Audio-textual Architecture for Robust Spoken Language Understanding

Anonymous ACL submission

Abstract

Tandem spoken language understanding (SLU) systems suffer from the so-called automatic speech recognition (ASR) error propagation. In this work, we investigate how such problem impacts state-of-the-art NLU models such as BERT (Bidirectional Encoder Representations from Transformers) and RoBERTa. Moreover, a multimodal language understanding (MLU) system is proposed to mitigate SLU performance degradation due to error present in ASR transcripts. Our solution combines an encoder network to embed audio signals and the state-of-the-art BERT to process text transcripts. A fusion layer is also used to fuse audio and text embeddings. Two fusion strategies are explored: a pooling average of probabilities from each modality and a similar scheme with a fine-tuning step. The first approach showed to be the optimal solution to extract semantic information when the text input is severely corrupted whereas the second approach was slightly better when the quality of ASR transcripts was higher. We found that as the quality of ASR transcripts decayed the performance of BERT and RoBERTa also decayed, compromising the overall SLU performance, whereas the proposed MLU showed to be more robust towards poor quality ASR transcripts. Our model is evaluated on five tasks from three SLU datasets with different complexity levels, and robustness is tested using ASR outputs from three ASR engines. Results show that the proposed approach effectively mitigates the ASR error propagation problem across all datasets.

1 Introduction

Speech signals carry out the linguistic message, with speaker intentions, as well as his/her specific traits and emotions. As depicted in Figure 1-a, to extract semantic meaning from audio, tandem spoken language understanding (SLU) uses a pipeline that starts with an automatic speech recognizer (ASR) that transcribes the linguistic information into text, and a natural language understanding (NLU) module that interprets the ASR textual output. Such solutions offer several drawbacks (Serdyuk et al., 2018)(Bastianelli et al., 2020). First, the NLU relies on ASR transcripts to attain the semantic information. Because the ASR is not error-free, the NLU module needs to deal with ASR errors while extracting the semantic information (Simonnet et al., 2017)(Zhu et al., 2018)(Simonnet et al., 2018)(Huang and Chen, 2020). This is a major issue as error propagation significantly affects the overall SLU performance as shown in (Bastianelli et al., 2020).

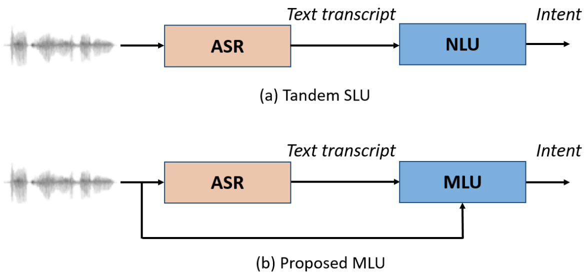

Figure 1: Tandem SLU vs proposed SLU architectures. The former relies solely on ASR transcripts to extract semantics whereas the latter fuses audio and text data to improve robustness of the SLU system.

Another drawback of such approaches is the fact that the two modules (ASR and NLU) are optimized independently with separate objectives (Serdyuk et al., 2018)(Agrawal et al., 2020). While the ASR is trained to transcribe the linguistic content, the NLU is optimized to extract the semantic information, commonly from clean text (Huang et al., 2020). Hence, the tandem approach is not globally optimal for the SLU task. To overcome this, end-to-end SLU (e2e SLU) solutions have been proposed as an alternative to the ASR-NLU pipeline (Haghani et al., 2018)(Lugosch et al., 2019). As pointed out in (Bastianelli et al., 2020), a recurrent problem of e2e SLU solutions is the scarcity of publicly available resources which leads to sub-optimal performance.

In this paper, we are interested in improving the robustness of tandem SLU systems. As depicted in Figure 1-b, this can be achieved by replacing the NLU by the so-called multimodal language understanding (MLU) module. Such MLU-based solution fuses text transcripts with their corresponding speech signal. We evaluate two fusion strategies. One based on a pooling average of probabilities from each modality and a similar approach with a fine-tuning step. The fusion is performed on the outputs of the text and speech encoders. Our results show that, for an error-free ASR, combining text and speech while extracting meaning from the user's utterance provides results as good as the tandem solution based on state-of-the-art NLUs. Experiments also show that our solution leads to SLU robustness as it mitigates performance degradation caused by noisy ASR transcripts. To confirm that, the SLU robustness was assessed on three SLU datasets with different complexity: (1) the Fluent Speech Command (FSC) dataset (Lugosch et al., 2019); (2) the SNIPS dataset (Saade et al., 2019); and (3) the recent released and challenging Spoken Language Understanding Resource Package (SLURP) dataset (Bastianelli et al., 2020). We also tested our solution using ASR trascripts from three off-the-shelf ASR engines. The contribution of this work can be summarized as follows. First, we show that state-of-the-art models, such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), although very successfull, are susceptible to the ASR error propagation problem. Second, to overcome that, we propose a multimodal architecture that uses speech information to leverage the performance of traditional tandem SLU solutions. Third, we show that such approach confers robustness to SLU solutions in presence of low quality ASR text transcription.

