Interspeech 2024
1-5 September 2024, Kos, Greece

AR-NLU: A Framework for Enhancing Natural Language Understanding Model Robustness against ASR Errors

Emmy Phung[1*], *Harsh Deshpande*[1*], *Ahmad Emami*[1], *Kanishk Singh*[2]

[1]JP Morgan Chase & Co., USA
[2]Columbia University, USA

{emmy.phung, harshsaiprasad.deshpande, ahmad.emami}@jpmchase.com
ks4038@columbia.edu

Abstract

A major challenge with pipeline spoken language understanding systems is that errors in the upstream automatic speech recognition (ASR) engine adversely impact downstream natural language understanding (NLU) models. To address this challenge, we propose an ASR-Robust NLU (AR-NLU) framework that extends a pre-existing NLU model by training it simultaneously on two input streams: human generated or gold transcripts and noisy ASR transcripts. We apply contrastive learning to make the model learn the same representations and predictions for both gold and ASR inputs, thereby enhancing its robustness against ASR noises. To demonstrate the effectiveness of this framework, we present two AR-NLU models: a **R**obust **I**ntent **DE**tection (RIDE) and **A**SR-**R**obust **BI**-encoder for **N**ame**D** **E**ntity **R**ecognition (AR-BINDER). Experimental results show that our proposed AR-NLU framework is applicable to various NLU models and significantly outperforms the original models in both sequence and token classification tasks.
Index Terms: spoken language understanding, ASR error robustness, contrastive learning

1. Introduction

There are two common approaches to extract downstream analytics from speech, such as intent detection or named entity recognition (NER): one is an end-to-end (E2E) spoken language understanding (SLU) system and the other is a pipeline SLU system. An E2E SLU system directly maps speech signal inputs to SLU outputs [1, 2]. A pipeline SLU system consists of two sub-systems: an ASR engine that transforms speech audio signals to text and an NLU model that extracts intents or entities from text transcripts. While recent research in SLU shows more focus and advancements in E2E SLU [3], this system presents some challenges. First, training an E2E SLU system requires a substantial amount of speech data with task-specific labels (i.e. intent and entity labels), which is expensive to annotate. In contrast, in a pipeline system, one can leverage existing ASR and NLU models that are pre-trained on much larger datasets and fine-tune them for a specific task, which often requires less data. Second, due to privacy concerns, there are instances where the audio is unavailable for system design, leaving only the anonymized ASR transcripts as accessible input data to downstream NLU models [4]. Pipeline SLU allows ASR and NLU components to be developed and deployed separately for such cases. Because of these advantages, pipeline SLU remains a practical approach.

This decoupled design, however, encounters a critical problem: ASR transcription errors can adversely impact NLU models that are pre-trained on formal text documents [5]. Prior approaches to improve ASR transcription quality include n-best re-ranking [6], lattice re-scoring [7, 8], and ASR-robust representation learning [5, 9]. There have been limited research studies that focus on improving the downstream NLU models only. One such study improves model performance on noisy ASR text by training the NLU model on both gold and ASR text inputs and forcing the final layer to generate similar outputs for each input [10].

In this paper, we propose an ASR-Robust NLU (AR-NLU) training framework that enhances NLU model robustness against ASR transcription errors. This framework holds promise for multiple downstream tasks, including sequence and token classification. Our proposed framework extends the work in [10] by first training the text encoder to learn the same representations for both gold and ASR inputs, via contrastive learning, then forcing the classification layer to generate similar outputs for each input. We demonstrate the use of this training framework in two SLU tasks with a different base model for each: BERT [11] for intent detection, a sentence classification task, and BINDER [12] for NER, a token classification task. Our key contributions include 1) the ASR-Robust NLU training framework, which is applicable to multiple downstream SLU sequence and token classification tasks, and operates without the need for audio inputs at inference, and 2) two ASR-robust NLU models that demonstrate significant improvements in intent detection and NER tasks.

2. Related Work

2.1. Intent detection

ASR errors can be categorized as deletion, insertion or substitution [13]. Here is an example illustrating how ASR errors affect downstream intent classification task. Suppose the ground-truth text is "play the weekend", ASR text could be "the weekend" ("play" is missing - deletion error) or "pay the weekend" ("play" is mis-transcribed as "pay" - substitution error). Since "play" is an important keyword to signal the speaker's intent in this case, such an error would challenge an NLU model trained on regular text data, to predict the $play\_music$, the correct intent label.

