Training a Generalizable End-to-End Speech-to-Intent Model

Emmy Phung, Cora Hyun Jung, Sujeong Cha, Wendy Hou
New York University
60 5th Ave, New York, NY 10011
{mtp363, hj1399, sjc433, wh916}@nyu.edu

Abstract

A conventional approach to spoken language understanding (SLU) has a pipeline structure where a speech signal is converted into written texts (transcription) by an automatic speech recognition (ASR) module, which is then followed by a natural language understanding (NLU) module performing intent classification or slot-filling. However, more recent research works have been geared towards end-to-end SLU where a single model determines intents or slot-filling directly from a speech input without the need for intermediate transcriptions. Because such a system requires a lot of training data and it's expensive to produce labeled domain-specific data, our goal is to train a good generalizable model on existing corpora and then fine-tune on a new corpus. To tackle this problem, we present three methods, 1) data augmentation where we manipulate audio files' tempo to increase the size of training data, 2) model architecture modification where we modify the number of hidden units and GRUs, and finally 3) transfer learning where the model is first pre-trained by one corpus and then fine-tuned on another corpus. The results show that the model performance has been improved by data augmentation and transfer learning when some of the target dataset is included in the pre-training stage.

1 Introduction

While a conventional spoken language understanding (SLU) system maps audio to text and extracts intents from the text transcription, our interest lies in developing an end-to-end (E2E) speech-to-intent (S2I) system that directly outputs intent from a speech audio input without producing text transcription. Because training such a system requires a huge amount of domain-specific labeled data, it is hard to adapt an E2E SLU model to a data-scarce domain. Our approaches to tackle this problem are 1) using data augmentation to generate more training data, 2) improving the model architecture, and 3) training a generalizable E2E S2I model on a combination of existing corpora and then domain-adapt it to a previously unseen corpus.

The baseline model is adopted from Lugosch et al. [3], which was originally pre-trained on the LibriSpeech dataset, a large automatic speech recognition (ASR) dataset which contains more than 200,000 words [6], and fine-tuned on Fluent Speech Commands (Fluent.ai), a dataset of home device control commands.

For our project, we do not limit ourselves to the Fluent.ai dataset but also search for other corpora that have intent labels and audio data. We have successfully preprocessed and established new benchmarks for four datasets from three domains using the baseline model by Lugosch. The Fluent.ai dataset and the SNIPS dataset include home device control commands. The Airline Travel Information Systems (ATIS) dataset comes from the airline travel information domain. The Appen Medical (medical) dataset has questions about medical symptoms.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Our first approach, data augmentation, successfully improves the test accuracy on ATIS, SNIPS and medical datasets when compared to the baseline results, establishing a set of benchmarks. In our second approach, we attempt to modify the model architecture, but this does not improve the baseline results. Finally, transfer learning, which is our primary hypothesis, has brought back a positive result and beaten our benchmark.

2 Related Work

As stated in the introduction, one of the obstacles in training a well-performing E2E SLU model is data scarcity. Since speech signals are high-dimensional and highly variable, training deep models and learning these hierarchical representations without a large amount of training data is difficult [3]. There have been many approaches to tackle this problem by exploiting abundant data from the relevant tasks (e.g. ASR, NLU).

Huang, et al.[2] from IBM Research AI suggest two methods to take advantage of NLU text resources, which are 1) an S2I model can be jointly trained with a text-to-intent (T2I) model to closely match the acoustic embeddings with BERT-based text embeddings and 2) a multi-speaker text-to-speech (TTS) system can be used to create synthetic S2I data.

Another common practice is to utilize transfer learning, which attempts to transfer knowledge from other sources to benefit the new task. The benefits of transfer learning for SLU tasks include faster convergence, more robust results, and less data sensitivity [7]. Lugosch et al. [3] have shown that pre-training upstream modules (phoneme and word) of an E2E SLU model on LibriSpeech dataset increases the model's accuracy on the downstream task (intent classification) and shortens the training time. The authors also show that we can benefit more from unfreezing parts of the pre-trained layers during the fine-tune stage (i.e. unfreezing word and intent layers) than freezing them all [3].

