MTL-SLT: Multi-Task Learning for Spoken Language Tasks

Zhiqi Huang[1], **Milind Rao**[2], **Anirudh Raju**[2], **Zhe Zhang**[2], **Bach Bui**[2], **Chul Lee**[2]

[1]Peking University, China
[2]Amazon Alexa, USA
zhiqihuang@pku.edu.cn, {milinrao, ranirudh}@amazon.com

Abstract

Language understanding in speech-based systems has attracted extensive interest from both academic and industrial communities in recent years with the growing demand for voice-based applications. Prior works focus on independent research by the automatic speech recognition (ASR) and natural language processing (NLP) communities, or on jointly modeling the speech and NLP problems focusing on a single dataset or single NLP task. To facilitate the development of spoken language research, we introduce MTL-SLT, a multi-task learning framework for spoken language tasks. MTL-SLT takes speech as input, and outputs transcription, intent, named entities, summaries, and answers to text queries, supporting the tasks of spoken language understanding, spoken summarization and spoken question answering respectively. The proposed framework benefits from three key aspects: 1) pre-trained sub-networks of ASR model and language model; 2) multi-task learning objective to exploit shared knowledge from different tasks; 3) end-to-end training of ASR and downstream NLP task based on sequence loss. We obtain state-of-the-art results on spoken language understanding tasks such as SLURP and ATIS. Spoken summarization results are reported on a new dataset: Spoken-Gigaword.

	Input
Speech	I am going to the airport tomorrow, please turn off bedroom light at nine thirty pm.
Q1	When should I turn off bedroom light?
Q2	When do I go to the airport?
	Output
Sum.	turn off the bedroom light
Intent	hue_lightoff
Slots	[date : tomorrow], [time : nine thirty pm] [house_place : bedroom],
Ans1.	nine thirty pm
Ans2.	tomorrow

Table 1: An example of multiple spoken language tasks. Given input utterances in the form of speech, the ASR-NLP system can provide a summary of the speech (summarization), intent detection and named entity recognition (language understanding) and answer textual queries. The spoken question answering task requires additional questions as input.

1 Introduction

The wide deployment of voice controlled computing has led to extensive interest in spoken language tasks in recent years (Saade et al., 2019; Bastianelli et al., 2020; Li et al., 2018). For instance, spoken language understanding aims to extract the semantics from user queries (Chung et al., 2021; Kim et al., 2021a; Lai et al., 2021), spoken question answering aims to predict the answer given the spoken context (You et al., 2021; Kuo et al., 2020). The rapid development of spoken language tasks have followed dataset releases (Zhang et al., 2020; Liu et al., 2019) and the evolution of pre-trained models (Devlin et al., 2019; Lewis et al., 2020; Chuang et al., 2020).

Multi-task learning (MTL) (Caruana, 1997) focuses on simultaneously solving multiple related tasks and has attracted much attention in recent years. Compared with single-task learning, it can reduce the training and inference time while improving generalization performance and prediction accuracy by learning a shared representation across related tasks. Prior works show the effectiveness of MTL while they only focus on multiple text-based tasks/datasets (e.g., MT-DNN (Liu et al., 2019; Wang et al., 2019)) or multiple speech-based tasks/datasets (e.g., SpeechStew (Chan et al., 2021)). Also, some works (Raju et al., 2021; Rao et al., 2021) prove the effectiveness of considering speech information when performing NLP tasks. Thus, as can be seen in Figure 1, we argue that it is helpful when extend these MTL approaches to spoken language tasks (i.e., ASR-NLP-shared).

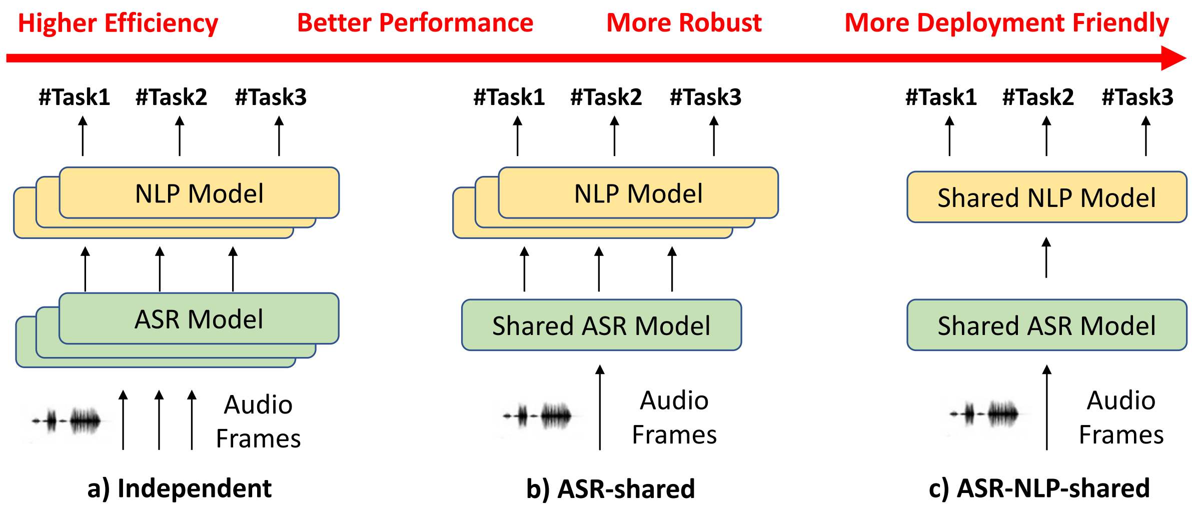

Figure 1: Different Implementations of Spoken Language Tasks.

