \times I \times 2d}$, and $\oplus$ denotes the concatenation operation.

4.2.4 Sequence-Level Multi-Intent Classifier

MID is formulated as a sequence-level multi-label prediction task, where a single utterance may contain multiple intent labels simultaneously. Let the intent label space contain I possible intent classes. For each utterance, the ground-truth intents are represented using a *multi-hot* vector $\mathbf{y} \in \{0, 1\}^I$, where $y_i = 1$ represents that the i-th intent is present, while $y_i = 0$ indicates that the intent does not occur. This representation allows multiple intent labels to be active for a single utterance during training. We leverage the sequence-level intents with token-level slot predictions by concatenating them with the pooled output of the encoder. We integrate a dense neural network layer that applies the ReLU activation function before predicting the intents using the Sigmoid activation function.

$$y_m^l = g(Wl[h_{cls} \oplus y_k^s] + b^l) \quad (4)$$

$$y_i = \text{Sigmoid}(W_i[y_m^l] + b_i) \quad (5)$$

Here, W_i is a matrix in $\mathsf{R}^{j \times I \times d}$, where each dimension of y_i in $\mathsf{R}^{j \times I}$ signifies the likelihood of an intent label.

4.3 Training Constraints

For our experimentation, we used Python as the programming language and the PyTorch framework. We trained the S3MIC model using the computational strength of NVIDIA A-100 GPU. We use the English uncased BERT-Large model [11]. The model configuration consists of 24 layers, where each layer contains 1024 hidden units and 16 attention heads. This gives a strong base for our tasks. We fixed some parameter settings for training to optimize consistently and effectively. The highest sequence length, training group size, and drop-out ratio were set at 50, 32, and 0.2, respectively. We also employed an additional feedforward layer to the decoder of the Multi-Intent Classifier. The sequence length was set to 50 to allow a fair comparison with prior work. Since most utterances in the datasets are short, this length is enough to capture the full context without affecting performance. In the training, S3MIC was made to go through 20 epochs for full study and adjustments. The loss weights α and β adjust the balance between the intent classification loss and the slot labeling loss during joint optimization, as shown in the training algorithm. In our experiments, we follow the same configuration as the SLIM model and set α = 2 and β = 1 to ensure fair comparison with prior work. In our model formulation, Equation 3 shows that there is an activation function represented by g, which is an essential part of the training process of a multi-intent classifier. We experimented with both ReLU (Rectified Linear Unit) and PReLU (Parametric Rectified Linear Unit) activation mechanisms to see how the model performs. Generally, PReLU provides increased learning flexibility by allowing the network to adapt small negative values, while ReLU is simpler and tends to be more stable. In our experiments, both activation functions worked well and produced consistently good results.

