Rethinking End-to-End Evaluation of Decomposable Tasks: A Case Study on Spoken Language Understanding

Siddhant Arora[*] *Alissa Ostapenko*[*] *Vijay Viswanathan*[*] *Siddharth Dalmia*[*]
Florian Metze *Shinji Watanabe* *Alan W Black*

Language Technologies Institute, Carnegie Mellon University, USA
{siddhana,aostapen,vijayv,sdalmia}@cs.cmu.edu

Abstract

Decomposable tasks are complex and comprise of a hierarchy of sub-tasks. Spoken intent prediction, for example, combines automatic speech recognition and natural language understanding. Existing benchmarks, however, typically hold out examples for only the surface-level sub-task. As a result, models with similar performance on these benchmarks may have unobserved performance differences on the other sub-tasks. To allow insightful comparisons between competitive end-to-end architectures, we propose a framework to construct robust test sets using coordinate ascent over sub-task specific utility functions. Given a dataset for a decomposable task, our method optimally creates a test set for each sub-task to individually assess sub-components of the end-to-end model. Using spoken language understanding as a case study, we generate new splits for the Fluent Speech Commands and Snips SmartLights datasets. Each split has two test sets: one with held-out utterances assessing natural language understanding abilities, and one with held-out speakers to test speech processing skills. Our splits identify performance gaps up to 10% between end-to-end systems that were within 1% of each other on the original test sets. These performance gaps allow more realistic and actionable comparisons between different architectures, driving future model development. We release our splits and tools for the community.[1]
Index Terms: spoken intent prediction, end-to-end evaluation, generalization, challenge set, Fluent Speech Commands, Snips

1. Introduction

Complex, real-world tasks, such as the spoken language understanding (SLU) tasks of spoken intent prediction and spoken language translation, comprise of hierarchies of simpler sub-tasks. Spoken intent prediction combines automatic speech recognition (ASR) to process audio, followed by natural language understanding (NLU) to classify an utterance to a particular intent (intent prediction) [1]. Similarly, speech translation involves an ASR task followed by machine translation (MT) to translate a transcription of the input audio [2].

Deep, end-to-end models [3–8] are adopted for these complicated tasks due to advancements in model architectures and computing capabilities. End-to-end architectures typically outperform traditional, modular architectures without requiring domain expertise or feature engineering [9]. Moreover, end-to-end models avoid the error propagation arising from traditional approaches [10]. However, traditional modular or cascade architectures, naturally structured into sub-components that each address a specific sub-task, are more straightforward to evaluate. End-to-end models cannot quantify performance of decomposed sub-tasks [11], blurring the lines between the individual sub-tasks. Using pre-trained systems for some sub-tasks further reduces the chance of errors propagating to the downstream subtasks [12–14]. Thus, it is important to explicitly evaluate each sub-component of an end-to-end network.

Prior datasets for decomposable problems, however, often test only the top-level subtask. For example, the Fluent Speech Commands (FSC) [4] and Air Travel Information System (ATIS) [15] SLU benchmarks are open-speaker but not open-utterance, and thus, only effectively test speaker generalizability. Moreover, train-test overlap is a problem in modern question answering datasets [16]. In paradigms like encoder-decoder modeling or speech translation [11, 17–19], neural network components each solve different logical functions which combine to solve the final task. Therefore, standard benchmarks may effectively test a particular sub-network of a given system, masking any weaknesses of other model sub-components and providing an inflated estimates of model performance.

To address this, we present a dataset-agnostic framework for evaluating end-to-end model on decomposable tasks. Using spoken intent prediction as a case study, we focus on two popular benchmarks, FSC [4] and Snips SmartLights (Snips) [20,21] datasets. We provide evidence that the original test splits do not fairly evaluate the ASR and NLU subtasks of spoken intent prediction. Using our framework, we propose robust *Unseen* and *Challenge* splits that each contain two test sets: one test set with held-out speakers, and one with held-out utterances. For the Challenge set, we use coordinate ascent with speaker- and transcript-specific utility metrics to explicitly test for generalization to diverse speakers and varied phrasings of intents. Our experiments show the new test splits can amplify accuracy differences by up to 10% between sub-components of several state-of-the art models for spoken intent prediction, offering more in-depth analyses of strengths and weaknesses of various end-to-end modeling approaches. These splits have the potential to drive future modeling innovation, not only for this task, but for any similarly decomposable task.