A typical intent classification model consists of two components: 1) a text encoder (for an NLU system) or an acoustic encoder (for an E2E speech-to-intent system) and 2) a classifier. Recent research has been focused on E2E approach, which involves utilizing a powerful pre-trained acoustic encoder and fine-tuning it for this specific task [14]. Other works attempted to improve the upstream ASR engine by training it simultaneously on multiple tasks, such as predicting tokens and durations [15] or predicting tokens, slots, and intent classes [2]. A novel

*These authors contributed equally.

10.21437/Interspeech.2024-1292

approach to enhance the performance of intent detection model on noisy ASR transcript involves training a cross-modal system. This system comprises an acoustic encoder and a text encoder, designed to project speech and gold transcript inputs of an utterance into a shared latent space, then tying the acoustic embedding and text embedding together via a triplet loss [4, 16]. By doing so, these systems leverage the semantically powerful pre-trained text encoder to refine the training of the acoustic encoder. However, both systems require access to audio at inference, which is not always available. One of the rare attempts to improve the NLU component of a pipeline intent detection system is by forcing the classification layer to produce the same predicted class probability distribution for both gold and ASR input streams using Kullback-Leibler (KL) divergence loss [10]. Our proposed RIDE architecture, demonstrated in Figure 1, effectively addresses the limitations of [4, 16] and extends [10] by: 1) training the text encoder on both gold and ASR text inputs, using contrastive learning and 2) training the classifier to learn the same class probability distributions for both inputs, using KL divergence loss.

2.2. Named entity recognition

For entity recognition, ASR errors pose a greater challenge because named entities are less common words that may not exist in the vocabulary or training data used for ASR, hence they are especially susceptible to transcription errors. For example, a company named "EBRD" could be mis-transcribed as "IBID" or a person named "Oleg" as "Eg". Because of these errors, NER labels annotated on gold transcripts (gold labels) are incompatible with ASR transcripts, thereby should not be used to evaluate NER model predictions on ASR transcripts [16]. In this work, we obtain pseudo-labels on ASR transcripts to account for such differences, similar to [16]. One difference, however, is that we use a label-transfer algorithm without human supervision, which will be explained in Section 4.1.

One approach to expand the robustness of traditional NER models [17] against ASR errors includes modifying the loss function used in the CRF layer to account for missing or uncertain token-label pairs in ASR inputs [16]. Recent works, leveraging the semantically powerful BERT encoder, replace CRF with a linear token classification layer [11]. A novel approach in NER is Bi-Encoder for Named Entity Recognition (BINDER) which uses the distance between an input text representation and an entity type representation as a dynamic threshold to predict entity spans [12]. This work has shown competitive performance over traditional NER models on several datasets, so we decide to use it as our base NER model. Our proposed model, ASR-Robust BINDER (AR-BINDER) builds upon the original BINDER model by training the text encoder simultaneously on both gold and ASR inputs, with the corresponding NER labels, and applying contrastive learning to tie the embeddings of ASR and gold text inputs together as well as to tie the embeddings of entity text coming from both inputs.

3. Model Architecture and Training Framework

3.1. Robust Intent Detection for ASR (RIDE)

As demonstrated in Figure 1, the proposed RIDE architecture consists of one text encoder (e.g. BERT) and a classifier with two fully connected layers. The text encoder takes in two input streams: a gold transcript and an ASR transcript, and generates gold and ASR text embeddings. The two embeddings are tied together in a shared embedding space via a triplet loss [4], which we refer to as the embedding loss. The generated embeddings are subsequently fed into a shared intent classifier, which predicts an intent probability distribution for each input stream. Our second component in the RIDE architecture is the KL divergence loss that penalizes the difference between predicted distributions coming from the gold and ASR input streams, as inspired by [10].

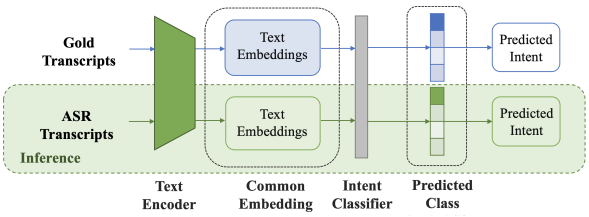

Figure 1: *Architecture of the RIDE model*

In summary, RIDE is trained using a combination of the following three losses:

$$L = L_{CE} + \epsilon_1 * L_E + \epsilon_2 * L_{KL}$$

where L_{CE} is the classification loss, specifically Cross Entropy (CE) loss, L_E denotes the embedding loss, and L_{KL} denotes the KL divergence loss, described in [10]. ϵ_1 and ϵ_2 are hyperparameters that determine the weights for these loss terms.

