INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

PhonemeBERT: Joint Language Modelling of Phoneme Sequence and ASR Transcript

Mukuntha Narayanan Sundararaman[*], *Ayush Kumar, Jithendra Vepa*

Observe.AI, India

mukunthas@observe.ai, ayush@observe.ai, jithendra@observe.ai

Abstract

Recent years have witnessed significant improvement in ASR systems to recognize spoken utterances. However, it is still a challenging task for noisy and out-of-domain data, where ASR errors are prevalent in the transcribed text. These errors significantly degrade the performance of downstream tasks such as intent and sentiment detection. In this work, we propose a BERT-style language model, referred to as **PhonemeBERT** that learns a joint language model with phoneme sequence and ASR transcript to learn phonetic-aware representations that are robust to ASR errors. We show that PhonemeBERT leverages phoneme sequences as additional features that outperform word-only models on downstream tasks. We evaluate our approach extensively by generating noisy data for three benchmark datasets - Stanford Sentiment Treebank, TREC and ATIS for sentiment, question and intent classification tasks respectively in addition to a real-life sentiment dataset. The results of the proposed approach beats the state-of-the-art baselines comprehensively on each dataset. Additionally, we show that PhonemeBERT can also be utilized as a pre-trained encoder in a low-resource setup where we only have ASR-transcripts for the downstream tasks.

Index Terms: language modelling, phoneme, asr, bert

1. Introduction

With the proliferation of voice-enabled technologies spoken language understanding (SLU) has become significantly ubiquitous. The general modus-operandi of SLU systems is to convert voice into text using an ASR engine and use Natural Language Understanding (NLU) on the transcribed text to comprehend the speaker's intents and requests. Despite advancements in ASR systems, domain adaptation and word recognition in noisy setups remain a big challenge. In a typical SLU system that operates on the ASR outputs, these errors degrade the performance of the system on the downstream tasks [1].

The approaches tried by scientific community to address the errors in the ASR system can be broadly categorized into four groups: a) *Modelling word confidence*: Liu et al. [2] proposes a BERT model that jointly encodes the word confidence network and the dialog context. Ladhak et al. [3] proposes LatticeRNN to encode the ambiguities of the ASR recognition for the intent classification task; b) *ASR correction*: Weng et al. [4] presents a contextual language correction on ASR outputs jointly with modelling LU task that learns from ASR n-best transcriptions. Mani et al. [5] use machine translation technique for domain adaptation to correct ASR mistakes in medical conversations; c) *End-to-End SLU*: Serdyuk et al. [6] explore the possibility to extend the end-to-end ASR learning to include NLU component and optimize the whole system for SLU task while Ghannay et al. [7] study end-to-end named entity and semantic concept extraction from speech to circumvent the errors arising from the ASR pipeline; d) *Phoneme enhanced representations*: Yenigalla et al. [8] use word2vec generated phoneme embedding for emotion recognition task. Fang et al. [9] propose word2vec based approaches to learn phoneme embeddings capturing pronunciation similarities of phonemes to make classification robust to ASR errors.

Our work falls in the category of *Phoneme enhanced representations* to learn representations that are robust to ASR transcription errors. In one of the earlier works, Yenigalla et al. [8] trained word2vec based phoneme and ASR embeddings independent of each other for SLU tasks. Fang et al. [9] propose multiple variants of phoneme embedding which either require alignments between the reference transcript and the phoneme sequence (*p2va*) or learns phoneme embedding in isolation to the ASR transcript (*p2vc, s2s*). Additionally, the authors derive phoneme sequence obtained on top of the ASR transcripts. We note two main drawbacks in these methods: *a)* Learning embeddings from isolated sequences prohibits leveraging the complementary information present in the ASR transcript and phoneme sequence; *b)* Due to direct conversion of words in ASR transcript to phoneme sequences, the errors from ASR transcript are propagated to the phoneme sequence which induces redundant errors that can lead to sub-optimal results.

