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

FANS: Fusing ASR and NLU for on-device SLU

Martin Radfar, Athanasios Mouchtaris, Siegfried Kunzmann, and Ariya Rastrow

Alexa Machine Learning, Amazon, USA
{radfarmr,mouchta,kunzman,arastrow}@amazon.com

Abstract

Spoken language understanding (SLU) systems translate voice input commands to semantics which are encoded as an intent and pairs of slot tags and values. Most current SLU systems deploy a cascade of two neural models where the first one maps the input audio to a transcript (ASR) and the second predicts the intent and slots from the transcript (NLU). In this paper, we introduce FANS, a new end-to-end SLU model that fuses an ASR audio encoder to a multi-task NLU decoder to infer the intent, slot tags, and slot values directly from a given input audio, obviating the need for transcription. FANS consists of a shared audio encoder and three decoders, two of which are seq-to-seq decoders that predict non null slot tags and slot values in parallel and in an auto-regressive manner. FANS neural encoder and decoders architectures are flexible which allows us to leverage different combinations of LSTM, self-attention, and attenders. Our experiments show compared to the state-of-the-art end-to-end SLU models, FANS reduces ICER and IRER errors relatively by 30% and 7%, respectively, when tested on an in-house SLU dataset and by 0.86% and 2% absolute when tested on a public SLU dataset.

Index Terms: Spoken language understanding (SLU), Automatic speech recognition (ASR), Natural language understanding (NLU), sequence-to-sequence, neural transformer, encoder-decoder, intent-slot

1. Introduction

Spoken language understanding (SLU) systems translate the meaning of a spoken utterance to a format that is readable by machine [1–3]. Today's commercial neural SLU systems deploy a cascade of ASR and NLU to infer semantic information from audio [4]. In these models, after a spoken utterance is transcribed by an ASR module, an NLU system decodes the desired NLU output from the transcript. This disjoint SLU framework has several drawbacks: First, ASR and NLU are optimized independently and NLU performance highly depends on ASR accuracy. Second, many ASR errors occurred in words that have null slot tags and, therefore, have no impact on NLU performance. Accordingly, training ASR to get a full transcript is sub-optimal for NLU. Third, disjoint SLU models restrict the model compression and sharing parameters end-to-end, a requirement for on-device SLU where computational resource is limited.

In this paper, we go beyond disjoint SLU paradigm and introduce an end-to-end SLU system which is optimized to maximize the NLU accuracy directly from audio. The outputs of an SLU system are often: an intent and two sequences that contain slot tags and values. In recent years, several end-to-end SLU approaches have beed proposed [5–23] which can be classified into two broad categories: The models in the first category approach SLU as a multi-label classification problem aiming mostly to predict the intent of a spoken utterance [5,9,10,24,25]. In [24], stacked layers of biGRU along with MLP are used to classify intent directly from audio files. In [5], a SincNet layer [26] is augmented to the architecture proposed in [24] and leverage a pre-training strategy to further improve accuracy. In addition to RNNs, convolutional neural networks (CNNs) also have been used alone or joint with RNNs for speech-to-intent classification [10, 25]. In [25], a cascade of CNN and GRU layers is used with different training strategy called Reptile, to infer the intents. In [27], a multi-modal model was proposed in which a pre-trained text embedding incorporated as an auxiliary model along with a biLSTM audio encoder to predict intent.

The second category of SLU models infer not only intent but also slot tags and slot values. There have been different proposals, though less than the first category, to achieve this goal [12, 17–22, 28, 29] . In [18], an SLU model is proposed that leverages text BERT pre-training that is jointly optimized with ASR. In [19], a jointly trained ASR and NLU is proposed by introducing an interface to couple ASR and NLU neural models. In this model, ASR and NLU are first pre-trained and then fine-tuned on the SLU task. In [30], an SLU model is proposed that adapts the pre-trained CTC- and attention- based ASR modules to directly predict NLU output. In [20], a transformer-based SLU is proposed which is shown to be a competitive E2E SLU models but this architecture assumes the slot values are associated to a fixed, ordered, pre-defined slot tags. In [21], another transformer based SLU model was proposed that leverages self-supervised pre-trained acoustic features, pre-trained model initialization that allows to use large training data and fine tuned on SLU data. In [12], multi-task learning is leveraged for the SLU task. The authors proposed different model architectures to decode ASR and NLU outputs alone or together. In this multi-task learning scheme, one task is to decode the transcript and the other task is to decode serialized semantics (slot values and slot tags). They also proposed a direct model to decode the serialized slot tag and slot values.