As the data and codebase used in [3] are publicly available, we use this as our primary code base and extend it via the following approach: in addition to phoneme and word modules pre-trained with LibriSpeech, we also pre-train the final intent layer with an existing S2I corpus and then fine-tune the model with the target S2I corpus.

3 Problem Definition and Model Explanation

Our project goal is to train a generalizable E2E S2I model that can be easily adapted to other data-scarce domains. We take audio files as our input data and output intent classification. Conventionally, there are two approaches to solve an S2I problem, intent classification and slot-filling. In this project, we are focusing on intent classification to benefit the relative simplicity.

Lugosch's baseline model is a GRU-based deep neural network composed of three modules, 1) phoneme module, 2) word module, and 3) intent module. The phoneme and word modules are pre-trained on LibriSpeech. Since the number of unique words is too large, a label is assigned to the 100,000 most common words. The phonemes are used as intermediate targets so that speech segments with no word label can also be pre-trained. The model is pre-trained using Montreal Forced Aligner [4] to obtain phoneme- and word-level alignments for 960 hours [3]. The features learned from the pre-trained phoneme and word modules are used as input to the intent module, which is trained using intent-labeled datasets.

3.1 Phoneme Module

The phoneme module takes the input audio signal and outputs the hidden representations of the predicted phonemes, where the logits are computed by a linear classifier. The module is composed of a SincNet layer, followed by multiple convolutional layers, GRU layers, pooling, and dropout layers.

3.2 Word Module

The word module takes the hidden representations of the phonemes as input and outputs the hidden representations of words. The word module is composed of pre-trained layers including GRU layers with dropout and pooling, and predicts word targets using a linear classifier. However, these predicted

word targets are discarded to save memory, and the module outputs the hidden representations of words instead.

3.3 Intent Module

The intent module takes hidden representations of words and outputs the predicted intents. The module is composed of a GRU layer, followed by max-pooling to reduce the sequence of outputs into the logits corresponding to the number of intent slot values, which is fixed to be three. Later in the paper, we will discuss how we modify the intent module structure, such as increasing or decreasing the number of hidden units and adding more GRU layers.

3.4 Model Configurations

We experiment with unfreezing the weights of previously frozen modules and fine-tuning them by backpropagation. Based on Lugosch's codebase, we use four different configurations, 1) **No Pre-training**: the model weights are randomly initialized and trained from scratch, 2) **No Unfreezing**: the model keeps all pre-trained layers and weights, 3) **Unfreeze Word Layers**: we unfreeze the word and intent layers and fine-tune their weights, 4) **Unfreeze All Layers**: we unfreeze both the phoneme, word and intent layers and fine-tune all of the weights.

4 Data

4.1 Fluent.ai

The Fluent.ai dataset is the dataset containing spoken English commands for a smart home or virtual assistant, such as "put on the music" or "turn up the heat in the kitchen". There are 248 unique transcriptions mapping to 31 unique intents. The dataset was obtained from the Fluent.ai website [1]. Initially, the data are randomly split into train, valid, and test datasets in a way that each dataset contains unique speakers (Data Split Method 1, Appendix Table 1). However, in this split, all of the utterances are repeated across the three datasets, which may cause overfitting. Thus, we re-split the dataset in a way that each dataset contains unique utterances (Data Split Method 2, Appendix Table 2). Also, we lowercase and merge the intent slots into one label to create a uniform label format across all datasets in our transfer learning experiments.

4.2 ATIS

The ATIS dataset contains English speech commands to make flight reservations, such as "I'd like to book the flight from San Francisco to New York." There are 5,488 unique utterances mapping to 22 unique intent labels. Because the ATIS dataset doesn't offer a validation dataset, we split the train data to obtain the validation dataset, reflecting roughly the same distribution as the original train data and matching the ratio of the Fluent.ai dataset. We also remove two intent labels and the corresponding three utterances in the test dataset that don't appear in the training dataset (Appendix Table 3).