To address these drawbacks, we propose PhonemeBERT that jointly models the phoneme and ASR sequence with an added benefit of producing better results in a low-resource setup. The main contributions of this work are:

1. PhonemeBERT: A method to jointly model ASR transcripts and phoneme sequences using a BERT-based pre-training setup is proposed.
2. Extensive experiments are carried out on benchmark and real life dataset to show the method's effectiveness. Results show that joint language model in PhonemeBERT can leverage phoneme sequences as complementary features, making it robust to ASR errors.
3. Pre-trained PhonemeBERT can be effectively used as word-only encoder in a low-resource downstream setup where phoneme sequences are not available, still producing better results than word-only language model.
4. We also release our generated datasets used in the work for research usages: `https://github.com/Observeai-Research/Phoneme-BERT`

2. Proposed Methodology

In the proposed work, we aim to learn representations that are robust to ASR errors. To accomplish this objective, we utilize the phoneme sequence in addition to the words (ASR transcript) to train a joint language model (LM), named *Phone-*

[*]Work done during internship at Observe.AI

http://dx.doi.org/10.21437/Interspeech.2021-1582

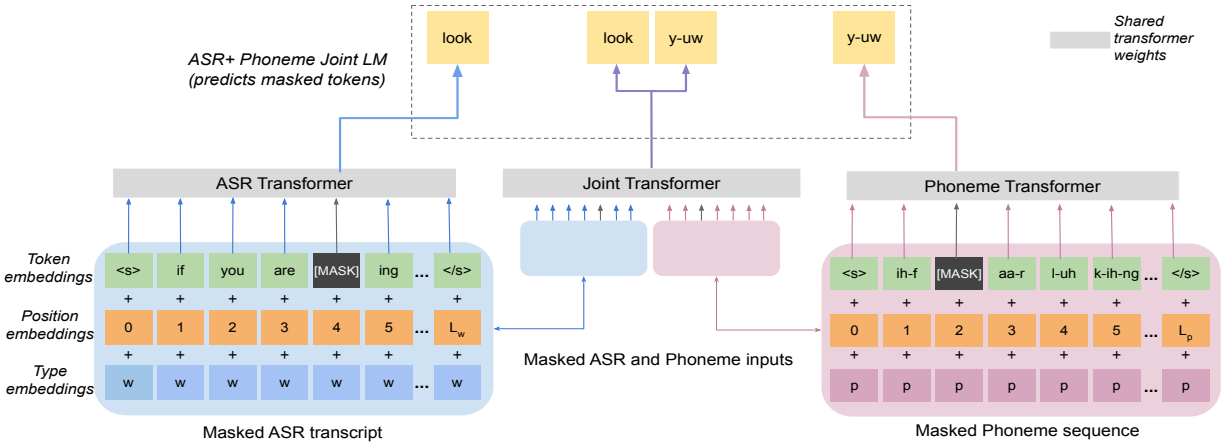

Figure 1: *Proposed architecture of PhonemeBERT that jointly models phoneme sequence and ASR transcript.*

meBERT (Figure 1). The vanilla masked language modelling (MLM) setup is trained by predicting the masked tokens using the contextual information from the surrounding tokens. We extend this setup to a joint MLM training that predicts masked tokens in both ASR transcript and the phoneme sequence leveraging the coupled information from the two sequences. In a joint MLM setup, we use a joint transformer to feed both ASR transcript and phoneme sequence to predict the masked tokens. To predict a token masked in the word sequence, the model can either attend to surrounding word tokens or to the phoneme sequence, encouraging the model to align the word and phoneme representations, making the word representations more phonetic-aware. The joint modelling of word and phoneme helps the model to leverage the phoneme context if the word context is insufficient to infer the masked token in the word sequence and vice-versa. In addition to joint loss, we also use shared transformers for ASR, phoneme and combined sequence represented by ASR transformer, phoneme transformer and joint transformer respectively in Figure 1. Shared transformers ensures that shared representations are learnt that facilitate predicting masked token with joint as well as individual contexts while not adding complexity to the overall architecture. The proposed joint setup is optimized with a training objective comprising three loss functions: ASR MLM loss, phoneme MLM loss and joint MLM loss as defined in Eqn 1:.

$$\mathcal{L} = \sum_{i=1}^{|A_{mask}|} \mathcal{L}_{mlm}(a_i|\hat{A}) + \sum_{j=1}^{|P_{mask}|} \mathcal{L}_{mlm}(p_j|\hat{P}) + \sum_{k=1}^{|A_{mask}+P_{mask}|} \mathcal{L}_{joint}(t_k|\hat{A},\hat{P}) \quad (1)$$

where, $\mathcal{L}(t_i|\hat{T})$ represents the MLM loss for predicting token i in using information from masked sequence $\hat{T} \in (A, P)$ representing masked ASR transcript and phoneme sequence respectively. The loss is accumulated over all masked tokens (A_{mask}, P_{mask}) across both ASR transcript and phoneme sequence.