2. Motivation

In the following section, we introduce the spoken intent prediction task and discuss limitations of existing SLU benchmarks.

2.1. Task Definition

Spoken intent prediction maps a spoken command (e.g. "Turn on the lights in the kitchen") and to a discrete, actionable set of slots (Action: "Activate"), (Object: "Lights"), (Location: "Kitchen"). This task challenges a model's speech recognition and semantic processing abilities: a good SLU model must generalize to new speakers *and* to new phrasings of similar intents.

[*] Equal Contribution. [1] We call our software package MASE (Multi-Aspect Subtask Evaluation): https://MASEeval.github.io/.

Table 1: *% WER values for Google ASR [24] on speakers with first language English and non English. (S, I, and D refer to substitution, insertion and deletion errors, respectively)*

First Language Spoken	% S	% I	% D	% WER
English	2.0	0.7	2.4	10.8
Non English	6.3	2.2	7.7	27.8

The FSC dataset [4] tests a model's ability to predict intents from commands used with a home voice assistant. Following traditional speech processing paradigms, the FSC test set consists of audio from speakers unseen during training. Although this test split measures generalization to new speakers, the test set fails to explicitly test generalization to new utterances. As Table 2 illustrates, the training set provides 100% coverage of the transcripts seen during test time. Snips, which is similar in content to FSC, does not release official splits, but typical split creation approaches [9,21,22] do not explicitly consider testing generalizability for each sub-task independently and instead use random splits which can lead to overtly optimistic estimate [23]. This can also mask performance gaps in thesub-tasks.

In the real world, we expect systems to understand the same commands spoken in different ways by speakers of diverse backgrounds. Thus, it is important to hold out new utterances to more robustly assess a model's semantic processing ability [25]. Moreover, open-speaker test sets should assess model generalizability to diverse demographics. In FSC, for example, all held-out speakers are native English speakers, while accented speakers are seen only during training time. To understand how this affects spoken language understanding evaluation, we used Google's ASR system [24] to generate transcripts from audio files in the dataset and computed Word Error Rates using the gold transcripts. Table 1 illustrates that the ASR model's WER on audio from speakers whose first language is not English is twice as high as the WER on audio from native English speakers. To develop technologies that are inclusive to different speaker demographics, it is important to create benchmarks that are representative of these diverse backgrounds.

3. Methodology

We discuss our approach for creating the open-utterance and open-speaker test sets for the *Unseen* and Challenge splits. The *Challenge* split uses additional constraints to make both test sets more difficult and realistic.

3.1. Dataset Optimization

We construct test splits using coordinate ascent [26, 27] over sub-task driven utility functions. Each coordinate direction corresponds to the test set assignment (either the open-speaker or open-utterance test sets) of a block of datapoints in the dataset. We first select an open-speaker test set, then choose the open-utterance test set. Finally, we randomly distribute the remaining instances into training and validation sets, preserving the original size ratio and intent distributions of these sets [4].

3.2. Unseen Split

We use the two functions to generate unseen-speaker and unseen-utterance splits with desirable qualities.

Unseen Speaker Set The FSC dataset contains speakers of various ages, native languages, English fluency levels, and genders, but the original, open-speaker test set is not representative of these groups. To ensure we are testing on speakers of diverse backgrounds, we minimize the symmetrised Kullback-Leibler (KL) divergence [28] between the discrete distributions of speaker demographics in the training and test sets.