The remainder of this document is organized as follows. In Section 2, we review the related work on SLU and multimodal approaches. Section 3 presents the proposed method. Section 4 describes our experimental setup and Section 5 discusses our results. Section 6 gives the conclusion and future works.

2 Related Work

Joint ASR+NLU optimization. One drawback of tandem SLU solutions is that the ASR and the NLU are optimized separately. The literature offers different approaches to mitigate this problem. For example, in (Kim et al., 2017), the authors jointly train an online SLU and a language model. They show that a multi-task solution that learns to predict intent and slot labels together with the arrival of new words can achieve good performance in intent detection and language modeling with a small degradation on the slot filling task when compared to independently trained models. In (Haghani et al., 2018), the authors propose to jointly optimize both ASR and NLU modules to improve performance. Several e2e SLU encoder-decoder architectures are explored. It is shown that better performance is achieved when an e2e SLU solution that performs domain, intent, and argument predictions is jointly trained with an e2e model that learns to generate transcripts from the same audio input. This study provides two important considerations. First, joint optimization induces the model to learn from errors that matter more for SLU. Second, the authors also found from their experimental results that direct prediction of semantics from audio, neglecting the ground truth transcript, leads to sub-optimal performance.

End-to-end SLU. Recently, we have witnessed an increasing interest in minimizing SLU latency as well as the joint optimization problem with end-to-end (e2e) SLU models. Such solutions bypass the need of an ASR and extracts semantics directly from the speech signal. In (Lugosch et al., 2019), for example, the authors introduce the FSC dataset and present a pre-training strategy for e2e SLU models. Their approach is based on using ASR targets, such as words and phonemes, that are used to pre-train the initial layers of their final model. These classifiers once trained are discarded and the embeddings from the pre-trained layers are used as features for the SLU task. The authors show that improved performance on large and small SLU training sets was achieved with the proposed pre-training approach. Similarly, in (Chen et al., 2018), the authors propose to fine-tune the lower layers of an end-to-end CNN-RNN based model that learns to predict graphemes. This pre-trained acoustic model is optimized with the CTC loss and then combined with a semantic model to predict intents. A relevant and more recent research is presented in (Mhiri et al., 2020). In this work, the proposed speech-to-intent model is built based on a global max-pooling layer that allows for processing speech signals of varied length, also with the ability to process a given speech segment while receiving an upcoming segment from the same speech. In (Potdar et al., 2021), an end-to-end streaming SLU framework is proposed. With a unidirectional LSTM architecture, optimized with the alignment-free CTC loss, and pre-trained with the cross-entropy criterion, the authors show that their solution can predict multiple intentions in an online and incremental way. Their results are comparable to the performance of start-of-the-art non-streaming models for single-intent and multi-intent classification.

Multimodal SLU. A recurrent problem of e2e SLU solutions is the limited number of publicly available resources (i.e. semantically annotated speech data) (Bastianelli et al., 2020). Because there are much more NLU resources (i.e. semantically annotated text without speech), many efforts have been made towards transfer learning techniques that enable the extraction of acoustic embeddings that borrow knowledge from state-of-the-art language models such as BERT (Devlin et al., 2018). In (Huang et al., 2020), for instance, the authors propose two strategies to leverage performance of e2e speech-to-intent systems with unpaired text data. The first method consists of two losses: (1) one that optimizes the entire network based on text and speech embeddings, extracted from their respective pretrained models, and are used to classify intents; and (2) another loss that minimizes the mean square error between speech and text representations. This second loss only back-propagates to the speech branch as the goal is to make speech embeddings resemble text embeddings. The second method is based on a data augmentation strategy that uses a text-to-speech (TTS) system to convert annotated text to speech. In (Sarı et al., 2020), the authors show that the performance of a speech-only E2ESLU model can be improved by training the model with non-parallel audio-textual data. For that, the authors propose a multiview learning technique based on two unimodal branches consisting of an encoder for each modality. The unimodal branches receive either text or speech as input in order to produce the output. The authors first train the text branch as more resources are available. After, the classifier is frozen and the speech encoder is trained. As the final step, both branches are fine-tuned using parallel data and the shared classifier.