To train PhonemeBERT, the model is initialized with weights from RoBERTa [10]. RoBERTa is a robustly optimized pre-trained LM using masked language modeling (MLM) as the pre-training objective. The training sequence in the proposed model contains the ASR transcript concatenated with the phoneme sequence (Figure 1). We utilize Byte Pair Encoding (BPE) that represents the vocabulary of words and phonemes. BPE has been shown to be an effective method to handle large vocabularies with sub-word units. It has also been shown to work well with sub-words based on phonemes [11]. We use a byte-level BPE vocabulary like in RoBERTa [10], which allows using spaces (pauses in the phoneme sequence) as part of the BPE tokens. To represent phoneme sequences, we train our model with a vocabulary consisting of 600 phoneme sub-word units constructed over the phoneme sequences from the pre-training corpus. Finally, we represent each BPE-token using three embeddings: *token, position* and *type embeddings*. Token embeddings for word sequence are initialized from RoBERTa base model, while phoneme-BPE token embeddings are randomly initialization and trained. Position embedding is initialized from RoBERTa base model. The position embeddings for the phoneme sequence start at 0 to enable soft yet unsupervised alignment between word and phoneme tokens. Following the training regime of RoBERTa, we sample randomly 15% of the BPE tokens from the word and phoneme sequences and replace them by `[MASK]`. Of the selected tokens, 80% are replaced with `[MASK]`, 10% are left unchanged, and 10% are replaced by a randomly selected vocabulary token of the same type (word or phoneme). In addition to token and position embedding, we randomly initialize a *type embedding* denoting whether the token is coming from a 'word' or a 'phoneme' sequence.

Table 1: *Statistics of pre-training dataset*

Dataset	#Sentences	Speech Hours	Vocab Size
LibriSpeech	35411	360*h*	37468
AmazonReviews	75000	684*h*	99583
SQuAD v1.1	50000	278*h*	134287

2.1. Pre-training Dataset

With a primary objective to learn generic representations for the noisy data, we choose three datasets from different domains to accomplish the pre-training step (Table 1). We use, LibriSpeech corpus [12], Amazon reviews [13] and Squad v1.1 [14] for pre-training step. Specifically, we use LibriSpeech train-clean-360 corpus, while we randomly sample the Amazon reviews and

SQuAD dataset to collect specified number of sentences across the multiple topics and paragraphs. Once we collect the raw text corpus, we follow a similar strategy as described in Fang et al. [9] to create noisy speech corpus. We use Amazon Polly [§] to convert the raw text to speech. Next, we apply Speech Synthesis Markup Language (SSML) tags to the audio to change the prosody of the produced speech. We also add ambient noise [§] to the speech to make the data align with real-life data. For a specific speech file, we apply one SSML tag and one ambient noise. Once the noisy speech data is generated, we use Amazon Transcribe to transcribe the audio. Since our hypothesis is to create a language model that is robust to ASR errors, we generate data spanning across different levels of word-error-rate (WER). We choose to reject any transcription that has WER less than 5% or more than 40%. We additionally use more than one noise level for 25% of the pre-training corpus to capture characteristics of word and phoneme sequences at different noise levels. The mean WER for the generated corpus is 30.79%.

2.2. Phoneme sequence generator

To address the drawback of accumulating the errors in ASR transcript by directly converting it into phoneme sequence, we generate phoneme sequence from a separate phonemic listen-attend-spell (LAS) model explained in [15]. The phonemic LAS model is a sequence-to-sequence model with phoneme as its output unit. The primary objective to use an independent acoustically trained phoneme-generator is to make it non-biased with language modelling and focus only on acoustic information in the speech. We aim to retain the complementary information in the word and phonemes sequences especially for noisy speech. We use LibriSpeech data to train the phoneme model achieving the phoneme error rate (PER) of 10.2 on test-other data while 3.6 on test-clean data. Further specific details of phoneme LAS model is beyond the scope of this work.

2.3. PhonemeBERT as encoder in low-resource setup

In many practical setups, we do not always have an advantage of phoneme sequence available to us. The only resort in this case is to fine tune a vanilla word-only model on the downstream task. We demonstrate that PhonemeBERT can be used as a pre-trained encoder which when fine-tuned on the downstream task with word-only input leads to a better result than a vanilla setup (discussed in Section 3.4). Thus, a jointly pre-trained PhonemeBERT learns phonetic-aware representations which outperform word-only representations.