Unseen Utterance Set When selecting unique utterances to hold out from training, we minimize the symmetrised KL divergence of the discrete intent label distributions between training and test sets. Utterances with the same intent are semantically similar, ensuring the semantic distributions of training and test sets match. We also minimize the KL divergence of the discrete distributions of transcript lengths between training and test sets.

3.3. Challenge Split

In addition to the constraints defined in the previous section, we define speaker-specific and transcript-specific utility functions to quantify the "hardness" for each subtask. When optimized, these functions create more challenging, realistic held-out sets. Notably, these test sets may capture dataset outliers due to noisy recordings, labeling errors, or poorly aligned data. Thus, we recommend using them in addition to the Unseen splits. The proposed utility functions are specific to spoken language tasks, but could be replaced with arbitrary task-specific objectives.

Challenge Speaker Set We compute the Word Error Rate, measured by insertion, deletion, and substitution errors, of Google's ASR model [24] to identify particularly challenging utterances. However, high WER is not always indicative of a reasonably hard example. According to previous work [29], substitution errors reflect confusions in ASR systems. Alignment errors, indicated by an increase in insertion and deletion errors and a large deviation between these quantities, are a sign of poor data quality. To produce a challenging speaker set without compromising data quality, we use the following utility function, U_{WER}:

$$U_{\text{WER}} = S - \alpha\,|I - D| - \beta\,I - \gamma\,D$$

where S, I and D refer to substitution, insertion, and deletion rates, respectively, such that we maximize S while minimizing I, D and their deviation, $|I - D|$. α, β and γ are hyperparameters. We empirically observe $\alpha = 0.05$, $\beta = 0.05$ and $\gamma = 0.4$ work well for the FSC dataset, producing a challenging split without compromising on test-set data quality.

Challenge Utterance Set A dataset with many unique n-grams makes the SLU task more difficult [25]. Thus, we create splits that minimize the n-gram overlap between our train and test set. We choose the Sentence BLEU [30] score as a proxy for n-gram overlap and use it in the following utility function:

$$U_{\text{BLEU}} = -BP * \exp(\sum_{i=1}^{4} \alpha_i \log(p_i))$$

where p_i is the modified precision [30] for each n-gram, α_i weighs the respective importance of the i^{th}-gram overlap, and *BP* is the brevity penalty [30] penalizing shorter sentences. Transcripts in the FSC dataset are 3-5 words in length, thus, considering only 1-gram and 2-gram overlap (i.e. $\alpha_1 = 0.5, \alpha_2 = 0.5, \alpha_3 = 0.0, \alpha_4 = 0.0$) worked well for holding out unique n-grams not seen during training.

Finally, to ensure both *Unseen* and *Challenge* speaker sets test generalization only to new speakers (and not utterances), we maximize the n-gram overlap with each split's respective training utterances. As Table 2 shows, our constructed speaker test sets have 100% n-gram overlap with their respective training sets. Unlike the original splits, in which test speakers are less demographically diverse than training set speakers, our new

Table 2: *Comparing data statistics and different models compared on original and proposed splits for the Fluent Speech Commands dataset. Speaker and Utterance Coverage refer to the percentages of test set speakers and utterances, respectively, observed in the training set. "Speaker KL" is the symmetrised Kullback-Leibler divergence of speaker demographic distributions between training and test sets. We ensure that our proposed splits have roughly the same # of examples in each test set as in the split proposed in [4]. We also construct different variants of the Unseen split by changing the random seed of our algorithm and report the standard deviation.*