In this paper, we develop multi-task learning methods to optimize spoken summarization, spoken question answering, spoken language understanding (intent classification and slot filling), as well as speech recognition on multiple spoken language datasets. An example of an application with these four tasks can be seen in Table 1. Note that instead of performing experiments only on understanding task (e.g., Feng et al. (2021)), we also consider harder generation task into our framework, whose data distribution has significant difference to classification task (Observation can be witnessed from Figure 2, the purple points are far away from the other data).

A primary challenge with audio as an input modality is the impact of speech recognition errors and acoustic noise on spoken language tasks. To mitigate this, our approach jointly optimizes pretrained speech recognition and language models for semantic metrics of interest and we train across multiple language tasks. The various language tasks and the impact of multi-task training can be visualized in the clustering plot of the hidden state of a pretrained language model in Figure 2. We demonstrate our results using listen-attend-spell (LAS) (Chan et al., 2016) speech recognition model and a BART (Lewis et al., 2020) based NLP model.

Overall, the main contributions are as follows:

- We propose a MTL-SLT framework to effectively joint train an ASR model and an NLP model on multiple spoken language tasks.

- Experimental results show that our proposed multi-task learning framework is state-of-the-art on spoken language understanding tasks. Training multiple language tasks followed by task-specific finetuning yields optimal models. Jointly training ASR and NLP with policy gradient methods improves metrics on all spoken language tasks.

- We prepare a spoken summarization dataset based on the Gigaword dataset (Rush et al., 2015) using a multi-speaker text-to-speech (TTS) model. The performance of the introduced spoken-summarization task with the MTL framework is studied.

- Our approach extends to multiple NLP tasks, providing improvements in an end-to-end spoken language learning setting. We make our code and data publicly available for researchers to accelerate the development of related spoken language tasks.

2 Related Work

MTL MTL aims to improve the performance on a set of primary tasks through an inductive bias (Caruana, 1997) introduced by additional training objectives on auxilliary tasks. MTL has also been used to train several tasks jointly, without the notions of primary and auxilliary tasks (McCann et al., 2018). MTL approaches for deep learning include hard parameter sharing where the entire layers and parameters are shared between tasks; and soft parameter sharing, where each task has it's own model parameters but the distance between the model parameters is regularized to help the task-specific parameters to be similar (Ruder, 2017).

Pre-trained Models The paradigm of pre-training a language model (LM) followed by task-specific fine-tuning has been shown to obtain remarkable performance on many NLP tasks. BERT (Devlin et al., 2019) pre-trains deep bidirectional representations from unlabeled text and showed competitive performance on the GLUE (Wang et al., 2019) benchmark. This provided a base for researchers to build upon, leading to several extensions and rapid progress in the space of pre-trained LMs. The MultiTask Deep Neural Network (Liu et al., 2019) is one such extension with multi-task learning across all GLUE tasks. The paper argues for improved domain transfer by performing standard BERT pretraining, followed by multi-task learning and task-specific fine-tuning. BERT has been leveraged for various NLP tasks, for e.g. the effectiveness of BERT for the summarization task was explored by Liu and Lapata (2019). The performance of text generation tasks have been approaching a near-human level by virtue of pre-trained encoder-decoder models, such as BART (Lewis et al., 2020) and T5 (Raffel et al., 2020).

Spoken Language Tasks Spoken language tasks include standard NLP tasks with speech-input instead of text-input. Speech recognition errors can impact the performance of downstream NLP systems. Recently, Feng et al. (2021) proposed the ASR-GLUE benchmark, augmented 6 NLP tasks from GLUE with speech generated from Google TTS, and analyzed the robustness of NLP to ASR errors. However, all 6 tasks are sentence-level classification problems, and the models did not utilize MTL framework. Chung et al. (2021) introduced a speech-language joint pre-training framework for SLU tasks. The paper showed the effectiveness of the joint pre-training method with experiments on four classification tasks, i.e., intent detection, dialog act classification, spoken sentiment analysis and spoken question answering. Prior works for SLU show the impact of speech recognition errors on downstream Natural Language Understanding (NLU) performance and propose joint training of ASR and NLU to improve overall performance (Rao et al., 2021). Kim et al. (2021b) introduced a speech-based benchmark for task-oriented dialogue systems, specifically targeting the problems of multi-domain dialogue state tracking and knowledge grounded dialogue modeling, and showed that well-behaved models trained on written conversations do not perform well on spoken data.

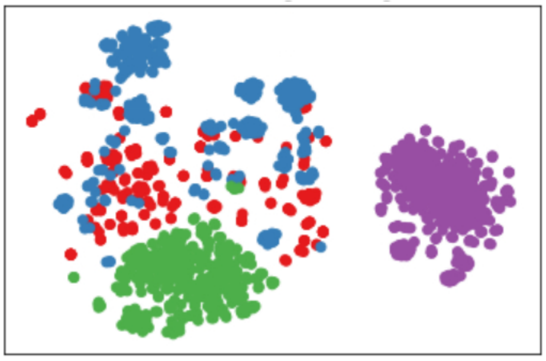
Figure 2: T-SNE Visualization of BART's last hidden state features. Red and blue represent ATIS and SLURP datasets, green denotes Spoken-SQuAD dataset, purple denotes Spoken-Gigaword dataset.

3 Approach

3.1 Architecture of MTL-SLT

Figure 3 shows the proposed MTL framework which consists of three different modules, i.e., the ASR model, the NLP model and the interface between them. In this work, the MTL-SLT uses the LAS architecture for ASR and BART for NLP.

