ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-6654-0540-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICASSP43922.2022.9747477

SLIM: EXPLICIT SLOT-INTENT MAPPING WITH BERT FOR JOINT MULTI-INTENT DETECTION AND SLOT FILLING

Fengyu Cai[1] *Wanhao Zhou*[1] *Fei Mi*[2] *Boi Faltings*[1]

[1] Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
[2] Huawei Noah's Ark Lab, Shenzhen, China

ABSTRACT

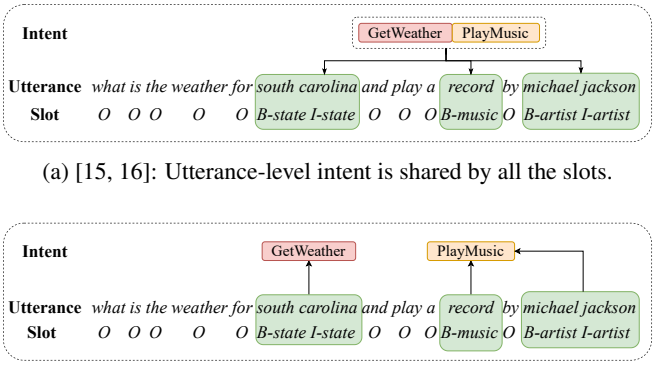
Utterance-level intent detection and token-level slot filling are two key tasks for spoken language understanding (SLU) in task-oriented systems. Most existing approaches assume that only a single intent exists in an utterance. However, there are often multiple intents within an utterance in real-life scenarios. In this paper, we propose a multi-intent SLU framework, called SLIM, to jointly learn multi-intent detection and slot filling based on BERT. To fully exploit the existing annotation data and capture the interactions between slots and intents, SLIM introduces an explicit slot-intent classifier to learn the many-to-one mapping between slots and intents. Empirical results on three public multi-intent datasets demonstrate (1) the superior performance of SLIM compared to the current state-of-the-art for SLU with multiple intents and (2) the benefits obtained from the slot-intent classifier.

***Index Terms*—** Spoken Language Understanding, Multi-intent Classification, Slot Filling

1. INTRODUCTION

Spoken language understanding (SLU) is an essential task for task-oriented dialog (ToD) systems [1]. It contains two major sub-tasks, **intent detection (ID)** and **slot filling (SF)**. Take *"Listen to Westbam album Allergic on Google music"* as an example. The task of ID is to identify the intent ("*PlayMusic*") of the utterance. SF is a sequence labeling task to predict the slot for each token, which are `[O, O, B-artist, O, B-album, O, B-service, I-service]` using the "Inside-outside-begining" (IOB) tagging format [2]. Traditional techniques often tackle these two tasks separately [3, 4, 5]. Recently, models that learn these two tasks jointly achieve better performance by capturing semantic dependencies between slots and intents [6, 7, 8, 9, 10, 11, 12, 13].

The aforementioned methods assume only a single intent in an utterance. However, multiple intents often exist in one utterance in real-life scenarios [14, 15, 16]. Regardless of single or multiple intents, the relationship between slots and intent(s) is many-to-one. In single-intent utterances, all slots share the same intent; in multi-intent scenarios, however, different slots may correspond to different intents. In Fig. 1, the slot *"south carolina"* corresponds to *GetWeather*, while slots *"record"* and *"michael jackson"* should be linked to *PlayMusic*. Existing multi-intent SLU models [15, 16] inherit the single-intent pattern, i.e., the utterance-level intent label distribution is shared by all the slots (Fig. 1a). Therefore, the relationship between slots and intents are not utilized even though they are **already** annotated in most SLU datasets.



(a) [15, 16]: Utterance-level intent is shared by all the slots.

(b) SLIM: Explicit slot-level intent is used by each slot.

Fig. 1: Comparison between approaches of utilizing intent information for different slots.