	Dataset Statistics				E2E SLU Model [4] Test Accuracy			
Fluent Speech Command Test Set	Speaker Coverage	Utterance Coverage	Speaker KL	Test Size	No Pretraining	w/ Pretrained ASR (Frozen)	Finetune Word + Intent Layers	Finetune All Layers
Original Split	0%	100%	0.88	3793	96.8	98.5	99.1	97.2
Random Split	100%	100%	<0.01	3793	94.6	96.2	97.2	95.8
Unseen Split (Spk.)	0%	100%	0.01	3366	92.0 (±0.4)	92.9 (±0.2)	94.2 (±0.3)	93.9(±0.4)
Unseen Split (Utt.)	100%	0%	<0.01	3971	78.1 (±1.3)	86.0 (±0.7)	88.2 (±0.9)	88.3(±2.0)
Challenge Split (Spk.)	0%	100%	0.01	3349	87.2	90.9	92.3	91.1
Challenge Split (Utt.)	100%	0%	<0.01	4204	68.2	73.4	78.3	74.1

splits effectively minimize this distributional gap, as shown by the "Speaker KL" column. Each new test set has similar size to the original test set, and its distribution of utterance lengths is kept close to that of the training set to limit distribution shift.

4. Experiments

4.1. Comparing end-to-end SLU systems

We compare four different models from [4] on the *Original*, *Unseen*, and *Challenge* splits, as well as a stratified *Random* split (stratified over all intent labels). The four models are based on a three-stage neural architecture consisting of a phoneme layer, word layer, and intent layer. Each model uses different pretraining and finetuning schemes: using no pretraining, using a frozen pretrained ASR model (i.e. finetuning only the intent layers), finetuning only word and intent layers, or finetuning all layers. When pretraining, the phoneme and word modules are pretrained on the LibriSpeech dataset [31]. Using the Original test split, we successfully reproduced the results [4] for each of these freezing and unfreezing schedules.

Using the speaker and utterance test sets we create, we can highlight sub-task-level performance differences across the four models. As Table 2 illustrates, our Unseen and Challenge splits reveal that all models are better at generalizing to new speakers than to new utterances. However, all models achieve at least 3% lower accuracy on the Unseen and Challenge speaker sets compared with the Original held-out speaker set, indicating that current SLU models still do not generalize well to diverse speaker demographics. The results on the Challenge utterance set indicate that all models are significantly worse at generalizing to unique phrases of the same intent, suggesting an opportunity for enhancing semantic processing abilities of SLU models.

Our splits are also useful for comparing configurations of the same model. Intuitively, pretraining phoneme layers to detect phonetic patterns should help generalize to unseen speakers and utterances. However, Table 2 shows that on the Original and Random splits, there are small performance gaps between pretrained and non-pretrained models (1-2%, or ~50 test set examples), suggesting pretraining offers limited value, considering the resources it requires. In contrast, the performance gap becomes significant in the Unseen utterance set and both the speaker and utterance Challenge splits. The model without pretraining performs 10.1% worse than the best pre-trained model (pretraining with finetuned word and intent layers) in both the Unseen and Challenge utterance sets, corresponding to ≈460 more mistakes. The gaps are smaller in the speaker Challenge set, suggesting a non-pretrained ASR model generalizes better to new speakers than to new words or phonemes. These results corroborate previous findings [25] that finetuning models to the dataset's distinct acoustic and linguistic patterns improves generalization to new phrasings. Finally, we change the random seeds used to create the Unseen split to test the robustness of our methods. The relatively low standard deviations in performance, as seen in Table 2, illustrate that our method is stable.

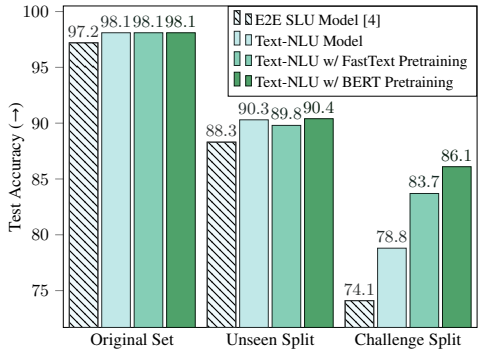

Figure 1: *Comparing text-based NLU models with the "Finetune All Layers SLU" baseline on the original and proposed utterance test sets. "Text-NLU Model" refers to a text-based NLU system using randomly initialized word embeddings.*

