



Jointly Encoding Word Confusion Network and Dialogue Context with BERT for Spoken Language Understanding

Chen Liu^{*}, Su Zhu^{*}, Zijian Zhao, Ruisheng Cao, Lu Chen[†] and Kai Yu[†]

MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai

{chris-chen, paul2204, 1248uu, 211314, chenlusz, kai.yu}@sjtu.edu.cn

Abstract

Spoken Language Understanding (SLU) converts hypotheses from automatic speech recognizer (ASR) into structured semantic representations. ASR recognition errors can severely degenerate the performance of the subsequent SLU module. To address this issue, word confusion networks (WCNs) have been used as the input for SLU, which contain richer information than 1-best or n-best hypotheses list. To further eliminate ambiguity, the last system act of dialogue context is also utilized as additional input. In this paper, a novel BERT based SLU model (WCN-BERT SLU) is proposed to encode WCNs and the dialogue context jointly. It can integrate both structural information and ASR posterior probabilities of WCNs in the BERT architecture. Experiments on DSTC2, a benchmark of SLU, show that the proposed method is effective and can outperform previous state-of-the-art models significantly.

Index Terms: spoken language understanding, word confusion network, dialogue context, Transformer, BERT

1. Introduction

The spoken language understanding (SLU) module is a key component of spoken dialogue system (SDS), parsing user utterances into corresponding semantic representations (e.g., dialogue acts [1]). For example, the utterance “*I want a high priced restaurant which serves Chinese food*” can be parsed into a set of semantic tuples “*inform(price_range=expensive), inform(food=Chinese)*”. In this paper, we focus on SLU with semantic labels in the form of *act(slot=value)* triplets (i.e., unaligned annotations), which does not require word by word annotations. Both discriminative [2, 3, 4, 5] and generative [6, 7] methods have been developed to extract semantics from ASR hypotheses of the user utterance.

SLU systems trained on manual transcripts would get a dramatic decrease in performance when applied to ASR hypotheses [8]. To eliminate ambiguity caused by ASR errors, two kinds of input features can be exploited to enhance SLU models: (1) ASR hypotheses and (2) dialogue context information. 1) Considering the uncertainty of ASR hypotheses, previous works utilized ASR 1-best result [3, 7, 9, 10], N-best lists [3, 11, 12], word lattices [13, 14, 15] and word confusion networks (WCNs) [16, 3, 17, 18, 19, 20] for inputs to train an SLU

model. Masumura et al. [20] proposed a fully neural network based method, *neural ConfNet classification*, to encode WCNs. It first obtains bin (each bin contains multiple word candidates and their posterior probabilities of ASR hypothesis in the same time step) vectors by the weighted sum of all word embeddings in each bin separately, and then exploits a bidirectional long short-term memory recurrent neural network (BLSTM-RNN) to integrate all bin vectors into an utterance vector. Nevertheless, bin vectors are extracted locally, ignoring contextual features beyond certain bins. 2) Furthermore, the last system dialogue act [1] can be utilized to track context [3, 5], and provide some implications about the user intent under noisy conditions. However, utterance and context are independently encoded by different models to generate the final representation, resulting in a lack of interaction between them.

Recently, pre-trained language models, such as GPT [21] and BERT [22], have been successfully adopted in various NLP tasks. Huang et al. [15] adapted GPT for modeling word lattices, where lattices are represented as directed acyclic graphs. However, GPT is modeled as a unidirectional Transformer [23] and neglects context in the future, thus less expressive than BERT. Although both word lattices and WCNs contain more information than N-best lists, WCNs have been proven more efficient in terms of size and structure [16].

To these ends, we propose a novel BERT (Bidirectional Encoder Representations from Transformers) [22] based SLU model to jointly encode WCNs and system acts, which is named WCN-BERT SLU. It consists of three parts: a BERT encoder for jointly encoding, an utterance representation model, and an output layer for predicting semantic tuples. The BERT encoder exploits posterior probabilities of word candidates in WCNs to inject ASR confidences. Multi-head self-attention is applied over both WCNs and system acts to learn context-aware hidden states. The utterance representation model produces an utterance-level vector by aggregating final hidden vectors. Finally, we add both discriminative and generative output layers to predict semantic tuples. To the best of our knowledge, this is the first work to leverage the structure and probabilities of input tokens in BERT. Our method is evaluated on DSTC2 dataset [24], and the experimental results show that our method can outperform previous state-of-the-art models significantly.

