ARoBERT: An ASR Robust Pre-Trained Language Model for Spoken Language Understanding

Chengyu Wang, Suyang Dai, Yipeng Wang, Fei Yang, Minghui Qiu, Kehan Chen, Wei Zhou, and Jun Huang

***Abstract*—Spoken Language Understanding (SLU) aims to interpret the meanings of human speeches in order to support various human-machine interaction systems. A key technique for SLU is Automatic Speech Recognition (ASR), which transcribes speech signals into text contents. As the output texts of modern ASR systems unavoidably contain errors, mainstream SLU models either trained or tested on texts transcribed by ASR systems would not be sufficiently error robust. We present ARoBERT, an ASR Robust BERT model, which can be fine-tuned to solve a variety of SLU tasks with noisy inputs. To guarantee the robustness of ARoBERT, during pretraining, we decrease the fluctuations of language representations when some parts of the input texts are replaced by homophones or synophones. Specifically, we propose two novel self-supervised pre-training tasks for ARoBERT, namely Phonetically-aware Masked Language Modeling (PMLM) and ASR Model-adaptive Masked Language Modeling (AMMLM). The PMLM task explicitly fuses the knowledge of word phonetic similarities into the pre-training process, which forces homophones and synophones to share similar representations. In AMMLM, a data-driven algorithm is further introduced to mine typical ASR errors such that ARoBERT can tolerate ASR model errors. In the experiments, we evaluate ARoBERT over multiple datasets. The results show the superiority of ARoBERT, which consistently outperforms strong baselines. We have also shown that ARoBERT outperforms state-of-the-arts on a public benchmark. Currently, ARoBERT has been deployed in an online production system with significant improvements.**

***Index Terms*—ASR robust representation learning, pre-trained language model, spoken language understanding.**

I. Introduction

SPOKEN Language Understanding (SLU) is the task of interpreting and understanding the meanings of human speeches. Typical SLU tasks include slot filling [1], user intent classification [2] and speech event detection [3]. To solve SLU tasks, a variety of methods apply Automatic Speech Recognition (ASR) to transcribe human speeches into texts before training downstream SLU models. By combining ASR systems and SLU models, the meanings of speech signals and textual contents can be extracted and utilized in applications with rich human-machine interactions.

Due to the importance of speech-to-text conversion, the ASR task has been extensively addressed in both research and industrial communities, mostly by means of designing deep speech-to-text neural networks [4], [5]. Notable neural network architectures for ASR systems include Deep Speech 2 [6], the Speech Transformer [7], the wav2vec self-supervised systems [8] and many others. Despite the success, the transcripts generated by ASR systems unavoidably contain errors, such as the substitution of words by their homophones or synophones. The errors produced by ASR systems can easily propagate to the downstream SLU models [9], [10]. In the literature, several lines of research have been proposed to address the problem of the low robustness of SLU models. (i) ASR error detection methods [11]–[13] recognize ASR errors in transcribed texts. (2) Several approaches [14]–[16] modify the encoder-decoder architectures in ASR systems to improve the language correctness and fluency when the systems decode the latent representations into discrete text outputs. When the two types of approaches are applied to SLU tasks, users must modify existing ASR systems, resulting in additional technical burdens. It would be more desirable if there is *an end-to-end SLU model that generates robust predictions from ASR-transcribed texts directly*.

Recently, the emergence of large-scale Pre-trained Language Models (PLMs) has significantly improved the performance of various language understanding tasks [17], including BERT [18], ALBERT [19], GPT-3 [20] and many others. It is thus straightforward to adopt PLMs for building an end-to-end SLU model. The learning process involves a PLM pre-training stage and a PLM fine-tuning stage for downstream SLU tasks. However, if the representations learned during PLM pre-training are not robust to ASR errors, the performance of PLM fine-tuning would suffer to a large extent. There exist a few studies on learning domain-invariant representations, more robust to ASR errors [21]–[23]. Yet none of the prior studies consider ASR error robustness in pre-training language models for boosting the performance of a variety of SLU tasks.

To bridge the gap, we present *ARoBERT* (short for ASR Robust Bidirectional Encoder Representation from Transformers)

Manuscript received July 21, 2021; revised December 12, 2021 and February 1, 2022; accepted February 3, 2022. Date of publication February 24, 2022; date of current version March 30, 2022. This work was supported in part by the Open Research Projects of Zhejiang Lab under Grant 2019KD0AD01/004 and in part by the Natural Science Foundation of Zhejiang Province under Grant LQ21F020004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhijian Ou. *(Corresponding author: Minghui Qiu.)*

Chengyu Wang is with Zhejiang Lab, Hangzhou, Zhejiang 311121, China, and also with Alibaba Group, Hangzhou, Zhejiang 311121, China (e-mail: chywang2013@gmail.com).

Suyang Dai, Yipeng Wang, Minghui Qiu, Kehan Chen, Wei Zhou, and Jun Huang are with Alibaba Group, Hangzhou, Zhejiang 311121, China (e-mail: suyang.dsy@alibaba-inc.com; wyp209764@alibaba-inc.com; minghuiqiu@gmail.com; kehan.ckh@alibaba-inc.com; fayi.zw@alibaba-inc.com; huangjun.hj@alibaba-inc.com).

Fei Yang is with the Zhejiang Lab, Hangzhou, Zhejiang 311121, China (e-mail: yangf@zhejianglab.com).

Digital Object Identifier 10.1109/TASLP.2022.3153268

in this paper. Since the majority of ASR errors are on word homophones or synophones, we design ARoBERT *to decrease the fluctuations of language representations when part of the input texts are replaced by their homophones or synophones*. In ARoBERT, inspired by BERT [18], a stack of transformer encoders [24] are employed to learn input token representations. Then, the transformer encoders in ARoBERT should tolerate the ASR errors that are phonetically similar to those without errors. Apart from the classical Masked Language Modeling (MLM) task [18], we further propose two new self-supervised tasks for pre-training ARoBERT, namely Phonetically-aware Masked Language Modeling (PMLM) and ASR Model-adaptive Masked Language Modeling (AMMLM), briefly summarized as follows:

- *PMLM:* This task is a significant extension to MLM such that the model suffers from a smaller loss when it incorrectly predicts the masked tokens to be the homophones or synophones of the actual words. Hence, homophones and synophones share similar representations. In PMLM, heuristic-based phonetic similarities are injected into the loss function as *knowledge priors*.
- *AMMLM:* As different ASR systems may have diverse types of errors, simple phonetic heuristics may have low coverage of ASR errors. We further extend PMLM such that the loss function can be adaptive to particular ASR models. Specifically, we introduce a data-driven algorithm to extract ASR errors as the *seed error set*. The errors are then generalized and fused into a novel loss function AMMLM. In this way, the pre-trained ARoBERT model can fit specific errors created by particular ASR systems that can not be captured by heuristics used in PMLM.

As for fine-tuning, ARoBERT utilizes the same training paradigm as that of BERT [18], such that it can be directly applied to various PLM-based approaches for SLU tasks without any modifications. In the experiments, we evaluate the effectiveness of the ARoBERT model over the benchmark of the Chinese Audio-Textual SLU Challenge (CATSLU)[1] and two real-world, labeled datasets in industrial applications. The results show the superiority of ARoBERT over strong baselines, and confirm ARoBERT's capacity of preserving ASR error robustness in language representation learning. We have also deployed ARoBERT in an online production system and observed significant improvements, compared to existing online systems.