To this end, we propose SLIM (**SL**ot-**I**ntent **M**apping) that fully exploits the existing annotations in datasets to explicitly model the relationship between slots and intents. We add a slot-intent classifier to predict the intent label for each slot of an utterance (Fig. 1b) on top of the state-of-the-art pre-trained model [8] based on BERT [17] to jointly tackle ID and SF. In experiments, we compare SLIM to a wide range of SLU techniques on two simulated datasets (MixATIS and MixSNIPS [16]) and a real-world dataset (DSTC4 [14]). Empirical results demonstrate that SLIM achieves better performance compared to the current state-of-the-art *without* using extra annotation data. We analyze and reveal that the explicit slot-intent mapping module indeed helps the model learn faster and better. To our best knowledge, this is the first study to explicitly link slots to their intents for better SLU in task-oriented dialog systems.

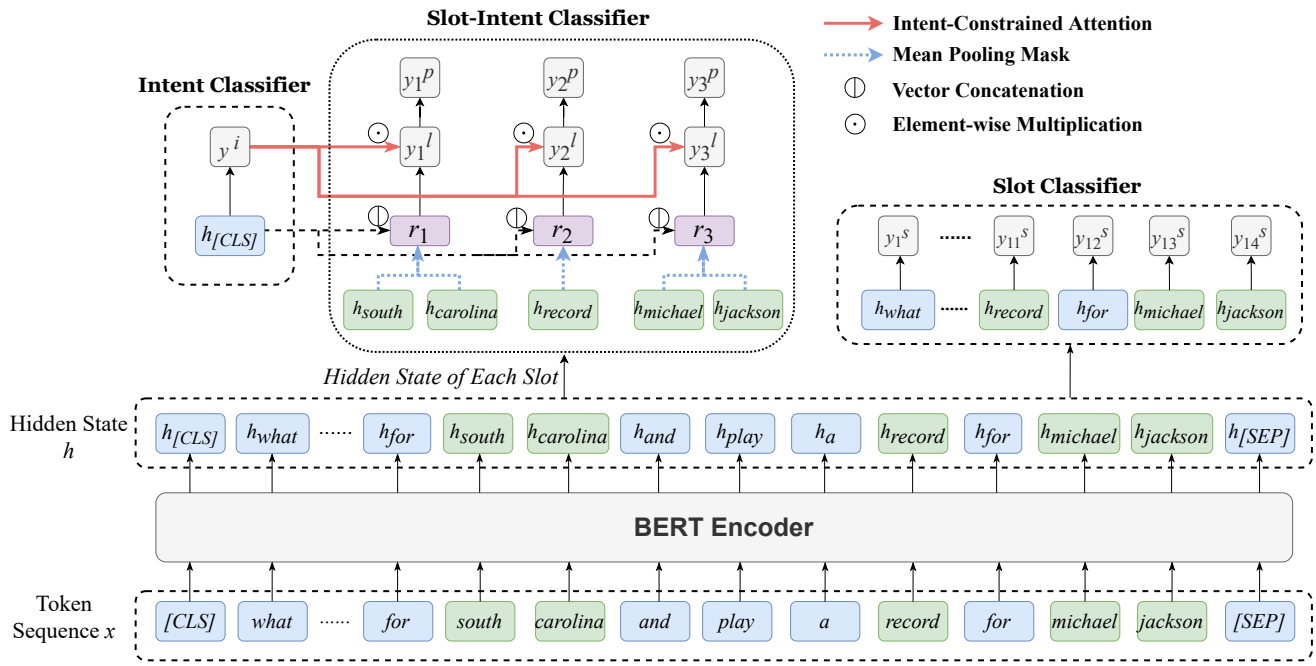


Fig. 2: Illustration of our model **SLIM**. The lower part is the BERT encoder. On the top, from left to right, they are the intent classifier (for ID), the slot-intent classifier, and the slot classifier (for SF) respectively.

2. APPROACH

2.1. Problem Setting

For an input utterance x with token sequence $x = (x_1, ..., x_n)$, the multi-intent SLU task is composed of (1) *utterance-level intent detection*: predict the multi-label intents $I_x \subset I$ of the utterance, where I is the set of possible intents, and (2) *token-level slot filling*: predict the slot label for each token of the input utterance from a set T of possible slots. Different from existing single-intent SLU task, the following two *assumptions* are made:

(a) An utterance x has **at least one** utterance-level intent, i.e, $|I_x| \geq 1$.