2. WCN-BERT SLU

In this section, we will describe the details of the proposed framework, as shown in Fig. 1. For each user turn, the model takes the corresponding WCN and the last system act as input, and predicts the semantic tuples at turn level.

^{*} equal contribution

[†] corresponding authors

We thank the anonymous reviewers for their thoughtful comments. This work has been supported by the National Key Research and Development Program of China (Grant No. 2017YFB1002102) and Shanghai Jiao Tong University Scientific and Technological Innovation Funds (YG2020YQ01).

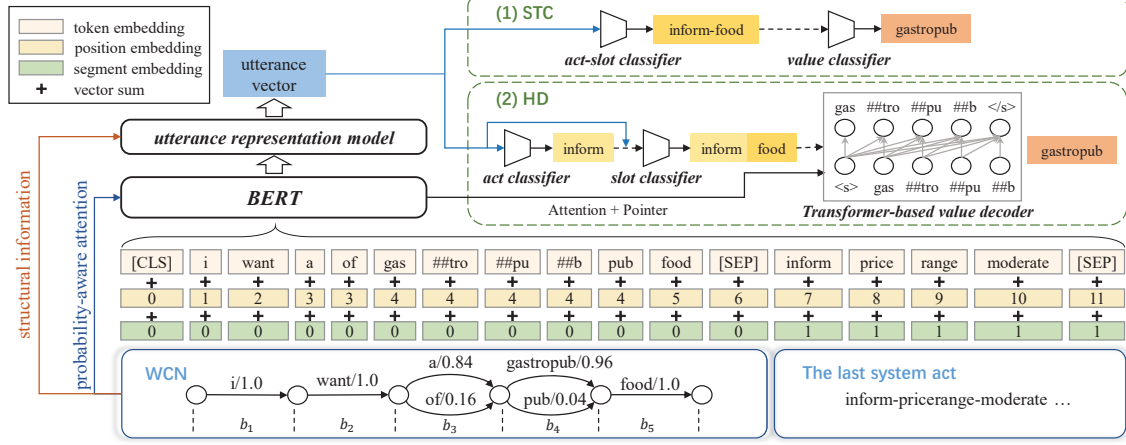


Figure 1: An overview of our proposed WCN-BERT SLU, which contains a BERT encoder, an utterance representation model, and an output layer. First, the WCN and the last system act are arranged as a sequence to be fed into the BERT encoder. The token-level BERT outputs are then integrated into an utterance-level vector representation. Finally, either a discriminative (semantic tuple classifier, STC) or generative (hierarchical decoder, HD) approach is utilized in the output layer for predicting the act-slot-value triplets.

2.1. Input representation

The WCN is a compact lattice structure where candidate words paired with their associated posterior probabilities are aligned at each position [25]. It is commonly considered as a sequence of word bins $B = (b_1, \dots, b_M)$. The m -th bin can be formalized as $b_m = \{(w_m^1, P(w_m^1)), \dots, (w_m^{I_m}, P(w_m^{I_m}))\}$, where I_m denotes the number of candidates in b_m , w_m^i and $P(w_m^i)$ are respectively the i -th candidate and its posterior probability given by ASR system. The WCN is flattened into a word sequence, $w^{\text{WCN}} = (w_1^1, \dots, w_1^{I_1}, \dots, w_M^1, \dots, w_M^{I_M})$.

System acts contain dialogue context in the form of “act-slot-value” triplets. We consider the last system act just before the current turn, $A = ((a_1, s_1, v_1), \dots, (a_K, s_K, v_K))$, where K is the number of triplets, a_k , s_k and v_k are the k -th act, slot and value, respectively. It is also arranged as a sequence $w^{\text{SA}} = (a_1, s_1, v_1, \dots, a_K, s_K, v_K)$. Act and slot names are tokenized, e.g., “pricerange” is split into “price” and “range”.