In summary, we make the following major contributions in this paper:[2]

- We formally introduce the ARoBERT model for solving various SLU tasks. To the best of our knowledge, our work is the first to incorporate ASR error robustness into the pre-training process of deep neural language models.
- We propose two novel self-supervised learning tasks for pre-training ARoBERT, namely PMLM and AMMLM. The former incorporates heuristic-based phonetic similarities, while the latter considers a data-driven algorithm to be adaptive to ASR errors.
- Extensive experiments over multiple datasets and different types of downstream SLU tasks prove the effectiveness of ARoBERT, outperforming strong baselines.

The rest of this paper is organized as follows. Section II summarizes the related work. Details of ARoBERT and experimental results are presented in Sections III and IV. We discuss the limitations and extensions of our work in Section V. Finally, we draw the conclusion and discuss the future work in Section VI.

II. Related Work

In this section, we briefly summarize the related work from the literature in the following two aspects: PLMs and SLU tasks. Specifically, we focus on how to learn ASR robust representations to improve the performance of SLU models.

A. Pre-Trained Language Models

In recently years, the rapid development of large-scale PLMs has boosted the research of Natural Language Processing (NLP) [17]. After the PLMs are pre-trained over massive corpora, only a simple fine-tuning process is required to produce models for downstream NLP tasks. ELMo [25] is one of the early works that learns deep contextual representations. BERT [18] is probably the most influential PLM that encodes words by a stack of transformer encoders. In BERT, two self-supervised objectives are employed for pre-training, i.e., MLM and next sentence prediction. RoBERTa [26] improves the pre-training process of BERT by various optimization techniques. ALBERT [19] reduces the sizes of BERT-style models by parameter sharing and factorization. StructBERT [27] considers the syntactic structures of languages for pre-training. A limitation of these PLMs is that they utilize vanilla transformer encoders that have fixed-length contexts. For longer sequences, Transformer-XL [28] learns the long-term contextual dependencies by recurrence and relative positional encoding. Apart from using transformer encoders only, the encoder-decoder architecture has also been employed in PLMs, such as T5 [29]. The GPT-3 model [20] uses the transformer decoder architecture to generate texts for various texts.

Despite the success, we notice that existing PLMs learn token representations based on clean texts with high accuracy. To the best of our knowledge, the proposed ARoBERT model is the first to incorporate ASR error robustness into pre-training, in order to improve the performance of downstream SLU tasks.

B. Spoken Language Understanding

SLU is the process of understanding the semantic meanings of utterances in human speeches. Typical SLU tasks include slot filling [1], user intent classification [2], speech event detection [3] and many others. After human speeches are transcribed into texts by ASR systems, NLP models (especially PLMs) can be used to solve these SLU problems by fine-tuning.

[1][Online]. Available: https://sites.google.com/view/catslu/home/

[2]The source codes, the pre-trained ARoBERT model and the sampled datasets will be publicly available at: https://github.com/alibaba/EasyTransfer/tree/master/scripts/arobert.

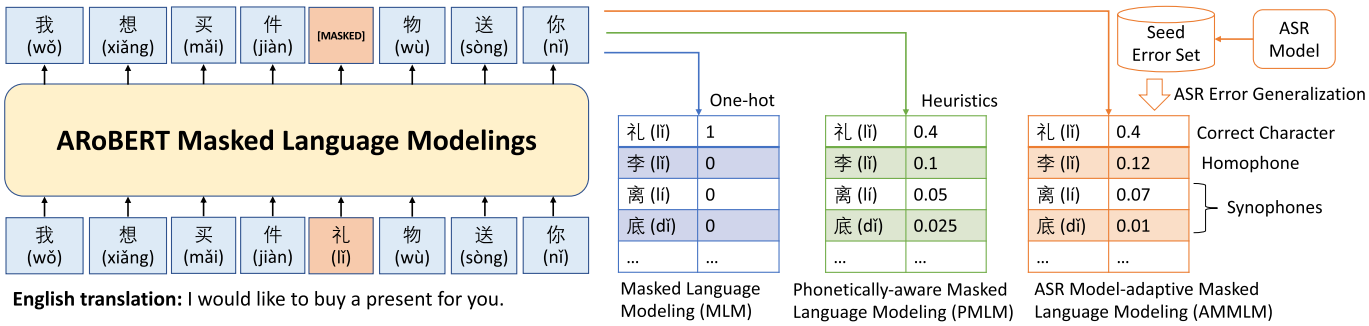


Fig. 1. Illustration of the pre-training tasks in ARoBERT. In the figure, each Chinese character is accompanied by its Chinese phonetic symbols (named *pinyin*) to show its pronunciation. (Best viewed in color.).

Different from standard NLP tasks, a unique challenge for SLU is that, some input texts of SLU contain errors generated by ASR systems. If the underlying models are trained with clean texts, they may perform poorly when the inputs for prediction contain ASR errors. Similarly, models trained using noisy texts may suffer from performance degradation over a dataset without ASR errors [22]. In order to improve the robustness of such models, several types of methods have been proposed. As texts with and without ASR errors for the same task can be viewed as different "domains," several transfer learning approaches have been proposed. For example, Wang *et al.* [30] employ transfer learning to learn domain-invariant features between the two types of texts. Tan *et al.* [31] use the minimum edit distances between ASR results and correct candidates to rebuild the tuples affected by ASR errors. The work [10] utilizes adversarial learning to train the SLU model encoders for ASR error adaptation. Ruan *et al.* [22] add an additional loss in SLU models to minimize the distribution differences between prediction outputs of correct and incorrect texts. One potential disadvantage is that it requires the labeled, training data with both inputs. Yet a few works focus on end-to-end SLU, which perform SLU tasks directly from audio inputs and are less vulnerable to ASR errors [32]–[35].

Another stream of research aims to learn word embeddings considering phonetic information for SLU tasks. SpellGCN [36] integrates phonological and visual similarities of Chinese characters into language models. Phoneme2vec [37] learns the similarities between phonemes that benefit downstream SLU tasks, without considering the semantic similarities between words. Confusion2vec [38] uses both semantic similarities and phonetic ambiguities to learn word embeddings. The confusion information is leveraged in [23], which fine-tunes the ELMo [25] model to make it more robust to ASR errors. Different from existing works, ARoBERT is one of the first attempts to integrating ASR ambiguities into the pre-training stages of PLMs, which makes it capable of addressing a variety of downstream SLU tasks.

III. ARoBERT: The Proposed Model

In this section, we formally present ARoBERT in detail. We begin with an overview of the three pre-training tasks used in ARoBERT. After that, the detailed techniques of ARoBERT are elaborated.

A. An Overview of ARoBERT

ARoBERT shares the same transformer encoder architecture as that of BERT [18] to learn token representations. It differentiates itself from BERT-style models in that it incorporates rich phonetic knowledge during pre-training. Specifically, the transformer encoders should tolerate the ASR errors that are phonetically similar to correct transcripts without errors.

In Fig. 1, we give an illustrative example on three pre-training tasks of ARoBERT, namely Masked Language Modeling (MLM), Phonetically-aware MLM (PMLM), and ASR Model-adaptive MLM (AMMLM). Denote the loss functions of the three tasks as $\mathcal{L}_{MLM}$, $\mathcal{L}_{PMLM}$ and $\mathcal{L}_{AMMLM}$, respectively. The overall loss function of ARoBERT is defined as follows:

$$\mathcal{L} = \mathcal{L}_{MLM} + \lambda_1 \mathcal{L}_{PMLM} + \lambda_2 \mathcal{L}_{AMMLM}, \tag{1}$$

where λ_1 and λ_2 are balancing hyper-parameters. In the following, we describe the three tasks in detail.