In this paper, we propose, FANS, a new end-to-end neural architecture that fuses the ASR encoder directly to the NLU decoder. FANS infers intent, slot tags, and slot value directly from audio with no dependency on transcription. In contrast to [12], FANS avoids serialized semantics by deploying two parallel decoders to infer only non null slot tags and their corresponding slot values. In addition, FANS does not use transcription which is used in [12] as an auxiliary task to help predict semantic. Using attention mechanism, FANS learns to attend to audio segments that are solely relevant to NLU. FANS uses predicted intent as a prior to help better slot tag prediction. FANS model architecture leverages both LSTM and transformer in the encoder and decoder with different attention mechanisms to get best of both worlds. FANS learns parameters that are optimized for the NLU output rather than ASR so that makes FANS more accurate, compact, and a promising candidate for on-device SLU. We evaluate the performance of FANS on both public and in-house SLU datasets. Our experiments show a significant performance improvement when FANS is compared with the state-

http://dx.doi.org/10.21437/Interspeech.2021-793

of-the-art-models.

The rest of this paper is organized as follows: In Section 2, we describe and formalize FANS architecture with its variants. In Section 3, we demonstrate the superiority of FANS over a competitive model using both in-house and public datasets. Finally, in Section 4, we draw the conclusions.

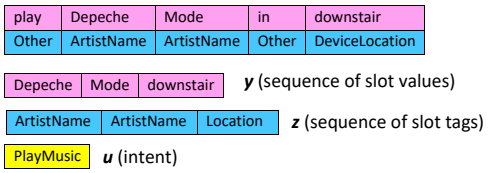

Figure 1: *The FANS output target consists of an intent and two sequences that contain pair-wise match slot tag and slot values. We discard all slot values associated with slot labeled as Other— note that we also use the term Null interchangeably in the text.*

2. FANS Architecture

2.1. FANS input and output structures

The input audio signal is transferred to a time-frequency space represented by $\boldsymbol{x} = [x_1, \ldots, x_i, \ldots, x_T]^\mathsf{T}$ where $x_i \in \mathcal{R}^D$, T and D denote the number of frames and the dimension of the time-frequency space, respectively; also T denotes the transpose operation. Each audio signal, $\boldsymbol{x}$, has a transcript and a slot tag annotation represented by $\hat{\boldsymbol{y}} = [\hat{y_1}, \ldots, \hat{y_j}, \ldots, \hat{y_L}]^\mathsf{T}$ and $\hat{\boldsymbol{z}} = [\hat{z_1}, \ldots, \hat{z_j}, \ldots, \hat{z_L}]^\mathsf{T}$, respectively, where L is the number of words in the transcript and $z_j \in \{\text{vocab}(\mathcal{Z}) \cup Null\}$. In other words, each word in the transcript belongs either to a slot tag from a slot tag dictionary or it is labelled as $Null$ (or Other). An example is demonstrated in Figure 1 in which the transcript is $\hat{\boldsymbol{y}} = [play, Depeche, Mode, in, Downstairs]^\mathsf{T}$ and $\hat{\boldsymbol{z}} = [Null, ArtistName, ArtistName, Null, DeviceLocation]^\mathsf{T}$. For the SLU task, we are only interested in inferring non $Null$ slot tags and words in the transcripts associated with non $Null$ slot tags—we call them slot values. Accordingly, we exclude all $Null$ tags and words associated with the $Null$ tags to generate two new sequences: $\boldsymbol{y} = [Depeche, Mode, downstairs]^\mathsf{T}$ and $\bar{\boldsymbol{z}} = [ArtistName, ArtistName, DeviceLocation]^\mathsf{T}$. As such, the target output would be the slot tag $\boldsymbol{y} = [y_1, \ldots, y_j, \ldots, y_M]^\mathsf{T}$, and the slot value $\bar{\boldsymbol{z}} = [\bar{z_1}, \ldots, \bar{z_j}, \ldots, \bar{z_M}]^\mathsf{T}$, where M is the length of the sequence after discarding $Null$ tags in the tag sequence and corresponding slot values. In addition to $\boldsymbol{y}$ and $\bar{\boldsymbol{z}}$, there is an intent, $u \in \text{vocab}(\mathcal{U})$, corresponding to each $\boldsymbol{x}$. We insert u as the first element into $\bar{\boldsymbol{z}}$ to get $\boldsymbol{z} = [u; \bar{\boldsymbol{z}}]^\mathsf{T}$. Hence, we generate two sequences $\boldsymbol{y}$ and $\boldsymbol{z}$ which comprise the output of FANS and together tuple $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ forms the input and outputs of FANS, where $\boldsymbol{x}$ is a vector of audio features, $\boldsymbol{y}$ a vector of slot values, and $\boldsymbol{z}$ is a vector of slot tag plus intent.