(b) Each slot $s_m = \{x_{m_1}, ..., x_{m_j}\}$ is a set of j tokens, and it is mapped to a specific slot-level intent **belonging to** the utterance-level intents, i.e, $i_m \in I_x$.

2.2. Model

The proposed model (SLIM) contains a shared encoder and three classifiers for different tasks.

2.2.1. Encoder

We use BERT [17] as token sequence encoder of our model. The utterance is tokenized by standard BERT tokenizer with a special token [CLS] prepended and [SEP] appended. The output $h = (h_{\text{cls}}, h_1, ..., h_n, h_{\text{sep}})$ of BERT's encoder will be utilized for three classifiers, where $h_k \in \mathbb{R}^d$.

2.2.2. Intent Classifier and Slot Classifier

Utterance-level intent detection is accomplished by the intent classifier. To classify the intent y^i, we use *Sigmoid* as the activation function after feeding h_{cls} into an output network:

$$y^i = Sigmoid(W^i h_{\text{cls}} + b^i), \tag{1}$$

where $W^i \in \mathbb{R}^{|I| \times d}$, and each dimension of $y^i \in \mathbb{R}^{|I|}$ represents the probability of an intent label. In the slot filling task, to predict the slot y_k^s at position k, we apply *Softmax* after feeding h_k into a separate slot classification network as:

$$y_k^s = Softmax(W^s h_k + b^s), \tag{2}$$

where $W^s \in \mathbb{R}^{|T| \times d}$, and $y_k^s \in \mathbb{R}^{|T|}$ is the slot probability distribution for token x_k with $k \in \{1, \ldots, n\}$. Intent and slot classifier are formulated similarly as [8], except that intent detection is formulated as a multi-label classification task.

2.2.3. Slot-Intent Classifier

To explicitly capture the relation between slots and intents, we predict the slot-level intent for each slot s_m. First, we compute the representation r_m of a slot $s_m = \{x_{m_1}, ..., x_{m_j}\}$ by a *mean pooling* of the token representations in this slot by: $r_m = 1/j \sum_{i=1}^{j} h_{m_i}$. Afterward, we concatenate the

global utterance representation h_{cls} with r_m and compute an **unconstrained** slot-intent prediction y_m^l as:

$$y_m^l = Softmax(W^l[h_{\text{cls}}|r_m] + b^l), \tag{3}$$

where $W^l \in \mathbb{R}^{|I|\times 2d}$, and $|$ indicates the operation of vector concatenation. To better align the above slot-intent prediction with the predicted utterance intent (assumption (b)), we propose an *intent-constrained attention*. It computes a final **constrained** slot-intent prediction y_m^p by an element-wise multiplication between the utterance-level intent prediction y^i and unconstrained slot-intent prediction y_m^l as:

$$y_m^p = y^i \odot y_m^l \tag{4}$$

2.3. Training Objective

The overall objective of SLIM is to maximize $p(y^i, y^s, y^p|x)$, and it can be decomposed as:

$$\prod_m \underbrace{p(y^i, y^s_{s_m}, y^p_m|x)}_{\text{inside each slot}} \underbrace{\prod_{j\notin\cup_m s_m} p(y^i, y^s_j|x)}_{\Gamma\text{: outside the slots}} \tag{5}$$

$$= \prod_m p(y^i, y^s_{s_m}|x)\cdot p(y^p_m|y^i, y^s_{s_m}, x)\cdot\Gamma \tag{6}$$

$$\propto p(y^i|x)\prod_{k=1}^{n} p(y^s_k|x)\prod_m p(y^p_m|y^i, y^s, x) \tag{7}$$

$$= \underbrace{p(y^i|x)}_{\text{ID}}\underbrace{\prod_{k=1}^{n} p(y^s_k|x)}_{\text{SF}}\underbrace{\prod_m p(y^p_m|y^i, y^s_{s_m}, x)}_{\text{Slot-Intent Classification}} \tag{8}$$