B. Pre-Training Tasks in ARoBERT

1) Masked Language Modeling: As Liu *et al.* [26] show, MLM is more effective for BERT pre-training, compared with next sentence prediction. Hence, in ARoBERT, we employ MLM as a basic pre-training task. Before we introduce PMLM and AMMLM, it is necessary to take a closer look at the mechanisms of MLM. Let the underlying PLM be parameterized by θ. The vocabulary set is denoted as V. Assume an arbitrary token (with m as the index in the vocabulary V) is masked for model prediction. The *token-wise* MLM loss $\mathcal{L}_{MLM}(m)$ is defined as:

$$\mathcal{L}_{MLM}(m) = -\sum_{i=1}^{|V|} y_{i,m} \cdot \log \Pr(i, m|\theta), \tag{2}$$

which is the cross-entropy between an one-hot vector $\vec{y}_m$ (with the m-th element to be 1 and otherwise 0) and the model's prediction probability distribution. $y_{i,m}$ is the i-th element of $\vec{y}_m$ and $\Pr(i, m|\theta)$ is the probability of m being the i-th token in the vocabulary V, predicted by the PLM parameterized by θ.

The overall loss function $\mathcal{L}_{MLM}$ is the sum of the losses of all masked tokens in the corpus.

2) Phonetically-Aware Masked Language Modeling: As described, MLM is unable to encode the phonetic knowledge of words. In ARoBERT, we further define the *token-wise* PMLM loss $\mathcal{L}_{PMLM}(m)$:

$$\mathcal{L}_{PMLM}(m) = -\sum_{i=1}^{|V|} y'_{i,m} \cdot \log \Pr(i, m|\theta), \tag{3}$$

where $y'_{i,m}$ integrates the phonetic knowledge that requires the model to approximate. Denote the phonetic similarity between the i-th and m-th tokens in the the vocabulary V as $sim(i, m)$. A naive approach to define $y'_{i,m}$ is by setting $y'_{i,m} \propto sim(i, m)$. However, it ignores the semantic relations between the two words. Additionally, this practice significantly enlarges the size of the pre-training data, hence increasing the computational complexity. To specify, during pre-training, $|V|$ numeric values (i.e., $y'_{i,m}$ for $i = 1, \ldots, |V|$) are required to feed to the model for each masked token. In ARoBERT, for each masked token m, we retrieve the top-k most phonetically similar tokens. Denote the collection of indices of these tokens as C_m. The value $y'_{i,m}$ is then defined as follows:

$$y'_{i,m} = \begin{cases} \mathcal{M} & m = i \\ \frac{(1-\mathcal{M}) \cdot sim(i,m)}{\sum_{j \in C_m} sim(j,m)} & i \in C_m \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{M}$ is a pre-defined constant $(0 < \mathcal{M} < 1)$. Thus $\mathcal{L}_{PMLM}(m)$ can be re-written by the following formula:

$$\begin{aligned} \mathcal{L}_{PMLM}(m) = & -\mathcal{M} \cdot \log \Pr(m, m|\theta) \\ & - \sum_{i \in C_m} y'_{i,m} \cdot \log \Pr(i, m|\theta). \end{aligned} \tag{5}$$

Compared to $\mathcal{L}_{MLM}(m)$, $\mathcal{L}_{PMLM}(m)$ gives a higher tolerance of incorrectly predicting a token to be its homophones or synophones, with degrees linearly proportional to the phonetic similarity between the two words. Hence, the representations learned by ARoBERT is less sensitive to ASR errors. Similar to MLM, the loss function $\mathcal{L}_{PMLM}$ is the sum of the losses $(\mathcal{L}_{PMLM}(m))$ of all masked tokens.

A remaining problem is how to compute the phonetic similarity $sim(i, m)$ properly. In Mandarin, the pronunciation of a Chinese character can be represented by the phonetic symbols (named *pinyin*).[3]. Unlike English where vowels and consonants form the pronunciation of words, Chinese phonetic symbols mostly consist of three components: *initials*, *finals* and *tones*. A simple example is shown in Fig. 2. We compute the phonetic similarity $sim(i, m)$ as follows:

$$\begin{aligned} sim(i,m) = & \alpha_1 \cdot \mathbf{1}\left(initial(i) = initial(m)\right) \\ & + \alpha_2 \cdot \mathbf{1}\left(final(i) = final(m)\right) \\ & + (1-\alpha_1-\alpha_2) \cdot \mathbf{1}\left(tone(i) = tone(m)\right), \end{aligned} \tag{6}$$

[3][Online]. Available: https://en.wikipedia.org/wiki/Pinyin

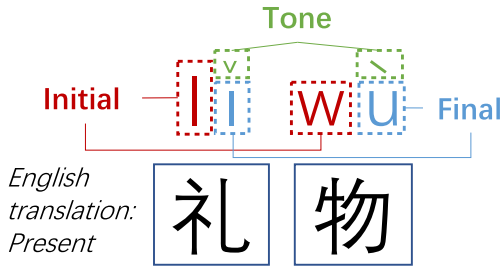


Fig. 2. Examples of the three components in Chinese phonetic symbols (*pinyin*). (Best viewed in color.).

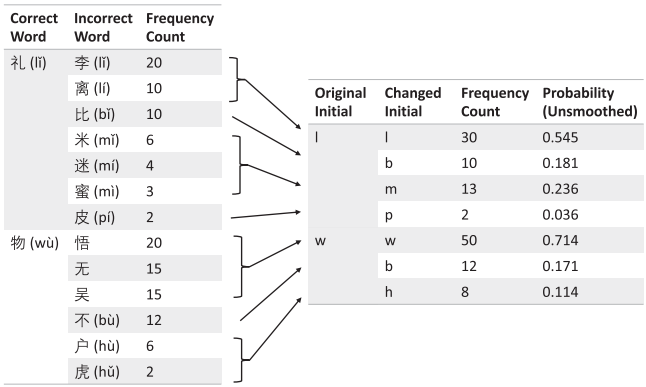
where $0 < \alpha_1 < 1$, $0 < \alpha_2 < 1$ and $0 < \alpha_1 + \alpha_2 < 1$. $\mathbf{1}(\cdot)$ is the indicator function that returns 1 if the input Boolean expression is true and otherwise 0. $initial(\cdot)$, $final(\cdot)$ and $tone(\cdot)$ represent the respective phonetic components of the underlying Chinese characters. We find that the above approach is heuristic-based, serving as the *knowledge prior* for pre-training ARoBERT. Implementation details will be described in the experiments.

3) ASR Model-Adaptive Masked Language Modeling: The heuristic-based PMLM task has a relatively strong assumption that phonetic similarities directly relate to ASR errors. However, it is not always the case in real-world applications. The AMMLM pre-training task is complementary to PMLM, which aims to learn robust representations that can fit errors generated by ASR models.

Formally, let S be the sequence of texts that an ASR model generates from human speeches, and S' be the corresponding texts that have been corrected by human annotators. Based on the alignments between S and S', inspired by [23], we generate a *seed error set* $\mathcal{W} = \{(m, m')\}$ such that the m-th token in the vocabulary V can be incorrectly substituted with the m'-th token by the underlying ASR model. One disadvantage of the construction of $\mathcal{W}$ is that it requires the tedious work of human correction of transcribed texts. To minimize the amount of human labor and make the discovered errors more generalized to unseen texts, we further propose an *ASR error expansion* algorithm, described below.