Different from our output structure, in [12], the slot tag and slot values are serialized—what the authors called serialized semantics; for our example, the serialized semantics given by $\hat{\boldsymbol{s}} = [ArtistName, Depeche, ArtistName, Mode, DeviceLocation, Downstairs]^\mathsf{T}$

2.2. FANS model

FANS consists of a shared audio encoder and two sequence decoders: slot tag and slot value decoders and one intent decoder whose parameters are shared with the slot tag decoder (Figure 2). The neural structure of the encoder and the decoders can be any types of known neural architectures but here we use different combinations of LSTM and self-attention—self-attention is technically the self-attention mechanism used in Transformers [31] . In addition, FANS deploys an attender that helps the decoders to attend to the encoder for better prediction. We use both additive attention [32] and cross-attention [31].

The shared audio encoder transforms $\boldsymbol{x}$ to the higher level feature representations $\boldsymbol{h}^{\text{enc}} = [h_1, \ldots, h_i, \ldots, h_T]^\mathsf{T}$ where $h_i \in \mathcal{R}^K$, and K is the dimension of the encoder output feature space. Given the input vectors, previous decoded outputs, and the model parameters $\Omega = \{\Omega^{\text{enc}}, \Omega^{\text{dec-tag}}, \Omega^{\text{dec-value}}, \Omega^{\text{dec-intent}}\}$, FANS infers the slot values, slot tags, and intent by maximizing the following posterior probabilities:

$$\begin{aligned}
\overset{*}{y}_i &= \underset{\text{vocab}(\mathcal{Y})}{\text{argmax}}\ p(y_i|\boldsymbol{x}, f_{\text{att}}(\boldsymbol{h}^{\text{enc}}, \boldsymbol{y}_{-i}), \Omega^{\text{enc}}, \Omega^{\text{dec-tag}}), \\
\overset{*}{z}_i &= \underset{\text{vocab}(\mathcal{Z})}{\text{argmax}}\ p(z_i|\boldsymbol{x}, f_{\text{att}}(\boldsymbol{h}^{\text{enc}}, \boldsymbol{z}_{-i}, u), \Omega^{enc}, \Omega^{\text{dec-value}}), \\
\overset{*}{u} &= \underset{\text{vocab}(\mathcal{U})}{\text{argmax}}\ p(u|\boldsymbol{x}, \boldsymbol{h}^{\text{enc}}, \Omega^{enc}, \Omega^{\text{dec-intent}}),
\end{aligned} \tag{1}$$

where the subscript $-i$ represents all previous decoded tokens before the ith token. Also $f_{\text{att}}(\cdot,\cdot)$ denotes the attender function which is either the cross attention of the form:

$$f_{CA}(\boldsymbol{h}^{\text{enc}}, \boldsymbol{q}_{-i}) = \text{softmax}\Big(\frac{\boldsymbol{w}_q \boldsymbol{q}_{-i} \boldsymbol{h}^{\text{enc}\mathsf{T}} \boldsymbol{w}_k{}^\mathsf{T}}{\sqrt{K'}}\Big)\boldsymbol{w}_v \boldsymbol{h}^{\text{enc}} \tag{2}$$

where $\boldsymbol{q}_{-i} = \boldsymbol{y}_{-i}$ for the slot value decoder and $\boldsymbol{q}_{-i} = \boldsymbol{z}_{-i}$ for the slot tag decoder; also, $\boldsymbol{w}_q$, $\boldsymbol{w}_k$, and $\boldsymbol{w}_v$ are learnable query, key, and value matrices that transform the input vector into a K'-dimensional space. The additive attention given by

