Multimodal 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 problem. Additionally, as the ASR is not optimized to extract semantics, but solely the linguistic content, relevant semantic cues might be left out of its transcripts. In this work, we propose a multimodal language understanding (MLU) architecture to mitigate these problems. Our solution is based on two compact unidirectional long short-term memory (LSTM) models that encode speech and text information. A fusion layer is also used to fuse audio and text embeddings. Two fusion strategies are explored: a simple concatenation of these embeddings and a cross-modal attention mechanism that learns the contribution of each modality. The first approach showed to be the optimal solution to robustly extract semantic information from audio-textual data. We found that attention is less effective at testing time when the text modality is corrupted. Our model is evaluated on three SLU datasets and robustness is tested using ASR outputs from three off-the-shelf ASR engines. Results show that the proposed approach effectively mitigates the ASR error propagation problem for 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, 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).

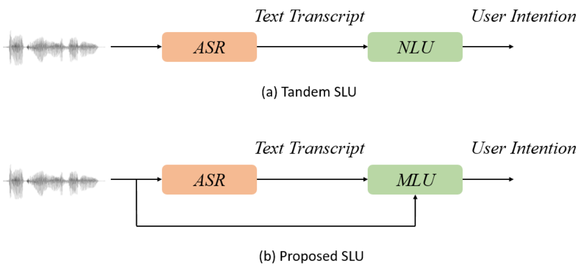


Figure 1: Tandem SLU vs proposed SLU architectures. The former relies solely on ASR transcripts to extract semantics whereas the latter fuses audio and text data to improve robustness of the SLU system.

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 (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, 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 simple concatenation of text and audio embeddings and the other one base on a cross-modal attention layer. 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 can help to improve performance. Experiments also show that our solution leads to SLU robustness as it helps to mitigate performance degradation caused by noisy ASR transcripts. To confirm that, the SLU robustness was assessed on three SLU datasets with different complexity: (1) the SNIPS dataset (Saade et al., 2019); (2) the Fluent Speech Command (FSC) dataset (Lugosch 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 propose a multimodal architecture that uses speech information to leverage the performance of traditional tandem SLU solutions. Second, we show that such approach confers robustness to SLU solutions by mitigating performance degradation due to ASR error propagation.

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 e2e SLU 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.

3 Proposed Model

3.1 Spoken Utterance Classification

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:

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

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:

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

3.2 Architecture Overview

We adopt two compact unidirectional long short-term memory (LSTM) encoders to process speech and text modalities independently. As shown in Figure 2, the LSTM speech encoder receives wav2vec embedded features as input and fine-tunes the speech representation for the downstream SLU task. Likewise, word2vec text embeddings are provided to the LSTM text encoder which further enhance the text representation. The whole model then consists of a speech encoder, a text encoder, and a fusion layer. Instead of an over-parameterized LSTM encoders, we choose a compact approach that we detail in the following sections.

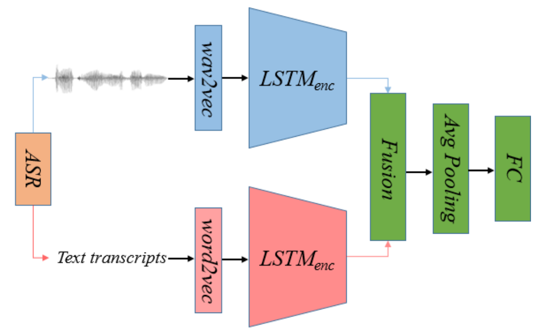


Figure 2: Diagram depicting the proposed multimodal language understanding (MLU) architecture used to predict semantic labels from audio-textual data. As fusion strategies, we explore (1) a simple concatenation of the two modalities and (2) a soft alignment between speech and text modalities which is achieved using a cross-modal attention layer that projects the speech onto the text space.

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 unsupervised 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 (Schneider et al., 2019). 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, $c_i = g(z_i, ..., z_v)$. Note that c_i represents roughly 210ms of audio context with each step i comprising a 512-dimensional feature vector (Schneider et al., 2019).