4.2. Gap between SLU and NLU

Using the utterance test sets of the Unseen and Challenge splits, we identified that SLU systems struggle to effectively capture lexical and semantic information. As an ablation study, we used gold transcripts to train and test the intent prediction component of the end-to-end model [4] in isolation (keeping all word and phoneme layers frozen). As a baseline, we train a text-based intent classification model initialized with random word embeddings that are finetuned during training. To incorporate semantic information into the word representations, we extract two types of word embeddings: (1) pretrained FastText [32] embeddings and (2) contextual BERT embeddings [33]. In Figure 1, we compare the baseline and semantically-enhanced text NLU models with the "Finetune All Layers" SLU model of [4]. Figure 1 illustrates that BERT pretraining can boost the accuracy of the intent subcomponent by 12% on the Challenge utterance set. These differences are not so apparent in the Unseen split, which is not as semantically challenging because it does not

Table 3: *Adding semantic word embeddings to the SLU system has only a minor effect (<1%) on the the Original split and proposed unseen-speaker splits. On the unseen-utterance splits, we see a magnified performance gap (>2%), in* ***bold****.*

E2E SLU Model [4]	Original	Unseen		Challenge	
		Spk.	Utt.	Spk.	Utt.
Pretrained ASR (Frozen)	98.5	92.9	86.0	90.9	73.4
+ FastText Pretraining	98.7	92.7	**88.3**	90.0	**75.5**

Table 4: *% WER values of the Google ASR system [24] on the original and proposed speaker test sets, and the corresponding accuracy of the Pretrained ASR (Frozen) model. (S, I, and D refers to substitution, insertion, and deletion, respectively.)*

Test Split	% S	% I	% D	% WER	SLU Acc.
Original	1.0	0.5	0.6	6.5	98.5
Unseen Speaker	2.1	1.0	2.5	12.1	92.9
Challenge Speaker	3.2	1.2	2.6	13.9	90.9

explicitly minimize n-gram overlap [25]. There is still a 2% gap between the SLU and NLU models' performance on the Unseen utterance split, suggesting that pretraining embeddings helps enhance semantic understanding.

Based on the results of our ablation study, we extend the frozen pretrained ASR model, the best reported SLU model from [4], with FastText embeddings. At each audio frame, the ASR module predicts a distribution over words; we use this compute a weighted average FastText word embedding [34] and pass it to the intent layer. Table 3 illustrates that enriching the ASR outputs with semantic information gives very minor improvements on the original test set, but provides >2% improvement to the unseen-utterance sets of both Unseen and Challenge splits, consistent with the experiments in the previous section.

4.3. Analyzing proposed utility functions

In Section 4.1, we illustrated how our optimized splits can distinguish model performance. We now verify our utility functions can effectively quantify the complexity of each subtask.

Word Error Rate As in Section 2.1, we use WER of Google's ASR system [24] to quantify the difficulty of our splits. Table 4 illustrates that WER is twice as high on the proposed speaker-diverse splits as on the original splits. Substitution errors are most prominent in Challenge set, indicating that we create a hard test set without necessarily compromising on data quality.

N-gram overlap For each proposed split, we compute the average BLEU score for the test set relative to the training set. Table 5 highlights that our test splits have much lower n-gram overlap with their training sets. Minimizing n-gram overlap while preserving intent distributions of training and test sets further tests a model's generalization to new phrasings of the same intents.

4.4. Extending to the Snips SmartLights dataset

To illustrate our methodology is dataset agnostic, we extend our approach to Snips SmartLights, a popular SLU dataset [35, 36].

Snips SmartLights dataset is unseen-utterance by design because all utterances are unique. Thus, we create a single Unseen test set that holds out speakers and utterances. We optimize both speaker and utterance utilities defined in Section 3.2 to create the split. Using the WER and n-gram based utilities defined in Section 3.3, we create separate speaker and utterance Challenge test sets. Following the Challenge test setup of FSC, we increase the speaker test set's n-gram overlap set with its train set to match that of the random split. We do not control for n-gram overlap in the Unseen split since it holds out both speakers and utterances. As a result, our Challenge speaker set may be easier than the Unseen split for SLU models. Moreover, Snips is a smaller dataset, so the Snips Challenge set's train, valid, speaker test, utterance test ratios are 75:10:7.5:7.5 as compared to 80:10:10 in the baseline random split [9].