3. Experiments and Results

3.1. Dataset

For experimental evaluation of the approach, we use three benchmark datasets as described below (Table 2). We use the exact same setup to generate the noisy version of the dataset and the corresponding ASR transcript and phoneme sequences.

- SST-5: This is Stanford Sentiment Treebank dataset [16] containing the labels in five point scale from very negative to very positive sentiment.
- TREC: This is a question classification dataset [17] with coarse (6 classes) and fine-grained (50 classes) label set.

[§] https://aws.amazon.com/polly

[§] www.pacdv.com/sounds/ambience_sounds.html

Table 2: *Statistics of downstream task dataset*

Dataset	#Class	Avg. Length	Train	Test	WER
SST-5	5	17.38	8534	2210	31.86
TREC	6, 50	8.89	5452	500	32.93
ATIS	22	11.14	4978	893	29.11

- ATIS: This is an audio recordings of people making flight reservations [18] with intent recognition as one of the downstream tasks. We only use datapoints with single label in our experiment.

3.2. Implementation Details: Pre-training and Fine-tuning

The PhonemeBERT model is jointly trained on ASR transcript augmented with phoneme sequence in batch size of 192 on $4 \times V100$ GPU for 50 epochs. Once we have pre-trained PhonemeBERT, we fine-tune it for the downstream tasks with task specific cross-entropy loss. The downstream models are fine-trained on the PhonemeBERT model for 20 epochs. The checkpoint with the best validation score is used to report the test results. For cases where a pre-defined validation set is not provided, we use 20% of the training set as the validation set.

3.3. Baselines and Ablations

We define various baseline systems and ablations to compare the performance against PhonemeBERT which is jointly trained on ASR transcript (w_{asr}) and corresponding generated phoneme sequence (p_{sp}):

- **Oracle**: This baseline sets the upper limit of the results obtained by fine-tuning RoBERTa by training and testing on the original (w_{clean}) corpus for downstream tasks.
- **B1**: In this baseline, we fine-tune RoBERTa base model directly on the downstream tasks with task specific ASR transcripts.
- **B2**: We further pre-train the RoBERTa base model on our ASR transcripts from the pre-training corpus with word-only MLM task [10]. We then fine-tune the model for each noisy-downstream task using word-only input.
- **B3**: In this setup, we pre-train a joint LM on word and phoneme sequence along with fine tuning it on downstream tasks with both information fed to the model. The phoneme sequence (p_{asr}) in this case is generated directly from the text using Phonemizer[§] tool.
- **A1**: This ablation model refers to using PhonemeBERT as word-only encoder in the low-resource setup where only ASR transcript is available for downstream tasks.
- **A2**: This setup is similar to A1 except that A2 uses only phoneme sequence for the downstream tasks.
- **A3**: In this setup, we train a PhonemeBERT-like model without a joint loss function i.e, with separate word-only and phoneme-only loss.

3.4. Results and Analysis

The comparative results of the proposed approach against various baselines are presented in Table 3. *PhonemeBERT* trained with joint modelling framework of ASR transcript and phoneme

[§] https://github.com/bootphon/phonemizer

Table 3: *Comparison of the proposed model, PhonemeBERT, with baselines and ablation setups.*

Baseline	Pre-Training (Feature, Joint Loss-T/F)	Task Fine-Tuning Feature	Results (Accuracy/Macro-F1) SST-5	TREC-6	TREC-50	ATIS
Oracle	No (-, F)	w_{clean}	56.60 / 54.03	87.20 / 97.64	91.80 / 86.35	99.37 / 94.78
B1	No (-, F)	w_{asr}	43.12 / 42.02	82.60 / 80.82	76.20 / 58.70	94.87 / 81.67
B2	Yes (w_{asr}, F)	w_{asr}	44.52 / 40.84	85.40 / 83.53	77.80 / 59.64	95.05 / 81.51
B3	Yes ($w_{asr} + p_{asr}$, T)	$w_{asr} + p_{asr}$	46.24 / 38.97	86.20 / 84.39	80.80 / **67.45**	96.75 / 80.36
PhonemeBERT	Yes ($w_{asr} + p_{sp}$, T)	$w_{asr} + p_{sp}$	**46.74 / 43.95**	**87.00 / 85.34**	**81.40** / 65.16	**97.25 / 84.15**
Ablation Study						
A1	Yes ($w_{asr} + p_{sp}$, T)	w_{asr}	46.70 / 41.54	85.80 / 83.85	79.60 / 62.77	95.62 / 79.80
A2	Yes ($w_{asr} + p_{sp}$, T)	p_{sp}	38.77 / 34.17	83.40 / 83.94	75.20 / 62.03	96.25 / 79.98
A3	Yes ($w_{asr} + p_{sp}$, F)	$w_{asr} + p_{sp}$	46.10 / 43.16	84.60 / 83.47	79.60 / 62.05	96.00 / 79.81