3.4 LSTM Speech Encoder

A single-layer LSTM is used to further improve the speech representation from wav2vec for the downstream SLU task. Thus, it takes wav2vec embeddings as input and is optimized to output semantic labels such as slot values and intents. The feature dimension 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 wav2vec feature representation, 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.

3.5 LSTM Text Encoder

The text encoder takes word embeddings as input and is trained on the downstream task to output semantic labels in a similar way as the speech encoder. A single-layer LSTM is adopted to capture temporal context from the input text representation, as shown bellow:

$$\mathbf{h}_j = LSTM(\mathbf{e}_j), j \in \{1...M\} \tag{5}$$

where $\mathbf{e}_i$ is a sequence of 256-dimensional word representation, with j being the word index in a sentence. The hidden states of the unidirectional LSTMs are represented by $\mathbf{h}_j$ which is a 512-dimensional feature representation.

3.6 Cross-modal Fusion Layer

The cross-modal fusion layer receives output from the speech and text encoders. Note that the feature representation from these encoders, $\bar{\mathbf{s}}_i$ and $\mathbf{h}_j$, are 512-dimensional vectors with different timestep lengths, denoted by N and M, respectively, for speech and text modalities. The cross-modal fusion layer then comprises a simple concatenation of speech and text embeddings, as shown below:

$$\mathbf{o} = [\text{mean-pooling}(\bar{\mathbf{s}}_i), \text{mean-pooling}(\mathbf{h}_j)] \tag{6}$$

where $\mathbf{o}$ is a fixed-length vector attained after applying average pooling on the hidden states of $\bar{\mathbf{s}}_i$ and $\mathbf{h}_j$. As suggested in (Lin et al., 2020), mean-pooling can be used to attain the high-level semantic representation within an utterance. In our case, it also solves the alignment issue between speech and text modalities as they are based on length. Note that $\mathbf{o}$ undergoes a linear transformation prior to computing softmax with cross entropy for classification, as follow:

$$\bar{\mathbf{o}} = W^{\top}\mathbf{o}, \bar{\mathbf{o}} \in R^L \tag{7}$$

$$p_l = \frac{e^{\bar{o}_l}}{\sum_{k=1}^{L} e^{\bar{o}_k}} \tag{8}$$

$$\mathcal{L} = -\sum_{l=1}^{L} y_l \log p_l \tag{9}$$

where W is a matrix with trainable parameters and $\bar{o}_l$ is the l-th element in $\bar{o}$, and y_l is 1 for the ground-truth label and 0 otherwise.

3.7 Cross-modal Attention Fusion Layer

The cross-modal attention layer investigated here receives output from the speech and text encoders, $\bar{\mathbf{s}}_i$ and $\mathbf{h}_j$. The motivation to apply the attention mechanism is two fold. First, it helps to optimize the model taking into account the contribution of

	SLURP	SNIPS	FSC
# 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).

each modality for the downstream task. Moreover, it develops a context matrix of attention weights that are used to learn the soft alignment between speech and text modalities, as proposed in (Xu et al., 2019). This is attained by projecting the speech representation onto the text space. For instance, the attention weight between the speech frame i and the word embed j can be calculated using the hidden state $\bar{\mathbf{s}}_i$ of the speech LSTM encoder and the hidden state $\mathbf{h}_j$ of the text LSTM encoder (Xu et al., 2019), as shown bellow:

$$a_{j,i} = \tanh(\mathbf{u}^\top \bar{\mathbf{s}}_i + \mathbf{v}^\top \mathbf{h}_j + b) \tag{10}$$