4.3 SNIPS

We use the "smart light" subset of the SNIPS dataset, which contains English spoken commands related to smart light appliances, such as "Can you turn on the light in the kitchen, please?". It contains 1,660 unique utterances spoken by 21 real speakers, mapping to six unique intent labels. For the synthetic dataset, Facebook's VoiceLoop was used to synthesize the audios. Twenty-two synthetic speakers were used to synthesize all 1,660 utterances. The dataset was split into five folds for cross-validation. We combined four folds to be a training dataset, and the fifth fold is split into half, to make the validation and test datasets. Also, just like the Fluent.ai dataset, we lower-cased and merged the intent labels into one slot, to minimize the variability across corpora for transfer learning experiments (Label 2), which will be described later in the paper (Appendix Table 4).

Table 1: Model performance before and after data augmentation

	ATIS	Aug. ATIS	SNIPS	SNIPS w/ Synth.	Medical	Aug. Medical
No Pre-training	71.0%	88.0%	-	-	5.0%	5.0%
No Unfreezing	94.0%	94.0%	83.2%	84.6%	24.0%	26.0%
Unfreeze Word	95.0%	**96.0%**	87.4%	**88.0%**	38.0%	48.0%
Unfreeze All	94.0%	94.0%	-	-	63.0%	**72.0%**

4.4 Medical

The Medical dataset contains English utterances for common medical symptoms like "knee pain" or "headache," totaling 8.5 hours. The dataset was obtained from the Appen website [5]. Each utterance was created by 124 unique speakers based on a given symptom. An example utterance could be "I have this strange rash on my arm", labeled with the intent "skin issue". There are 384 unique utterances mapped to 25 unique intent labels. We find that several utterances are heard across datasets. In the future, we may consider re-splitting the dataset in a way that no utterances is repeated across the datasets to test the model's generalizability to predict the intent given the utterances that it hasn't heard before (Appendix Table 5).

5 Hypothesis and Experimentation

5.1 Data Augmentation

For our smaller datasets (i.e. ATIS, SNIPS, and Medical), we hypothesize that data augmentation which increases the amount of training data can improve our model's performance.

5.1.1 ATIS

A closer investigation of the experiments on ATIS reveals that ATIS has a class imbalance issue, where the intent "flight" is associated with around 74% of all the utterances. The **No Pre-training** experiment result shows that the model does not learn from the audio input but simply predicts the majority class. Therefore, data augmentation on ATIS does not only aim to increase the dataset's volume but also solves its class imbalance issue.

Sound eXchange (SoX) is a powerful sound processing module, which can manipulate the attributes of an input audio file. Using SoX, we randomly change the original audio's tempo, to create different versions of the original audio files. To balance our classes, only one copy with a random tempo is created for each audio belonging to the majority intent, "flight". In the same fashion, 14 copies are created for the second majority intent, "airfare", and 19 copies for the rest of the classes. The augmented ATIS dataset has more than 30,000 utterances and no majority class. As a result, the benchmark for ATIS increases from 95% to 96% (Table 1).

No Pre-training benefits the most (Table 1) because in this configuration the model has to learn ATIS acoustics from scratch. As data augmentation balances out the intents for ATIS, the model is forced to learn the acoustics of ATIS utterances and stops predicting the majority intent, thus resulting in a significant improvement in the performance.

5.1.2 SNIPS

The SNIPS dataset comes with an extension dataset, which includes synthetic audio files generated from synthetic speakers as described in section 4. Comparing the results of SNIPS with and without synthetic data, we raise the benchmark for SNIPS from 87.4% to 88% (Table 1).

5.1.3 Medical

For the Medical dataset, four different versions with random tempos are created for each original audio file. After augmentation, the benchmark for Medical increases from 63% to 72% (Table 1).