Table 5: *BLEU score values for unigram, bigram, trigram and 4-gram for Original and proposed (utterance) test sets. Utterances shorter than order of a given n-gram were removed.*

Test Split	N-gram overlap				SLU
	1	2	3	4	Acc.
Original	100.0	100.0	100.0	100.0	98.5
Unseen Utterance	98.0	87.4	73.4	71.7	86.0
Challenge Utterance	91.0	69.9	66.4	66.6	73.4

Table 6: *Evaluating models on original and proposed splits for the Snips SmartLights dataset. Snips does not provide default splits, so we compare against a random split.*

E2E SLU Model [4]	Rand. [9] Split	Unseen Split	Challenge Split (Spk.)	Challenge Split (Utt.)
No Pretraining	60.4	27.3	37.8	45.2
w/ Pretrained ASR (Frozen)	83.2	78.5	73.2	67.4
Finetune Word + Intent Layers	88.0	80.9	82.6	75.3
Finetune All Layers	85.0	75.0	74.8	78.5

Using the same models as Section 4.1, we compare the performance on our proposed splits against a random split [9] in Table 6 (Snips does not release official splits). We observe similar results as in the FSC setting. Pretraining ASR models improves speaker test set performance by nearly 40-50% for the Unseen and Challenge splits. Moreover, finetuning improves performance for all splits, especially on the Challenge utterance set, on which both finetuned models achieve nearly 12% gains in performance. Thus, we illustrate that we can easily extend our approach to another SLU benchmark, and see effects consistent with those on the FSC dataset.

5. Conclusions

We present a novel, dataset-agnostic methodology for constructing splits for decomposable tasks, casting the construction of splits as an optimization problem over dataset-level utility functions. We release *Unseen* and *Challenge* splits for the FSC and Snips datasets to the community, and show evidence that these splits can amplify performance differences between sub-components of models. We recommend the use of the *Unseen* splits for testing in-domain performance and the *Challenge* splits for more extreme out-of-domain generalization scenarios. As our methodology is task-agnostic, we encourage the extension of our re-splitting method to other decomposable tasks, such as speech translation or visual question answering.

6. Acknowledgements

This work was supported in part by the National Science Foundation under Grant No. IIS2040926, the NSF SaTC Frontier project(CNS-1914486) [37], Bridges PSC (ACI-1548562, ACI-1445606) and an AWS Machine Learning Research Award.

7. References

[1] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may i help you?" *Speech communication*, vol. 23, no. 1-2, pp. 113–127, 1997.

[2] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Interspeech 2017, 18st Annual Conference of the International Speech Communication Association*, 2017.

[3] Y. Qian, X. Bian, Y. Shi, N. Kanda, L. Shen, Z. Xiao, and M. Zeng, "Speech-language pre-training for end-to-end spoken language understanding," *arXiv preprint arXiv:2102.06283*, 2021.

[4] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech Model Pre-Training for End-to-End Spoken Language Understanding," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, 2019.

[5] Y. Huang, H. K. Kuo, S. Thomas, Z. Kons, K. Audhkhasi, B. Kingsbury, R. Hoory, and M. Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," in *ICASSP*, 2020.

[6] H.-K. J. Kuo, Z. Tüske, S. Thomas, Y. Huang, K. Audhkhasi, B. Kingsbury, G. Kurata, Z. Kons, R. Hoory, and L. Lastras, "End-to-End Spoken Language Understanding Without Full Transcripts," in *Interspeech,*. ISCA, 2020.

[7] Y. Chen, W. Lu, A. Mottini, E. Li, J. Droppo, Z. Du, and B. Zeng, "Top-down attention in end-to-end spoken language understanding," in *ICASSP 2021*, 2021.