As seen, the pronunciations of Chinese characters are largely determined by their *initials* and *finals*. Hence, it is necessary to discover how *initials* and *finals* can be replaced when ASR errors occur. We do not consider the substitution of *tones* here for two reasons. i) In PMLM, when α_1 and α_2 are relatively large for the computation of the heuristic-based phonetic similarity, two Chinese characters already have high similarity if they share the same initials and finals. Hence, the knowledge of tone similarity is largely captured by PMLM. ii) The inclusion of tones for ASR error expansion may largely expand the parameter space of the probabilistic distributions, making them less generalized to unseen cases. Denote $\mathcal{I}$ and $\mathcal{F}$ as the collections of all initials and finals, repectively. For any two initials $p, q \in \mathcal{I}$, we compute the *initial substitution probability* by the following formula:

$$\Pr(q|p) = \frac{\#(p, q) + \epsilon}{\sum_{\tilde{p} \in \mathcal{I}} \#(\tilde{p}, q) + |\mathcal{I}| \cdot \epsilon}, \tag{7}$$



Correct Word	Incorrect Word	Frequency Count
礼 (lǐ)	李 (lǐ)	20
	离 (lí)	10
	比 (bǐ)	10
	米 (mǐ)	6
	迷 (mí)	4
	蜜 (mì)	3
	皮 (pí)	2
物 (wù)	悟	20
	无	15
	吴	15
	不 (bù)	12
	户 (hù)	6
	虎 (hǔ)	2

Original Initial	Changed Initial	Frequency Count	Probability (Unsmoothed)
l	l	30	0.545
	b	10	0.181
	m	13	0.236
	p	2	0.036
w	w	50	0.714
	b	12	0.171
	h	8	0.114

Fig. 3. Toy example of the generation process of the initial substitution probabilistic distribution.

where $\#(p, q)$ is the frequency count where a word m's initial p is substituted by the initial q of another word m' in $\mathcal{W}$, and ϵ is the pre-defined smoothing factor (we set $\epsilon = 1e-3$ in default). Toy examples of the computation of initial substitution probabilities can be found in Fig. 3.

Similarly, for two finals $r, s \in \mathcal{F}$, we also have the *final substitution probability*, defined as follows:

$$\Pr(s|r) = \frac{\#(r,s) + \epsilon}{\sum_{\tilde{r}\in\mathcal{F}} \#(\tilde{r}, s) + |\mathcal{F}| \cdot \epsilon}, \tag{8}$$

where $\#(r, s)$ is the frequency count where a word m's final r is substituted by the final s of another word m' in $\mathcal{W}$.

Based on the two probabilistic distributions, it is straightforward to expand the *seed error set* for the computation of the AMMLM loss. We define the *substitution score* from the m-th token to the i-th token (denoted as $subs(i, m)$) as follows:

$$subs(i,m) = \begin{cases} \Pr(final(i)|final(m)) & initial(i) = initial(m) \\ \Pr(initial(i)|initial(m)) & final(i) = final(m) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

We does not consider the situation where both the initial and the final are different as such cases are very rare in our Mandarin ASR system. Let $\mathcal{L}_{AMMLM}(m)$ be the *token-wise* AMMLM loss with:

$$\mathcal{L}_{APMLM}(m) = -\sum_{i=1}^{|V|} y''_{i,m} \cdot \log \Pr(i, m|\theta), \tag{10}$$

where $y''_{i,m}$ fuses the mined and generalized knowledge from the *seed error set* made by an ASR model. Similar to PMLM, we also consider the top-k tokens with the highest substitution scores when we compute $y''_{i,m}$:

$$y''_{i,m} = \begin{cases} \mathcal{M} & m = i \\ \frac{(1-\mathcal{M})\cdot subs(i,m)}{\sum_{j\in\tilde{C}_m} subs(j,m)} & i \in \tilde{C}_m \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $\tilde{C}_m$ is the collection of tokens with the top-k substitution scores w.r.t. the m-th token. It is trivial to see that:

$$\mathcal{L}_{AMMLM}(m) = -\mathcal{M} \cdot \log \Pr(m, m|\theta) - \sum_{i\in\tilde{C}_m} y''_{i,m} \cdot \log \Pr(i, m|\theta). \tag{12}$$

Similar to PMLM, the loss function of the AMMLM task is the sum of the token-wise losses ($\mathcal{L}_{APMLM}(m)$) of all the masked tokens in the pre-training corpus. In the implementation, for all the Chinese characters in the ARoBERT vocabulary V, we have computed all the scores $y''_{i,m}$ and $y'_{i,m}$ w.r.t. the PMLM and AMMLM tasks before model pre-training. The total number of scores ($y''_{i,m}$ and $y'_{i,m}$) is $2k \cdot |V|$. Hence, during the pre-training process, the optimization algorithm only needs to access the corresponding values in the memory, which makes the pre-training process of ARoBERT highly efficient.

4) Updating ARoBERT Through Time: We further analyze how ARoBERT should be updated when the underlying ASR systems change. In ARoBERT, we have three parts in the loss function, namely $\mathcal{L}_{MLM}$, $\mathcal{L}_{PMLM}$ and $\mathcal{L}_{AMMLM}$. We can see that the optimization of $\mathcal{L}_{MLM}$ and $\mathcal{L}_{PMLM}$ is not ASR model-specific. Hence, the change of ASR systems does not affect the values of $\mathcal{L}_{MLM}$ and $\mathcal{L}_{PMLM}$. In contrast, $\mathcal{L}_{AMMLM}$ is related to specific ASR systems.

C. Fine-Tuning ARoBERT for SLU Tasks

As ARoBERT employs transformer encoders to learn ASR robust representations, it is easy to fine-tune ARoBERT to solve various SLU tasks. For example, one can follow the same procedure of BERT fine-tuning [18] when ARoBERT is applied for user intent classification. For a few complicated tasks such as slot filling, post-processing steps are required to generate the complete results. Readers can refer to the experiments for the implementation details for slot filling over the CATSLU Challenge. We do not further elaborate.

IV. Experiments

In this section, we conduct extensive experiments to evaluate the proposed ARoBERT over multiple datasets and SLU tasks. We compare ARoBERT with strong baselines to make the convincing conclusion. We also deploy ARoBERT in an online production system to demonstrate its usefulness in real-world applications.

A. Datasets and Experimental Settings

1) Datasets: To our knowledge, the only publicly available SLU dataset containing texts with ASR errors in the Mandarin language is provided by the CATSLU Challenge [39], which are slot filling tasks divided into four domains: map, music, weather and video. For each domain, texts containing ASR errors, together with their correct slot values have been divided into training, development and testing sets by the organizers. The dialogue-related statistics of CATSLU datasets are also shown in Table II.

TABLE I
STATISTICS OF DATASETS USED FOR EVALUATION

Dataset	#Train	#Dev.	#Test	#Test (ASR Error)
ARoMatch	36,292	9,294	6,691	979
ARoTopic	7,487	2,495	2,497	278
CATSLU (Map)	5,093	921	1,578	-
CATSLU (Music)	2,189	381	676	-
CATSLU (Weather)	341	378	2,660	-
CATSLU (Video)	205	195	1,641	-

"#Test" for *ARoMatch* and *ARoTopic* refers to "#Test (Base)".