Table 4: *Accuracy comparison of the baseline (B1, B2) and proposed model (P: PhonemeBERT) at different WER range.*

WER	TREC-50			SST-5		
	B1	B2	P	B1	B2	P
10-20	80.00	**81.43**	80.00	46.94	**51.47**	50.94
20-30	81.82	82.82	**84.85**	43.62	43.62	**46.29**
30+	69.53	72.53	**77.52**	41.04	40.57	**43.50**
Overall	76.20	77.80	**81.40**	43.12	44.52	**46.74**

Table 5: *Results on Observe.AI's sentiment classification task*

Model	Precison	Recall	F1
B1	60.32	79.29	68.51
B2	65.06	76.54	70.33
A1	67.28	77.18	71.79
PhonemeBERT	**69.16**	**76.16**	**72.49**

sequences outperforms other models across all datasets. The proposed model comprehensively beats the model pre-trained on only ASR transcripts (*B2*) by upto 3.6% in TREC-50 on accuracy. This shows that a language model trained on noisy ASR transcript alone is not robust to ASR errors as compared to the one trained with both ASR transcripts and phoneme sequences. Additionally, PhonemeBERT also outperforms *B3* which justifies our hypothesis that a model trained on phoneme sequence generated directly over ASR transcripts yields sub-optimal results.

Furthermore, in a low-resource setup (*A1*) where phoneme is not available for the downstream task, the model performs better than a word-only model (*B2*) fine-tuned on the downstream task. This result justifies our hypothesis that joint modelling helps with learning a phonetic aware language model that is more powerful in noisy conditions. We also evaluate the setup using only phoneme sequence for downstream fine-tuning (*A2*). The results show a drop in evaluation scores for all but one dataset. We note a sharp decline in results for SST-5 and TREC-50 dataset compared to other results. This can be explained by the fact that both of these two tasks need understanding of semantics in the text than syntactic information. Phoneme sequences do not capture the semantic information, rather they are syntactic representation of the utterances and hence there is a loss of performance phoneme-only downstream setup. Finally, to study the impact of joint-modelling framework, we pre-train a model with independent word-MLM and phoneme-MLM tasks *A3*, where each task sees the context from respective sequence only. PhonemeBERT performs better than the model trained on independent MLM tasks with big gains in F1-score on TREC-6, TREC-50 and ATIS datasets.

In a separate analysis, we categorize the test set of *TREC-50* and *SST-5* into buckets based on the WER score of each instance and compare the performance of the trained models (Table 4) on each bucket. We note that for a lower WER ($10-20$), the baseline pre-trained on only ASR transcripts performs best. However, at higher WER ($>= 20$), the performance of the two models (*B1, B2*) decline steeply, while PhonemeBERT outperforms the word-only models and is less susceptible to the noise. We speculate that since pre-training data has an average WER greater than 20, PhonemeBERT learns to better model the data belonging to higher WERs while degrading a bit on lower WER ranges. Nevertheless, this reinforces the fact that we need a better modelling ability to address the robustness to ASR errors. For WER $>= 30$, PhonemeBERT outperforms the next-best model (*B2*) by 4.99% and 2.93% in accuracy for TREC-50 and SST-5 respectively. We additionally compare the proposed model on Observe.AI's sentiment classification dataset (Table 5). This dataset comprise of call center voice conversations with labels as negative or positive sentiment displayed by the customer on a call. The overall training and test size of the sample is 10492 and 3198 respectively with an average WER of 24.76. The results show that the joint LM setup outperforms word-only baseline *B2* by 2.16 points. PhonemeBERT when used in the low-resource setup (*A1*) outperforms word-only setup (*B2*) by 1.46 F1 score. This demonstrates the applicability of the proposed method in the practical setups too.

4. Conclusions and Future Work

In this work, we propose a joint modelling of phoneme sequence and ASR transcript to build a language model robust to ASR errors. We demonstrate that the proposed joint learning setup performs better than any strong baselines in downstream tasks. Through ablation study, we show that the proposed setup can also work in a low resource setup, still producing better results than model pre-trained with only ASR transcript. Our analysis suggests that the method is significantly better at higher WER ranges. In future, we would like to extend the work to tasks such as entity recognition and question answering for the noisy data. Additionally, we would also like to explore the possibility of using a larger corpus for the pre-training step.