Algorithm 1. S3MIC for joint MID and SF

procedure S3MIC TRAINING WITH BERT(Sentence S, Slot Label Set L, Intent Label Set I)
    Initialize BERT model B
    for each training iteration do
        Sample mini-batch of sentences $S_1, S_2, \ldots, S_m$ from training data
        for each sentence S_i do
            Tokenize S_i into subword tokens T_i using BERT tokenizer
            Add [CLS] at the start and [SEP] at the end of T_i
            Pass tokenized sentence T_i through BERT model B to obtain embeddings E_i
            Split embeddings E_i into slot-specific E_{slot} and intent-specific E_{intent} subspaces:
                $E_{\text{slot}} = W_{\text{slot}} \cdot E_i + b_{\text{slot}}$
                $E_{\text{intent}} = W_{\text{intent}} \cdot E_{[\text{CLS}]} + b_{\text{intent}}$
            Apply linear layer and softmax on E_{slot} to get slot label predictions $\hat{L}_{\text{slot}}$:
                $\hat{L}_{\text{slot}} = \text{softmax}(W_{\text{slot}} \cdot E_{\text{slot}} + b_{\text{slot}})$
            Concatenate slot classifier output $\hat{L}_{\text{slot}}$ with $E_{[\text{CLS}]}$ to provide input to multi-intent classifier:
                $E_{\text{intent combined}} = \text{concatenate}(\hat{L}_{\text{slot}}, E_{[\text{CLS}]})$
            Apply linear layer and softmax on $E_{\text{intent\_combined}}$ to get intent label predictions $\hat{I}$:
                $\hat{I} = \text{softmax}(W_{\text{intent}} \cdot E_{\text{intent\_combined}} + b_{\text{intent}})$
        end for
        Compute slot loss L_{slot} using cross-entropy between $\hat{L}_{\text{slot}}$ and true slot labels
        Compute intent loss L_{intent} using cross-entropy between $\hat{I}$ and true intent labels
        Combine losses with a weighted sum:
            $L_{\text{total}} = \alpha \cdot L_{\text{slot}} + \beta \cdot L_{\text{intent}}$
        Compute gradients of L_{total} for all trainable parameters through backpropagation
        Adjust model parameters with the selected optimizer (e.g., Adam)
    end for
end procedure

5. Analysis of Experimental Results

This section briefly summarizes the experimental results. We discuss the metrics on which the success of the S3MIC model is based. Afterward, the results are discussed so as to highlight the overall performance of the model. Besides discussing statistical indicators, we also reflect on the broader implications of the results, especially regarding their adequacy in addressing the problem at hand and their possible applications in reality. These results not only provide a confirmation of the effectiveness of S3MIC but should also prove to have major practical implications.

5.1 Evaluation Metrics

We compare results on the most widely adopted evaluation metrics for ID and SF, which are Slot F1 score, intent accuracy, and overall semantic frame accuracy.

F1 Score: To determine the efficiency of SF, we apply the F1 score. F1 scores tell us about both the correctness and comprehensiveness of the slot filling. In slot filling evaluation, correctness is determined by exact matches between predicted slots and the expected ones. We calculate precision (Pt), recall (Rt), and the Slot F1-score (F1-scoret) at the token level as follows:

$$\text{Precision } P_t = \frac{\text{True Positive }_t}{\text{True Positive }_t + \text{ False Positive }_t}$$

$$\text{RecallR}_t = \frac{\text{True Positive }_t}{\text{True Positive }_t + \text{ False Negative }_t}$$

$$\text{F1-score }_t = 2 \times \frac{P_t \times R_t}{P_t + R_t}$$

where - True Positive$_s$: Number of correctly predicted slots. - False Positive$_s$: Number of slots incorrectly predicted. - False Negative$_S$: Number of slots not predicted but are actual slots. The F1 score reported in our experiments is micro-averaged, which aggregates the contributions of all classes when computing precision and recall.

Intent Accuracy: We use accuracy to see how well intent detection is working. It represents the percentage of sentences for which the system correctly predicts the intended purpose or goal. By denoting N = total number of utterances in the dataset and C_u = number of correctly predicted utterances, the Intent Accuracy can be calculated as:

$$Intent\_Acc = \frac{C_u}{N}$$

Semantic Frame Accuracy: We use overall accuracy to see how often both the intent and slot are predicted correctly in a sentence. This measure looks at intent recognition and slot labeling together. For evaluating the overall performance when both slots and intents are correctly predicted, we can compute the joint accuracy as:

$$\text{SeFr Acc (Slots and Intents) } = \frac{N_{\text{correct}}}{N_{\text{total}}}$$

where N_{total} = Total number of utterances in the dataset and N_{correct}: Number of utterances where both slots and intents are correctly predicted.