$$f_{AA}(\boldsymbol{h}^{\text{enc}}, \boldsymbol{q}_{i-1}) = \sum_{j=1}^{T} \text{softmax}\big(g(\boldsymbol{q}_{i-1}, \boldsymbol{h}_j^{\text{enc}})\big)\boldsymbol{h}_j^{\text{enc}} \tag{3}$$

where $\boldsymbol{q}_{i-1} = \boldsymbol{y}_{i-1}$, the hidden state just before emitting $\boldsymbol{y}_i$ for the slot value decoder and $\boldsymbol{q}_{i-1} = \boldsymbol{z}_{i-1}$, the hidden state just before emitting $\boldsymbol{z}_i$ for the slot tag decoder. Also, $g(\cdot)$ is a feed-forward neural network. Given N training samples drawn from distribution $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$, the model parameters Ω are learned by maximizing the likelihood function

$$\begin{aligned}
\overset{*}{\Omega} = \underset{\Omega}{\text{argmax}} \sum_{\Lambda} \Big(&\log p(\mathcal{Y}|\mathcal{X}, \Omega^{\text{enc}}, \Omega^{\text{dec-tag}}) \\
&+\lambda_1 \log p(\mathcal{Z}|\mathcal{X}, \mathcal{U}, \Omega^{enc}, \Omega^{\text{dec-value}}) \\
&+\lambda_2 \log p(\mathcal{U}|\mathcal{X}, \Omega^{enc}, \Omega^{\text{dec-intent}}) \Big)
\end{aligned} \tag{4}$$

where $\Lambda = (\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \mathcal{U})$ is the set of the training data and λ_1 and λ_2 are two tuning parameters that can control the contribution of each likelihood function during the training process.

3. Experiments

3.1. Datasets and evaluation metrics

We use two SLU datasets: a public and an in-house. Compared to ASR and NLU datasets that contain thousands of hours

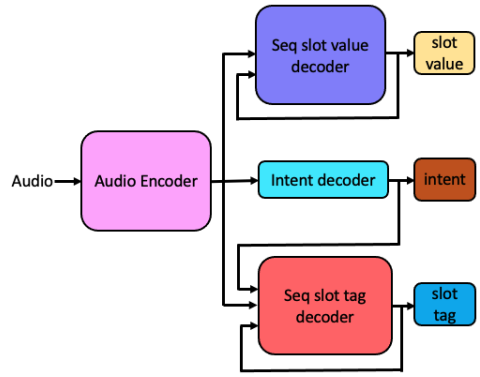

Figure 2: *FANS architecture. The model consists of a shared audio encoder, three decoders, and an attender .*

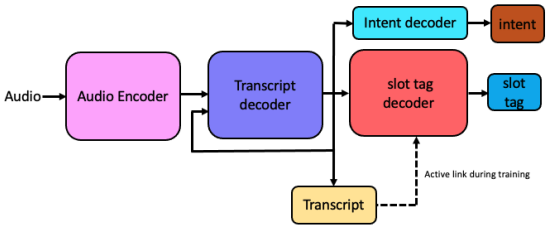

Figure 3: *A high-level block-diagram of cascade ASR-NLU SLU models.*

of transcribed audio files and numerous text, current popular public SLU datasets have limited audio files. Three popular ones are FSC, containing 30,000 utterances [5], SNIP containing 14,000 utterances [33], and ATIS, containing 4,500 utterances [34] which makes training SLU models end-to-end very challenging. We opted for FSC [5] to be able to leverage end-to-end training. This dataset contains 11 intents, 3 slot tags, and slot values belong to a vocabulary of size 100. Because FSC is not annotated for intent, we consider the action labels as intents and the object and location labels as slot tags. The SNIP and ATIS datasets are too small to be used for E2E training so these datasets usually are used as fine tuning datasets when models are pre-trained with large ASR and NLU corpa separately.

We also used a subset of 500 hours of de-identified in-house SLU data in which the data were split into train (80%), evaluation (10%) and test (10%) sets. Our SLU dataset contains 57 intents and 175 slot tags and the slot values belong to a vocabulary of size ∼100,000. We used 64-dimensional log short time Fourier transform vectors obtained by segmenting the utterances with a Hamming window of the length 25 ms and frame rate of 10 ms. The 3 frames are stacked with a skip rate of 3, resulting in 192-dimensional input features; these features are normalized using the global mean and variance normalization.