$$\alpha_{j,i} = \frac{e^{a_{j,i}}}{\sum_{t=1}^{N} e^{a_{j,t}}} \tag{11}$$

$$\tilde{\mathbf{s}}_j = \sum_i \alpha_{j,i} \bar{\mathbf{s}}_i \tag{12}$$

with $\mathbf{u}$, $\mathbf{v}$ and b being learnable parameters. In Eq. (11) the normalized attention weight, $\alpha_{j,i}$, is attained, representing the soft alignment strength between the j-th word and the i-th speech frame. Note that the alignment between speech feature vectors corresponding to the j-th word is the weighted summation of hidden states from the speech LSTM econder which is denoted by $\tilde{\mathbf{s}}_j$ in Eq. (12). The final part comprises an average pooling, as described bellow:

$$\tilde{\mathbf{o}} = \text{mean-pooling}([\tilde{\mathbf{s}}_1, ..., \tilde{\mathbf{s}}_M]) \tag{13}$$

where $\tilde{\mathbf{o}}$ is a fixed-length vector, attained after applying average pooling on $\tilde{\mathbf{s}}_j$, that undergoes a linear transformation similar to the one discussed in the previous section.

4 Experimental Setup

4.1 Datasets

Three SLU datasets are used in our experiments. The reader is referred to Table 1 for partial statistics of audio samples for these datasets. 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 comprises 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”. In our experiments, we defined intent as the combination of action and object, which led to a total of 15 different intent labels. Location was defined as slot, which led to a total of 8 different slot labels. More details about the dataset can be found in (Lugosch et al., 2019).

SNIPS is the second dataset considered. 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 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.,

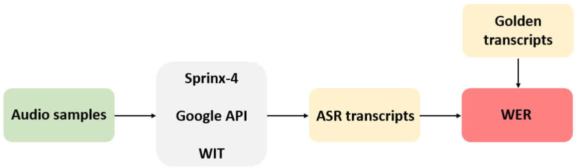


Figure 3: Pipeline for generating ASR text transcripts.

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 10 times lexically richer than both FSC and SNIPS. SLURP also provides 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 3. This is particularly important to assess the robustness of SLU models when golden transcripts are not available.

4.3 Noise Injection

Introducing noise into a neural network input is a form of data augmentation that improves robustness and leads to better generalization (Coulombe, 2018). To increase the robustness of our proposed model, we injected noise word into the training set. We used lexical replacement which consists of proposing one or more words that can replace a given word. Thus, we choose a random word from the vocabulary V with the main constraint to not be the target word w in an utterance. This was achieved by perturbing golden transcripts by adding, dropping, or replacing a few words in a sentence. During training, we randomly selected 30 % of sentences within a batch to be corrupted with noise. Moreover, only 1/3 of words within a sentence were corrupted.

4.4 Experimental Settings

Our network is trained on mini-batches of 16 samples over a total of 200 epochs. Early-stopping

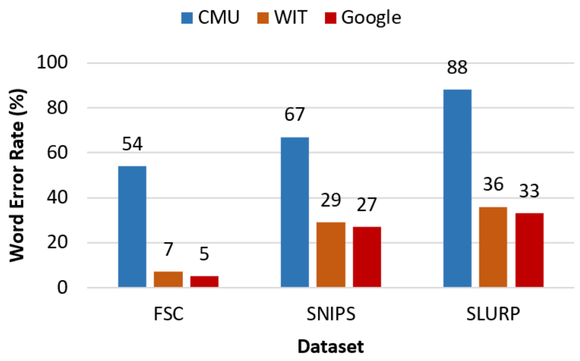


Figure 4: Word error rate (WER) based on true ASR engines (cmu, google, cloud and wit) for the three investigated datasets.

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 was trained using the Adam optimizer (Kingma and Ba, 2014), with the initial learning rate set to 0.0001. 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 4 models trained to predict semantic labels: (1) the NLU baseline; (2) the E2E SLU; (3) the MLU and (4) the MLU(ATT) that uses attention mechanism. The first model is based on the text LSTM encoder and is trained with text-only (TO). The second model is based on the speech LSTM encoder and is trained with speech-only (SO). The two MLU models are based on text and speech and use output embeddings from the pre-trained LSTM encoders mentioned before.