3.1.1. Embedding Loss

Our motivation behind this loss term is to force the text encoder to generate similar embeddings for both the gold and ASR text inputs, thereby improving its robustness against ASR noise. This can be achieved via a triplet loss, which moves the ASR text embedding of the current example closer to the gold text embedding of the positive example and away from that of the negative example. Here is how we form a triplet: for an utterance u, we randomly sample a positive utterance u_p from the same intent class, as well as a negative utterance u_n from a different intent class. We obtain E_u^{ASR}, an embedding from the ASR transcript of this utterance, and two gold embeddings, E_p^{GOLD} and E_n^{GOLD}, from u_p and u_n, respectively. The triplet loss is then defined as follows:

$$L_E = max(0, \beta + d(E_u^{ASR}, E_p^{GOLD}) - d(E_u^{ASR}, E_n^{GOLD}))$$

where, $d(A, B)$ is the cosine distance between two embeddings, A and B, and β is the margin [16].

3.2. Robust NER for ASR

AR-BINDER extends the architecture of its base model, BINDER [12], by adding an ASR input stream along with the original gold text input. The model consists of two text encoders: an entity type encoder to produce entity type representations ($[CLS]_{entity\_type}$) based on entity descriptions, and a text encoder to produce sentence and token representations of the input text. Note that the sentence representation ($[CLS]_{input}$) plays an important role in the original BINDER architecture as it determines a threshold for entity prediction: a span is predicted as entity span if the distance between the span representation and entity type representation is smaller than that between the input and entity type representation, as shown in Figure 2.

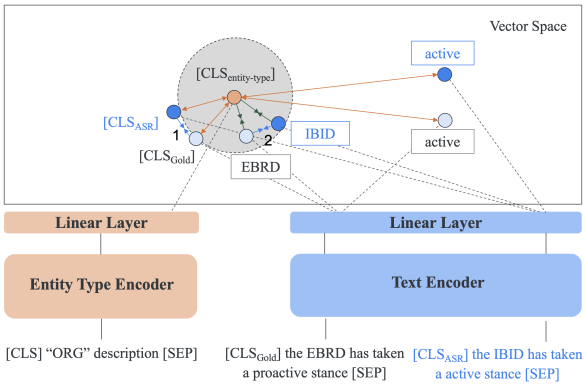

Figure 2: *Architecture of ASR-Robust BINDER*

In AR-BINDER, we apply the same BINDER architecture for both gold and ASR text input streams, to obtain $[CLS]^{GOLD}_{input}$ and $[CLS]^{ASR}_{input}$, and use the distance between each of them and $[CLS]_{entity\_type}$ as the thresholds for entity prediction. Therefore, our first extension to the BINDER architecture involves doubling its input streams to accommodate ASR training inputs and adding ASR loss terms, denoted L^{ASR}_B, for this input stream (1). Next, we apply embedding loss to tie the embeddings of the gold and ASR transcript of the same utterance, captured by *distance 1* in Figure 2 (2). Combining (1) and (2), we obtain AR-BINDER I. Finally, we extend the same concept to entity span embeddings, training the encoder to learn the same representation for gold entity spans and ASR pseudo-labeled entity spans, captured by *distance 2* in Figure 2 (3). AR-BINDER II is our final architecture, comprising of (1), (2), and (3).

In summary, the training loss for AR-BINDER models for the NER task is the sum of the following four components.

$$L = L^{GOLD}_B + L^{ASR}_B + \epsilon_1 * L_{CLS} + \epsilon_2 * L_{Span}$$

where L^{GOLD}_B and L^{ASR}_B represent BINDER losses applied on gold and ASR input streams. L_{CLS} denotes the cosine distance between the sentence embedding of gold and ASR text inputs of the same utterance, which we refer to as embedding loss. L_{Span} is the embedding loss on the gold and ASR entity span embeddings. ϵ_1 and ϵ_2 are hyper-parameter of the AR-BINDER model, determining the weights of the two added loss terms.