We evaluate the SLU models using two well-known metrics: intent classification error rate (ICER) and interpretation error rate (IRER). These metrics are defined as follows. Let I denote the number of utterances with miss-recognized intent. Let E denote the number of utterances that contains any of miss-recognized intent, slot tags, or slot values; e.g., if we have an utterance with one intent, 10 slot values and 10 slot tags and only one of these 21 (10+10+1) entities is miss-recognized, we count one error for the entire utterance. Hence: ICER $= \frac{I}{N}$ and IRER $= \frac{E}{N}$ where N is the total number of utterances. It should be noted for the in-house SLU data, we report performance in terms of relative increase in ICER (RI-ICER) and IRER (RI-IRER) against the best model, meaning that we set IRER and ICER to zero for the best model and compute the relative increase in ICER and IRER for other models when compared to the best model.

3.2. Model parameters

We built three different versions of FANS by deploying LSTM and self-attention (SA) [31] neural structures in the encoder and decoders. We also used cross-attention (CA)[31] or additive attention (AA) [32] as the attender. We tried to keep all models within the same size to make comparison not biased on the size of the model. We compared FANS with the multi-task and direct A2I models proposed in [12] .

The three versions of FAN are as follows: 1- FAN-a: an LSTM based encoder, two LSTM based decoders, and an additive attention attender; 2- FAN-b: an LSTM based encoder, two self-attention based decoders, a cross-attention attender; 3- FAN-c: a self-attention based encoder and two self-attention based decoders, and a cross-attention attender. In FAN-a, we used a 3×612 LSTM in the encoder and two decoders. In FAN-b, we used a 4×768 LSTM in the encoder and 3 layers of self-attention layers in each decoder. In FAN-c, we used 12 self-attention layers in the encoder, 5 self-attention layers in the slot value decoder, and 3 self-attention layers in the slot tag decoder. The input dimension (d-model) was set to 256. For all self-attention layers, we used 4 heads and a 2,048-hidden unit feedforward net is used. For the multi-task and direct A2I in [12], the size of the encoder and decoder were set to 3×612 and 4×612, respectively. We trained these models using 500 hours of data in-house data.

For the FSC dataset, we built smaller models to prevent overfitting as the FSC dataset is very small and not complex. We observed models of size $\sim$ 3 millions deliver good results and don't suffer from overfitting. For FSC, the FANS models were built as follows: In FAN-a, we used a 1×256 LSTM in the encoder and two decoders. In FAN-b, we used a 2×232 LSTM in the encode and 2 layers of self-attention layers in each decoder with 1 head and a 800-hidden unit feedforward net. In FAN-c, we used 2 self-attention layers in the encoder, 1 self-attention layer in the slot value decoder, and 1 self-attention layer in the slot tag decoder with 4 heads and a 1024-hidden unit feedforward net. The input dimension (d-model) for all these models was set to 128. For the multi-task and direct A2I in [12], the size of the encoder and decoder were set to 1×256 and 1×350 LSTM, respectively.

The self-attention sub-space dimension (the column dimension of the query, key and value matrices) was set to 64. The dropout rate and label smoothing were set to 0.1. We used Adam optimizer with β_1 = 0.9, β_2= 0.98, and ϵ = 1e-9. The learning curve was chosen by setting the pre-defined factor k and warm-up rate w to 0.99 and 16,000, respectively. We used step size of 1000 and the model trained until no improvement observed. For all experiments, λ_1 and λ_2 were set to 1. For the cascade ASR-NLU model we used 12/4 and 7/2 layers self-attention in the encoder and decoder of ASR for large/small models, respectively. For the NLU model, we used a transformer encoder with 2 layers that was trained on the full transcript and the NLU output including $Null$ tags . The large models were trained using four machines each of which has eight GPUs and the small models were trained on one GPU.

Table 1: *Results in terms of relative increase in ICER (RI-ICER) and IRER (RI-IRER) against the best model; The last column gives the model size in terms of the number of parameters in million; In this experiments we used our in-house SLU dataset. SA: self-attention, CA: cross attention [31] and AA: additive attention[32] .*

Model	Encoder	Decoder	Attender	*RI-ICER (%)*	*RI-IRER(%)*	Size (M)
A2I- multi-task [12]	LSTM	LSTM	AA	30	7	31
A2I-direct-model [12]	LSTM	LSTM	AA	39	21	29
FANS-a	LSTM	LSTM	AA	25	16	31
FANS-b	LSTM	SA	CA	22	11	30
FANS-c	SA	SA	CA	0	0	30
Joint ASR NLU	SA	SA	CA	47	35	30