5.2 Overall Results

We prioritize overall accuracy because it reflects whether both the intents and the corresponding slot labels for an utterance are predicted correctly. Table 3 indicates that our model outperforms MTLN-GP, achieving a 1.8% increase in Intent Accuracy and a 0.21% improvement in Slot F1 score on MixATIS. It would not be fair to compare S3MIC with MTLN-GP on MixSNIPS, as overall accuracy for MTLN-GP is not available, and the other two metrics cause a dilemma. Although the Joint Bi-GAT model [45] achieves a slightly higher Intent Accuracy on the MixSNIPS dataset, S3MIC achieves a superior 86.28 SeFr Acc, indicating a substantial improvement in overall sentence understanding. S3MIC surpasses SLIM by 4.9% overall accuracy on MixATIS and 3.52% on MixSNIPS. For fair comparison,

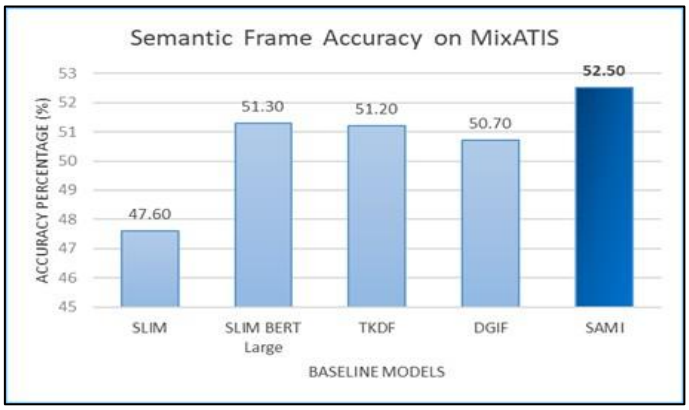
we experiment with SLIM using BERT Large, which increases the overall accuracy of SLIM by 3.7% on MixATIS and 1.36% on MixSNIPS. S3MIC outperforms SLIM with BERT Large by 1.2% on MixATIS and 0.92% on MixSNIPS, which indicates the effectiveness of our approach.

Table 3. Comparison with Baseline Models

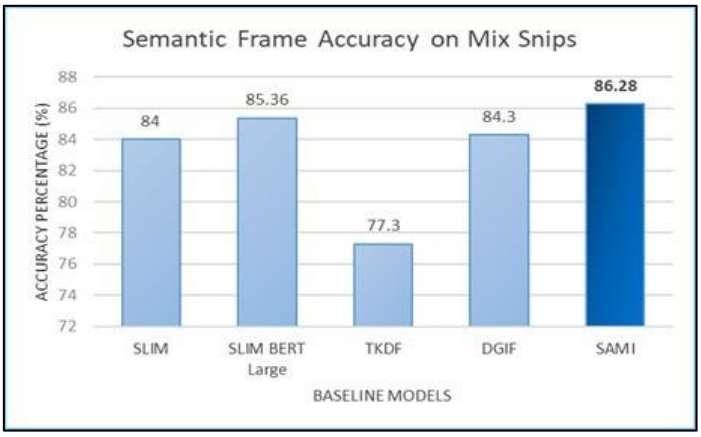
Model	MixATIS			MixSNIPS		
	Slot F1	Intent Acc	SeFr Acc	Slot F1	Intent Acc	SeFr Acc
Bi-Model [6]	85.5	72.3	39.1	86.8	95.3	53.9
SF-ID [6]	87.7	63.7	36.2	89.6	96.3	59.3
Stack-Propagation (1) [6]	86.6	76.0	42.8	93.9	96.4	75.5
Stack-Propagation (2) [6]	87.4	71.9	41.0	93.2	94.6	71.9
Joint Multiple ID-S [6]	87.5	73.1	38.1	91.0	95.7	66.6
AGIF [6]	88.1	75.8	44.5	94.5	96.5	76.4
SDJN [3]	88.2	77.1	44.6	94.4	96.5	75.7
DIF [43]	88.2	75.8	49.3	94.4	95.3	75.9
CKEM [42]	89.7	77.1	46.6	96.7	96.1	79.7
SLIM [1]	88.5	78.3	47.6	96.5	97.2	84.0
SLIM (PACL) [41]	87.3	81.9	50.4	96.8	96.9	85.1
SLIM (BERT Large)	88.47	79.4	51.3	96.78	96.44	85.36
GL-GIN [46]	88.3	76.3	43.5	94.9	95.6	75.4
MTLN-GP [2]	88.4	79.6	–	96.7	97.9	–
DGIF [24]	88.5	83.3	50.7	95.9	97.8	84.3
TKDF [23]	89.8	78.4	51.2	94.6	97.4	77.3
Inter-DSCP [44]	81.86	52.22	-	96.66	92.0	-
Joint Bi-GAT [45]	89.6	78.4	49.9	96.2	98.6	78.3
S3MIC (Proposed Model)	88.61	81.4	52.5	96.92	97.72	86.28

