INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Bi-directional Joint Neural Networks for Intent Classification and Slot Filling

Soyeon Caren Han[*]*, Siqu Long*[*]*, Huichun Li, Henry Weld, Josiah Poon*

School of Computer Science, University of Sydney
Sydney, NSW, Australia
{Caren.Han, slon6753, huli7926, hwel4188, Josiah.Poon}@sydney.edu.au

Abstract

Intent classification and slot filling are two critical tasks for natural language understanding. Traditionally the two tasks proceeded independently. However, more recently joint models for intent classification and slot filling have achieved state-of-the-art performance, and have proved that there exists a strong relationship between the two tasks. In this paper, we propose a bi-directional joint model for intent classification and slot filling, which includes a multi-stage hierarchical process via BERT and bi-directional joint natural language understanding mechanisms, including intent2slot and slot2intent, to obtain mutual performance enhancement between intent classification and slot filling. The evaluations show that our model achieves state-of-the-art results on intent classification accuracy, slot filling F1, and significantly improves sentence-level semantic frame accuracy when applied to publicly available benchmark datasets, ATIS (88.6%) and SNIPS (92.8%).
Index Terms: Intent classification, Slot filling

1. Introduction

The introduction of deep learning architectures has shown great success in Natural Language Understanding (NLU), including two main tasks, slot filling and intent classification, which aim to extract the user's intent and to identify semantic constituents from the natural language utterance. Traditionally the two are processed independently. Slot filling can be treated as a sequence labeling task that annotates an input word sequence $[x_1, x_2, \ldots, x_n]$ with a corresponding slot label sequence $[y_1^s, y_2^s, \ldots, y_n^s]$, each from a set of slot labels $\mathbb{S}$ [1]. Intent detection can be considered as a classification problem to map the input sequence to an intent label y^I from a set of intent labels $\mathbb{I}$. Traditional independent models in pipeline usually suffer from error propagation since such models do not consider the mutual relationship between the two tasks. This mutual relationship between intent classification and slot filling can be illustrated with the example. With a word *westbam* being identified as slot *artist*, the utterance query is more likely to be classified as music related, say *PlayMusic*, than from another area, like *BookRestaurant*. Conversely, with the intent recognised as *PlayMusic*, the slot categories are more likely to be related to music (e.g. *artist*, *album*) than other fields, like *restaurant_type*. A large body of work [2, 3, 4, 5, 6, 7, 8] has confirmed that conducting intent classification and slot filling simultaneously achieves a synergistic effect and improve performance over independent models. Earlier joint models typically adopted a single RNN model to fill slots and then detect an intent with a shared objective or, a detected intent informed the filling of slots. Such models achieved decent performance but did not entirely and/or explicitly consider the interrelationship between the intent and slot.

[*] equal contribution

In this paper, we introduce the bi-directional joint model for intent classification and slot filling, which contains a multi-stage hierarchical process using BERT (Bi-directional Encoder Representations from Transformers) [9] and a bi-directional NLU mechanism. The proposed model produces initial intent and slot probabilities from BERT word embeddings, then applies the Bi-directional NLU mechanism to obtain the mutual enhancement between intent classification and slot filling. Our Bi-directional NLU mechanism contains two models: intent2slot for slot filling and slot2intent for intent classification. The intent2slot draws on the intent probability distribution, utilising the semantic information of the whole sequence while labeling each word. The slot2intent takes in the sequence label probability distributions as supplementary information for intent classification. To jointly model intent classification and slot filling, the model is trained on the joint loss of slot labeling and intent classification. The contributions are: 1) A bi-directional joint NLU mechanism with intent2slot and slot2intent models achieves full synergistic effects for joint intent classification and slot filling; 2) The proposed model introduces a multi-stage hierarchical process integrating Transformer-based Modelling and a Bi-directional NLU mechanism.