To feed WCNs and system acts into BERT together, we denote the input token sequence as $w = (w_1, \dots, w_T) = [\text{CLS}] \oplus \text{Tok}(w^{\text{WCN}}) \oplus [\text{SEP}] \oplus \text{Tok}(w^{\text{SA}}) \oplus [\text{SEP}]$, where \oplus concatenates sequences together, $\text{Tok}(\cdot)$ is the BERT tokenizer which tokenizes words into sub-words, $[\text{CLS}]$ and $[\text{SEP}]$ are auxiliary tokens for separation, $T = |\text{Tok}(w^{\text{WCN}})| + |\text{Tok}(w^{\text{SA}})| + 3$. Considering the structural characteristics of WCN, i.e., multiple words compete in a bin, we define that all words (in fact sub-words after tokenization) in the same bin share the same position ID.

Eventually, the BERT input layer embeds w into d_x -dimensional continuous representations by summarizing three embeddings [22] as follows:

$$x_t = E_{\text{token}}(w_t) + E_{\text{pos}}(w_t) + E_{\text{seg}}(w_t), x_t \in \mathbb{R}^{d_x} \quad (1)$$

where $E_{\text{token}}(\cdot)$, $E_{\text{pos}}(\cdot)$ and $E_{\text{seg}}(\cdot)$ stand for WordPiece [26], positional and segment embeddings, respectively.

2.2. BERT encoder

BERT consists a series of bidirectional Transformer encoder layers [23], each of which contains a multi-head self-attention module and a feed-forward network with residual connections [27]. For the l -th layer, assuming the input is $x^l =$

(x_1^l, \dots, x_T^l) , the output is computed via the self-attention layer (consisting of H heads):

$$e_{ij}^{l,h} = \frac{(W_Q^{l,h} x_i^l)^\top (W_K^{l,h} x_j^l)}{\sqrt{d_x/H}}; \quad \alpha_{ij}^{l,h} = \frac{\exp(e_{ij}^{l,h})}{\sum_{k=1}^T \exp(e_{ik}^{l,h})}$$

$$z_i^{l,h} = \sum_{j=1}^T \alpha_{ij}^{l,h} (W_V^{l,h} x_j^l); \quad z^l = \text{Concat}(z_i^{l,1}, \dots, z_i^{l,H})$$

$$\tilde{x}_i^{l+1} = \text{LayerNorm}(x_i^l + \text{FC}(z_i^l))$$

$$x_i^{l+1} = \text{LayerNorm}(\tilde{x}_i^{l+1} + \text{FC}(\text{ReLU}(\text{FC}(\tilde{x}_i^{l+1}))))$$

where FC is a fully-connected layer, LayerNorm denotes layer normalization [28], $1 \leq h \leq H$, $W_Q^{l,h}, W_K^{l,h}, W_V^{l,h} \in \mathbb{R}^{(d_x/H) \times d_x}$.

WCN probability-aware self-attention We propose an extension to the self-attention mechanism to consider the posterior probabilities of tokens in WCN. Following the notations in section 2.1, for each input token sequence $w = (w_1, \dots, w_T)$, we define its corresponding probability sequence as $p = (p_1, \dots, p_T)$, with

$$p_t = \begin{cases} P(w_t) & w_t \in \text{Tok}(w^{\text{WCN}}) \\ 1.0 & \text{otherwise} \end{cases} \quad (2)$$

where $P(w_t)$ denotes the ASR posterior probability of a token. Note that the probability of a sub-word equals that of the original word in WCN. Probabilities of tokens in the system acts are defined as 1.0. Now we inject ASR posterior probabilities into the BERT encoder by changing the computation of $e_{ij}^{l,h}$:

$$e_{ij}^{l,h} = (W_Q^{l,h} x_i^l)^\top (W_K^{l,h} x_j^l) / \sqrt{d_x/H} + \lambda^{l,h} \cdot p_j \quad (3)$$

where $\lambda^{l,h}$ is a trainable parameter.

Finally, token-level representations are produced after the stacked encoder layers, denoted as $o = (o_1, \dots, o_T)$.