Explicit slot–intent mapping approaches, such as those proposed in [1], assume that the relationship between slots and intents can be modeled through predefined associations. While this design improves interaction between SF and ID, it may prevent the model from fully leveraging the rich contextual representations produced by the encoder. In practice, slots and intents often have complex and context-sensitive relationships, which makes it difficult to represent all dependencies through explicit mappings alone. Furthermore, strong reliance on such mappings can introduce coupling between the SF and ID tasks, where errors in one component may affect the other during joint prediction.

In the proposed approach, the slot classifier output provides additional semantic signals to the intent classifier. By incorporating slot-level representations into the intent prediction module, the model can leverage entity-level information alongside the global sentence context captured by the encoder. Specifically, the slot classifier outputs are concatenated with the encoder's h[CLS] representation before intent prediction. While the h[CLS] vector captures the overall contextual meaning of the utterance, it may not explicitly encode localized slot-level semantics. Integrating slot representations, therefore, enriches the input to the intent classifier and improves its ability to distinguish between intents, particularly in ambiguous or multi-intent scenarios.



(a) Semantic Frame Accuracy on MixATIS



(b) Semantic Frame Accuracy on MixSnips

Figure 6. Comparative Analysis of Semantic Frame Accuracy Across Leading Models

5.3 Impact of the S3MIC Framework

To further evaluate our slot-aware strategy, we compare it with the standard SLIM model and the SLIM variant using BERT Large. Figure 6b exhibits the experimental results between the top five benchmark results. Compared to the baseline models, S3MIC clearly performs better. The SLIM model performs better than TKDF [23] and DGIF [24] on the MixSNIPS dataset, highlighting the benefit of incorporating slot–intent interactions over distillation-based methods. However, approaches that rely primarily on explicit slot–intent associations may not fully exploit the contextual slot representations generated by the encoder, which can limit intent prediction in complex multi-intent scenarios. S3MIC addresses this limitation by integrating slot contextual representations with the intent classifier. By propagating slot-level semantic signals alongside the encoder's contextual representation, the model is able to utilize richer information during intent prediction. As a result, S3MIC outperforms both SLIM and SLIM with BERT Large, demonstrating the effectiveness and robustness of the proposed S3MIC framework.

5.4 Impact of BERT Large

We evaluate the regular SLIM model against the SLIM with BERT Large in order to confirm the efficacy of applying the pretrained BERT Large. SLIM with BERT Large achieves higher Semantic Frame accuracy than SLIM models on both MixATIS and MixSNIPS datasets, indicating the effectiveness of employing the BERT Large model. A total of 345M parameters are present in BERT Large. The decoder part of S3MIC consists of 65.8 M parameters