In equation 8, the first two terms are the objectives of ID and SF, and the last term is the objective of slot-intent mapping. ID is trained with binary cross-entropy loss for multi-intent detection, while the two other terms are trained with regular cross-entropy loss. Because the slot-intent prediction y_m^p is conditional on the predicted utterance intent y^i and slot label $y^s_{s_m}$, parameters for ID and SF will also be updated when training the slot-intent classifier. The loss of SLIM is a weighted sum of losses from these three classifiers. An overview model pipeline of SLIM is illustrated in Fig. 2.

3. EXPERIMENT AND ANALYSIS

3.1. Data

We evaluate our approach on three multi-intent SLU datasets. **MixSNIPS** [16] contains 39,776/2,198/2,199 utterances for train/validation/test. It is created based on the Snips personal voice assistant [18]. **MixATIS** [16] is constructed from ATIS [19], containing 13,161/759/828 utterances for train/validation/test. **DSTC4** [14] contains multi-intent human to human dialogues with 5,308/2,098/1,865 utterances for train/validation/test. Table 1 summarizes the statistics of three datasets.

# of Sentences / # of Intent(s) / Dataset	1	2	3	4
MixSNIPS	8,277	22,499	9,000	-
MixATIS	1,118	8,444	3,599	-
DSTC4	3,963	1,240	100	5

Table 1: Summary of number of utterances with different numbers of intents in MixSNIPS, MixATIS and DSTC4.

Hyper-parameter	Search Range
Dropout Rate	{0, 0.1, **0.2**, 0.3, 0.4}
Learning Rate	{1e-5, **5e-5**, 1e-4, 5e-4}
Loss Weight of Intent Classifier	{0.5, **1**, 2}
Loss Weight of Slot classifier	{0.5, 1, **2**}
Loss Weight of Slot-Intent Classifier	{0.5, **1**, 2}

Table 2: Hyper-parameter search range of **SLIM**. **Bold** numbers indicate our choice of hyper-parameters.

3.2. Training Details

We use English uncased BERT-Base model, containing 12 layers, 768 hidden states, and 12 heads. The max sequence length is 50, and training batch size is 32. Hyper-parameters of SLIM are tuned based on the semantic frame accuracy on validation set, as shown in Table 2. Dropout rate is 0.2 for the output layers of all three classifiers. In Eq. (8), the losses of ID and slot-intent classifier are weighted by 1, and the loss of SF is weighted by 2. We train SLIM for maximum 20 epochs, and the early stop patience is 3 epochs. For the slot-intent classifier, we use slots provided by ground truth during training, while the predicted slots are used during inference.

3.3. Baselines and Evaluation Metrics

We compare SLIM with both single-intent and multi-intent models. When evaluating single-intent models on the multi-intent task, we follow [15, 16] to concatenate multiple intents with '#' into a single intent. On MixATIS and MixSNIPS, we compare previous single-intent models: Bi-Model [20], SF-ID [10], Stack-Propagation [11] and recent multi-intent models: Joint Multiple ID-SF [15], AGIF [16]. On DSTC4, we compare SLIM with state-of-the-art single-intent model Stack-Propagation [11] and multi-intent model AGIF [16].

We evaluate the performance of different methods using three metrics: F1 score of slot filling (Slot F1), accuracy of intent detection (Intent Acc), and semantic frame accuracy (SeFr Acc) as in [15] and [16]. "SeFr Acc" considers the prediction of an utterance to be correct when slots and intents are *all* accurate, and we consider it as a main metric in our experiments.