3.2.1. *Entity Span Embedding Loss*

We first need to align the entities in gold and ASR transcripts within our training inputs. To obtain these gold and ASR entity pairs, we develop a label-transfer algorithm, which, as mentioned, is used to infer NER spans on ASR transcripts based on the entities labeled on gold transcripts (see 4.1). Next, we apply contrastive loss to tie the gold and ASR entity span pairs:

$$L_{Span} = \sum_{(i_s,i_e,j_s,j_e)\in \mathbf{I}_U} d(E_{i_s,i_e}, E_{j_s,j_e})$$

where I_U represents the set of all gold-ASR entity pairs in utterance U, in which a gold entity i and its paired ASR entity j are specified by their start and end indices, denoted by (i_s, i_e) and (j_s, j_e). E_{i_s,i_e} represents the span embeddings of entity i.

Table 1: *Test Accuracy on Intent Detection Task*

Model	Best config (ϵ_1,ϵ_2)	E2E Accuracy
BERT$_{GOLD}$	-	73.52
BERT$_{ASR}$	-	75.72
BERT$_{GOLD\&ASR}$	-	71.06
Ruan et al [10]	-	77.16
RIDE I	(1, 0)	77.04
RIDE II	(1, 5)	**78.05**

4. Experiments and results

4.1. Datasets

We perform our experiments on two publicly available datasets: SLURP for intent detection and SLUE for NER.

SLURP — Spoken Language Understanding Resource Package [18] is an end-to-end speech-to-intent labeled dataset consisting of 72K recordings and a set of 91 intent classes such as *play_music*, *calender_set* and *weather_query*.

SLUE — Spoken Language Understanding Evaluation dataset [19] provides new transcriptions and annotations on subsets of VoxCeleb and VoxPopuli. We use the VoxPopuli dataset with NER labels, consisting of 8,500 thousands examples and 17 entity types, including $PERSON, DATE, ORG$. We use the normalized text as our gold transcripts.

ASR Transcripts — For both datasets, we use out-of-the-box AWS Transcribe service[1] to generate ASR transcripts. The word error rate (WER) of transcribing SLURP is 22% and that of transcribing SLUE is 19%, which are on par with observed performance for these datasets [18, 19]. We follow the provided train, validation, and test splits. For NER task, as mentioned, we develop a label-transfer algorithm to obtain pseudo NER labels on ASR text. We first align our gold and ASR transcripts, then search for gold entities, which are annotated labels on gold transcripts, in the aligned ASR text using string matching and position matching. Our threshold to find an entity match is that the Levenshtein distance between an entity span found in ASR text and the original gold entity cannot exceed half the length of the gold entity. We report our final results on both sets: ASR pseudo-label (ASR_p set) for NER performance evaluation only, and gold label set ($Gold$) for end-to-end SLU system evaluation.

4.2. Metrics

Intent detection is a multi-class classification problem. Following existing benchmarks on SLURP, we evaluate and benchmark our models using overall accuracy.

For NER task, we use the NER evaluation metrics provided in the SLUE toolkit[2]. We use micro-average F1 (F1) across all entities as our primary metrics.

4.3. Results

4.3.1. *Intent detection*

Table 1 summarizes our experimental results on intent detection. Our baselines are BERT$_{GOLD}$, BERT$_{ASR}$, BERT$_{GOLD\&ASR}$, which are BERT with a linear classifier, trained on three different datasets: gold transcripts, ASR transcripts, and a combination of both. Note that we do not include

[1] https://aws.amazon.com/transcribe

[2] https://github.com/asappresearch/slue-toolkit/tree/main

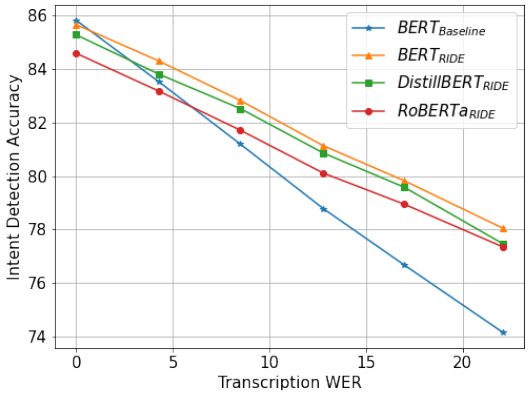

Figure 3: *Performance of RIDE models at different WERs*

Table 2: *Test F1 Scores on Named Entity Recognition Task*

Model	Best config (ϵ_1,ϵ_2)	ASR_p (NER Eval)	*Gold* (E2E Eval)
SLUE baseline[3]	-	-	47.4
BERT	-	72.1	69.6
BINDER	-	72.2	69.7
AR-BINDER I	(40, 0)	73.6	71.0
AR-BINDER II	(40, 60)	**73.9**	**71.4**