2. Related work

Handling the two sub-tasks in a pipeline has achieved decent results, but suffers from error propagation; a misidentified intent can prompt seemingly matching but incorrect slots in the second step, or vice versa. Models that work on the tasks simultaneously are proposed, many of which implicitly model joint distributions via joint loss back-propagation [2, 3, 10]. However, explicitly capturing the relationships between intents and slots is beneficial. Goo [11] used a slot-gate attention-based joint model that enables the learned intent result to assist the slot filling process and performs well in semantic accuracy due to this global consideration. Zhang [12] proposed a capsule neural network-based joint model that has slot prediction inform intent, and feedback from intent to slot prediction. Chen [13] proposed to use the contextual BERT model to jointly learn the two tasks. However, they simply make prediction based on BERT output hidden states without further exploration of the potential relationship between the two tasks. Stack-Propagation-BERT [14] uses an intent2slot architecture with BERT encoding and using stack propagation. Stack propagation in multi-task architectures provides a differentiable link from one task to the other rather than performing each one in parallel.

3. Methodology

3.1. Input embedding layer

The input data to intent detection and slot filling tasks is user utterances in the form of text sentences, which are typically

http://dx.doi.org/10.21437/Interspeech.2021-2044

tokenised into sequences of word tokens. The output for slot filling is a sequence of slot labels, the length of which exactly matches the number of tokens in the input. For intent detection the output is a single intent label. The sets of distinct slot labels and intent labels are converted to numerical representations by mapping them to integers. Let n be the maximum input sequence length in the dataset. All input sequences are post-padded to length n with a [PAD] token. Slot label sequences are also padded to length n with a padding label. We use the pre-trained BERT-BASE model for numerical representation of the input sequences. The feature size $N = n + 2$ is fed to the model, to include the BERT [CLS] and [SEP] tokens.

3.2. NLU modelling layer

BERT gives a contextual, bi-directional representation of input tokens. These representations will be used to compute the initial intent probability and slot probabilities. The BERT model we used is "bert_uncased_L-12_H-768_A-12"; that is, using uncased tokens, with 12 encoder stacks, final hidden layer of dimension D_E = 768, and 12 attention heads. Each stack contains a multi-head attention module and a 2-layer feed forward neural network (FFNN). Within the attention layers there is dropout on attention with probability 0.1. The FFNN contains two dense, linear layers of size 3072 and 768 respectively. A GELU activation function is used for the first layer only. Dropout with probability 0.1 is applied. A residual connection of the attention output is added to the output of the second layer. This addition is normalized and becomes the input for the next encoder stack, and also the final output of the current encoder stack. The output shape is $[N, D_E] \in \mathbb{R}^{N \times D_E}$. We add one more dropout layer with dropout rate 0.1 on the encoder output to further mitigate potential overfitting.

3.3. Bi-directional NLU layer

On top of the NLU Modelling Layer, we propose the Bi-directional NLU layer that contains two models: intent2slot for slot filling and slot2intent for intent classification. The intent2slot draws on the intent probability distribution, utilising the semantic information of the whole sequence while labeling each slot. The slot2intent takes in the sequence label probability distributions as supplementary information for intent classification. To jointly model intent classification and slot filling, the model is trained on the joint loss of slot labeling and intent classification.

Intent2Slot model The intent2slot model aims to draw the intent probability by extracting the semantic information of the whole sequence and utilising it to aid detection of a slot label for each word. Let D_{IP} and D_{SP} be the number of distinct intent labels and slot labels respectively. Let the encoder output of size $[N,D_E]$ be denoted $(h_1, h_2, \ldots, h_N)$ where each h_i is the $D_E = 768$ dimension final hidden layer state of each token. Here h_1 is the encoding of the [CLS] token which is trained to classify the entire sequence. In order to utilise the intent probability P_I for slot filling we first take h_1 and pass it through a dense layer then apply a softmax:

$$P_I = \text{softmax}(h_1 \cdot W_I^T + b_I) \tag{1}$$

This is a dense layer with D_{IP} nodes. Hence $W_I \in \mathbb{R}^{D_{IP} \times D_E}$ is a matrix of weights and $b_I \in \mathbb{R}^{D_{IP}}$ is a vector of biases. $P_I \in \mathbb{R}^{D_{IP}}$ is thus a vector of probabilities. We next broadcast P_I to the length of sequence to be labelled:

$$\hat{P}_I = \text{repeat}(P_I), N-1 \text{ times} \tag{2}$$

Then for the intent2slot we take the hidden state sequence of the remaining $N-1$ tokens, each of dimension D_E:

$$H = (h_2, h_3, \ldots, h_N) \tag{3}$$

We concatenate each element of H with P_I and feed it into a softmax classifier to get the slot probability for each word:

$$S = (H \oplus \hat{P}_I) \cdot V_S^T + m_S \tag{4}$$

We have a matrix $S \in \mathbb{R}^{(N-1) \times D_{SP}}$. This represents $N-1$ vectors of size $[D_E + D_{IP}]$ each being a BERT encoding concatenated with the intent probability. For each one there is a dense layer of size D_{SP}. Hence $V_S \in \mathbb{R}^{(N-1) \times D_{SP} \times (D_E + D_{IP})}$ is an array of $N-1$ matrices of weights and $m_S \in \mathbb{R}^{(N-1) \times D_{SP}}$ is an array of $n-1$ vectors of biases. The $N-1$ linear outputs feed to $N-1$ softmaxes, each with an argmax to get a predicted slot label for each element of the input sequence.

$$S_n = \text{argmax}(\text{softmax}(S^n)), n \in \{2, \ldots, N\} \tag{5}$$

where S^n is the nth slice of S on the first axis.

Slot2Intent model The slot2intent model aims to take the sequence label probability distributions as additional information in intent detection. This can be viewed as a dual of the intent2slot model. Similarly to the intent2slot model, for intent classification we first take the final hidden state sequence of the tokens excluding the first special token [CLS], each of dimension D_E:

$$H = (h_2, h_3, \ldots, h_N) \tag{6}$$

For each $h_n \in H$ we have a dense layer with D_{SP} nodes. Then we use softmax to get the slot probabilities P_{S_n}:

$$P_{S_n} = \text{softmax}(h_n \cdot W_{S_n}^T + b_{S_n}), n \in \{2, \ldots, N\} \tag{7}$$

Here each $W_S \in \mathbb{R}^{D_{SP} \times D_E}$ is a matrix of weights and each $b_S \in \mathbb{R}^{D_{SP}}$ is a vector of biases. Each $P_{S_n} \in \mathbb{R}^{D_{SP}}$ is thus a vector of slot label probability distributions for a token of the input sequence after [CLS].

In order to fuse the slot probabilities with the intent hidden state, we flatten the slot probabilities as:

$$\hat{P}_S = \text{flatten}(P_{S_n}), n \in \{2, \ldots, N\} \tag{8}$$

This is a vector of length $D_{SP}(N-1)$.

In the slot2intent model, we concatenate the final hidden state of the first special token [CLS], h_1, with $\hat{P}_S$ and feed it into a softmax classifier to get the intent classification:

$$I = \text{argmax}(\text{softmax}((h_1 \oplus \hat{P}_S) \cdot V_I^T + m_I)) \tag{9}$$

Here $V_I \in \mathbb{R}^{D_{IP} \times (D_E + D_{SP}(N-1))}$ is a matrix of weights, and $m_I \in \mathbb{R}^{D_{IP}}$ is a vector of biases.

Joint optimisation To jointly model intent classification and slot filling, the model is trained on the sum of slot labeling loss L_S and of intent classification loss L_I (i.e. $L = L_I + L_S$). The intent loss for a sample is:

$$L_I = -\sum_{i \in \mathbb{I}} I(y_g = i) \log y_i \tag{10}$$

where $I(y_g = i)$ is the indicator function that the gold intent for the sample is i, y_i is the output from the softmax classifier

in Equation (9) corresponding to intent i for the sample. The slot loss for a sample is

$$L_S = -\sum_{n=2}^{N}\sum_{s\in\mathbb{S}} I(y_g^n = s)\log y_s^n \tag{11}$$

where $I(y_g^n = s)$ is the indicator function that token n has gold slot label s and y_s^n is the output from the softmax classifier in Equation (5) corresponding to intent s for token n of the sample.

4. Experiment

In this section, we first evaluate our bi-directional joint intent classification and slot filling model against the current state-of-the-art models. Then we conduct an ablation study to reveal the contributions from the input embedding layer, NLU modeling layer and the bi-directional NLU layer of our model to the final performance.

4.1. Datasets

We ran our experiments on two widely-used and publicly available benchmark datasets. ATIS is a single-domain, single utterance dataset. It consists of flight information query utterances. The dataset consists of 4,478 training, 500 validation, and 893 testing samples respectively. There are 120 slot labels and 21 intents. SNIPS[1] is collected from the SNIPS voice assistant. In the SNIPS dataset, the training set contains 13,084 utterances, the validation set contains 700 utterances, and another 700 utterances are used as the test set. There are 72 slot labels and 7 intent types. The dataset is more complicated than the ATIS since it contains more than one domain and a large vocabulary.