Table 2: *Resutls in terms of ICER and IRER, and model size; In these experiments, we used public FSC SLU dataset [5].*

Model	Encoder	Decoder	Attender	*ICER (%)*	*IRER(%)*	Size (M)
A2I- multi-task [12]	LSTM	LSTM	AA	2.05	3.45	2.2
A2I-direct-model [12]	LSTM	LSTM	AA	2.37	3.28	2.3
FANS-a	LSTM	LSTM	AA	2.14	3.03	2.2
FANS-b	LSTM	SA	CA	0.71	1.03	2.3
FANS-c	SA	SA	CA	2.03	3.9	2.3
Joint ASR NLU	SA	SA	CA	1.56	14.49	2.31

3.3. Results

Table 1 and 2 show the RI-ICER/RI-IRER and ICER/IRER results which are obtained by using the in-house and FSC datasets, respectively. As shown in Table 1, the best performance is pertained to FANS-c which deploys self-attention both in the encoder and the decoders with a cross attention attender—this model is basically a transformer with three decoders. We observed 30 % and 7 % relative reduction in ICER and IRER compared to the multi-task A2I [12]. We also observe averaging the ICER and IRER , all variants of FANS outperform three other models. Similar to what is reported in [12], we also observed the A2I direct model underperforms compared to the A2I multi-task model suggesting that in the serialized semantic models having a transcript learner helps the NLU decoder. It is also seen that the joint ASR NLU model (Figure 3) exhibits worst performance confirming that the performance of these non E2E SLU models highly depends on separately pre-trained ASR and NLU models. A closer look at slot tag and slot values reveals the slot values prediction is more challenging as the number of slot values are far exceeding the number of slot tags. None of the models compared here are pre-trained.

We also compared FANS with other models when we used the public FSC SLU dataset as shown in Table 2. In order to avoid over-fitting, we reduced the size of the models to $\sim$ 3 millions. Using the FSC dataset, we observed FANS-b outperforms all other models with 0.86 % and 2.00 % ICER and IRER reduction, respectively. We speculate two reasons as why FANS-b outperforms FANS-c on FSC: First, the utterances in FSC are very short so the attention mechanism in the encoder is not as effective as when we deal with longer utterances contained in the larger datasets. Second, FSC is not semantically complex and has very low entropy with repetitive utterances and small number of intent and slot tags; as such, for this dataset, a smaller model is sufficient to provide satisfactory results. We also observe the gap between ICER and IRER is smaller than large data set; this is however mainly because FSC has much smaller number of slot tags compared to the large dataset (3 vs 175). We also observe IRER for the joint ASR NLU model is strikingly high; one explanation is that the 20 hours FSC is not enough to train the ASR model which requires to deliver transcript to the NLU modules with as low WER as possible. Overall, the results given in Table 1 and 2 suggest that multi-task learners for slot tag and slot values is a better strategy than serializing them into one stream and using one learner.

4. Conclusions

In this paper, we introduce FANS, a new neural architecture for end-to-end SLU. FANS fuses ASR and NLU modules to decode outputs that matters for NLU. FANS obviates the need for recognizing the transcript, allowing decoders to attend to the parts of input acoustics that are relevant for inferring slot tags and slot values. We showed that using different learners for slot tags and slot values leads to better performance compared to when all semantic information serialized. Majority of words in ASR transcripts are associated with null slot tags; therefore what the NLU module expects from the audio encoder is a high level representation for words corresponds to non null slot tags. FANS architecture learns to only focus on these words. We showed through experiments on both in-house data and public datasets that FANS outperforms competitive neural SLU models. The compact structure of FANS and its high accuracy make it a promising candidate for on-device SLU.

5. References

[1] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, 2005.

[2] G. Tur and R. D. Mori, "Spoken language understanding: Systems for extracting semantic information from speech," *Wiley*, 2011.

[3] A. Bhargava, A. Çelikyilmaz, D. Z. Hakkani-Tür, and R. Sarikaya, "Easy contextual intent prediction and slot detection," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8337–8341, 2013.

[4] M. A. Larson and G. J. F. Jones, "Spoken content retrieval: A survey of techniques and technologies," *Found. Trends Inf. Retr.*, vol. 5, no. 4-5, pp. 235–422, 2012.