on MixATIS and 60.4M parameters on MixSNIPS, which aggregate to 410.8M and 405.4M parameters respectively. More parameters lead to utilizing more resources and more time. However, S3MIC takes about 1 hour 20 minutes and 2 hours 58 minutes to train on MixATIS and MixSNIPS respectively, which is a decent amount of time given the complexity of models these days. Prior approaches, such as Stack-Propagation, Joint Multiple ID-SF, and AGIF report decoding latencies of 34.5 s, 45.3 s, and 48.5 s, respectively, while the non-autoregressive GL-GIN model achieves 4.2 ms latency [46]. In comparison, S3MIC achieves 6.01 ms per utterance with a batch size of 32 and 4.9 ms with a batch size of 64. S3MIC was trained on an NVIDIA A100 GPU (40GB), while inference latency was measured on an NVIDIA RTX 3060 GPU (12GB). It is worth noting that GL-GIN experiments were conducted on GeForce RTX 2080Ti and TITAN Xp GPUs. Despite evaluating inference on comparatively modest hardware, S3MIC maintains competitive latency while achieving higher Semantic Frame Accuracy, showing a good balance between computational efficiency and semantic understanding performance.

6. Conclusion

The paper proposes a Slot-Aware Semantic Signaling framework for Multi-Intent Classification, abbreviated as S3MIC. The proposed S3MIC framework is a new method for enhancing the performance of MID and SF integrated learning. The proposed method differs from other methods in that, during the intent prediction step, slot logs are used at the slot level. In the proposed method, the use of slot logs at the slot level enables the model to process multiple intents in a sentence more efficiently. As the proposed method maintains a simple model structure and is easier to integrate into existing integrated models, it is effective. It was evaluated using these datasets. The proposed S3MIC method achieved high accuracy when evaluated using the MixATIS and MixSNIPS datasets. Compared to other benchmark models, the proposed S3MIC method was found to have an accuracy of 52.5% when evaluated using the MixATIS dataset, and 86.28% when evaluated using the MixSNIPS dataset.

Acknowledgement

We thank the Computer Science and Engineering Department, Institute of Technology, Nirma University, for supporting this research.

References

[1] Cai, Fengyu, Wanhao Zhou, Fei Mi, and Boi Faltings. "Slim: Explicit Slot-Intent Mapping with Bert for Joint Multi-Intent Detection and Slot Filling." In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, 7607-7611.

[2] Wan, Xue, Wensheng Zhang, Mengxing Huang, Siling Feng, and Yuanyuan Wu. "A Unified Approach to Nested and Non-Nested Slots for Spoken Language Understanding." Electronics 12, no. 7 (2023): 1748.

[3] Chen, Lisong, Peilin Zhou, and Yuexian 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), IEEE, 2022, 7612-7616.

[4] Yu, Dian, Luheng He, Yuan Zhang, Xinya Du, Panupong Pasupat, and Qi Li. "Few-Shot Intent Classification and Slot Filling with Retrieved Examples." In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, 734-749.

[5] Wu, Di, Liang Ding, Fan Lu, and Jian Xie. "SlotRefine: A Fast Non-Autoregressive Model for Joint Intent Detection and Slot Filling." In Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), 2020, 1932-1937.

[6] Qin, Libo, Xiao Xu, Wanxiang Che, and Ting Liu. "AGIF: An Adaptive Graph-Interactive Framework for Joint Multiple Intent Detection and Slot Filling." In Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, 1807-1816.

[7] Qin, Libo, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. "A Stack-Propagation Framework with Token-Level Intent Detection for Spoken Language Understanding." In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), 2019, 2078-2087.

[8] Gangadharaiah, Rashmi, and Balakrishnan Narayanaswamy. "Joint Multiple Intent Detection and Slot Labeling for Goal-Oriented Dialog." In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, 564-569.

[9] Wang, Yu, Yilin Shen, and Hongxia Jin. "A Bi-Model Based RNN Semantic Frame Parsing Model for Intent Detection and Slot Filling." In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), 2018, 309-314.

[10] Chen, Qian, Zhu Zhuo, and Wen Wang. "Bert for Joint Intent Classification and Slot Filling." arXiv preprint arXiv:1902.10909 (2019).

[11] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding." In Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, 4171-4186.