4.2. Settings

The BERT model we used is "bert_uncased_L-12_H-768_A-12". We used 0.1 dropout rate and initialize the weights with standard deviation of 0.02. We set the maximum sequence length (including special tokens) to 50 and train our model on the training set of the two datasets. For all experiments, the learning rate is 5e-5 and the optimizer is the same Adam optimizer used for BERT pre-training. We applied 10 epochs to ATIS and 20 to SNIPS. We applied the same hyper-parameters one of our baselines [13].

Following the literature we measure intent accuracy, span based slot f1[2], and semantic accuracy wherein all slots and intent are correctly predicted for a positive result.

4.3. Experiment 1: Joint model overall evaluation

As shown in the Table 1 we achieved better performance on all tasks for both datasets. For ATIS, the intent accuracy and the sentence-level semantic accuracy are improved by 3.6% and 5.2% respectively compared to the highest from the previous non-BERT-based state-of-the-art model [12]. Comparatively, the slot F1 was improved by a smaller amount, being 1.1%. Similarly for SNIPS, the sentence-level semantic accuracy was increased by 11.9%, while the intent accuracy and slot filling F1 were improved by 1.9% and 5.4% respectively. We also found that our proposed model out-performs BERT-based joint models, including Joint BERT [13] and Stack-Propagation with BERT[14]. Note that Joint BERT is a pre-print and the performance is not stable based on different learning rate, and Stack-Propagation-BERT set the default epoch as 300.

Table 1: *Natural language understanding (NLU) performance on ATIS and SNIPS-NLU datasets (%).*

Model	ATIS (10 epoch)			SNIPS (20 epoch)		
	Slot (f1)	Intent (acc)	Sent. (acc)	Slot (f1)	Intent (acc)	Sent. (acc)
Joint Seq [3]	94.3	92.6	80.7	87.3	96.9	73.2
Attn. [10]	94.2	91.1	78.9	87.8	96.7	74.1
S-Gated F. [11]	94.8	93.6	82.2	88.8	97.0	75.5
S-Gated I. [11]	95.2	94.1	82.6	88.3	96.8	74.6
Capsule NN [12]	95.2	95.0	83.4	91.8	97.3	80.9
Joint Bert [13]	96.1	97.5	88.2	97.0	98.6	92.8
Stack-Prop.[14]	96.1	97.5	88.6	97.0	99.0	92.9
Our Model	**96.3**	**98.6**	**88.6**	**97.2**	**99.2**	92.8

We found that the proposed bi-directional contextual contribution (slot2intent, intent2slot) is effective and outperformed baseline models. For the ATIS dataset, the intent label distribution is imbalanced, ranging from a highest frequency of 3309 samples for *atis_flight* to a lowest of 1 for *atis_cheapest*. This may negatively impact the overall intent classification performance. In addition, the number of slots in ATIS (120) is significantly larger than SNIPS (72) and those slots in ATIS only cover one domain which results in a high number of slots being mutually shared by multiple intents. For example, the *atis_flight* intent contains 66 slot types, 52 of which are repeated in other intents. The efficacy of the contextual clues provided by slot2intent flow in our proposed model might be undermined by such overlapping slots across different intents. Despite all this, utilizing those slots unique to specific intents still improves the intent classification performance, as is verified by the results. On the other hand, the slot filling F1 score improved by a relatively smaller amount. This may be related to the longer sequence length and larger number of slots to be labelled for a single intent in ATIS. However, the improvement can still validate the effectiveness of the intent2slot flow, and indicates that providing the intent-related contextual information may help with narrowing down from the large slot pool to much smaller candidate groups.

Compared to ATIS, SNIPS is more complex regarding its cross-domain topics. For example, intents *BookRestaurant*, *GetWeather* and *PlayMusic* are from different domains. Due to this, the vocabulary size is much larger than in ATIS. The diversity of vocabulary in SNIPS for each intent provides useful information for intent classification and its intent distribution in the training dataset is well balanced. These factors point to why the intent classification accuracy in SNIPS is relatively higher than in ATIS. In addition to this, SNIPS has relatively small number of overlapping slots (only 11 slots are mutually shared with other intents, while ATIS has 79 such slots). With less overlapping slots, our slot2intent flow further provides discriminated contextual information regarding slots, resulting in a high accuracy of 99.2% for intent classification.