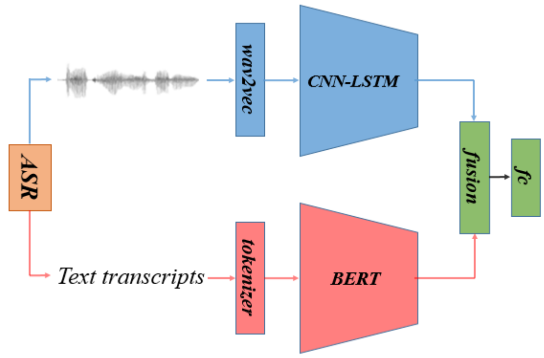

Figure 2: Diagram depicting the proposed multimodal language understanding (MLU) architecture used to predict semantic labels from audio-textual data.

3 Methodology

In this section, we start formally describing our task. We then present the proposed architecture and finalize introducing two strategies for performing the fusion of multimodal features.

3.1 General Principles

As a special case of SLU, spoken utterance classification (SUC) aims at classifying the observed utterance into one of the predefined semantic classes $L = \{l_1, ..., l_k\}$ (Masumura et al., 2018). Thus, a semantic classifier is trained to maximize the class-posterior probability for a given observation, $W = \{w_1, w_2, ..., w_j\}$, representing a sequence of tokens. This is achieved by the following probability:

$$L^* = \arg\max_L P(L|W, \theta) \quad (1)$$

where θ represents the parameters of the end-to-end neural network model. In this work, our assumption is that the robustness of such network can be improved if an additional modality, $X = \{x_1, x_2, ..., x_n\}$, representing acoustic features, is combined with the text transcript. Thus, Eq. (1) can be re-written as follow:

$$L^* = \arg\max_L P(L|W, X, \theta) \quad (2)$$

3.2 Architecture Overview

The proposed architecture consists of a speech encoder based on the pre-trained speech model, a convolutional module and a LSTM layer. As shown in Figure 2, the convolutional module and LSTM layer receive wav2vec embedded features as input and fine-tunes the speech representation for the

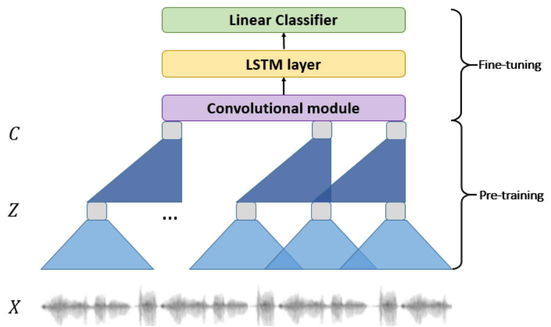

Figure 3: Models architecture combines the pre-trained wav2vec with a convolutional and lstm layers and a linear classifier.

downstream SLU task. This is referred to as our E2ESLU.

The text encoder, on the other hand, is based on the pretrained BERT. The encoders are trained separately on the downstream task. After the models optimization, late fusion is adopt to combine the two modalities.

3.3 Wav2vec Embeddings

We use the wav2vec model to extract deep semantic features from speech. While state-of-the-art models require massive amount of transcribed audio data to achieve optimal performance, wav2vec is an self-supervised pre-trained model trained on a large amount of unlabelled audio (Schneider et al., 2019). The motivation to adopt wav2vec relies on the fact that the model is able to learn a general audio representation that helps to leverage the performance of downstream tasks (Baevski et al., 2020). Thus, given an audio signal, $x_i \in \mathcal{X}$, a five-layer convolutional neural network, $f : \mathcal{X} \rightarrow \mathcal{Z}$, is applied in order to obtain a low frequency feature representation, $z_i \in \mathcal{Z}$, which encodes about 30 ms of audio at every 10 ms. Following, a context network, $g : \mathcal{Z} \rightarrow \mathcal{C}$, is applied to the encoded audio and adjacent embeddings, $z_i, ..., z_v$, are used to attain a single contextualized vector, $c_i = g(z_i, ..., z_v)$. A causal convolution of 512 channels is applied to the encoder and context networks and normalization is performed across the feature and temporal dimensions for each sample. Note that c_i represents roughly 210ms of audio context with each step i comprising a 512-dimensional feature vector (Baevski et al., 2020).