the results from [14, 15], and other benchmarks on SLURP because they use an E2E approach, which is not comparable to ours. To compare our work with Ruan et al [10], we replicate their work by having only KL loss, freezing the embedding loss, in RIDE but one difference is that we use BERT encoder instead of Bi-LSTM. The results show that RIDE I (using embeddings loss only) outperforms the baseline performance on ASR test set by a significant margin (4.8%), and performs on par with Ruan et al. RIDE II, where we have both KL divergence and embedding loss, achieves the best performance.

We assess RIDE's robustness across various levels of ASR quality. To achieve this, we create n evaluation sets with different WERs by varying the ratio at which we randomly correct ASR errors in our test set. We further examine the effectiveness of our RIDE framework to different base models, including BERT, RoBERTa [20] and DistillBERT [21]. The baseline we choose for this comparison is BERT trained on gold transcripts only. Graph 3 indicates that the performance of the baseline model deteriorates more quickly and significantly as WER increases, compared to models trained with RIDE framework. At a WER of 15-20%, the gap between our proposed models and the baseline is up to 5%. These experiments prove that our AR-NLU system consistently outperforms the baseline at all levels of WERs and is most beneficial when ASR quality is poor.

4.3.2. Named entity recognition

For a fair comparison, we juxtapose our model with that from the SLUE benchmark[3] with the most similar architecture: a pipeline system with ASR WER of 18.4% and using BERT for NER. For our two base models, BERT and BINDER, we experiment with three different sets of training data: gold transcripts with gold NER labels, ASR transcripts with pseudo-labels, and a combined dataset of both. Since our focus is on the NER performance only, not E2E, we select the best model based on its performance on ASR pseudo-label test set (ASR_p) and only report the best results: BERT baseline is achieved when trained on the combined dataset and BINDER baseline is achieved when trained on ASR training data. On the E2E metrics, our base models outperform the SLUE benchmarks by a strong margin, even when our ASR quality is worse.

Table 2 shows that among three training regimes for AR-BINDER, AR-BINDER I outperforms existing benchmarks and both baselines by a strong margin. AR-BINDER II is the best performer on both ASR_p set and E2E evaluation, outperforming the baselines by about 2.5% on both sets. It should be noted that the reported metric is F1 averaging on a wide set of entity types, among which, some are less challenging for both ASR and NER models, such as CARDINAL (i.e. one, two) and ORDINAL (i.e. first, second). This may explain why we have observed on-par performances between BERT and BINDER baselines. Therefore, to understand the impact of our robust training regime, we perform analysis on a subset of challenging entities, where ASR engine is more likely to make mistakes. On two entity classes with the highest character error rate (CER), PERSON and ORG, AR-BINDER II outperforms BERT baseline model by 7% and BINDER baseline model by 5%. This indicates that our AR-NLU framework yields the greatest benefits when it comes to challenging entity types.

5. Discussion

Our proposed AR-NLU training framework shows value in the intent detection task by improving the model's ability to accurately identify speakers' intents despite ASR errors. However, its impact may not be as clear for the NER task because mistranscribed entities, despite being correctly identified, may not be used directly. We believe that recognition of such entities is beneficial to other downstream analytics. Some could take advantage of correctly recognized entity types, with some tolerance for entity spelling errors. These errors could also be overcome by some further downstream tasks, such as entity linking [22]. Furthermore, while the quality of a label-transfer algorithm cannot be as good as as human annotation, the improvements on E2E metrics in Table 2 indicate that our ASR pseudo-label test results align with the gold test results, and models benefit from training on ASR pseudo-label sets. This approach allows us to automatically generate ASR pseudo-labels on transcripts produced by multiple ASR engines.