2.3. Utterance representation

The output hidden vector corresponding to the $[\text{CLS}]$ token, i.e., o_1 , is usually applied to represent the whole utterance. Additionally, we propose to gather the other hidden vectors by considering the structural information of the WCN.

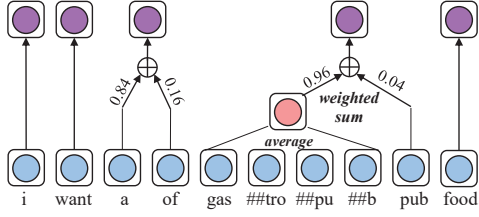


Figure 2: Illustration of the feature aggregation. Only the WCN part of an input token sequence is presented. The token-level features are in blue while bin-level features are in purple.

Firstly, the token-level hidden vectors of the WCN part are aggregated into bin-level through the following two steps: (1) BERT sub-word vectors belong to an input word are averaged to be the word vector; (2) word-level vectors of each bin are then weighted and summed to get a bin vector. An example of the feature aggregation of the WCN part is illustrated in Fig. 2, while the features corresponding to the system acts are unchanged. After aggregation, we get new feature vectors $u = (u_1, \dots, u_{T'})$, where $T' = M + |\text{Tok}(w^{SA})| + 3$ (M is the number of bins) and $T' \leq T$.

Then, we summarize the bin-level features with a self-attentive approach as follows:

$$u' = \sum_{t=2}^{T'} \left[\frac{\exp(\bar{v}^\top \tanh(W_a u_t + b_a))}{\sum_{j=2}^{T'} \exp(\bar{v}^\top \tanh(W_a u_j + b_a))} \cdot u_t \right] \quad (4)$$

where $b_a, \bar{v} \in \mathbb{R}^{d_x}$ and $W_a \in \mathbb{R}^{d_x \times d_x}$ are trainable parameters. The final utterance representation is obtained by concatenating u' with the hidden state of [CLS], i.e., $r = \text{Concat}(u_1, u')$.

2.4. Output layer

To evaluate the validity and portability of the encoding model, we apply both discriminative (Sec. 2.4.1) and generative (Sec. 2.4.2) approaches for predicting the act-slot-value labels.

2.4.1. Semantic tuple classifier (STC)

Upon the final utterance representation r , we apply a binary classifier for predicting the existence of each act-slot pair, and a multi-class classifier over all possible values existing in the training set for each act-slot pair¹. Therefore, this method cannot predict values unseen in the training set.

2.4.2. Transformer-based hierarchical decoder (HD)

To improve the generalization capability of value prediction, we follow Zhao et al. [7] to construct a hierarchical decoder consisting of an act classifier, a slot classifier, and a value generator. However, there are two main differences listed as follows:

1) Acts and slots are tokenized and then embedded by the BERT embedding layer as additional features for the slot classifier and the value generator. For each act a or slot s , sub-word level vectors from BERT embedding layer are averaged to get a single feature vector (e_a for a and e_s for s).

2) We replace the LSTM-based value generator with a Transformer-based one [23]. Tokenized values are embedded by the BERT embedding layer and generated at the sub-word

¹For act-slot pairs requiring no value (like *thankyou*, *goodbye*, *request-phone*, *request-food* etc.), the value classification is omitted.

level. Therefore, we tie the BERT’s token embeddings with the weight matrix of the linear output layer in the value generator.

3. Experiments

We experiment on the dataset from the second Dialog State Tracking Challenge (DSTC2) [24], which contains 11677, 3934, 9890 utterances for training, validation and testing, respectively. In order to shorten flattened WCN sequences, we prune WCNs by removing interjections² and word candidates with probabilities below a certain threshold (0.001 as recommended [19]). The evaluation metrics are F_1 score of *act-slot-value* triplets and utterance-level accuracy. We do not assume that all value candidates of each slot are known in advance.