3.4 Convolutional LSTM Speech Encoder

In order to fine-tune the pre-trained wav2vec for the downstream task, a convolutional module and a LSTM layer is added on top of the context network, followed by a linear classifier that projects the hidden states from the LSTM into a set of L semantic labels. The architecture is depicted in Figure 3. Our convolution module is inspired in (Gulati et al., 2020) and consists of a gating mechanism mechanism, a point-wise convolution and a gated linear unit (GLU), which is followed by a single 1-D depthwise convolution layer. Batchnorm is deployed just after the convolution to aid training deep models. A single-layer LSTM is also used to further improve the speech representation and was found to be relevant for the downstream SLU task. The feature dimension in the LSRM layer is controlled with a projection layer as shown bellow:

$$\mathbf{s}_i = LSTM(\mathbf{c}_i), i \in \{1...N\} \tag{3}$$

$$\bar{\mathbf{s}}_i = W_{sp}\mathbf{s}_i \tag{4}$$

where $\mathbf{c}_i$ is the sequence of 512-dimensional feature representation from the convolutional layer, with i being the frame index. The hidden states of the unidirectional LSTM is represented by $\mathbf{s}_i$ which is a 1024-dimensional representation that undergoes a projection layer, W_{sp}, leading to $\bar{\mathbf{s}}_i$. The projection layer is an alternative LSTM architecture, proposed in (Sak et al., 2014), that minimizes the computational complexity of LSTM models. In our architecture, we project a 1024-dimensional features to half of this dimension. Thus, during the fine-tuning phase the speech encoder is optimized to output semantic labels using wav2vec embeddings as input.

3.5 Probability aggregation

In order to classify semantic labels using both audio and text information, we aggregate the output probabilities given by each modality for each class. Thus, multimodal predictions are attained based on the class with the highest averaged confidence. To achieve this, we first fine-tuned the speech encoder described in Section 3.4 and the BERT$_{large}$ model separately. We investigated two strategies. The first one, referred to as MLU$_{avg}$, is the cross entropy of the avaraged probabilities as described below:

$$p_l = \frac{e^{\bar{o}_l}}{\sum_{k=1}^{L} e^{\bar{o}_k}} \tag{5}$$

where $\bar{o}$ is the averaged probability for each class. In the second approach, we use the aggregated prob-

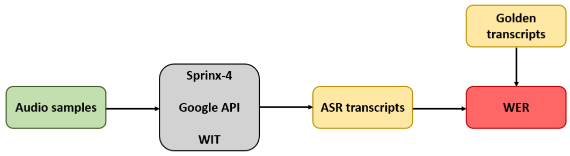

Figure 4: Pipeline for generating ASR text transcripts.

abilities to compute the cross-entropy loss in order to back-propagate it through our speech encoder.

4 Experimental Setup

In this section, the datasets used in our experiments are presented as well as the ASR engines adopted to investigate the impact of ASR error propagation on SLU. We then present our data augmentation strategy based on noise injection, followed by the experimental settings description.

4.1 Datasets

Three SLU datasets are used in our experiments. The reader is referred to Table 1 for partial statistics covering number of speakers, number of audio files, duration (in seconds), and utterance average length (in seconds). The first is the FSC dataset which comprises single-channel audio clips sampled at 16 kHz. The data was collected using crowdsourcing, with participants requested to cite random phrases for each intent twice. It contains about 19 hours of speech, providing a total of 30,043 utterances cited by 97 different speakers. The data is split in such a way that the training set contains 14.7 hours of data, totaling 23,132 utterances from 77 speakers. Validation and test sets comprise 1.9 and 2.4 hours of speech, leading to 3,118 utterances from 10 speakers and 3,793 utterances from other 10 speakers, respectively. The dataset has a total of 31 unique intent labels resulted in a combination of three slots per audio: action, object, and location. The latter can be either "none", "kitchen", "bedroom", "washroom", "English", "Chinese", "Korean", or "German". More details about the dataset can be found in (Lugosch et al., 2019).

SNIPS is the second dataset considered here. It contains a few thousand text queries. Recordings were crowdsourced and one spoken utterance was collected for each text query in the dataset. There are two domains available: smartlights (English) and smartspeakers (English and French). In our experiments only the former was used as it comprised only English sentences. With a reduced vocabulary size of approximately 400 words, the data contains

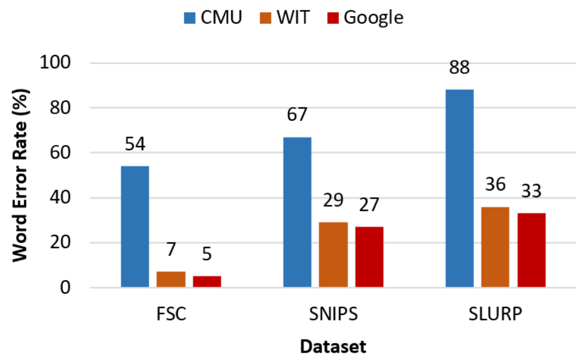

Figure 5: Word error rate (WER) based on true ASR engines (cmu, google, cloud and wit) for the three investigated datasets.