Table 2: Test Accuracy for different numbers of hidden units

Hidden Units	Pre-trained Phonemes+Words	Pre-trained Phonemes+Words+Intent
64	79.4%	-
128	**84.4%**	**82.2%**
256	81.0%	79.6%
512	79.8%	80.6%

Table 3: Test accuracy for different numbers of GRU layers

	Pre-trained Phonemes+Words	Pre-trained Phonemes+Words +1 Intent Layer	Pre-trained Phonemes+Words +2 Intent Layers
1 GRU block	**84.4%**	**82.2%**	-
2 GRU blocks	80.0%	79.8%	81.0%

We conclude that data augmentation generally improves the model's performance by bringing in more training data. It can be especially helpful in balancing the dataset's class distribution to create robust results and forcing the model to learn the acoustics of the target dataset.

5.2 Model Modification

In the original model design, the final intent module contains only one GRU layer, and we hypothesize that increasing the complexity of the intent module would increase the model's learning capacity and enable it to better capture the input signals.

We first attempt to manipulate the number of hidden units. The values that we experiment with are 64, 128, 256, and 512. Different from our hypothesis, the results show that increasing the number of hidden units lowers the test accuracy, and a moderate number of hidden units (128) yields the best performance regardless of the pre-trained amount.

Analyzing these results, we speculate that a more complicated model might need a lower learning rate because the optimization space has become more complex. With a high learning rate, the gradient descent is more likely to be stuck at a local minimum. However, our experiments with lower learning rates did not show any improvements (Appendix Table 6).

While the first modification is to "broaden" the network, the next set of experiments deal with a "deeper" network. We added another block of GRU to the final intent module. However, our hypothesis is also not supported as the new accuracy does not beat the result of the default setting (Table 3).

5.3 Transfer Learning

We also investigate the impact of transfer learning on model performance, which is our primary focus for this project. We hypothesize that pre-training an E2E S2I model on other corpora would improve the performance when domain-adapting to the target dataset. To validate this hypothesis, we attempt to pre-train the model on datasets from both different and similar domains.

5.3.1 Pre-training on a different domain

In the first set of experiments, we pre-train the model on Fluent.ai and then fine-tune it on the target dataset, ATIS. This is to verify if pre-training on the domain that's not related to the domain of the target dataset still transfers the knowledge. In these experiments, the Data Split Method 1 is used for the Fluent.ai dataset. The best performing pre-training configuration is **No Unfreezing**, which is used to generate the results below.

As a result, the best test accuracy (95%) achieved is the same as the stand-alone ATIS model's best performance (Table 4). The reason that the knowledge from the pre-trained model isn't transferred could be that the domains of two datasets are very different, with the Fluent.ai dataset belonging

Table 4: Test accuracy for ATIS (Pre-trained on Fluent.ai)

Fine-tune on ATIS	Original Benchmark	Benchmark for Pre-train on Fluent.ai
No Unfreezing	94.0%	**95.0%**
Unfreeze Word Layer	**95.0%**	**95.0%**
Unfreeze All Layers	94.0%	**95.0%**

Table 5: Test accuracy for SNIPS (Pre-trained on Fluent.ai or Fluent.ai + SNIPS)

Fine-tune Config.	Original Benchmark	Test Accuracy for Pre-trained on Fluent.ai	Test Accuracy for Pre-trained on Fluent.ai + SNIPS
No Unfreezing	80.2%	78.2%	**88.6%**
Unfreeze Word	86.6%	84.6%	88.0%
Unfreeze All	80.0%	75.2%	81.4%

to smart home speech commands and the ATIS dataset belonging to the flight reservation domain. Another possible reason can be the way the Fluent.ai dataset is split. Because the model has already heard the utterances in the test dataset in the training stage, there is the possibility that the model has overfitted the Fluent.ai dataset and has not been generalized well. We will proceed with the experimentations by using the Data Split Method 2 for the Fluent.ai dataset and using the target dataset that has a similar domain as the pre-training dataset.