[12] Haffner, Patrick, Gokhan Tur, and Jerry H. Wright. "Optimizing SVMs for Complex Call Classification." In 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03)., vol. 1, IEEE, 2003, I-I.

[13] Raymond, Christian, and Giuseppe Riccardi. "Generative and Discriminative Algorithms for Spoken Language Understanding." In Interspeech 2007-8th Annual Conference of the International Speech Communication Association. 2007.

[14] Hemphill, Charles T., John J. Godfrey, and George R. Doddington. "The ATIS Spoken Language Systems Pilot Corpus." In Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990. 1990.

[15] Coucke, Alice, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro et al. "Snips Voice Platform: An Embedded Spoken Language Understanding System for Private-By-Design Voice Interfaces." arXiv preprint arXiv:1805.10190 (2018).

[16] Castellucci, Giuseppe, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. "Multi-Lingual Intent Detection and Slot Filling in a Joint Bert-Based Model." arXiv preprint arXiv:1907.02884 (2019).

[17] Zhang, Xiaodong, and Houfeng Wang. "A Joint Model of Intent Determination and Slot Filling for Spoken Language Understanding." In IJCAI, vol. 16, no. 2016, 2016, 2993-2999.

[18] Hakkani-Tür, Dilek, Gökhan Tür, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. ”Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional Rnn-Lstm.” In Interspeech, 2016, 715-719.

[19] Zhang, Chenwei, Yaliang Li, Nan Du, Wei Fan, and Philip S. Yu. "Joint Slot Filling and Intent Detection Via Capsule Neural Networks." In Proceedings of the 57th annual meeting of the association for computational linguistics, 2019, 5259-5267.

[20] Liu, Yijin, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu. "Cm-Net: A Novel Collaborative Memory Network for Spoken Language Understanding." In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), 2019, 1051-1060.

[21] Goo, Chih-Wen, Guang Gao, Yun-Kai Hsu, Chih-Li Huo, Tsung-Chieh Chen, Keng-Wei Hsu, and Yun-Nung 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), 2018, 753-757.

[22] Li, Changliang, Liang Li, and Ji Qi. ”A Self-Attentive Model with Gate Mechanism for Spoken Language Understanding.” In Proceedings of the 2018 conference on empirical methods in natural language processing, 2018, 3824-3833.

[23] Cheng, Xuxin, Zhihong Zhu, Wanshi Xu, Yaowei Li, Hongxiang Li, and Yuexian Zou. ”Accelerating Multiple Intent Detection and Slot Filling Via Targeted Knowledge Distillation.” In Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, 8900-8910.

[24] Zhu, Zhihong, Weiyuan Xu, Xuxin Cheng, Tengtao Song, and Yuexian Zou. ”A Dynamic Graph Interactive Framework with Label-Semantic Injection for Spoken Language Understanding.” In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2023, 1-5.

[25] Mesnil, Grégoire, Xiaodong He, Li Deng, and Yoshua Bengio. ”Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding.” In Interspeech, 2013, 3771-3775.

[26] Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. ”Natural Language Processing (Almost) from Scratch.” Journal of Machine Learning Research 12 (2011): 2493-2537.

[27] Yao, Kaisheng, Geoffrey Zweig, Mei-Yuh Hwang, Yangyang Shi, and Dong Yu. ”Recurrent Neural Networks for Language Understanding.” In Interspeech, 2013, 2524-2528.

[28] Mikolov, Tomas, Stefan Kombrink, Anoop Deoras, Lukar Burget, and Jan Cernocky. ”Rnnlm-Recurrent Neural Network Language Modeling Toolkit.” In Proc. of the 2011 ASRU Workshop, 2011, 196-201.

[29] Ravuri, Suman V., and Andreas Stolcke. ”Recurrent Neural Network and LSTM Models for Lexical Utterance Classification.” In Interspeech, 2015, 135-139.