TABLE II
DIALOGUE-RELATED STATISTICS OF CATSLU DATASETS

Dataset	#Users	#Utterances	#Slots
Map	1,788	7,592	24
Music	268	3,246	20
Weather	276	3,379	22
Video	227	2,041	28

To fully evaluate the effectiveness of ARoBERT, we also construct two new datasets from real-world industrial applications. The first dataset *ARoMatch* is generated from a hotline service from a popular e-commerce platform in China (i.e., Alibaba). Each instance is in the form of "user query, product name, class label" triples, where a user queries about products on the platform via hotline. The task is to predict whether a user query that is transcribed by an ASR system matches the product. This is modeled as a binary classification task in our work. The second dataset *ARoTopic* is a fine-grained topic classification task generated from the same platform, which aims to predict the fine-grained topic of a commercial audio (which has also been transcribed by the ASR system) in the fashion domain. In *ARoTopic*, there exist 11 different fine-grained class labels, such as *Promotions*, *Style*, *Material*, *Color & Patterns*, etc.

The dataset settings of *ARoMatch* and *ARoTopic* are significantly different (and possibly more challenging) from those in the CATSLU Challenge. The training and development sets of *ARoMatch* and *ARoTopic* are manually constructed to have relatively low ASR error rate of 10% approximately. To test the model performance on relatively clean texts and texts with high ASR error rates. The testing sets of *ARoMatch* and *ARoTopic* are split into two parts: "Test (Base)" and "Test (ASR Error)". "Test (Base)" has similar ASR error rates as the training and development sets. "Test (ASR Error)" is selected by crowd-sourced workers such that every transcript has ASR errors which might harm the model performance. Each instance is manually labeled by crowd-sourced workers with original audios and transcribed texts provided. Readers can refer to the statistics of all the datasets in Table I.

2) Experimental Settings: In the implementation, we pre-train two ARoBERT models in different sizes: i) the tiny version (2 layers, with the dimension size 128) and ii) the base version (12 layers, with the dimension size 768). The former is used in *ARoMatch* and *ARoTopic* to ensure fast inference speed for online applications, while the latter is used in the CATSLU challenge. The underlying ASR system is a commercial system.[4] The vocabulary and the pre-training settings are the same as that of the Chinese version of BERT [18].[5] For fine-tuning on *ARoMatch* and *ARoTopic*, we use Adam as the optimizer [40]. The hyper-parameter settings for model pre-training are: $\lambda_1 = \lambda_2 = 0.5$, $\mathcal{M} = 0.4$, $k = 30$ and $\alpha_1 = \alpha_2 = 0.4$. We also tune the models (with different learning rates and epochs) and report the results in the experiments. During model fine-tuning, we use the Adam optimizer with the batch size to be 32. The learning rate and the number of epochs are tuned over the development sets, with detailed results reported in the experiments.

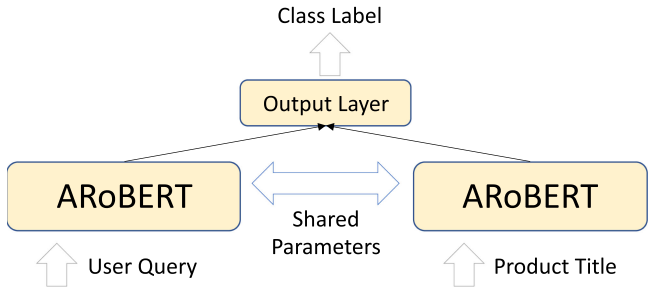


Fig. 4. Two-tower version of ARoBERT for query-title matching.

All the algorithms are implemented with TensorFlow based on the EasyTransfer platform [41] and trained with NVIDIA Tesla V100 GPUs. For evaluation, we use accuracy and F1 for *ARoMatch* and the CATSLU Challenge as the metrics. As *ARoTopic* involves multiple classes, we report accuracy and weighted F1 (which is weighted by the support of different classes) as the evaluation metrics.

B. General Performance Comparison

In this section, we report the general performance of ARoBERT and baselines over *ARoMatch* and *ARoTopic*. Because *ARoMatch* is generally a text matching task, we treat DAM [43], HCNN [44], SpokenVec [23] and BERT [18] as strong baselines. Specifically, DAM [43] is a decomposable attention model that learns the relations of textual inputs by attention mechanisms. HCNN [44] employs both representation and interaction learning for the text pairs. SpokenVec [23] learns ASR-robust contextualized word embeddings for SLU. Apart from the conventional fine-tuning approach for BERT [18] and ARoBERT, we also implement the two-tower versions of BERT and ARoBERT that uses two BERT/ARoBERT models with shared parameters to learn representations of the query and the product name separately, and the circle loss as the matching loss function. A simple example of the two-tower version of ARoBERT is shown in Fig. 4. As for *ARoTopic*, we employ the following baselines: TextCNN [45], TextRCNN [46] DGCNN [47], SpokenVec [23] and BERT [18]. Specially, baselines: TextCNN [45] employs CNN blocks with different sizes for sentence classification. TextRCNN [46] integrates CNN and RNN blocks as text encoders. DGCNN [47] is a strong

[4][Online]. Available: https://www.alibabacloud.com/product/intelligent-speech-interaction

[5][Online]. Available: https://github.com/google-research/bert

TABLE III
TESTING RESULTS OF AROBERT OVER THE CATSLU CHALLENGE (%)

Method	Map		Music		Weather		Video		Average	
	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1	Acc.
Official Baselines										
Rule Baseline [39]	37.92	40.43	77.39	49.26	85.52	75.38	78.25	45.28	69.77	52.58
Neural Baseline [39]	77.61	74.65	81.57	71.15	85.25	78.16	75.18	57.53	77.90	70.37
Strong Baselines for Slot Filling										
Transfer Learning [30]	84.27	79.48	88.10	78.74	83.9	70.77	87.22	82.54	85.9	77.88
Multi-classification BERT [31]	87.43	83.08	91.53	82.40	92.24	86.95	91.71	81.17	90.72	83.40
*BiLSTM + Acoustic/Domain Knowledge [42]**	*89.00*	*84.54*	*93.42*	*86.69*	*93.70*	*88.80*	*92.84*	*84.28*	*92.24*	*86.07*
Top Systems in CATSLU										
Top-F1 System (Leaderboard w/o. audio)	88.07	83.84	92.84	84.91	94.16	88.80	93.04	83.91	92.03	85.37
*Top-F1 System (Leaderboard w. audio)**	*89.28*	*84.47*	*95.53*	*86.09*	*93.88*	*89.02*	*93.04*	*83.91*	*92.86*	*85.78*
ARoBERT (Our approach)	**89.56**	**86.19**	**93.14**	**85.21**	**94.46**	**89.70**	**93.08**	**84.10**	**92.56**	**86.30**

Methods with * use additional audio resources as features and hence are not directly comparable with us.