ASR Model Unlike previous works on spoken language tasks (SLT) that obtain transcriptions using existing ASR systems/tools (Feng et al., 2021; Li et al., 2018), in our approach, the ASR model is updated with the training of end-to-end spoken language tasks. To address this, we generate the ASR transcriptions from a LAS model explained in (Rao et al., 2021; Chan et al., 2016), and pre-trained it on the LibriSpeech dataset (Panayotov et al., 2015) following previous works (Lugosch et al., 2019).

Enc-Decoder NLP Model Bidirectional and Auto-Regressive Transformers (BART) (Lewis et al., 2020) uses a separate bidirectional encoder and autoregressive decoder similar to BERT (Devlin et al., 2019) except that (1) BART's decoder incorporates cross attention over the final encoder layer and (2) BART's encoder does not use a feed-forward dense layer for word prediction. The BART model can be used to perform both language understanding (i.e., intent classification) and language generation (i.e., summarization) problems at the same time, we refer to it as an NLP model in this work. We use the same pre-trained BART-base model as the original paper, which includes 6 transformer layers in the encoder and decoder.

Spoken Language Interface The interface exposes relevant outputs from the ASR model to the

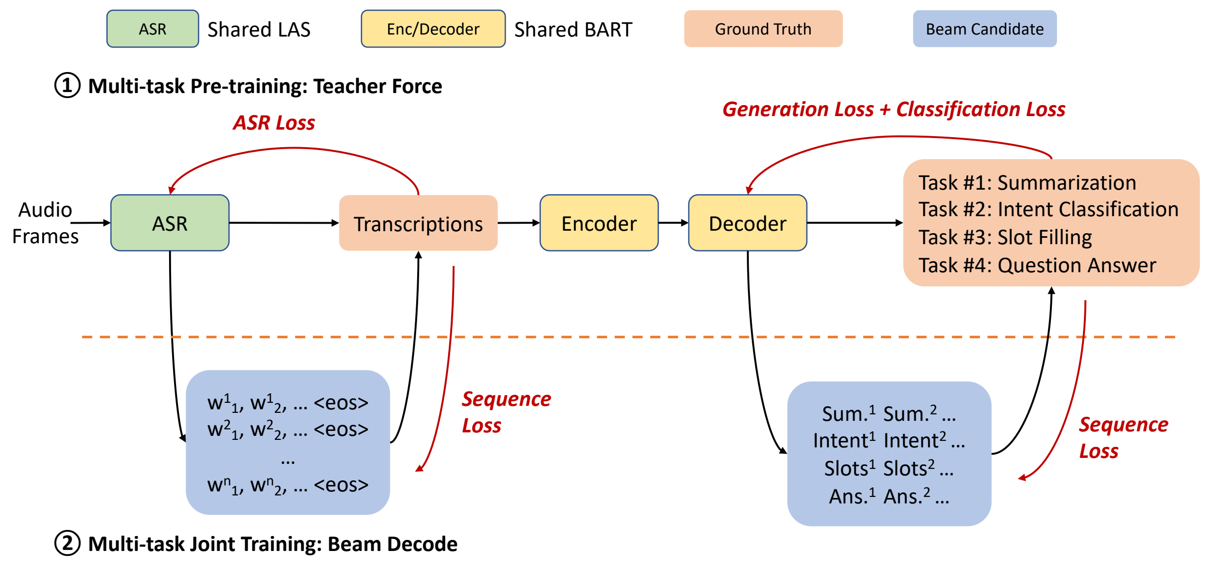

Figure 3: Our proposed MTL framework for LAS-BART-based Spoken Language Models. The model consists of an ASR system to generate transcription for the input audio frames, and an encoder-decoder system to generate intents, slots, answers, summarizations for different tasks. They share parameters of LAS, BART encoder and decoder, and are first trained on multiple tasks with ASR Loss, Generation Loss and Classification Loss; then the two systems are jointly trained with Sequence Loss.

downstream NLP model. Prior works have proposed rich interfaces that expose neural embeddings from ASR in addition to the text recognition (Rao et al., 2020). In this work, we use a simple text interface i.e. the best text recognition hypothesis from the output of ASR as the input to the NLP models. We leverage pre-trained models for both ASR and NLP. Inspired by (Rao et al., 2021; Raju et al., 2021), we introduce sequence loss training for the joint ASR-NLP system that allows direct optimization of non-differentiable SLT metrics. Specifically, we consider the error rate of ASR, summarization, QA, intent classification and slot filling as the SLT metrics.

3.2 Joint MTL Training Strategy

The MTL Training Strategy can be divided into three steps.

Backbone Pre-training The ASR model is first pre-trained for the speech recognition task using the LibriSpeech dataset. The NLP model uses the pre-trained BART (Lewis et al., 2020) model which is trained to reconstruct corrupted text.

MTL Pre-training Our joint pre-training on multiple tasks falls into the paradigm of multi-task learning (MTL). Training details of the MTL-SLT can be seen in Algorithm 1, in the training stage, we take turns to load the training data of these pre-training tasks. For example, we update model parameters on a batch of training instances from the first task, and then update parameters on a batch of training instances of the second task, and the process repeats. Note that, according to our preliminary experimentation, the effect of different orders of carrying out these pre-training tasks is negligible.

Post Fine-tuning After pre-trained with MTL objective, the MTL model is further fine-tuned on each dataset with few training steps to improve the performance.