In our experiments, we use English uncased BERT-base model, which has 12 layers of 768 hidden units, and 12 attention heads. During training, Adam [29] is used for optimization. We select the initial learning rate from {5e-5, 3e-5, 2e-5}, with a warm-up rate of 0.1 and an L2 weight decay of 0.01. The maximum norm for gradient clipping is set to 1. The dropout rate is set to 0.1 for the BERT encoder and 0.3 for the utterance representation model and the output layer. The model is trained for 50 epochs and saved according to the best performance on the validation set. Each experimental setting was run five times with different random seeds, and we report the average result.³

3.1. Main results

Table 1: F_1 scores (%) and utterance-level accuracies (%) of baseline models and our proposed model on the test set.

Models		Train→Test	F_1	Acc.
BLSTM+ Self-Attn	STC	manual→manual	98.56	97.29
		manual→1-best	83.57	74.91
		1-best→1-best	84.06	75.26
		1-best→10-best	84.92	75.70
		10-best→10-best	85.05	77.07
Neural ConfNet	STC	WCN→WCN	85.01	77.03
Lattice-SLU	STC	WCN→WCN	86.09	78.78
WCN-BERT (ours)	STC	WCN→WCN	87.91	81.14
	HD	WCN→WCN	87.33	80.74

As shown in Table 1, different types of inputs are applied for baselines, including *manual* transcriptions, ASR *1-best*, *10-best* lists, and *WCNs*. The baseline model for the first three types (*manual*, *1-best*, and *10-best*) is BLSTM with self-attention [20].⁴ BLSTM encodes the input sequence and gets the utterance representation r with the self-attention similar to Eq. (4). To test on *10-best* lists with the model trained on *1-best*, we run the model on each hypothesis from the list and average the results weighted by the ASR posterior probabilities. For direct training and evaluation on *10-best*s, the representation vector r is calculated as $r = \sum_{i=1}^{10} \gamma_i r_i$, where r_i is the representation vector of hypothesis i and γ_i is the corresponding ASR posterior probability.

For modeling WCNs, we follow the *Neural ConfNet Classification* [20] method. WCNs are fed into the model through

²Such as *uh*, *oh*, etc. according to Jagfeld et al. [19].

³The code is available at github.com/simple/WCN-BERT.

⁴In these baselines, word embedding is initialized with 100-dim GloVe6B [30]. The learning rate is set to 0.001, fixed during training. The maximum norm for gradient clipping is 5, and dropout rate is 0.3.

a simple weighted sum representation method, where all word vectors are weighted by their posterior probabilities and then summed. We also apply the *Lattice-SLU* method with GPT [15] on WCNs. The output layer of all baselines is STC (Sec. 2.4.1).

By comparing across the baselines, we find that performances become better with larger ASR hypotheses space. The *Neural ConfNet Classification* method can outperform the system trained and tested with 1-best, and it achieves comparable results to the 10-best system. With powerful pre-trained language models (GPT [21]), the *Lattice-SLU* beats the other baselines above. The last system act is not exploited in the baselines, which will be analyzed in the following ablation study.

Table 2: Comparison with prior arts on the DSTC2 dataset.

Models	F ₁ (%)
SLU2 [31]	82.1
CNN+LSTM_w4 [5]	83.6
CNN [6]	85.3
S2S-ATTN-PTR-NET [6]	85.8
HIERARCHICAL DECODING [7]	86.9
WCN-BERT + STC (ours)	87.9

By joint modeling WCNs and the last system act with powerful pre-trained BERT, our proposed framework outperforms the baselines significantly in F₁ score and utterance-level accuracy, and achieves new state-of-the-art performance on the DSTC2 dataset, as shown in Table 1 and Table 2.

3.2. Ablation study

In this section, we perform ablation experiments of our WCN-BERT SLU with both STC and HD, as presented in Table 3.

Table 3: F₁ scores (%) of ablation study on the test set.

Model variations	STC	HD
(a) WCN-BERT	87.91	87.33
(b) w/o WCN Prob	87.15	86.47
(c) $r = u_1$	87.75	86.99
(d) w/o BERT	86.12	86.01
(e) w/o system act	86.69	86.18
(f) w/o BERT and system act	85.85	85.06
(g) Replace system act with system utterance	88.01	87.28

Row (b) shows the results without considering WCN probabilities in the BERT encoder. In this case, the model lacks prior knowledge of ASR confidences, resulting in a significant performance drop (0.76% for STC and 0.86% for HD). By only considering the hidden state related to [CLS] as the utterance representation (row(c)), i.e., $r = u_1$, the F₁ scores decrease with both STC and HD, indicating that the structural information of WCN is beneficial for utterance representation.