5 Results

5.1 Impact of ASR Error Propagation on NLU

In Table 2, we investigate the impact of the ASR error propagation into our NLU model. 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 and we can observe a similar trend across all datasets and tasks. Performance decays as the number of ASR transcript samples increases. The performance on the FSC dataset is least affected by ASR outputs, specially when the transcripts from the commer-

Task	Engine	20 %	40 %	60 %	80 %	100 %
				FSC		
Intent	CMU	90.79	82.75	74.05	65.92	57.27
	WIT	99.23	98.44	97.83	96.75	96.38
	Google	99.23	98.44	98.15	97.41	96.94
Slot	CMU	95.75	91.85	87.34	82.41	77.92
	WIT	99.26	99.07	98.02	97.62	96.88
	Google	99.57	98.41	98.73	98.73	98.31
				SNIPS		
Intent	CMU	84.94	80.93	69.56	60.53	51.5
	WIT	93.64	94.31	91.63	90.63	88.29
	Google	95.31	90.96	89.29	87.28	83.61
				SLURP		
Scenario	CMU	73.71	62.52	49.89	38.85	30.46
	WIT	83.42	81.66	79.00	77.34	75.47
	Google	83.42	82.07	80.34	78.00	76.69
Action	CMU	71.77	60.90	49.21	37.78	29.22
	WIT	80.56	78.44	76.42	74.01	72.53
	Google	80.87	78.39	76.58	73.89	72.43
Intent	CMU	66.66	54.52	42.64	30.65	21.57
	WIT	76.11	73.18	70.54	67.71	65.77
	Google	76.47	73.74	71.29	68.72	66.86

Table 2: 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.

cial ASR engines were used. This is because the FSC is less challenging compared to the other two datasets, as discussed in (Bastianelli et al., 2020) and also shown in Figure 4. For the academic ASR engine, CMU, we observe a decay of 42 % for intent classification and 22 % for slot prediction. The NLU performance is also evaluated on the SNIPS dataset. We first notice a lower accuracy compared to the FSC dataset and this is due to the characteristic of SNIPS, i.e., less samples to train the neural network and overall a slightly more challenging dataset as observed in Table 2. The performance on the SLURP dataset is the most affected by noisy ASR transcripts. For the academic ASR engine, for example, performance can get as low as 21 %, for intent prediction, and as low as 30 % for scenario and action predictions, representing a decay in performance of approximately 72 %, 64 % and 64 %, respectively. As shown in Figure 3 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 intent, scenario and action predictions respectively.

Model	FSC		SNIPS	SLURP		
	Intent	Slots	Intent	Scenario	Action	Intent
E2ESLU	99.41	99.39	63.87	69.98	60.80	58.22
NLU	100.00	100.00	**95.98**	86.85	83.24	78.59
MLU	100.00	100.00	93.31	**87.67**	**84.26**	**78.72**
MLU(ATT	100.00	100.00	92.64	85.42	81.14	74.68

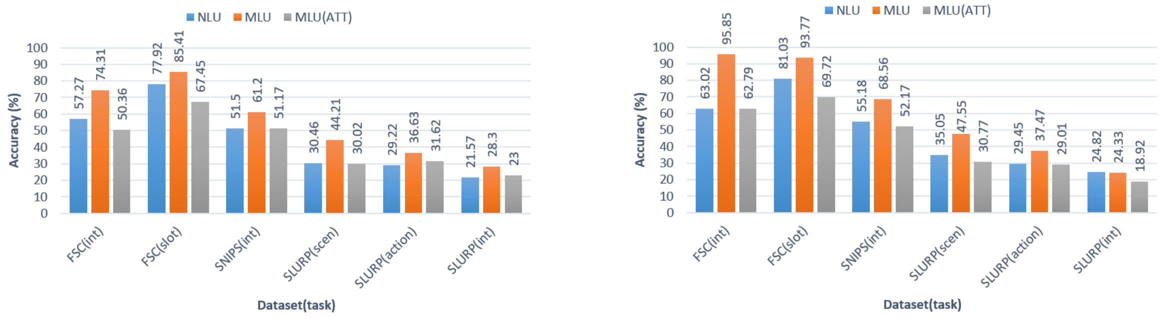
Table 3: Accuracy results for the SLURP, FSC and SNIPS datasets when gold transcripts are available for training and testing the NLU, MLU and the MLU with the attention mechanism.