5. References

[1] T. Desot, F. Portet, and M. Vacher, "SLU for voice command in smart home: Comparison of pipeline and end-to-end approaches," in *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore*. IEEE, 2019, pp. 822–829. [Online]. Available: https://doi.org/10.1109/ASRU46091.2019.9003891

[2] C. Liu, S. Zhu, Z. Zhao, R. Cao, L. Chen, and K. Yu, "Jointly encoding word confusion network and dialogue context with BERT for spoken language understanding," *CoRR*, vol. abs/2005.11640, 2020. [Online]. Available: https://arxiv.org/abs/2005.11640

[3] F. Ladhak, A. Gandhe, M. Dreyer, L. Mathias, A. Rastrow, and B. Hoffmeister, "Latticernn: Recurrent neural networks over lattices," in *Interspeech*. ISCA, 2016, pp. 695–699. [Online]. Available: https://doi.org/10.21437/Interspeech.2016-1583

[4] Y. Weng, S. S. Miryala, C. Khatri, R. Wang, H. Zheng, P. Molino, M. Namazifar, A. Papangelis, H. Williams, F. Bell, and G. Tür, "Joint contextual modeling for ASR correction and language understanding," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain*. IEEE, 2020, pp. 6349–6353. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9053213

[5] A. Mani, S. Palaskar, and S. Konam, "Towards understanding asr error correction for medical conversations," in *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*, 2020, pp. 7–11.

[6] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada*. IEEE, 2018, pp. 5754–5758. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8461785

[7] S. Ghannay, A. Caubrière, Y. Estève, N. Camelin, E. Simonnet, A. Laurent, and E. Morin, "End-to-end named entity and semantic concept extraction from speech," in *2018 IEEE Spoken Language Technology Workshop, SLT 2018, Athens, Greece*. IEEE, 2018, pp. 692–699. [Online]. Available: https://doi.org/10.1109/SLT.2018.8639513

[8] P. Yenigalla, A. Kumar, S. Tripathi, C. Singh, S. Kar, and J. Vepa, "Speech emotion recognition using spectrogram & phoneme embedding," in *Interspeech 2018*. ISCA, 2018, pp. 3688–3692. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1811

[9] A. Fang, S. Filice, N. Limsopatham, and O. Rokhlenko, "Using phoneme representations to build predictive models robust to ASR errors," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China*. ACM, 2020, pp. 699–708. [Online]. Available: https://doi.org/10.1145/3397271.3401050

[10] 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," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[11] A. Zeyer, W. Zhou, T. Ng, R. Schlüter, and H. Ney, "Investigations on phoneme-based end-to-end speech recognition," *CoRR*, vol. abs/2005.09336, 2020. [Online]. Available: https://arxiv.org/abs/2005.09336

[12] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Queensland, Australia*. IEEE, 2015, pp. 5206–5210. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7178964

[13] R. He and J. J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada*. ACM, 2016, pp. 507–517. [Online]. Available: https://doi.org/10.1145/2872427.2883037

[14] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100, 000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Texas, USA*, 2016, pp. 2383–2392. [Online]. Available: https://doi.org/10.18653/v1/d16-1264

[15] K. Irie, R. Prabhavalkar, A. Kannan, A. Bruguier, D. Rybach, and P. Nguyen, "On the choice of modeling unit for sequence-to-sequence speech recognition," in *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, G. Kubin and Z. Kacic, Eds. ISCA, 2019, pp. 3800–3804. [Online]. Available: https://doi.org/10.21437/Interspeech.2019-2277

[16] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP, Seattle, Washington, USA*. ACL, 2013, pp. 1631–1642. [Online]. Available: https://www.aclweb.org/anthology/D13-1170/

[17] X. Li and D. Roth, "Learning question classifiers," in *19th International Conference on Computational Linguistics, COLING 2002, Taipei, Taiwan*, 2002. [Online]. Available: https://www.aclweb.org/anthology/C02-1150/

[18] G. Tür, D. Hakkani-Tür, and L. P. Heck, "What is left to be understood in atis?" in *2010 IEEE Spoken Language Technology Workshop, SLT 2010, California, USA*. IEEE, 2010, pp. 19–24. [Online]. Available: https://doi.org/10.1109/SLT.2010.5700816