On the other hand, SNIPS has only 72 slots for the 7 intents. For a single intent, the maximum slot type count is 15 (only considering slots starting with '*B-*') for *BookRestaurant* while the minimum is only 2 for *SearchCreativeWork*. With the

[1] https://github.com/snipsco/nlu-benchmark/

[2] The Python implementation of *conlleval script* is used for F1-score: https://pypi.org/project/seqeval/

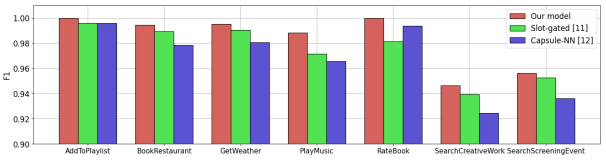

Figure 1: *Breakdown of intent detection performance of the proposed model against previous state-of-the-art models on SNIPS*

intent2slot flow providing topic-related clues for slot filling, our proposed model improves the slot filling performance. Compared to the ATIS result, this improvement is a little bit lower. This could be caused by ambiguity of slot value in SNIPS. For example, the location entity (e.g. 'Sydney') labelled as slot value for intent *flight_time* is quite obvious in ATIS because most of the location entities are labelled as CITY. However, the location entity labelled as a slot value for intent *GetWeather* in SNIPS can be labelled as CITY, STATE, or COUNTRY. This might be confusing to the machine when appearing alone without using an explicit dictionary.

4.4. Experiment 2: Intent classification evaluation

In order to further analyse the intent classification performance, we replicated the two most successful baseline models [11, 12] and trained them on the SNIPS dataset with 20 epochs, the same as for our proposed model. We measured the intent detection F1 score for each type of intent. As illustrated in Figure 1, we observe a similar trend for intent F1 scores among all models. It is clear that *SearchCreativeWork* and *SearchScreeningEvent* always get significantly lower F1 than the other intents, which are uniformly high. We investigated the slot distributions and found that both the lower performing intents tend to have non-discriminative slots in the training datasets. For example, *SearchCreativeWork* only has two slot types (starting with '*B-*'), both of which also appear frequently in utterances of other intents. Similarly, *SearchScreenEvent* has 3 out of its 7 slot types repeating in other intents. Further, one of the only two slot types for *SearchCreativeWork* also appears as a slot type for *SearchScreenEvent*. In addition, these two intents may actually be about the same topic when the creative work is a movie. Thus confusion may occur. After we further looked at the classification results of the three models, we found a similar pattern: most of the *SearchScreenEvent* were misclassified as *SearchCreativeWork* and the exact case rates of *SearchScreenEvent* misclassified as *SearchCreativeWork* are 11 out of 14 for Capsule-NN, 5 out of 7 for Slot-gated and 5 out of 5 for our proposed model. This causes the negative impact on precision and recall for the two intents in joint models.

Another notable point is that only for intent *RateBook* does Capsule-NN outperform Slot-gated model. This might be related to the relatively short length and regular pattern of the *RateBook* utterance. For instance, most of the utterances are similar to "Give/Rate [1-9] point/star to Bookname". This may lead to better slot filling performance. Additionally, *RateBook* has a higher proportion of unique slots compared to most of the other intents. Given that, providing the slot-related clue, as Capsule-NN does, may contribute to better intent classification compared to other intents. This may also help explain the highest accuracy for *RateBook* for our proposed model. It can be seen that our proposed model achieved better F1 for all intents, which validates the effectiveness of slot2intent as well as the mutual augmentation through bi-directional contextual flow.