By removing BERT (row (d)), we utilize vanilla Transformer to jointly encode WCNs and system acts but not fine-tune a pre-trained BERT. We use 100-dim word embeddings, initialized with Glove6B [30]. The results show that removing BERT brings about a remarkable decrease in F₁ score (1.79% for STC and 1.32% for HD).

Besides, we investigate the effect of dialogue context by removing the last system act from the input, as demonstrated in row (e). Results show that jointly encoding the WCN and the last system act improves the performance dramatically. It is a

fair comparison with the *Lattice-SLU* baseline in Table 1, both with pre-trained language models. The result implies that our model is much more effective, owing to (1) the capability of the bidirectional Transformer, which considers the future context, and (2) the different considerations of WCN structures.

In row (f), neither BERT nor the dialogue context is included. This also gives a fair comparison with the *Neural ConfNet Classification* method [20] at the model level (Transformer v.s. BLSTM). With STC as the output layer, the Transformer with WCN probability-aware self-attention mechanism is shown to have better modeling capability (an improvement of 0.84% in F₁) and higher training efficiency (over 6× faster).

Moreover, we replace the last system act with the last system utterance in the input (row (g)), which only causes a slight change (within 0.1%) in F₁ scores. This indicates that our model is also applicable to datasets with only system utterances, but no system acts.

3.3. Analysis of hierarchical decoder

As we can see from the previous results, models with hierarchical decoder (HD) perform worse than STC. However, the generative approach equips the model with generalization capability, and the pointer-generator network is beneficial to handle out-of-vocabulary (OOV) tokens. To analysis the generalization capability of the proposed model with hierarchical decoders, we randomly select a certain proportion of the training set to train our proposed WCN-BERT SLU with STC or HD. The validation and test set remain unchanged. Furthermore, we evaluate the F₁ scores of *seen* and *unseen* act-slot-value triplets, according to whether a triplet is seen in the training set.

Table 4: F₁ scores (%) of our WCN-BERT SLU on the test set with varying training size.

Train size	Models	overall	seen	unseen
1%	+ STC	62.9	67.1	0.0
	+ HD	71.6	77.0	29.7
5%	+ STC	78.7	80.2	0.0
	+ HD	81.5	83.1	35.3
10%	+ STC	81.3	81.9	0.0
	+ HD	82.9	83.8	22.9
100%	+ STC	87.9	88.2	0.0
	+ HD	87.3	87.5	4.8

As shown in Table 4, with the training size getting decreased, the overall performance of HD will not degrade sharply. Moreover, the hierarchical decoder is shown to have better generalization capability in the face of unseen labels.

4. Conclusion

In this paper, we propose to jointly encode WCN and dialogue context with BERT for SLU. To eliminate ambiguity caused by ASR errors, WCNs are utilized for involving ASR hypotheses uncertainties, and dialogue context implied by the last system act is exploited as auxiliary features. In addition, the pre-trained language model BERT is introduced to better encode WCNs and system acts with self-attention. Experimental results show that our method can beat all baselines and achieves new state-of-the-art performance on DSTC2 dataset. Except for encoding different ASR hypotheses, there are several studies about ASR error correction for SLU [32, 33], which will be our future work.