6 intents allowing to turn on or off the light, or change its brightness or color (Saade et al., 2019).

The recent released SLURP dataset is also considered in our experiments. It is a multi-domain dataset for end-to-end SLU and comprises approximately 72,000 audio recordings (58 hours of acoustic material), consisting of user interactions with a home assistant. The data is annotated with three levels of semantics: Scenario, Action and Intent, having 18, 56 and 101 classes, respectively. The dataset collection was performed by first annotating textual data, which was then used as golden transcripts for audio data collection. For that, 100 participants were asked to read out the collected prompts. This was performed in a typical home or office environment. Although SLURP offers distant and close-talk recordings, only the latter were used in our experiments. For more details about the dataset, the reader can refer to (Bastianelli et al., 2020).

	FSC	SNIPS	SLURP
# Speakers	97	69	177
# Audio files (headset)	30,043	2,943	34,603
# Audio files (Close-talk)	-	2,943	37,674
Duration [hs]	19	5.5	58
Avg. length [s]	2.3	3.4	2.9

Table 1: Statistics of audio samples for SLURP, SNIPS and FSC (Bastianelli et al., 2020).

Note that compared to other datasets, SLURP is much more challenging. The authors in (Bastianelli et al., 2020), directly compared SLURP to FSC and SNIPS in different aspects. For instance, SLURP contains 6x more sentences than SNIPS and 2.5x more audio samples than FSC. It also covers 9 times more domains and is 10 times lexically richer than both FSC and SNIPS. SLURP also provides

		FSC-I		SNIPS-I		SLURP-S		SLURP-A		SLURP-I	
Model	Modality	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
E2ESLU	S	95.20	95.21	63.54	63.41	63.88	63.88	57.28	56.77	50.28	50.05
BERT	T	99.99	100.00	98.26	98.26	91.98	92.07	90.24	90.19	86.59	86.38
RoBERTa	T	99.99	100.00	98.26	98.26	92.76	92.67	91.27	91.22	86.59	86.38
MLU_{avg}	S+T	99.97	99.97	94.09	94.10	89.95	89.75	89.95	89.75	84.61	83.92
MLU_{ft}	S+T	99.99	100.00	94.79	94.82	90.91	90.80	90.91	90.80	85.42	84.78

Table 2: Accuracy results for the SLURP, FSC and SNIPS datasets when gold transcripts are available for training and testing the NLU, MLU and the MLU with the attention mechanism.

a larger number of speakers compared to FSC and SNIPS. Next, we describe three ASR engines used to generate text transcripts. We also present the performance of these engines in terms of WER for each SLU dataset.

4.2 ASR engines

In order to evaluate the performance of our model in a more realistic setting, we simulate the generation of text transcripts from ASR engines as depicted in Figure 4. This is particularly important to assess the robustness of SLU models when golden transcripts are not available. The ASR systems adopted here are the open-source CMU SPHINX (Këpuska and Bohouta, 2017), developed at Carnegie Mellon University (CMU); the Google ASR API, which enables speech to text conversion in over 120 languages (Google, 2021); and the WIT engine (WIT, 2021), which is an online software platform that enables the development of natural language interfaces with support to more than 130 languages.

We evaluated the performance of these three ASR engines in terms of word error rate (WER) on three datasets and the results are presented in Figure 5. As expected, the SLURP revealed to be the most challenging dataset with the highest WER for all the three engines, followed by SNIPS and FSC. Note that, We chose datasets with different levels of complexity as well as ASR engines with diverse performance in order to evaluate our proposed MLU.

4.3 Experimental Settings

Our network is trained on mini-batches of 16 samples over a total of 200 epochs. Early-stopping is used in order to avoid overffiting, thus training is interrupted if the accuracy on the validation set is not improved after 20 epochs. Our model is trained using the Adam optimizer (Kingma and Ba, 2014), with the initial learning rate set to 0.0001 and a cosine learning rate schedule (Loshchilov and Hutter, 2016). Dropout probability was set to 0.3 and the parameter for weight decay was set to 0.002. Datasets are separated into training, validation and test sets and the hyperparameters are selected based on the performance on the validation set. All reported results are based on the accuracy on the test set.

Our experiments are based on 5 models: two NLU baselines based on BERT_{large} and RoBERTa_{large}; an E2ESLU; and two MLU proposed solutions, MLU_{avg} and MLU_{ft}. These models are trained to predict semantic labels for 5 tasks referred to as: FSC-I, SNIPS-I, SLURP-S, SLURP-A and SLURP-I. SLURP-S and SLURP-A denote scenario and action classification, respectively, and the remainder refer to intent classification.