[30] Mesnil, Grégoire, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He et al. ”Using Recurrent Neural Networks for Slot Filling in Spoken Language Understanding.” IEEE/ACM Transactions on Audio, Speech, and Language Processing 23, no. 3 (2014): 530-539.

[31] Liu, Bing, and Ian Lane. ”Recurrent Neural Network Structured Output Prediction for Spoken Language Understanding.” In Proc. NIPS Workshop on Machine Learning for Spoken Language Understanding and Interactions. 2015.

[32] Elman, Jeffrey L. ”Finding Structure In Time.” Cognitive science 14, no. 2 (1990): 179-211.

[33] Jordan, Michael I. ”Serial Order: A Parallel Distributed Processing Approach.” In Advances in psychology, vol. 121, North-Holland, 1997, 471-495.

[34] Xu, Puyang, and Ruhi Sarikaya. ”Convolutional Neural Network Based Triangular Crf for Joint Intent Detection and Slot Filling.” In 2013 IEEE workshop on automatic speech recognition and understanding, IEEE, 2013, 78-83.

[35] Yao, Kaisheng, Baolin Peng, Geoffrey Zweig, Dong Yu, Xiaolong Li, and Feng Gao. ”Recurrent Conditional Random Field for Language Understanding.” In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2014, 4077-4081.

[36] Yao, Kaisheng, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. ”Spoken Language Understanding Using Long Short-Term Memory Neural Networks.” In 2014 IEEE spoken language technology workshop (SLT), IEEE, 2014, 189-194.

[37] Vu, Ngoc Thang, Pankaj Gupta, Heike Adel, and Hinrich Schütze. ”Bi-Directional Recurrent Neural Network with Ranking Loss for Spoken Language Understanding.” In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2016, 6060-6064.

[38] Liu, Bing, and Ian Lane. ”Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling.” arXiv preprint arXiv:1609.01454 (2016).

[39] Liu, Bing, and Ian Lane. ”Joint Online Spoken Language Understanding and Language Modeling with Recurrent Neural Networks.” In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2016, 22-30.

[40] Qin, Libo, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting 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), IEEE, 2021, 8193-8197.

[41] Chen, Guanhua, Yutong Yao, Derek F. Wong, and Lidia S. Chao. ”A Two-Stage Prediction-Aware Contrastive Learning Framework for Multi-Intent NLU.” In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024, 1778-1788.

[42] He, Li, Jingxuan Zhao, Jianyong Duan, Hao Wang, and Xin Li. ”Conceptual Knowledge Enhanced Model for Multi-Intent Detection and Slot Filling.” IEICE TRANSACTIONS on Information and Systems 107, no. 4 (2024): 468-476.

[43] Wu, Di, Liting Jiang, Lili Yin, Kai Wang, Haoxiang Su, Zhe Li, and Hao Huang. ”Dual Level Intent-Slot Interaction for Improved Multi-Intent Spoken Language Understanding.” In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2024, 12301-12305.

[44] Qin, Libo, Qiguang Chen, Jingxuan Zhou, Jin Wang, Hao Fei, Wanxiang Che, and Min Li. ”Divide-Solve-Combine: An Interpretable and Accurate Prompting Framework for Zero-shot Multi-Intent Detection.” In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 23, 2025, 25038-25046.

[45] Chen, Shuxin, Xu Li, Jiaqi Wang, and Yu Zhang. ”Joint Model for Multi-Intent Spoken Language Understanding Based on Bidirectional Graph Attention Network and Enhanced With Large Language Models.” In International Journal of Innovative Computing, Information and Control, 2025.

[46] Qin, Libo, Fuxuan Wei, Tianbao Xie, Xiao Xu, Wanxiang Che, and Ting Liu. ”GL-GIN: Fast and Accurate Non-Autoregressive Model for Joint Multiple Intent Detection and Slot Filling.” In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, 178-188.