3.4. Overall Result

Table 3 summarizes different models' performances on MixATIS and MixSNIPS. We observe that SLIM outperforms

Model	MixATIS			MixSNIPS		
	Slot F1	Intent Acc	SeFr Acc	Slot F1	Intent Acc	SeFr Acc
Bi-Model† [20]	85.5	72.3	39.1	86.8	95.3	53.9
SF-ID† [10]	87.7	63.7	36.2	89.6	96.3	59.3
Stack-Propagation†§ [11]	86.6	76.0	42.8	93.9	96.4	75.5
Joint Multiple ID-SF‡ [15]	87.5	73.1	38.1	91.0	95.7	66.6
AGIF‡§ [16]	88.1	75.8	44.5	94.5	96.5	76.4
SLIM (w/o slot-intent classifier)	85.6	77.1	46.3	96.2	96.8	82.3
SLIM (w/o intent-constrained attention)	87.2	75.6	46.4	96.5	96.0	83.6
SLIM	**88.5**	**78.3**	**47.6**	**96.5**	**97.2**	**84.0**

Table 3: Results on two multi-intent datasets. † or ‡ denotes single-intent or multi-intent model respectively. Results of models with † and ‡ are taken from [16]. Models with § indicate the previous state-of-the-art solutions. **Bold** numbers are the best results in each column.

Model	DSTC4		
	Slot F1	Intent Acc	SeFr Acc
Stack-Propagation†	56.4	35.8	20.7
AGIF‡	57.6	33.0	19.4
SLIM	**61.1**	**36.7**	**21.3**

Table 4: Results on DSTC4. † or ‡ denotes single-intent or multi-intent model respectively. **Bold** numbers are the best results in each column.

other baselines w.r.t. all three metrics, especially on the semantic frame accuracy. On Slot F1 and Intent Acc, SLIM outperforms the previous start-of-the-art (AGIF) by 0.4%-2.5%. More importantly, SLIM improves the semantic frame accuracy compared to AGIF by 3.1% and 7.6% on two datasets respectively. This result indicates that SLIM effectively improves the joint correctness for predicting intents and slots for understanding an utterance. Table 4 summarizes the results on the real-world DSTC4 dataset, and similar performance patterns can be observed as in Table 3. Moreover, SLIM also notably improves Slot F1 on this dataset (3.5% and 4.7% gain over the two baselines), indicating that SLIM can also strengthen token-level prediction in addition to the large utterance-level improvements in Table 3.

3.5. Ablation Study

We compare SLIM with two simplified versions, *w/o slot-intent classifier* [1] and *w/o intent-constrained attention*. The comparison to these two simplified versions aims to analyze how the slot-intent classifier and its intent-constrained attention affect the performance and the training process.

From Table 3 (Bottom), we can see that dropping these two components impairs the semantic frame accuracy of SLIM. Dropping slot-intent classifier mainly degrades Slot F1, and dropping intent-constrained attention mainly degrades Intent Acc. Without slot-intent classifier, slot prediction will lose the information from the local intent. And when removing only the intent-constrained attention, the sentence-level intent will be isolated from the slot-intent classification, losing partial supervision compared with full SLIM. Furthermore, we plot the learning curves of these three methods on MixATIS's validation set in Fig. 3. We could observe that SLIM converges **faster and better** compared to the two simplified versions, which reinforces the benefits of the slot-intent classifier with intent-constrained attention for the model learning process.

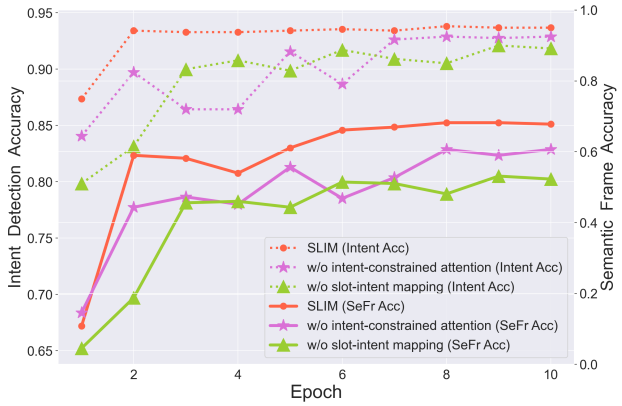


Fig. 3: Intent Acc and SeFr Acc on MixATIS validation dataset after each training epoch.