5.2 Combination of Speech and Text

In Table 3, we compare the performance of the NLU baseline, E2E SLU and the two MLU approaches. Across all datasets, the E2E SLU provided lower accuracy compared to the NLU and MLU solutions. This is expected such solutions are harder to train because speech signals accommodates not just variability due to the linguistic information, but also intra- and inter-speaker variability (Bent and Holt, 2017), and information from the acoustic environment. Not surprisingly, the FSC showed to be the easiest task with accuracy as high as 100 % for all modalities, with a slight decay for speech-only, achieving 99.41 % and 99.39 % accuracy for intent and slot classification, respectively. The gap between E2E SLU and the other modalities is more significant for the SNIPS and SLURP, with the former being linguistically more challenging. For instance, our TO model is able to achieve 95.98 % accuracy for intent classification on the SNIPS dataset while our SO model achieves only 63.87 %. Similar trend is observed for the SLURP tasks. Note that the MLU provides better performance when compared to the MLU(ATT). One explanation is that the speech features are noisier (comprising much more variability), and the attention weights tend to lean more towards text, neglecting complementary information from the speech signal. The MLU approach without attention also outperformed our NLU model for the SLURP dataset. The best performance for the SNIPS dataset was achieved with the NLU approach.

5.3 SLU Robustness Towards ASR Error Propagation

In Figure 5, we evaluate the robustness of the proposed MLU towards ASR error propagation. To evaluate a more realistic scenario, results are reported using 100 % of ASR output, i.e., we assume no access to golden transcripts during test. We considered two approaches during training: with



(a) Without noise injection

(b) With noise injection

Figure 5: SLU performance when ASR transcripts from the CMU ASR engine is used during test and training is performed (a) without noise injection and (b) with noise injection.

and without noise injection. In all experiments, we found that introducing noise during training (see Figure 5-b) was beneficial and helped to increase robustness. We also observed that our model was more valuable for low quality ASR transcripts attained from the academic ASR (i.e. CMU engine) and results are shown in Figure 5. For the commercial ASR engines, which provide higher quality transcripts, performance of the proposed MLU without attention is equivalent to text-only and slightly better in some cases.

5.4 Limitations and Future Work

A clear limitation of this work is its results towards the larger and 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 was to mitigate ASR error propagation into traditional NLU. The proposed model is based on two unidirectional LSTM encoders that learn speech and text representation, respectively. Two fusion approaches are explored and compared. The first one is based on a cross-modal attention mechanism, which is meant to align speech and text embeddings attained from the speech and text encoders. The second one is based on a simple concatenation of speech and text embeddings averaged over the LSTM timesteps. Performance is evaluated on 3 dataset, namely, SLURP, FSC and SNIPS. We also used three out-of-the-shelf 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 outperforms the text-only as well as the audio-only modules when golden transcripts are used as input. For instance, the proposed model achieves 87.16 %, 83.75 % and 79.18 % accuracy for scenario, action and intent classification in SLURP dataset, respectively, outperforming text-only and speech-only for the first two tasks. We also evaluated the robustness of our towards ASR transcripts. Results show that the proposed approach can robustly extract semantic information from audio-textual data.

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*.

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.

Claude Coulombe. 2018. Text data augmentation made simple by leveraging nlp cloud apis. *arXiv preprint arXiv:1812.04718*.

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*.

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.

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*.

Ting-En Lin, Hua Xu, and Hanlei Zhang. 2020. Discovering new intents via constrained deep adaptive clustering with cluster refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8360–8367.

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*.

Haiyang Xu, Hui Zhang, Kun Han, Yun Wang, Yiping Peng, and Xiangang Li. 2019. Learning alignment for multimodal emotion recognition from speech. *arXiv preprint arXiv:1909.05645*.

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.