5. References

- [1] S. Young, "CUED standard dialogue acts," *Report, Cambridge University Engineering Department, 14th October, 2007*.
- [2] F. Mairesse, M. Gasic, F. Jurcicek, S. Keizer, B. Thomson, K. Yu, and S. Young, "Spoken language understanding from unaligned data using discriminative classification models," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 4749–4752.
- [3] M. Henderson, M. Gašić, B. Thomson, P. Tsiakoulis, K. Yu, and S. Young, "Discriminative spoken language understanding using word confusion networks," in *2012 IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 176–181.
- [4] S. Zhu, L. Chen, K. Sun, D. Zheng, and K. Yu, "Semantic parser enhancement for dialogue domain extension with little data," in *2014 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 336–341.
- [5] L. M. R. Barahona, M. Gasic, N. Mrkšić, P.-H. Su, S. Ultes, T.-H. Wen, and S. Young, "Exploiting sentence and context representations in deep neural models for spoken language understanding," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, 2016, pp. 258–267.
- [6] L. Zhao and Z. Feng, "Improving slot filling in spoken language understanding with joint pointer and attention," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 426–431.
- [7] Z. Zhao, S. Zhu, and K. Yu, "A hierarchical decoding model for spoken language understanding from unaligned data," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7305–7309.
- [8] G. Mesnil, X. He, L. Deng, and Y. Bengio, "Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding," in *Fourteenth Annual Conference of the International Speech Communication Association*, 2013, pp. 3771–3775.
- [9] S. Zhu, O. Lan, and K. Yu, "Robust spoken language understanding with unsupervised asr-error adaptation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6179–6183.
- [10] H. Li, C. Liu, S. Zhu, and K. Yu, "Robust spoken language understanding with acoustic and domain knowledge," in *2019 International Conference on Multimodal Interaction*, 2019, pp. 531–535.
- [11] J.-P. Robichaud, P. A. Crook, P. Xu, O. Z. Khan, and R. Sarikaya, "Hypotheses ranking for robust domain classification and tracking in dialogue systems," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [12] O. Z. Khan, J.-P. Robichaud, P. A. Crook, and R. Sarikaya, "Hypotheses ranking and state tracking for a multi-domain dialog system using multiple ASR alternates," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [13] J. Švec, L. Šmídl, T. Valenta, A. Chýlek, and P. Ircing, "Word-semantic lattices for spoken language understanding," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5266–5270.
- [14] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "LatticeRnn: Recurrent neural networks over lattices," in *Seventeenth Annual Conference of the International Speech Communication Association*, 2016, pp. 695–699.
- [15] C.-W. Huang and Y.-N. Chen, "Adapting pretrained transformer to lattices for spoken language understanding," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 845–852.
- [16] D. Hakkani-Tür, F. Béchet, G. Riccardi, and G. Tur, "Beyond asr 1-best: Using word confusion networks in spoken language understanding," *Computer Speech & Language*, vol. 20, no. 4, pp. 495–514, 2006.
- [17] G. Tür, A. Deoras, and D. Hakkani-Tür, "Semantic parsing using word confusion networks with conditional random fields," in *Fourteenth Annual Conference of the International Speech Communication Association*, 2013, pp. 2579–2583.
- [18] X. Yang and J. Liu, "Using word confusion networks for slot filling in spoken language understanding," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
- [19] G. Jagfeld and N. T. Vu, "Encoding word confusion networks with recurrent neural networks for dialog state tracking," in *Proceedings of the Workshop on Speech-Centric Natural Language Processing*, 2017, pp. 10–17.
- [20] R. Masumura, Y. Ijima, T. Asami, H. Masataki, and R. Higashinaka, "Neural confnet classification: Fully neural network based spoken utterance classification using word confusion networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6039–6043.
- [21] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf, 2018.
- [22] 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*, 2019, pp. 4171–4186.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [24] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, 2014, pp. 263–272.
- [25] J. Feng and S. Bangalore, "Effects of word confusion networks on voice search," in *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 2009, pp. 238–245.
- [26] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [28] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.
- [29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [30] 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.
- [31] J. D. Williams, "Web-style ranking and slu combination for dialog state tracking," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 282–291.
- [32] R. Schumann and P. Angkititrakul, "Incorporating asr errors with attention-based, jointly trained rnn for intent detection and slot filling," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6059–6063.
- [33] Y. Weng, S. S. Miryala, C. Khatri, R. Wang, H. Zheng, P. Molino, M. Namazifar, A. Papangelis, H. Williams, F. Bell *et al.*, "Joint contextual modeling for asr correction and language understanding," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6349–6353.