5.3.2 Pre-training on a similar domain

For these experiments, we focus on Fluent.ai and SNIPS because they come from the same domain, which is smart-home device control commands. Since the two datasets have different intent labels, we are concerned that such differences might hinder transfer learning. As previously mentioned (Section 4.1), we also investigate Fluent.ai's train, validation, and test sets when noticing an extremely high test accuracy. We re-split this data before pre-training to make sure data leakage has been eliminated and then save the best pre-trained model based on validation accuracy.

For our experiments, we adopt the phoneme and word modules pre-trained on LibriSpeech, and continue our pre-training process on another dataset. We perform two sets of experiments, 1) pre-training on Fluent.ai and 2) pre-training on both Fluent.ai and some of SNIPS, which is also known as co-training. These two pre-trained models are then fine-tuned and evaluated on the entire SNIPS dataset. We develop two datasets for the pre-training stage: 1) one with only Fluent.ai data and 2) one with Fluent.ai plus roughly 50% of SNIPS data. The rationale for the second experiment is that, although coming from the same domain, the two datasets have different utterances so pre-training on one may not help the model learn the acoustics or semantics of the other. We hypothesize that adding some of the target dataset in the pre-training round will allow smoother adaptation to the target dataset. Though we have run several experiments with different combinations of configurations in the pre-training and fine-tuning rounds, we only report some of our best results in Table 5.

The results have validated our hypothesis that if we co-train Fluent.ai and some of SNIPS data in the pre-training round, the model can adapt better to the target dataset and give a higher test accuracy. This configuration raises the benchmark from 86.6% to 88.6%.

Our experiments also reveal that, when pre-trained on Fluent.ai, **Unfreeze Word** leads to the best results on the target dataset. This tells us that after pre-training the baseline model on LibriSpeech, the phoneme module weights can be directly transferred to an unseen dataset, but the word module wegiths need to be fine-tuned. If we pretrain on both Fluent.ai and SNIPS, **No Unfreezing** leads to the best result. We believe that adding some of the SNIPS data and co-training with Fluent.ai in the pre-training round allows the word module to pick up the semantics of SNIPS, and thus only the intent classifier needs further fine-tuning.

While obtaining promising results from pre-training on both Fluent.ai and SNIPS, we also acknowledge some shortcomings of our model. First, the cost of pre-training is still high compared to the amount of improvement we have obtained. Second, we also observe high variance in our model predictions despite setting a random seed. To mitigate this problem, we use a five-fold cross-validation.

Table 6: New benchmarks for all datasets

Dataset	Original Benchmark	Final Benchmark	Best Config.for Final Benchmark
Fluent.ai	99.0%	78.0%	Data Split Method 2 + Unfreeze Word
ATIS	95.0%	96.0%	Augmented + Unfreeze Word
SNIPS	87.4%	88.6%	Pre-trained on Fluent.AI + some SNIPS and Fine-tune on SNIPS including Synthetic Data
Medical	63.0%	72.0%	Augmented + Unfreeze All

We want to continue to tackle this in our future improvements of the model. Last but not least, we understand that it is hard to seek similar corpora within the same domain so allowing knowledge to be transferred across different domains is a critical research area for us to further investigate.

5.4 New Benchmarks for All Datasets

At the end of our experiments, we successfully establish new benchmarks for all four datasets. Our best configurations and test accuracies are summarized in Table 6.

6 Conclusions and Discussion

In conclusion, when we augment data, the performance has generally improved. Also, pre-training on similar datasets can improve the performance when we introduce some target training dataset in the pre-training stage.

The main contributions of this project include that we beat the benchmarks on ATIS by 1% through data augmentation, SNIPS by 2% through data augmentation and transfer learning, and Medical by 9% through data augmentation. As a result, we should use these new, improved benchmarks as a baseline in our future experiments.

In the future, we will explore new algorithms and architectures to improve performance and reduce run time. Some of the potential direction of our project includes incorporating BERT embeddings in the model to increase its learning ability and AdapterFusion [8] to make the model more generalizable. We also plan to continue working on this project, and eventually submit the paper to the InterSpeech 2021 conference in March.