3.3 Training Losses

There are three types of losses to be optimized in our framework, i.e., ASR loss, language task-specific losses and sequence losses. Our model is first trained by updating $\boldsymbol{\theta}_{ASR}$ based on the ASR loss, then trained by updating $\boldsymbol{\theta}_{NLP}$ for each downstream task. Finally, sequence loss training is employed to update both $\boldsymbol{\theta}_{ASR}$ and $\boldsymbol{\theta}_{NLP}$.

ASR Loss Given input audio sequence $\boldsymbol{x}$, the ASR system is trained by teacher-forcing the encoder-decoder network with the tokens of the ground truth transcript w with the loss function being $\mathcal{L}_{\text{asr}} = -\sum_{j=1}^{N} \log p(w_j|\boldsymbol{x}, w_{:j-1}; \boldsymbol{\theta})$.

Intent Detection For sentence-level classification problem, denote the sentence pooled represen-

Algorithm 1: Training a MTL-SLT model.

```
Parameter: Pre-trained LAS model and
  BART model $\theta$, random initialized task
  specific heads , epoch number $M$, task
  number $T$.
//Prepare the data for $T$ tasks.
for $t$ in $1, 2, ..., T$ do
    Pack the dataset $t$ into mini-batch: $D_t$.
end
// Multi-task Learning.
for epoch in $1, 2, ..., M$ do
    1. Merge all the datasets:
      $D = D_1 \cup D_2 ... \cup D_T$
    2. Shuffle $D$
    for $b_t$ in D do
        //$b_t$ is a mini-batch of task t.
        3. Compute loss : $L(\theta)$
        //Train the ASR and NLP tasks.
        $L(\theta)$ += $\mathcal{L}_{\text{asr}}$ for ASR
        $L(\theta)$ += $\mathcal{L}_{\text{gen}}$ for Summarization
        $L(\theta)$ += $\mathcal{L}_{\text{tagging}}$ for Slot Filling
        $L(\theta)$ += $\mathcal{L}_{\text{intent}}$ for Intent Detection
        $L(\theta)$ += $\mathcal{L}_{\text{qa}}$ for Question Answer
        if perform joint training then
            $L(\theta)$ += $\mathcal{L}_{\text{seq}}$ for ASR and NLP
        end
        4. Compute gradient: $\nabla(\theta)$
        5. Update model: $\theta = \theta - \epsilon\nabla(\theta)$
    end
end
```

tation as $\boldsymbol{e}$ from input ASR token sequence $\mathbf{w}$, and the correct intent label is c, the model infers c from $\boldsymbol{e}$. The negative log-likelihood loss is used for the classification loss $\mathcal{L}_{\text{intent}} = -\log p(c|\boldsymbol{e}; \boldsymbol{\theta})$.

Slot Filling For token-level classification problem, denote the slot sequence as s, the input as $\boldsymbol{v}$, and the sequence length as N, the negative log-likelihood loss is used for calculating slot loss $\mathcal{L}_{\text{tagging}} = -\sum_{j=1}^{N} \log p(s_j|\boldsymbol{v}, s_{:j-1}; \boldsymbol{\theta})$.

Summarization The summarization of $\boldsymbol{x}$ is defined as $\boldsymbol{y} = (y_1, \ldots, y_M)$. The model infers an appropriate $\boldsymbol{y}$ from $\boldsymbol{v}$. The generation loss $\mathcal{L}_{\text{gen}}$ is calculated with the negative log-likelihood loss $\mathcal{L}_{\text{gen}} = -\sum_{j=1}^{N} \log p(y_j|\boldsymbol{v}, y_{:j-1}; \boldsymbol{\theta})$.

Question Answering For question answering, we employ binary cross entropy loss on the sentence pooling representation $\mathcal{L}_{\text{has\_key}}$ and the span-based losses (Rajpurkar et al., 2016) on the sentence representation $\mathcal{L}_{\text{span}}$. The QA loss is $\mathcal{L}_{\text{qa}} = \mathcal{L}_{\text{has\_key}} + \mathcal{L}_{\text{span}}$.

Sequence Losses Inspired by reinforce framework (Prabhavalkar et al., 2018), sequence loss training enables end-to-end joint training of ASR and a downstream language task (Rao et al., 2021). Denote C as a joint sequence of ASR and NLP outputs, this is done by directly optimizing model parameters θ for the expected metric cost $M(c, c^*)$ over the distribution of candidate hypotheses. Here c^* is the ground-truth output and c is a model candidate. This is expressed as,

$$\mathcal{L}_{seq} = \mathbf{E}_{C \in \mathcal{C}}[M(C, c^*)] \quad (1)$$

$$\Rightarrow \nabla_\theta \mathcal{L}_{seq} = \nabla_\theta \mathbf{E}_{C \in \mathcal{C}}[M(C, c^*)] \quad (2)$$

$$\approx \nabla_\theta \sum_{c \in \bar{\mathcal{C}}} \bar{p}_\theta(c) M(c, c^*) \quad (3)$$

$$\approx \sum_{c \in \bar{\mathcal{C}}} M(c, c^*) \nabla_\theta \bar{p}_\theta(c). \quad (4)$$

Here, the approximation of the expectation in Eq. (3) is from using an n-best candidate set $\bar{\mathcal{C}}$ produced by the model with each candidate arising from a normalized probability $\bar{p}_\theta(c) = \frac{p_\theta(c)}{\sum_{c' \in \bar{\mathcal{C}}} p_\theta(c')}$. The probability of a candidate c is given by the combination of ASR and language task probabilities.

Sequence loss training is a policy gradient approach that jointly trains θ_{ASR} and θ_{NLP} by increasing the prediction probability of candidates with lower metric costs.