4. CONCLUSION

In this paper, we propose SLIM for multi-intent SLU in task-oriented dialog systems. Without any newly introduced annotated data, SLIM explicitly utilizes the existing relation between slots and intents by mapping slots to the corresponding intent via the slot-intent classifier and intent-constrained attention. Experimental results on three public datasets show that SLIM outperforms the previous state-of-the-art SLU models. Our findings may inspire future studies to better exploit the relationship between slots and intents for complicated SLU scenarios in task-oriented dialog systems.

[1]The model degenerates to [8] as mentioned in Sec. 2.2

References

[1] Gokhan Tur and Renato De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*, John Wiley & Sons, 2011.

[2] Xiaodong Zhang and Houfeng Wang, "A joint model of intent determination and slot filling for spoken language understanding," in *IJCAI*. 2016, pp. 2993–2999, IJCAI/AAAI Press.

[3] Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi, "Spoken language understanding using long short-term memory neural networks," in *SLT*. 2014, pp. 189–194, IEEE.

[4] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy, "Hierarchical attention networks for document classification," in *HLT-NAACL*. 2016, pp. 1480–1489, The Association for Computational Linguistics.

[5] Ngoc Thang Vu, "Sequential convolutional neural networks for slot filling in spoken language understanding," in *INTERSPEECH*. 2016, pp. 3250–3254, ISCA.

[6] Chih-Wen Goo, 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 *NAACL-HLT (2)*. 2018, pp. 753–757, Association for Computational Linguistics.

[7] Yijin Liu, Fandong Meng, Jinchao Zhang, Jie Zhou, Yufeng Chen, and Jinan Xu, "Cm-net: A novel collaborative memory network for spoken language understanding," in *EMNLP/IJCNLP (1)*. 2019, pp. 1051–1060, Association for Computational Linguistics.

[8] Qian Chen, Zhu Zhuo, and Wen Wang, "Bert for joint intent classification and slot filling," *CoRR*, vol. abs/1902.10909, 2019.

[9] Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S. Yu, "Joint slot filling and intent detection via capsule neural networks," in *ACL (1)*. 2019, pp. 5259–5267, Association for Computational Linguistics.

[10] Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song, "A novel bi-directional interrelated model for joint intent detection and slot filling," in *ACL (1)*. 2019, pp. 5467–5471, Association for Computational Linguistics.

[11] Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *EMNLP/IJCNLP (1)*. 2019, pp. 2078–2087, Association for Computational Linguistics.

[12] Libo Qin, 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 *ACL/IJCNLP (1)*. 2021, pp. 178–188, Association for Computational Linguistics.

[13] Libo Qin, 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, pp. 8193–8197.

[14] Seokhwan Kim, Luis Fernando D'Haro, Rafael E. Banchs, Jason D. Williams, and Matthew Henderson, "The fourth dialog state tracking challenge," in *IWSDS*. 2016, vol. 427 of *Lecture Notes in Electrical Engineering*, pp. 435–449, Springer.

[15] Rashmi Gangadharaiah and Balakrishnan Narayanaswamy, "Joint multiple intent detection and slot labeling for goal-oriented dialog," in *NAACL-HLT (1)*. 2019, pp. 564–569, Association for Computational Linguistics.

[16] Libo Qin, Xiao Xu, Wanxiang Che, and Ting Liu, "Towards fine-grained transfer: An adaptive graph-interactive framework for joint multiple intent detection and slot filling," in *EMNLP (Findings)*. 2020, pp. 1807–1816, Association for Computational Linguistics.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT (1)*. 2019, pp. 4171–4186, Association for Computational Linguistics.

[18] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *CoRR*, vol. abs/1805.10190, 2018.

[19] Charles T. Hemphill, John J. Godfrey, and George R. Doddington, "The ATIS spoken language systems pilot corpus," in *HLT*. 1990, Morgan Kaufmann.

[20] Yu Wang, Yilin Shen, and Hongxia Jin, "A bi-model based RNN semantic frame parsing model for intent detection and slot filling," in *NAACL-HLT (2)*. 2018, pp. 309–314, Association for Computational Linguistics.