From the various experiments conducted, we learn that class imbalance and small dataset size can seriously affect the model performance. Data augmentation can help 1) balance out class distribution, 2) generate more data, and 3) avoid overfitting. We've also learned good practices to design experiments like having other factors fixed and changing one factor at a time to have a controlled experiment where we can pinpoint the changes a factor brings. We've also gained firsthand experiences in tackling an SLU problem and familiarizing ourselves with the packages specifically designed for speech data.

7 Acknowledgments

We would like to acknowledge Professor Michael Alan Picheny of NYU and Drs. Hong-Kwang Kuo, Samuel Thomas, and Edmilson da Silva Morais of IBM Research for giving all the helpful suggestions and guidance throughout the project, as well as providing ATIS speech dataset. We would want to thank Loren Lugosch for making the codebase publicly available and promptly answering our questions. Thanks to Professor Cristina Savin, Elena Sizikova, and Wenda Zhou for their support and NYU CIMS for computing support.

References

[1] Fluent Speech Commands: A dataset for spoken language understanding research. (2020, April 03). Retrieved September 30, 2020, from https://fluent.ai/fluent-speech-commands-a-dataset-for-spoken-language-understanding-research/

[2] Huang, Y., Kuo, H., Thomas, S., Kons, Z., Audhkhasi, K., Kingsbury, B., Hoory, R., Picheny, M. (2020). Leveraging Unpaired Text Data for Training End-To-End Speech-to-Intent Systems. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). doi:10.1109/icassp40776.2020.9053281

[3] Lugosch, L., Ravanelli, M., Ignoto, P., Tomar, V. S., Bengio, Y. (2019). Speech Model Pre-Training for End-to-End Spoken Language Understanding. Interspeech 2019. doi:10.21437/interspeech.2019-2396

[4] Mcauliffe, M., Socolof, M., Mihuc, S., Wagner, M., Sonderegger, M. (2017). Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. Interspeech 2017. doi:10.21437/interspeech.2017-1386

[5] Medical Speech, Transcription, and Intent (English): Appen Datasets. (2020, November 04). Retrieved October 6, 2020, from https://appen.com/datasets/audio-recording-and-transcription-for-medical-scenarios/

[6] Panayotov, V., Chen, G., Povey, D., Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). doi:10.1109/icassp.2015.7178964

[7] Wang, D., Zheng, T. F. (2015). Transfer learning for speech and language processing. 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). doi:10.1109/apsipa.2015.7415532

[8] Pfeiffer, J., Kamath, A., Rücklé, A., Cho, K., Gurevych, I. (2020). AdapterFusion: Non-destructive task composition for transfer learning. https://arxiv.org/abs/2005.00247

Appendix

Table 1: Fluent.ai dataset statistics after Data Split Method 1

Splits	# of unique speakers	# of utterances	# hours
Train	77	23,132	14.7
Valid	10	3,118	1.9
Test	10	3,793	2.4
Total	**97**	**30,043**	**19.0**

Table 2: Fluent.ai dataset statistics after Data Split Method 2

Splits	# of unique utterances	# of utterances	# hours
Train	191	23,132	14.7
Valid	26	3,135	2.0
Test	31	3,776	2.3
Total	**248**	**30,043**	**19.0**

Table 3: ATIS dataset statistics

Splits	# of unique utterances	# of utterances
Train	4,057	4,378
Valid	525	598
Test	838	890
Total	**5,420**	**5,866**

Table 4: SNIPS dataset statistics

Splits	# of unique utterances	# of utterances
Train	1,302	1,328
Valid	146	150
Test	179	182
Synthetic	1,311	29,216
Total	**1,627**	**30,876**

Table 5: Medical dataset statistics

Splits	# of unique utterances	# of utterances
Train	702	5,894
Valid	306	383
Test	300	384
Total	**384**	**6,661**

Table 6: SNIPS test accuracy for different learning rates

Learning Rate	Hidden Units = 256	Hidden Units = 512
1e-3	79.6%	**80.6%**
1e-4	74.2%	76.6%
1e-5	77.2%	78.8%