In this work, we optimize for a composite metric which is a sum of metrics of interest, namely, word error rate (WER) for ASR task and a language task metric. The metrics for language task include: (1) rouge error rate for the summarization task, (2) exact match error rate and QA F1 error rate for question answering, and (3) intent and domain classification error rate as well as SLU-F1 error rate for the language understanding task. These metrics are further detailed in Sec. 4.3.

Sequence loss training can be done for an individual task and is used in conjunction with the cross-entropy losses defined earlier that acts as a regularizing term. It can also be combined with multi-task learning by applying task-appropriate sequence loss training to update relevant parameters for a batch from the merged dataset.

	Datasets	Spoken-Gigaword				Spoken-SQuAD			ATIS			SLURP		
Settings	**Models**	WER(↓)	R1(↑)	R2(↑)	RL(↑)	WER(↓)	EM(↑)	F1(↑)	WER(↓)	Acc(↑)	F1(↑)	WER(↓)	Acc(↑)	SLU-F1(↑)
1. ASR	LAS-*S*	22.63	-	-	-	27.11	-	-	4.52	-	-	18.00	-	-
	LAS-*M*	**21.20**	-	-	-	**26.40**	-	-	**3.03**	-	-	**16.53**	-	-
2. NLP	BART-*S*	-	43.12	25.72	40.78	-	56.30	65.82	-	97.63	96.19	-	87.24	85.10
	BART-*M*	-	**43.30**	**26.02**	**41.29**	-	**57.92**	**67.79**	-	**98.38**	**97.55**	-	**88.31**	**85.62**
3. Pipeline	*S* -> *S*	22.63	15.81	8.10	14.66	27.11	15.73	29.82	4.52	96.87	92.30	18.00	83.22	72.50
	M -> *M*	21.20	16.33	8.61	15.29	26.40	17.44	32.30	3.03	**98.11**	**94.19**	16.53	**83.75**	72.81
4. Jointly	*S* + *S*	20.66	16.02	9.24	14.90	25.10	21.98	36.78	2.74	96.90	93.11	16.14	81.87	73.88
	M + *M*	**19.80**	**16.90**	**9.88**	**15.78**	**22.89**	**23.31**	**41.26**	**2.55**	97.13	93.65	**15.81**	83.10	**74.49**

Table 2: Main results of different models and settings on different datasets. **BOLD BLACK** numbers are in the first place for ASR and NLP settings, **BOLD RED** numbers are in the first place for Pipeline and jointly settings. A (↓) means lower is better, and (↑) means higher is better. a) For evaluation, we choose four typical and large generation and understanding datasets, i.e., Spoken-Gigaword, Spoken-SQuAD, ATIS and SLURP. b) For trainging settings, ASR and NLP represent two independent systems for their own tasks. Pipeline means that the output transcriptions from the pre-trained ASR system are used as the input of the pre-trained NLP system. Jointly training means that the parameters of ASR and NLP system are jointly optimized through extra sequence losses. c) For models, we use LAS for ASR system and BART for NLP system empirically. Single models (*S*) are treated as baselines and trained only on their own task. MTL models (*M*) mean that parameters are shared across four tasks and trained together. S -> S means pipeline training of LAS-S and then BART-S. S + S refers to pre-trained LAS-S and BART-S which are further jointly trained with sequence loss.

4 Experiments

4.1 Datasets

We perform experiments on four datasets, three of which are existing public corpora (ATIS, SLURP, Spoken-SQuAD) and one is generated by us (Spoken-gigaword).

ATIS Airline Travel Information Systems (ATIS) (Hemphill et al., 1990; Shivakumar et al., 2019) is a widely used Spoken Language Understanding dataset for airline reservation, where the user's intent and utterance's slots are predicted given the input command.

SLURP SLURP (Bastianelli et al., 2020) is a recently released Spoken Language Understanding dataset. It is larger and more semantically complex compared to ATIS dataset. The SLURP is a collections of 72k audio recordings of single turn user interactions with a home assistant on 18 domains.

Spoken-SQuAD Spoken-SQuAD (Li et al., 2018) is a large extraction-based Spoken Question Answering (SQA) dataset, where the answer of question is predicted given corresponding context. For the dataset, the context is in the form of speech and text, while the question and the answer are in the form of text. The transcripts of Spoken-SQuAD are collected from SQuAD benchmark dataset (Rajpurkar et al., 2016).

Spoken-Gigaword Spoken-Gigaword is a large summarization dataset. It is formulated as a summary generation problem, where the general headlines are generated given articles. Considering that Gigaword is abstractive summaries generation dataset with large amount of data, it can provide possibility for designing data-driven models. The transcripts of Spoken-Gigaword are collected from Gigaword (Rush et al., 2015), the speech of Spoken-Gigaword are generated by existing TTS model.

4.2 Experimental Settings

For the MTL-SLT model, we use LAS as the ASR model, where the input audio features are 64-dim log-mel filterbank features computed over a 25 ms window, with 10 ms shifts, the text is tokenized into subword tokens using a unigram language model (Kudo, 2018) of vocabulary of 4500. We use BART-base as NLP model, which has 6 encoder layers and 6 decoder layers, a hidden size of 768, filter size of 3,072, and 12 attention heads. We apply the default hyper-parameters from prior works (Rao et al., 2021; Lewis et al., 2020) including the learning rate schedule.