TABLE IV
EXPERIMENTAL RESULTS OVER TESTING SETS OF *AROMATCH*

Method	Test (Base)		Test (ASR Error)	
	Acc.	F1	Acc.	F1
DAM [43]	0.799	0.680	0.556	0.427
HCNN [44]	0.842	0.732	0.548	0.326
BERT [18]	0.882	0.807	0.589	0.328
BERT [18] (Two-tower)	0.809	0.721	0.548	0.287
SpokenVec [23]	0.876	0.812	0.591	0.332
ARoBERT	**0.920**	**0.872**	**0.620**	**0.428**
ARoBERT (Two-tower)	0.832	0.755	0.572	0.386

TABLE V
EXPERIMENTAL RESULTS OVER TESTING SETS OF *AROTOPIC*

Method	Test (Base)		Test (ASR Error)	
	Acc.	Wgt. F1	Acc.	Wgt. F1
TextCNN [45]	0.644	0.637	0.486	0.474
TextRCNN [46]	0.653	0.645	0.507	0.498
DGCNN [47]	0.650	0.641	0.475	0.470
BERT [18]	0.666	0.660	0.482	0.481
SpokenVec [23]	0.658	0.654	0.492	0.502
ARoBERT	**0.671**	**0.668**	**0.532**	**0.532**

Gated-CNN model that achieves similar performance to BERT in several datasets with fast inference speed. We have also considered several adversarial learning approaches for learning ASR robust models (such as [10], [22]) as baselines. However, these methods require the labeled datasets of both clean texts and texts with ASR errors. Hence, these methods can not be directly applied to *ARoMatch* and *ARoTopic*. We tune ARoBERT and all the baselines over the development sets and report the performance on two types of testing sets: "Test (Base)" (with few ASR errors) and "Test (ASR Error)" (with high ASR error rates). The results are shown in Table IV and Table V.

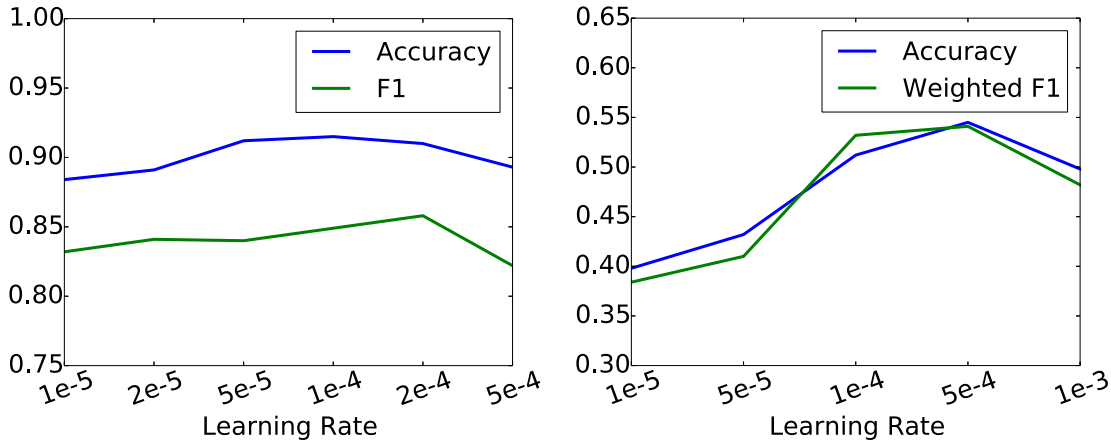
From the experimental results over "Test (Base)," we find that two PLMs (ARoBERT and BERT) without the two-tower settings consistently outperform all the other methods over the two SLU tasks. On *ARoMatch*, the F1 score of ARoBERT is higher than BERT by 6.5%, while on *ARoTopic*, the improvement is 0.8%. It shows that the ASR robust pre-training technique of ARoBERT is more important for text matching. As for the most challenging cases on "Test (ASR Error)," the performance of all models drops by a large margin, indicating there is still room for improvement on ASR error robustness. Compared to BERT, the improvements of ARoBERT in terms of F1 scores are 10.0% and 5.1%, much higher than those over "Test (Base)". Therefore, ARoBERT has a higher robustness level for ASR errors than BERT, together with other baselines.

To better understand the gap of between human performance and ARoBERT, we further ask crowd-sourced workers to label the "Test (ASR Error)" dataset of *ARoMatch* based on ASR transcripts only. The results are determined by majority vote. Overall, the human performance is 64.96% in terms of accuracy and 45.64% in terms of F1, which shows that this task is highly challenging, and ARoBERT is capable of generating near-human performance.

C. Results of the CATSLU Challenge

To our knowledge, the CATSLU Challenge [39] is the only public competition of ASR robust SLU for Mandarin. Hence, we evaluate ARoBERT over the challenge and compare it against strong published baselines and top systems in the competition. As CATSLU specifically focuses on slot filling tasks, we fine-tune the base ARoBERT model for named entity recognition, which detects key entities from inputs as candidates to fill in the slots. To generate the complete triples for slot filling, we match the detected entities with the ontology provided by the organizer. In CATSLU, both ASR transcribed texts and the original audios are provided for training and testing. We consider the following systems as baselines: two official baselines (rule-based and neural-based), two top F1 systems in the leaderboard (with and without the audio embeddings as features), and three published models as strong baselines [30], [31], [42]. Specifically, Wang *et al.* [30] employ transfer learning mechanisms for domain and ASR-error adaption, with multi-task BiLSTM models as the base models. Multi-classification BERT [31] is particularly designed for CATSLU, which uses different output components to generate different slots. Li *et al.* [42] leverage the BiLSTM networks with additional domain and acoustic knowledge to improve the model performance.

Following the guidance [39], we evaluate the model performance in terms of both F1 and accuracy. The results are shown in Table III. From the results, we can observe that ARoBERT consistently outperforms baselines and the top system without audio features in the competition. Specifically, ARoBERT even has a higher accuracy score than the top system and the method [42]



(a) Dataset: ARoMatch (b) Dataset: ARoTopic

Fig. 5. Parameter analysis over the two development sets when the learning rate varies.

TABLE VI
ABLATION STUDY ON THREE PRE-TRAINING TASKS OF AROBERT OVER "TEST (BASE)" AND "TEST (ASR ERROR)"

Method	ARoMatch		ARoTopic	
	Acc.	F1	Acc.	Wgt. F1
Dataset: Test (Base)				
ARoBERT (Full)	0.920	0.872	0.671	0.668
w/o. MLM	0.893	0.836	0.655	0.646
w/o. AMMLM	0.906	0.847	0.660	0.650
w/o. PMLM	**0.909**	**0.856**	**0.664**	**0.655**
Dataset: Test (ASR Error)				
ARoBERT (Full)	0.620	0.428	0.532	0.532
w/o. MLM	0.601	0.412	0.514	0.510
w/o. AMMLM	0.612	0.420	0.518	0.520
w/o. PMLM	**0.615**	**0.422**	**0.520**	**0.524**

TABLE VII
COMPARISON BETWEEN TINY AND BASE MODELS OVER "TEST (BASE)" AND "TEST (ASR ERROR)"

Model	ARoMatch		ARoTopic		#Param.
	Acc.	F1	Acc.	Wgt. F1	
Dataset: Test (Base)					
ARoBERT (Tiny)	*0.920*	*0.872*	*0.671*	*0.668*	*4.4M*
ARoBERT (Base)	0.942	0.904	0.686	0.682	110M
BERT (Tiny)	0.882	0.807	0.666	0.660	4.4M
BERT (Base)	*0.921*	*0.874*	*0.674*	*0.671*	*110M*
Dataset: Test (ASR Error)					
ARoBERT (Tiny)	*0.620*	*0.428*	*0.532*	*0.532*	*4.4M*
ARoBERT (Base)	0.684	0.487	0.592	0.590	110M
BERT (Tiny)	0.589	0.328	0.482	0.481	4.4M
BERT (Base)	*0.618*	*0.405*	*0.530*	*0.530*	*110M*

#Param. refers to the parameter number. Italic figures refer to the phenomenon where ARoBERT (Tiny) has similar performance to BERT (Base), with a smaller size.

with additional audio features as inputs. As for the F1 score, ARoBERT achieves comparable results with the top system with audio features and outperforms [42]. This clearly proves the effectiveness of ARoBERT for ASR robust SLU tasks, even without the usage of audio features. In the future, we plan to fuse rich audio features into ARoBERT.