Table 2: *Ablation study results on ATIS*

NLU Modeling	Embedding	Bidirectional NLU	Slot (f1)	Intent (acc)	Sent. (acc)
LSTM	GloVe	X	50.4	70.8	23.6
	GloVe	**O**	**51.4**	**72.2**	**25.2**
	word2vec	X	40.4	70.8	13.8
	word2vec	**O**	**44.5**	**72.0**	**16.3**
GRU	GloVe	X	54.5	70.8	25.5
	GloVe	**O**	**58.8**	**72.6**	**28.4**
	word2vec	X	45.0	70.8	18.1
	word2vec	**O**	**54.1**	**71.9**	**25.5**
BERT	BERT	X	94.8	96.2	86.1
	BERT	**O**	**96.3**	**98.6**	**88.6**

4.5. Experiment 3: Ablation study

In order to investigate the effect of the input embedding layer, NLU modeling layer and the bidirectional NLU layer, we also report ablation study results in Table 2 on the ATIS dataset. It can be seen that we replace the input embedding layer from BERT to GloVe or word2vec, the NLU Modelling layer from BERT to Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU). We also have each replaced input embedding layer and NLU Modelling Layer combinations with or without the bidirectional NLU layer. This aims to test the effectiveness of using mutual information in the model.

The results show that the combination 'BERT NLU modeling + BERT embedding + Bidirectional NLU layer' outperformed all other combinations by a large margin on both datasets. By changing the BERT NLU modeling and BERT embedding to other combinations of GRU/LSTM with GloVe/word2vec, it can be seen that the performance regarding all the three metrics on both datasets decrease dramatically, without regard to the use of the bidirectional NLU layer. This indicates that BERT helps with the two tasks in terms of both word embedding and NLU modeling aspects. Besides, it can be observed through comparing the model pairs in column 'Bidirectional NLU' that applying the bidirectional NLU contributes to overall performance improvement for all models. This validates the effectiveness and value of the bidirectional mutual information sharing for the joint task. This improvement is especially significant for GloVe and word2vec models, neither of which capture contextual information to the depth that BERT does.

5. Conclusion

In this paper, we propose a new and effective joint intent detection and slot filling model which integrates deep contextual embeddings and the transformer architecture. We further propose a Bi-directional NLU layer that contains two models: intent2slot for slot filling and slot2intent for intent classification. The model was applied to two real-world datasets and outperformed previous state-of-the-art results, using the same evaluation measurements: in intent detection, slot filling and semantic accuracy. The results indicate that our approach was more effective in capturing global interrelationships between the semantic components in the NLU tasks than existing joint models.

6. References

[1] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 78–83.

[2] D. Guo, G. Tur, W.-t. Yih, and G. Zweig, "Joint semantic utterance classification and slot filling with recursive neural networks," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 554–559.

[3] D. Hakkani-Tür, G. Tür, A. Celikyilmaz, Y.-N. Chen, J. Gao, L. Deng, and Y.-Y. Wang, "Multi-domain joint semantic frame parsing using bi-directional RNN-LSTM," in *Interspeech*. ISCA, 2016, pp. 715–719.

[4] Z. Zhao and Y. Wu, "Attention-based convolutional neural networks for sentence classification," in *Interspeech*. ISCA, 2016, pp. 705–709.

[5] C. Zhang, W. Fan, N. Du, and P. S. Yu, "Mining user intentions from medical queries: A neural network based heterogeneous jointly modeling approach," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 1373–1384.

[6] C. Xia, C. Zhang, X. Yan, Y. Chang, and P. Yu, "Zero-shot user intent detection via capsule neural networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3090–3099.

[7] Z. Zhang, Z. Zhang, H. Chen, and Z. Zhang, "A joint learning framework with bert for spoken language understanding," *IEEE Access*, vol. 7, pp. 168 849–168 858, 2019.

[8] H. Tang, D. Ji, and Q. Zhou, "End-to-end masked graph-based crf for joint slot filling and intent detection," *Neurocomputing*, 2020.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. 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, pp. 4171–4186.

[10] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *Interspeech*, pp. 685–689, 2016.

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

[12] C. Zhang, Y. Li, N. Du, W. Fan, and P. Yu, "Joint slot filling and intent detection via capsule neural networks," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5259–5267. [Online]. Available: https://www.aclweb.org/anthology/P19-1519

[13] Q. Chen, Z. Zhuo, and W. Wang, "BERT for joint intent classification and slot filling," 2019. [Online]. Available: http://arxiv.org/abs/1902.10909

[14] L. Qin, W. Che, Y. Li, H. Wen, and T. 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, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Hong Kong: Association for Computational Linguistics, 2019, pp. 2078–2087. [Online]. Available: https://doi.org/10.18653/v1/D19-1214