4.3 Experimental Metrics

In this section, we show the evaluation metrics for each tasks. For extractive question answering task (Rajpurkar et al., 2016), it is evaluated with two metrics: Exact Match (EM) to check whether the answer extracted by the model are exactly the same as the correct answer and F1 score to measure the degree of word overlap at token level. For summarization, we follow previous work (Rush et al., 2015) and use ROUGE-1 (unigrams), ROUGE-2 (bigrams), and ROUGE-L (longest-common substring) (Lin, 2004). For ATIS dataset, we evaluate it with intent classification accuracy and slot filling F1 score (Hemphill et al., 1990; Ruan et al., 2020). For SLURP dataset, we evaluate it with intent-domain classification accuracy and slot filling SLU-F1 score proposed in Bastianelli et al. (2020), which does not overly penalise misalignments caused by ASR errors.

4.4 Main Results

Results of different models and settings on four datasets are shown in Table 2.

ASR Taking word error rate (WER) as evaluation metric, we can see that the MTL has some advantages for the ASR task. From Setting 1, MTL helps improve the performance of LAS model on ASR when pooling data across tasks. From Setting 4, when jointly training with the NLP model, the MTL setting sees better performance than independently training ASR. Comparing the *S*+*S* in Setting 4 to the LAS-*S* and LAS-*M* in Setting 1, the improvements as per ASR from jointly training are (1.91% on average) larger than from MTL (1.28% on average), we attribute this to the optimization of ASR using sequence loss training for word error rate as well as related semantic metrics, similar conclusion can be witnessed in Rao et al. (2021).

NLP NLP system is different from the ASR system, in which all datasets are trained for same objective. For different NLP tasks, they share the backbone BART parameters and update their own task specific heads. From Table 2, we can see that BART-M has improvements over all independent models on all metrics, which proves the effectiveness of MTL in NLP system. Classification tasks see larger improvements than the generation tasks. In Setting 3 and Setting 4, NLP tasks can be further improved through jointly training, which shows the potential of sequence loss training in ASR-NLP system to make the system robust to acoustic noise. In Setting 4, *M*+*M* performs better than *S*+*S*, proving the effectiveness of MTL in ASR-NLP system.

Pipeline and Jointly Training Methods After pre-training the ASR and NLP model in single task mode or on multiple tasks, we have two methods to jointly use them, the pipeline method that is non-differentiable and the outputs of ASR system are directly treated as inputs of NLP system, and the jointly training method with sequence loss that is differentiable and can pass the gradient from NLP system to ASR system. From Table 2, we can see that results of different spoken language tasks in Setting 4 are better than in Setting 3, under both of independent training models and multi-tasks training models. Also multi-task trained models always perform better than independent trained models, no matter under pipeline setting or jointly training setting showing that both these effects are orthogonal and can complement one another.

Comparison with Existing Works We show the comparison results of our method to previous works on SLURP and ATIS in Table 4. Results are reported on the test set of ATIS and SLURP, as well as the development set of Spoken-SQuAD. From Table 3, because it is a recently released large SLU dataset, there are not too much previous works that we can refer, but we still get best performance compared the existing works to our knowledge.

Models	Acc	SLU-F1
Trained on text		
NLU* (Bastianelli et al., 2020)	84.84	-
NLU+ (Seo et al., 2021)	87.73	84.34
BART (Lewis et al., 2020)	88.00	85.49
Ours: MTL-Text	**88.31**	**85.62**
End-to-End trained		
ASR+ -> NLU+ (Seo et al., 2021)	82.93	71.12
Ours: MTL-SLT	**83.10**	**74.49**

Table 3: Comparison with existing works on SLURP. NLU* represents the results from SLURP paper. NLU+ represents the results from a recently released paper.

5 Analysis

5.1 Effect of MTL

MTL on ASR Chan et al. (2021) shows that by simply mixing multiple ASR datasets together, ASR model can perform better on each dataset, and can learn powerful transfer learning representation. Inspired by this, in our experiment, we would also like to investigate the performance change after employing multi-task training only on the experimented audio data and transcription. Specifi-

Models	Acc	F1
Trained on text		
Attention BiRNN (Liu and Lane, 2016)	91.10	94.20
Capsule-NLU (Zhang et al., 2019)	95.00	95.20
LIDSNet (Agarwal et al., 2021)	95.97	-
SF-ID Network (E et al., 2019)	96.60	95.60
SyntacticTF (Wang et al., 2021)	97.31	96.01
BERT SLU (Chen et al., 2019)	97.50	96.10
Stack-Prop. (Qin et al., 2019)	96.90	95.90
Stack-Prop. + BERT (Qin et al., 2019)	97.50	96.10
ASR Error Robust SLU (Ruan et al., 2020)	97.13	96.03
Ours: MTL-Text	**98.18**	**96.51**
End-to-End trained		
Phoneme-BERT (Sundararaman et al., 2021)	**97.25**	84.15
E2E SLP (Qian et al., 2021)	96.30	90.95
Pre-trained MTL (da Silva Morais et al., 2021)	96.60	91.20
Ours: MTL-SLT	96.92	**91.43**

Table 4: Comparison results on ATIS test set.

cally, during training, only the LAS model is shared across different tasks. Results can be seen in Setting 1 row LAS-M, in Table 2. We can see that after employing more data, LAS performs better on each dataset, which proves that it is effective to perform more data on ASR model.

MTL on NLP We can see from Table 2 that with multi-task training, BART performs better in both the text-based setting (i.e., BART) and jointly training setting (i.e., LAS-BART).