5 Results

In this section, we present our experimental results. We start comparing the performance of the 5 aforementioned models in presence of golden transcripts. We then discuss the effects of ASR error propagation on the NLU baselines. Finally, we present the benefits of combining speech and text to overcome ASR transcript errors.

5.1 Combination of Speech and Text

In Table 2, we present the performance of the NLU baselines, the E2ESLU and the two MLU approaches. Performance is compared in terms of accuracy and f1 scores. Across all datasets, the E2ESLU approach provides the lowest accuracy compared to the NLU and MLU solutions. This is expected as models based solely on speech are harder to train as speech signals carry out not just variability due to the linguistic content, but also intra- and inter-speaker variability (Bent and Holt, 2017), as well as information from the acoustic ambience. The FSC-I showed to be the easiest task with accuracy and f1 scores as high as 100 %

Task	Engine	20 %		40 %		60 %		80 %		100 %	
		BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa	BERT	RoBERTa
FSC-I	CMU	89.53	89.74	79.24	80.43	69.51	71.12	58.04	60.54	50.12	52.67
	WIT	99.02	98.89	97.70	97.49	96.43	96.32	95.79	95.35	94.92	94.35
	Google	99.24	99.29	98.53	98.63	97.89	98.01	97.10	97.17	96.47	96.65
SNIPS-I	CMU	88.87	89.32	79.18	80.11	71.29	72.16	60.62	61.41	53.43	56.21
	WIT	97.22	96.52	95.15	95.82	93.07	94.48	91.33	91.46	89.27	90.04
	Google	97.91	96.88	96.53	95.15	94.82	95.14	93.75	91.73	93.05	90.68
SLURP-S	CMU	81.85	81.92	71.31	71.73	59.68	60.19	48.77	49.73	38.70	40.32
	WIT	90.26	90.97	88.65	89.09	87.06	87.73	85.43	86.08	83.96	84.63
	Google	90.31	90.77	88.74	89.32	87.07	87.60	85.70	86.46	84.51	85.01
SLURP-A	CMU	80.02	80.56	69.53	69.79	58.02	59.03	46.06	47.03	36.05	37.27
	WIT	88.02	89.17	86.06	86.84	83.00	84.64	81.33	82.75	79.70	80.99
	Google	87.82	88.67	85.92	86.96	83.28	84.37	81.71	82.83	79.81	81.05
SLURP-I	CMU	76.14	76.66	64.82	65.34	52.99	53.22	41.69	41.95	30.91	31.43
	WIT	84.48	84.88	82.33	82.72	80.54	80.78	78.57	79.01	77.06	77.57
	Google	84.14	84.65	82.58	82.91	80.54	80.72	78.91	79.12	77.52	77.98

Table 3: Effect of mixing golden transcripts with varying amount of ASR transcript output on our NLU model. We investigate SLURP, FSC and SNIPS datasets as well as three ASR engines: CMU, WIT and Google.

for all modalities, with a slight decay for speech-only, achieving 95.20 % and 95.21 % in terms of accuracy and f1 scores, respectively. The gap between the E2ESLU performance and the other solutions is more significant for the SNIPS and SLURP tasks. For instance, BERT and RoBERTa are able to achieve 98.26 % accuracy and f1 scores for intent classification on the SNIPS dataset while E2ESLU model achieves only 63.54 % and 63.41. Similar trend is observed for the SLURP tasks. Note that the MLU$_{ft}$ provides better performance when compared to the MLU$_{avg}$. One explanation is that the speech features are noisier (comprising much more variability as discussed above), the fine-tuning approach tends to rely more on text rather than on complementary information from the speech signal. These results show that, when golden transcripts are available, BERT and RoBERTa will provide optimal performance compared to the E2ESLU and the MLU proposed in this work. Results also show that the MLU will not compromise the performance, providing slight decay in terms of accuracy and f1 score, specially for the datasets with more hours of training data, such as the FSC and SLURP.