[8] M. Dinarelli, N. Kapoor, B. Jabaian, and L. Besacier, "A data efficient end-to-end spoken language understanding architecture," in *ICASSP*, 2020.

[9] B. Agrawal, M. Müller, M. Radfar, S. Choudhary, A. Mouchtaris, and S. Kunzmann, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," *ArXiv*, vol. abs/2011.09044, 2020.

[10] M. Sperber and M. Paulik, "Speech Translation and the End-to-End Promise: Taking Stock of Where We Are," in *Association for Computational Linguistic (ACL)*, Seattle, USA, 2020.

[11] S. Dalmia, B. Yan, V. Raunak, F. Metze, and S. Watanabe, "Searchable hidden intermediates for end-to-end models of decomposable sequence tasks," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021.

[12] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Pre-training on high-resource speech recognition improves low-resource speech-to-text translation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[13] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[14] D. Wang and T. F. Zheng, "Transfer learning for speech and language processing," in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2015.

[15] C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

[16] P. Lewis, P. Stenetorp, and S. Riedel, "Question and answer test-train overlap in open-domain question answering datasets," *arXiv preprint arXiv:2008.02637*, 2020.

[17] T. Kano, S. Sakti, and S. Nakamura, "End-to-end speech translation with transcoding by multi-task learning for distant language pairs," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1342–1355, 2020.

[18] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, "Direct speech-to-speech translation with a sequence-to-sequence model," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association*, G. Kubin and Z. Kacic, Eds. ISCA, 2019.

[19] O. Weller, M. Sperber, C. Gollan, and J. Kluivers, "Streaming Models for Joint Speech Recognition and Translation," in *European Chapter of the Association for Computational Linguistic (EACL)*, 2021.

[20] A. Saade, A. Coucke, A. Caulier, J. Dureau, A. Ball, T. Bluche, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, and M. Primet, "Spoken language understanding on the edge," in *Energy Efficient Machine Learning and Cognitive Computing workshop, NeurIPS*, 2019.

[21] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," in *Privacy in Machine Learning and Artificial Intelligence workshop, ICML*, 2018.

[22] L. Qin, W. Che, Y. Li, H. Wen, and T. Liu, "A stack-propagation framework with token-level intent detection for spoken language understanding," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[23] A. Østerskov Søgaard, S. Ebert, J. Bastings, and K. Filippova, "We need to talk about random splits," in *Proceeding of the 2021 Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2021.

[24] "Google speech to text api," https://cloud.google.com/speech-to-text, accessed: 2021-03-15.

[25] J. P. McKenna, S. Choudhary, M. Saxon, G. P. Strimel, and A. Mouchtaris, "Semantic complexity in end-to-end spoken language understanding," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association*, 2020.

[26] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, vol. 151, pp. 3–34, 2015.

[27] D. Metzler and W. Bruce Croft, "Linear feature-based models for information retrieval," *Inf. Retr.*, 2007.

[28] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, 1951.

[29] Y. Gaur, W. S. Lasecki, F. Metze, and J. P. Bigham, "The effects of automatic speech recognition quality on human transcription latency," in *Proceedings of the 13th International Web for All Conference*, ser. W4A '16, 2016.

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2002.

[31] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *ICASSP 2015*, 2015.

[32] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," in *LREC 2018*, 2018.

[33] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[34] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, 2017.

[35] L. Lugosch, B. Meyer, D. Nowrouzezahrai, and M. Ravanelli, "Using speech synthesis to train end-to-end spoken language understanding models," *ICASSP 2020*, 2020.

[36] S. Bhosale, I. A. Sheikh, S. H. Dumpala, and S. K. Kopparapu, "End-to-end spoken language understanding: Bootstrapping in low resource scenarios," in *Interspeech*, 2019.

[37] "Usable privacy policy project 2017," https://usableprivacy.org/.