5.2 Effect of Sequence Loss

With the used sequence loss ($\mathcal{L}_{seq}$), we can train not only the ASR model NLP model independently, but also train both of them in an end-to-end manner. We compared the models with and without $\mathcal{L}_{seq}$, and the result are shown in Table 2. By using the $\mathcal{L}_{seq}$, we observe improvements in ASR and NLP metrics by 2-5%. Sequence loss training allows for the downstream language modelling task to be trained with potentially erroneous ASR hypotheses allowing for robustness to word errors. This also minimizes the domain shift that occurs from training (language task has the clean ground truth transcription as input) to inference (language task has ASR hypotheses as input) resulting in improved performance. Another impact of sequence loss training is that ASR is optimized for differentiable (eg. cross-entropy), non-differentiable (eg. WER) ASR losses along with arbitrary non-differentiable metrics of interest (eg. rouge scores, SLU-F1) of the downstream language task.

5.3 Effect of Post Fine-tuning

The post fine-tuning step described in 3.2 is important in our framework, because 1) it can eliminate differences between datasets arising from different domains; 2) the optimal performance of different datasets falls on different positions of a pareto-optimal surface, post fine-tuning can solve this problem without introducing more parameters. Effect of post fine-tuning can be seen in Table 5.

Models	ASR	Summarization-R1
MTL-ASR	21.20	-
w/o Post FT	23.13	-
MTL-Text	-	43.12
w/o Post FT	-	26.50
MTL-SLT	19.80	16.02
w/o Post FT	21.45	14.39

Table 5: Ablation study on Post Fine-tuning.

6 Conclusion

We proposed a multi-task learning framework for spoken language understanding tasks that take speech as input and produces (1) intents and named-entities in language understanding tasks, (2) abstract text summaries, or (3) question answering. This framework can be extended to other language tasks such as translation.

In this framework, we make use of pretrained ASR models and language models like BART and jointly train these layers across multiple language tasks. We demonstrate that this training across tasks coupled with task-specific post-finetuning produces significantly better results for ASR and BART separately. We made use of the sequence loss training framework to enable end-to-end training of ASR and BART to optimize for metrics of interest for the classification, sequence tagging, and generation tasks. This made the downstream language task robust to errors in ASR hypotheses that otherwise leads to performance degradation in pipelined ASR and language task systems.

We demonstrate state-of-the-art results on public corpora of SLURP and ATIS for spoken language understanding. We also prepare the Spoken-Gigaword dataset for abstractive summarization of speech.

References

Vibhav Agarwal, Sudeep Deepak Shivnikar, Sourav Ghosh, Himanshu Arora, and Yashwant Saini. 2021. Lidsnet: A lightweight on-device intent detection model using deep siamese network. *CoRR*, abs/2110.15717.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. SLURP: A spoken language understanding resource package. In *EMNLP*. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*.

William Chan, Daniel S. Park, Chris Lee, Yu Zhang, Quoc V. Le, and Mohammad Norouzi. 2021. Speechstew: Simply mix all available speech recognition data to train one large neural network. *CoRR*, abs/2104.02133.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

Yung-Sung Chuang, Chi-Liang Liu, Hung-yi Lee, and Lin-Shan Lee. 2020. Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering. In *INTERSPEECH*.

Yu-An Chung, Chenguang Zhu, and Michael Zeng. 2021. SPLAT: speech-language joint pre-training for spoken language understanding. In *NAACL-HLT*.

Edmilson da Silva Morais, Hong-Kwang Jeff Kuo, Samuel Thomas, Zoltán Tüske, and Brian Kingsbury. 2021. End-to-end spoken language understanding using transformer networks and self-supervised pre-trained features. In *ICASSP*.

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

Haihong E, Peiqing Niu, Zhongfu Chen, and Meina Song. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. In *ACL*.

Lingyun Feng, Jianwei Yu, Deng Cai, Songxiang Liu, Haitao Zheng, and Yan Wang. 2021. ASR-GLUE: A new multi-task benchmark for asr-robust natural language understanding. *CoRR*, abs/2108.13048.

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

Minjeong Kim, Gyuwan Kim, Sang-Woo Lee, and Jung-Woo Ha. 2021a. St-bert: Cross-modal language model pre-training for end-to-end spoken language understanding. In *ICASSP*, pages 7478–7482. IEEE.

Seokhwan Kim, Yang Liu, Di Jin, Alexandros Papangelis, Karthik Gopalakrishnan, Behnam Hedayatnia, and Dilek Hakkani-Tur. 2021b. " how robust ru?": Evaluating task-oriented dialogue systems on spoken conversations. *arXiv preprint arXiv:2109.13489*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL*.

Chia-Chih Kuo, Shang-Bao Luo, and Kuan-Yu Chen. 2020. An audio-enriched bert-based framework for spoken multiple-choice question answering. In *INTERSPEECH*.

Cheng-I Lai, Yung-Sung Chuang, Hung-Yi Lee, Shang-Wen Li, and James R. Glass. 2021. Semi-supervised spoken language understanding via self-supervised speech and language model pretraining. In *ICASSP*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880. Association for Computational Linguistics.

Chia-Hsuan Li, Szu-Lin Wu, Chi-Liang Liu, and Hung-yi Lee. 2018. Spoken squad: A study of mitigating the impact of speech recognition errors on listening comprehension. In *INTERSPEECH*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Bing Liu and Ian R. Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *INTERSPEECH*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *ACL*. Association for Computational Linguistics.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *EMNLP/IJCNLP*. Association for Computational Linguistics.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Interspeech*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP*, pages 5206–5210. IEEE.

Rohit Prabhavalkar, Tara N. Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjuli Kannan. 2018. Minimum word error rate training for attention-based sequence-to-sequence models. In *ICASSP*, pages 4839–4843. IEEE.

Yao Qian, Ximo Bian, Yu Shi, Naoyuki Kanda, Leo Shen, Zhen Xiao, and Michael Zeng. 2021. Speech-language pre-training for end-to-end spoken language understanding. In *ICASSP*.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *EMNLP/IJCNLP*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*.