5.2 Impact of ASR Error Propagation on NLU

In Table 3, we investigate the impact of ASR error propagation into the NLU baselines, BERT and RoBERTa. For this, transcripts sampled from CMU, WIT and Google ASR engines were mixed with golden transcript samples. This was performed only for the test set as we assume no access to golden transcripts in realistic scenarios (i.e., beyond laboratory settings). We can observe a similar trend across all three datasets and five tasks. Performance decays as the number of ASR transcript samples increases. The performance on the FSC dataset is the least affected by ASR outputs. This is due to the fact that the FSC is a much less challenging dataset compared to SNIPS and SLURP, as discussed in (Bastianelli et al., 2020) and also shown in Figure 5. Comparing the performance of BERT and RoBERTa when golden transcripts are available and when 100 % of transcripts are from the ASR engines, we observe a decay of roughly 50 % for the academic ASR and 3 % when using the two commercial ASR engines. The NLU performance is also evaluated on the SNIPS-I task. We notice lower f1 score compared to the FSC-I, which is due to the characteristic of SNIPS, i.e., less samples available to train the model and overall a more challenging dataset as observed in Figure 5. The performance on the SLURP dataset is the most affected by noisy ASR transcripts. For the academic ASR engine, for example, performance in terms of f1 scores can get as low as 30.91 %, for the SLURP-I task, and as low as 37.27 % and 40.32 % for SLURP-S and SLURP-A tasks, respectively. When compared to the performance attained with golden transcripts, this represents a decay of 65 %, 59 % and 56 %, respectively. As shown in Figure 4 and discussed in (Bastianelli et al., 2020), SLURP is a more challenging SLU dataset. For the other two comercial ASR engines, the impact of ASR transcripts are much lower but still exists for the SLURP dataset, representing a decay in terms of accuracy of roughly 15 %, 11 % and 12 % for the SLURP-I, ALURP-S and SLURP-A tasks, respectively.

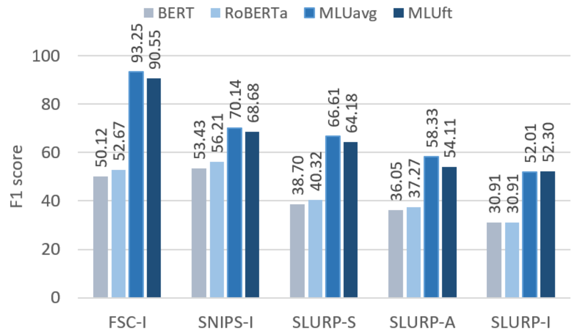

Figure 6: SLU performance when ASR transcripts from the CMU ASR engine is used during test.

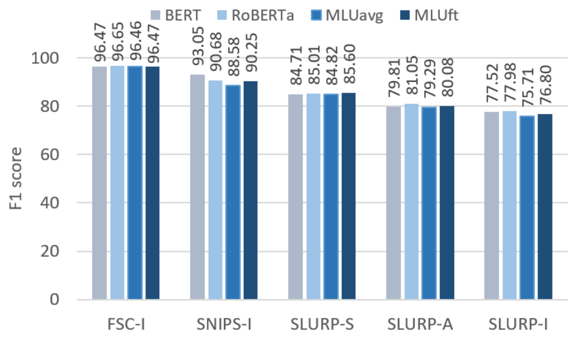

Figure 7: SLU performance when ASR transcripts from the google ASR engine.

5.3 SLU Robustness Towards ASR Error Propagation

In this section, we evaluate the robustness of the proposed MLU towards ASR error generated by the academic ASR engine, CMU, and by the commercial engine from Google. The results are presented respectively on Figures 6 and 7. As the commercial ASR engines have similar performance, we only present results from one of them. To evaluate a more realistic scenario, we assume no access to the golden transcripts during test. For all tasks, we observed that our model was more valuable for low quality ASR transcripts attained from the academic ASR (i.e. CMU engine), with the MLU$_{avg}$ providing better performance than the MLU$_{ft}$. We hypothesize that by finetuning the model tends to rely more on the text information. For the commercial ASR engine, which provide higher quality transcripts, performance of the proposed MLU is equivalent to text-only showing that it can be an alternative solution to mitigate the ASR error propagation without compromising performance when text transcripts are attained with high quality.

5.4 Limitations and Future Work

A limitation of this work is its results towards the more challenging SLURP dataset. Although we achieve competitive performance compared to the baseline results shared by the authors in (Bastianelli et al., 2020), results of our E2E SLU are way below. This corroborates with the findings in (Bastianelli et al., 2020), where several SOTA E2E SLU were tested and were not able to surpass the proposed modular (ASR+NLU) baselines as well. Note that the two baselines presented in (Bastianelli et al., 2020), are way more complex than our single-layer LSTM combined with word2vec embeddings. As for our MLU on the SLURP dataset, it was severely affected by the quality of the text transcripts.

As future work, we plan to propose a low-latency MLU architecture. We will adapt and evaluated the proposed MLU model for a streaming scenario where chunks of speech and text are processed in an online fashion and predictions of semantic labels are incrementally performed.