D. Detailed Model Analysis

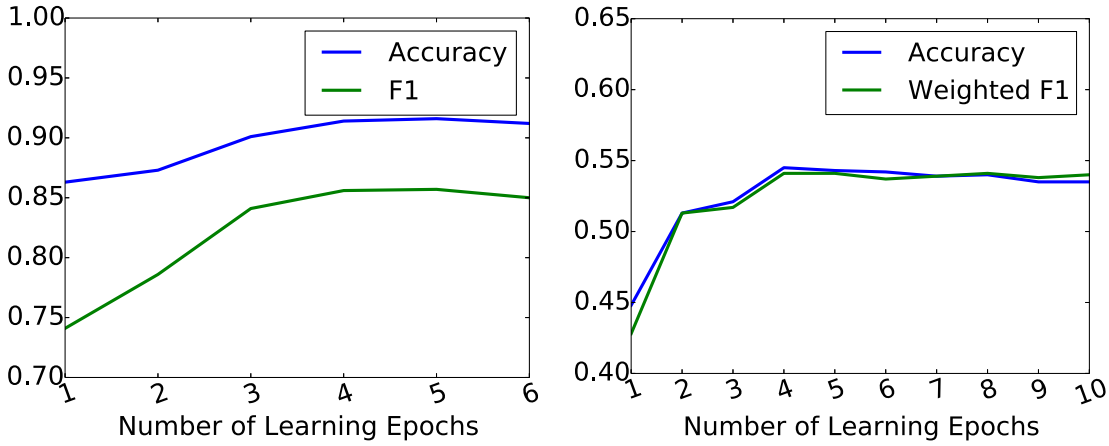
In this section, we further investigate the effectiveness of different techniques used in ARoBERT.

1) Ablation Study: In this experiment, we remove one loss function of ARoBERT at a time for pre-training, and report the performance of downstream applications for ablation study. The results are shown in Table VI. As seen, among the three tasks MLM, PMLM and AMMLM, MLM plays the dominant role. This indicates that although we wish to learn phonetic knowledge during pre-training, the semantic knowledge captured by MLM is still vital for the model and should not be ignored. Comparing PMLM and AMMLM, we find that AMMLM is slightly more effective than PMLM over *ARoMatch* and *ARoTopic*. Hence, our data-driven process of mining and generalizing ASR errors is more capable of fitting errors made by a given ASR model. Combining the three pre-training tasks together, ARoBERT achieves the best performance.

2) Parameter Analysis: We tune the learning rate and the number of learning epochs of ARoBERT. The performance on the two development sets is illustrated in Figs. 5 and 6. As seen, the best learning rates over the two SLU tasks are around $1e{-}4 \sim 2e{-}4$ and $5e{-}4$, respectively. When the number of learning epochs is concerned, the trend of the model performance is consistent across the two development sets. We suggest that a suitable choice of the learning epoch is 4∼5 for both tasks.

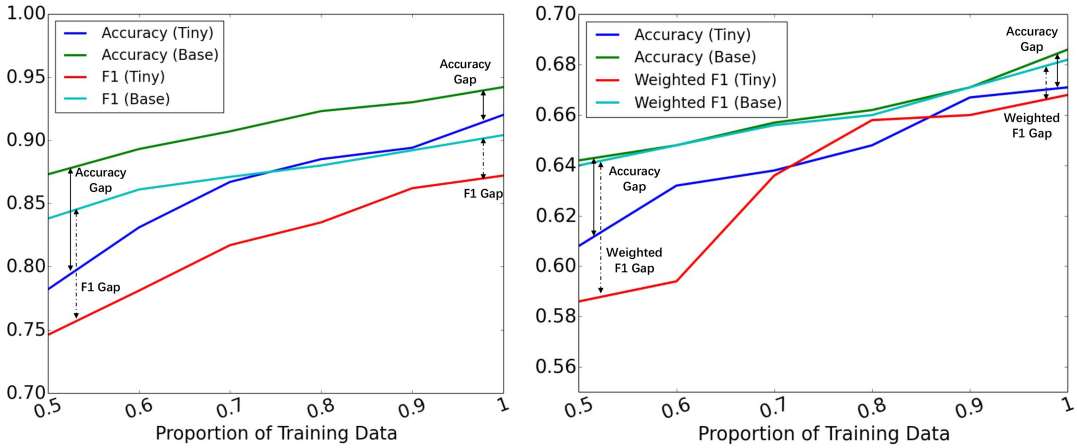
3) Learning With Larger Models: Because we use the tiny versions of BERT and ARoBERT previously to guarantee the fast online inference speed, we conduct an additional experiment to study how the performance changes when we use the base versions of BERT and ARoBERT. The result comparison is shown in Table VII.

From the experimental results, we have these observations. i) When the model size becomes larger, the performance increases correspondingly. This phenomenon holds for both BERT and ARoBERT. ii) The performance scores of ARoBERT (Tiny) and BERT (Base) are highly similar (refer to the italic accuracy and F1 scores in the table). This shows that by using the proposed pre-training technique in this work, we can use a much smaller ARoBERT model (with 4.4 M parameters) to replace the original



(a) Dataset: ARoMatch (b) Dataset: ARoTopic

Fig. 6. Parameter analysis over the two development sets when the learning epoch varies.



(a) Dataset: ARoMatch (b) Dataset: ARoTopic

Fig. 7. Performance gap between ARoBERT (Tiny) and ARoBERT (Base) when different proportions of training data are used.

large BERT model (with 110 M parameters) with similar performance. Therefore, the model size is significantly reduced by 25 times, which is particularly desirable for online applications.

We also compare the performance of ARoBERT with both base and tiny versions when the number of training instances varies. The results in Fig. 7 show that when the number of training instances is small, the gap between base and tiny versions becomes larger. This is possibly because smaller PLMs store less pre-trained knowledge with fewer parameters. When there are few training instances in downstream tasks, the pre-trained knowledge is more important for the generation of good results. Hence, larger PLMs perform better in these situations.

4) Learning With Different ASR Systems: In the set of experiments, we also report the results of ARoBERT using two different ASR systems from different providers. The first is the system described previously. The second is the transformer ASR model trained in our previous work [48], with a slight higher error rate than the default ASR model reported previously. Because the error rates w.r.t. "Test (Base)" are both low over the two ASR systems, we focus on the results on "Test (ASR Error)," with the results shown in Table VIII. Note that our transformer ASR model is referred as "Alternative". The results show that when the ASR system has a higher error rate, the performance of the ARoBERT slightly decreases, but not significant.