Anirudh Raju, Gautam Tiwari, Milind Rao, Pranav Dheram, Bryan Anderson, Zhe Zhang, Bach Bui, and Ariya Rastrow. 2021. End-to-end spoken language understanding using rnn-transducer ASR. *CoRR*, abs/2106.15919.

Milind Rao, Pranav Dheram, Gautam Tiwari, Anirudh Raju, Jasha Droppo, Ariya Rastrow, and Andreas Stolcke. 2021. DO as I mean, not as I say: Sequence loss training for spoken language understanding. In *ICASSP*.

Milind Rao, Anirudh Raju, Pranav Dheram, Bach Bui, and Ariya Rastrow. 2020. Speech to semantics: Improve ASR and NLU jointly via all-neural interfaces. In *INTERSPEECH*.

Weitong Ruan, Yaroslav Nechaev, Luoxin Chen, Chengwei Su, and Imre Kiss. 2020. Towards an ASR error robust spoken language understanding system. In *INTERSPEECH*.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *EMNLP*.

Alaa Saade, Joseph Dureau, David Leroy, Francesco Caltagirone, Alice Coucke, Adrien Ball, Clément Doumouro, Thibaut Lavril, Alexandre Caulier, Théodore Bluche, Thibault Gisselbrecht, and Maël Primet. 2019. Spoken language understanding on the edge. In *EMC2@NeurIPS*.

Seunghyun Seo, Donghyun Kwak, and Bowon Lee. 2021. Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding. *CoRR*, abs/2104.07253.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ-Skerrv Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In *ICASSP*.

Prashanth Gurunath Shivakumar, Mu Yang, and Panayiotis G. Georgiou. 2019. Spoken language intent detection using confusion2vec. In *INTERSPEECH*.

Mukuntha Narayanan Sundararaman, Ayush Kumar, and Jithendra Vepa. 2021. Phoneme-bert: Joint language modelling of phoneme sequence and asr transcript. *CoRR*, abs/2102.00804.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Jixuan Wang, Kai Wei, Martin Radfar, Weiwei Zhang, and Clement Chung. 2021. Encoding syntactic knowledge in transformer encoder for intent detection and slot filling. In *AAAI*.

Chenyu You, Nuo Chen, and Yuexian Zou. 2021. Knowledge distillation for improved accuracy in spoken question answering. In *ICASSP*.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip S. Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *ACL*.

Yu Zhang, James Qin, Daniel S. Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V. Le, and Yonghui Wu. 2020. Pushing the limits of semi-supervised learning for automatic speech recognition. *CoRR*, abs/2010.10504.

A Experiment Settings

A.1 Statistics of datasets

For the experimental datasets (Spoken-SQuAD, SLURP, ATIS), we follow the default train/dev/test splits from the original paper.

A.2 Hyperparameters

We show the detailed hyperparameters for the MTL Pre-training and Post Fine-tuning stages described in Section 3.2 of the proposed method on different datasets in Table 6.

	Pre-training	Fine-tuning
Speech Model Batch Size	16	16
Text Model Batch Size	16	16
Joint Training Model Batch Size	4	4
Learning Rate	$2e-5$	$2e-5$
Warmup Steps	0	0
Learning Rate Decay	Linear	Linear
Weight Decay	0	0
Gradient Clipping	1	1
Dropout	0.1	0.1
Attention Dropout	0.1	0.1
Training Steps	100k	20k

Table 6: Hyperparameters for the Pre-training and fine-tuning stages in training MTL-SLT on the four datasets.

B Spoken-gigaword Dataset

The detail statistics of the generated Spoken-Gigaword dataset are shown in Table 7. The articles and summarizations are acquired from gigaword headline generation dataset (Rush et al., 2015), we then generate the speech data for the articles using Tacotron2 (Shen et al., 2018) to extract feature and . Note that because the input article is noisy, which make it hard to generate proper speech, so we remove the ones with special symbols, and we remove the articles that have more than 30 words. The implementation is based on an open source library [1].

C Model Structure of NLP task with BART Model

As a pre-trained sequence-to-sequence denoising autoencoder, BART uses a standard Transformer-based neural machine translation architecture, which consists of 6 encoder and 6 decoder segments. In our work, we attribute each tasks with task specific classification head over the BART model. Specifically, for the Intent Detection task, we use the End-Of-Sentence (EOS) token on the last decoder layer to do the prediction; for the slot filling task, we predict the slot labels in BIO format after the last encoder layer; for the summarization task, generated sentences with EOS token at end are used to calculate the summarized loss; for the question answering task, EOS token in the last decoder layer is used to predict the answer.

[1] https://github.com/mozilla/TTS/

Types		Spoken-Gigaword
Training Set		249199
Validation Set		12578
Article	words	119M
	uni-words	110K
	aver length	14.6
	max length	30
	min length	11
Headline	words	31M
	uni-words	69K
	aver words	8.3
	max length	30
	min length	2

Table 7: Statistics of the generated Spoken-gigaword.

D E2E Spoken Question Answering

In Section 3.3, we mention the $\mathcal{L}_{\text{has\_key}}$ in Spoken Question Answering. Actually, Spoken-SQuAD is a dataset with all examples having answers. However, since the input context of each example is too long, if we process the input audio directly, the model's performance will be very poor. Thus, instead of processing the input audio directly, we first split the input into sentence-wise segments, and then during the training, we predict the answer on each sentence. Note that we have a classification head to determine whether this sentence contains the answer or not, and the loss over this classification head is $\mathcal{L}_{\text{has\_key}}$.