[5] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *arXiv preprint arXiv:1904.03670*, 2019.

[6] V. Renkens *et al.*, "Capsule networks for low resource spoken language understanding," *arXiv preprint arXiv:1805.02922*, 2018.

[7] Y.-P. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6189–6193.

[8] N. Tomashenko, A. Caubrière, Y. Estève, A. Laurent, and E. Morin, "Recent advances in end-to-end spoken language understanding," in *International Conference on Statistical Language and Speech Processing*. Springer, 2019, pp. 44–55.

[9] N. T. Vu, P. Gupta, H. Adel, and H. 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, pp. 6060–6064.

[10] C. Liu, J. Trmal, M. Wiesner, C. Harman, and S. Khudanpur, "Topic identification for speech without asr," *arXiv preprint arXiv:1703.07476*, 2017.

[11] Q. Chen, Z. Zhuo, W. Wang, and Q. Xu, "Transfer learning for context-aware spoken language understanding," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 779–786.

[12] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 720–726.

[13] 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.

[14] N. Tomashenko, C. Raymond, A. Caubriere, R. De Mori, and Y. Esteve, "Dialogue history integration into end-to-end signal-to-concept spoken language understanding systems," *arXiv preprint arXiv:2002.06012*, 2020.

[15] M. Dinarelli, N. Kapoor, B. Jabaian, and L. Besacier, "A data efficient end-to-end spoken language understanding architecture," *arXiv preprint arXiv:2002.05955*, 2020.

[16] P. Wang, L. Wei, Y. Cao, J. Xie, and Z. Nie, "Large-scale unsupervised pre-training for end-to-end spoken language understanding," in *ICASSP*. IEEE, 2020, pp. 7999–8003.

[17] C.-I. Lai, Y.-S. Chuang, H.-Y. Lee, S.-W. Li, and J. Glass, "Semi-supervised spoken language understanding via self-supervised speech and language model pretraining," *arXiv:2010.13826*, 2020.

[18] Y.-S. Chuang, C.-L. Liu, H.-Y. Lee, and L. shan Lee, "Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering," *arXiv:1910.11559*, 2019.

[19] M. Rao, A. Raju, P. Dheram, B. Bui, and A. Rastrow, "Speech to semantics: Improve asr and nlu jointly via all-neural interfaces," *arXiv preprint arXiv:2008.06173*, 2020.

[20] M. Radfar, M. Athanasios, and K. Siegfried, "End-to-end neural transformer based spoken language understanding," in *INTERSPEECH*, 2020, pp. 799–803.

[21] E. Morais, H.-K. J. Kuo, S. Thomas, Z. Tuske, and B. Kingsbury, "End-to-end spoken language understanding using transformer networks and self-supervised pre-trained features," *arXiv preprint arXiv:2011.08238*, 2020.

[22] Y. Qian, R. Ubale, V. Ramanaryanan, P. Lange, D. Suendermann-Oeft, K. Evanini, and E. Tsuprun, "Exploring asr-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 569–576.

[23] Y.-A. Chung, C. Zhu, and M. Zeng, "Splat: Speech-language joint pre-training for spoken language understanding," *arXiv:2010.02295*, 2021.

[24] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5754–5758.

[25] Y. Tian and P. J. Gorinski, "Improving end-to-end speech-to-intent classification with reptile," *arXiv preprint arXiv:2008.01994*, 2020.

[26] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," *SLT*, 2018.

[27] Y. Huang, H. Kuo, S. Thomas, Z. Kons, K. Audhkhasi, B. Kingsbury, R. Hoory, and M. Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7984–7988.

[28] L.-s. Lee, J. Glass, H.-y. Lee, and C.-a. Chan, "Spoken content retrieval—beyond cascading speech recognition with text retrieval," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 9, pp. 1389–1420, 2015.

[29] Y.-S. Chuang, C.-L. Liu, H.-Y. Lee, and L. shan Lee, "Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering," *arXiv:1910.11559*, 2020.

[30] H.-K. J. Kuo, Z. Tüske, S. Thomas, Y. Huang, K. Audhkhasi, B. Kingsbury, G. Kurata, Z. Kons, R. Hoory, and L. Lastras, "End-to-end spoken language understanding without full transcripts," *arXiv preprint arXiv:2009.14386*, 2020.

[31] 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.

[32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[33] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[34] P. Price, "Evaluation of spoken language systems: The atis domain," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.