6. Conclusion

We present AR-NLU, an ASR-Robust NLU training framework designed to enhance the robustness of NLU models against ASR errors. The cornerstone of this work lies in training the model simultaneously on both gold and ASR transcripts and applying contrastive learning to tie the input text embeddings, intermediate outputs (e.g. predicted class probability distribution) or embeddings of final outputs (e.g. entity spans) from the two input streams. This approach enables the text encoder and classifier to effectively adapt to ASR noises and perform robustly on ASR inputs as they do on clean transcripts. Our framework extends and significantly improves the performance of BERT-based models for intent detection and BINDER model for NER. More importantly, experimental results show that our framework provides the most value when handling poor-quality ASR transcripts and challenging entity types.

[3]https://asappresearch.github.io/slue-toolkit/leaderboard_v0.2.html

7. References

[1] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *ArXiv*, vol. abs/1904.03670, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:102352396

[2] M. Saxon, S. Choudhary, J. P. McKenna, and A. Mouchtaris, "End-to-end spoken language understanding for generalized voice assistants," in *Proc. Interspeech 2021*, 2021, pp. 4738–4742.

[3] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[4] S. Cha, W. Hou, H. Jung, M. Phung, M. Picheny, H.-K. J. Kuo, S. Thomas, and E. da Silva Morais, "Speak or chat with me: End-to-end spoken language understanding system with flexible inputs," in *Proc. INTERSPEECH 2021 – 22nd Annual Conference of the International Speech Communication Association*, 2021, pp. 4723–4727.

[5] C.-W. Huang and Y.-N. Chen, "Learning asr-robust contextualized embeddings for spoken language understanding," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8009–8013.

[6] A. Ogawa, M. Delcroix, S. Karita, and T. Nakatani, "Improved deep duel model for rescoring n-best speech recognition list using backward lstmlm and ensemble encoders," 09 2019, pp. 3900–3904.

[7] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "Latticernn: Recurrent neural networks over lattices," 09 2016, pp. 695–699.

[8] K. Wei, P. Guo, H. Lv, Z. Tu, and L. Xie, "Context-aware rnnlm rescoring for conversational speech recognition," in *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2021, pp. 1–5.

[9] P. G. Prashanth Gurunath Shivakumar, Mu Yang, "Spoken language intent detection using confusion2vec," in *Proc. Interspeech 2019*, 2019.

[10] W. Ruan, Y. Nechaev, L. Chen, C. Su, and I. Kiss, "Towards an asr error robust spoken language understanding system," in *Proc. INTERSPEECH 2020 – 21st Annual Conference of the International Speech Communication Association*, 2020.

[11] 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: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[12] S. Zhang, H. Cheng, J. Gao, and H. Poon, "Optimizing bi-encoder for named entity recognition via contrastive learning," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=9EAQVEINuum

[13] R. Errattahi, A. E. Hannani, and H. Ouahmane, "Automatic speech recognition errors detection and correction: A review," in *International Conference on Natural Language and Speech Processing*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:52272147

[14] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *ArXiv*, vol. abs/2111.02735, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:242757022

[15] H. Xu, F. Jia, S. Majumdar, H. Huang, S. Watanabe, and B. Ginsburg, "Efficient sequence transduction by jointly predicting tokens and durations," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.

[16] B. Agrawal, M. Müller, S. Choudhary, M. Radfar, A. Mouchtaris, R. McGowan, N. Susanj, and S. Kunzmann, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7157–7161.

[17] R. Panchendrarajan and A. Amaresan, "Bidirectional LSTM-CRF for named entity recognition," in *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, S. Politzer-Ahles, Y.-Y. Hsu, C.-R. Huang, and Y. Yao, Eds. Hong Kong: Association for Computational Linguistics, 1–3 Dec. 2018. [Online]. Available: https://aclanthology.org/Y18-1061

[18] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A spoken language understanding resource package," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 7252–7262. [Online]. Available: https://aclanthology.org/2020.emnlp-main.588

[19] S. Shon, A. Pasad, F. Wu, P. Brusco, Y. Artzi, K. Livescu, and K. J. Han, "Slue: New benchmark tasks for spoken language understanding evaluation on natural speech," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7927–7931, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:244463257

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[21] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[22] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, "Scalable zero-shot entity linking with dense entity retrieval," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 6397–6407. [Online]. Available: https://aclanthology.org/2020.emnlp-main.519