6 Conclusion

In this paper, we propose a multimodal language understanding (MLU) architecture, which combines speech and text to predict semantic information. Our main goal is to mitigate ASR error propagation into traditional NLU. The proposed model combines an encoder network to embed audio signals and the state-of-the-art BERT to process text transcripts. Two fusion approaches are explored and compared. A pooling average of probabilities from each modality and a similar scheme with a fine-tuning step. Performance is evaluated on 5 SLU tasks from 3 dataset, namely, SLURP, FSC and SNIPS. We also used three ASR engines to investigate the impact of transcript errors and the robustness of the proposed model when golden transcripts are not available. We first show that our model can achieve comparable performance to state-of-the-art NLU models. We evaluated the robustness of our towards ASR transcripts. Results show that the proposed approach can robustly extract semantic information from audio-textual data, outperforming BERT$_{large}$ and RoBERTa$_{large}$ for low quality text transcripts from the academic CMU ASR engine. For the commercial ASR engines, we show that the MLU can be an alternative solution as it does not compromise the overall SLU performance.

References

Bhuvan Agrawal, Markus Müller, Martin Radfar, Samridhi Choudhary, Athanasios Mouchtaris, and Siegfried Kunzmann. 2020. Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding. *arXiv preprint arXiv:2011.09044*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. Slurp: A spoken language understanding resource package. *arXiv preprint arXiv:2011.13205*.

Tessa Bent and Rachael F Holt. 2017. Representation of speech variability. *Wiley Interdisciplinary Reviews: Cognitive Science*, 8(4):e1434.

Yuan-Ping Chen, Ryan Price, and Srinivas Bangalore. 2018. Spoken language understanding without speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6189–6193. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Google. 2021. Google asr api. Accessed on 13-August-2021].

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Parisa Haghani, Arun Narayanan, Michiel Bacchiani, Galen Chuang, Neeraj Gaur, Pedro Moreno, Rohit Prabhavalkar, Zhongdi Qu, and Austin Waters. 2018. From audio to semantics: Approaches to end-to-end spoken language understanding. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 720–726. IEEE.

Chao-Wei Huang and Yun-Nung Chen. 2020. Learning asr-robust contextualized embeddings for spoken language understanding. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8009–8013. IEEE.

Yinghui Huang, Hong-Kwang Kuo, Samuel Thomas, Zvi Kons, Kartik Audhkhasi, Brian Kingsbury, Ron Hoory, and Michael Picheny. 2020. 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)*, pages 7984–7988. IEEE.

Veton Këpuska and Gamal Bohouta. 2017. Comparing speech recognition systems (microsoft api, google api and cmu sphinx). *Int. J. Eng. Res. Appl*, 7(03):20–24.

Young-Bum Kim, Sungjin Lee, and Karl Stratos. 2017. Onenet: Joint domain, intent, slot prediction for spoken language understanding. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 547–553. IEEE.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2016. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech model pre-training for end-to-end spoken language understanding. *arXiv preprint arXiv:1904.03670*.

Ryo Masumura, Yusuke Ijima, Taichi Asami, Hirokazu Masataki, and Ryuichiro Higashinaka. 2018. 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)*, pages 6039–6043. IEEE.

Mohamed Mhiri, Samuel Myer, and Vikrant Singh Tomar. 2020. A low latency asr-free end to end spoken language understanding system. *arXiv preprint arXiv:2011.04884*.

Nihal Potdar, Anderson R Avila, Chao Xing, Dong Wang, Yiran Cao, and Xiao Chen. 2021. A streaming end-to-end framework for spoken language understanding. *arXiv preprint arXiv:2105.10042*.

Alaa Saade, Joseph Dureau, David Leroy, Francesco Caltagirone, Alice Coucke, Adrien Ball, Clément Doumouro, Thibaut Lavril, Alexandre Caulier, Théodore Bluche, et al. 2019. Spoken language understanding on the edge. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing-NeurIPS Edition (EMC2-NIPS)*, pages 57–61. IEEE.

Haşim Sak, Andrew Senior, and Françoise Beaufays. 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*.

Leda Sarı, Samuel Thomas, and Mark Hasegawa-Johnson. 2020. Training spoken language understanding systems with non-parallel speech and text.

In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8109–8113. IEEE.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Dmitriy Serdyuk, Yongqiang Wang, Christian Fuegen, Anuj Kumar, Baiyang Liu, and Yoshua Bengio. 2018. Towards end-to-end spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5754–5758. IEEE.

Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, and Yannick Estève. 2018. Simulating asr errors for training slu systems. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Edwin Simonnet, Sahar Ghannay, Nathalie Camelin, Yannick Estève, and Renato De Mori. 2017. Asr error management for improving spoken language understanding. *arXiv preprint arXiv:1705.09515*.

WIT. 2021. Build natural language experiences. Accessed on 13-August-2021].

Su Zhu, Ouyu Lan, and Kai Yu. 2018. Robust spoken language understanding with unsupervised asr-error adaptation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6179–6183. IEEE.