TABLE VIII
RESULTS ON "TEST (ASR ERROR)" W.R.T. TWO ASR SYSTEMS

ASR System	ARoMatch		ARoTopic	
	Acc.	F1	Acc.	Wgt. F1
Default	0.620	0.428	0.532	0.532
Alternative	0.608	0.405	0.525	0.520

5) Case Studies: To facilitate deeper understanding of the ASR robustness of ARoBERT, we present the case studies of the two tasks *ARoMatch* and *ARoTopic*, shown in Tables IX and X respectively. These cases are correctly predicted by ARoBERT but incorrectly predicted by BERT. We find that without ASR robust pre-training, it is extremely challenging to make the correct predictions, for example, matching "compromise" with "man's slippers". The proposed ARoBERT model can be regarded as a "bridge" to implicitly "find" the correct word "slippers"

TABLE IX
CASE STUDIES OF PRODUCT MATCHING

User Query (ASR-transcribed)	User Query (Correct)	Matched Product Name
我说的是妥协。 *Wǒ shuō de shì* tuǒ xié. I mean compromise.	我说的是拖鞋。 *Wǒ shuō de shì* tuō xié. I mean slippers.	NIKE VICTORI ONE SLIDE男子拖鞋 *NIKE VICTORI ONE SLIDE nán zǐ tuō xié* NIKE VICTORI ONE SLIDE Man's Slippers
是的，我买的那件中医。 *Shì de, wǒ mǎi de nà jiàn* zhōng yī. Yes, that Chinese medicine I bought.	是的，我买的那件风衣。 *Shì de, wǒ mǎi de nà jiàn* fēng yī. Yes, that windbreaker I bought.	女装中长款风衣（2020年新款） *Nǚ zhuāng zhōng cháng kuǎn fēng yī (2020 nián xīn kuǎn)* Women's medium-length windbreaker (new in 2020)
我买的分泌液可以退吗? *Wǒ mǎi de* fēn mì yè *kě yǐ tuì mā?* Can I return the secretion I bought?	我买的粉底液可以退吗? *Wǒ mǎi de* fěn dǐ yè *kě yǐ tuì mā?* Can I return the foundation I bought?	阿玛尼大师造型轻垫粉底液 *Ā mǎ ní dà shī zào xíng qīng diàn fěn dǐ yè* Armani Designer Lift Foundation

The task is to match the user query (ASR-transcribed) with the product name. The correct user queries are for human reference only and unseen by our models. The second and third lines of each sample are Chinese phonetic symbols and English translations, respectively. The contents underlined are words with ASR errors.

TABLE X
CASE STUDIES OF FINE-GRAINED TOPIC CLASSIFICATION

Input Text (ASR-transcribed)	Input Text (Correct)	Topic (ARoBERT)	Topic (BERT)
买两件衣服可以打针。 *Mǎi liǎng jiàn yī fú kě yǐ* dǎ zhēng. Get an injection if you buy two clothes.	买两件衣服可以打折。 *Mǎi liǎng jiàn yī fú kě yǐ* dǎ zhé. Get a discount if you buy two clothes.	优惠活动 *Yōu huì huó dòng* Promotions	材质 *Cái zhī* Material
这款大衣是翻脸的。 *Zhè kuǎn dà yī shì* fān liǎn *de.* This coat has a turning face.	这款大衣是翻领的。 *Zhè kuǎn dà yī shì* fān lǐng *de.* This coat has a lapel collar.	领型 *Lǐng xíng* Collar style	材质 *Cái zhī* Material
这个设计是非常吸金的。 *Zhè gè shè jì shì fēi cháng* xī jīn *de.* This design is very money-making.	这个设计是非常吸睛的。 *Zhè gè shè jì shì fēi cháng* xī jīng *de.* This design is very attractive.	风格款式 *Fēng gé kuǎn shì* Style	颜色花纹图案 *Yán sè huā wén tú àn* Color & Patterns

The task is to predict the topic labels of input texts (ASR-transcribed). The correct input texts are for human reference only and unseen by our models. The topic labels predicted by BERT and ARoBERT are listed for comparison.

for "compromise" by assigning similar representations to two synophones "compromise" and "slippers".

E. Online Deployment

As our work is motivated by real-world applications in e-commerce, we briefly introduce how our model is deployed online to support these applications. Specifically, we have deployed the ARoBERT model in the hotline service in a popular e-commerce platform in China (i.e., Alibaba), which is used to retrieve the most possible product information from a customer's history orders that he/she would like to query. Unlike the experiments over *ARoMatch*, we are more concerned about the performance of ARoBERT on the top-1 ranking precision. In the implementation, we sort the history orders based on the classification logits generated by ARoBERT and return the product name and its information with the highest score. To evaluate the effectiveness of ARoBERT, we conduct an online A/B test to compare ARoBERT against the online production system, which directly uses named entity recognition to extract product names from ASR transcribed texts, and matches the extracted entities against all possible product names from history orders. The underlying model for the online production system is the vanilla BERT model.

The results are reported in Table XI, in terms of Precision@1 and the averaged response time per query. From the results, we find that ARoBERT improves the precision by a large margin. Additionally, by applying the 2-layer model, the online inference process is significantly more efficient, compared to the production system.

TABLE XI
A/B TEST RESULTS OF THE ONLINE DEPLOYMENT OF THE ARoBERT MODEL FOR PRODUCT NAME RANKING

Method	Precision@1	Response Time per Query (ms)
Production System (BERT)	0.546	121.74
Proposed Approach (ARoBERT)	**0.611**	**76.51**

V. LIMITATIONS AND EXTENSIONS

In this section, we further discuss limitations and extensions of ARoBERT, aiming to stimulate the research in this field.

A. Extending to Other Languages

In this work, we focus on Mandarin speech understanding only. However, we can make some simple adjustments to extend ARoBERT to other languages. Below we discuss a possible extension of ARoBERT to English language. Consider the vocabulary set V used in ARoBERT. In most cases, each token $i \in V$ denotes a Chinese character. Based on (6), we can compute the phonetic similarity between the two tokens, according to their initials, finals and tones. For non-Chinese languages (such as English), words in sentences are usually processed by WordPiece tokenizers [49]. For each token, we can use their phonetic embeddings to compute the similarities. The remaining parts of our model can be unchanged when it is applied to other languages. Interested readers can further refer to [50] for phonetic embeddings.

B. Extending to Other ASR Errors

Our work mostly addresses the substitution errors caused by ASR systems. This is because substitution errors account for over 90% of all the errors in Mandarin ASR [48]. Yet, the deletion and insertion errors may also occur. In this part, we further discuss how to address these errors by ARoBERT. During the fine-tuning process of ARoBERT, we can augment the training data by the deletion and insertion of tokens, and train the ARoBERT model over the augmented training set. In this way, ARoBERT will be more robust to all types of ASR errors, including substitution, deletion and insertion.

C. Extending to Text Generation Tasks

The backbone of the proposed ARoBERT is primarily based on BERT [18]. Hence, ARoBERT is capable of dealing with any downstream tasks that BERT can handle, with increased ASR error robustness. We also notice that, without the decoder architecture, ARoBERT can not be used for generation tasks, such as speech summarization and speech translation. One possible solution to these generation tasks is that we extend ARoBERT to other encoder-decoder based model architectures such as BART [51] and T5 [29] with similar pre-training techniques, which will be left as future work.

D. Incorporating Other Prior Knowledge

Apart from the heuristics used by PMLM, we suggest that there are other heuristics or prior knowledge that can be used for pre-training ARoBERT. For example, there may exist confusions between some initials or finals in the Mandarin language. In our work, we use Eq. (6) to compute the phonetic similarities $sim(i, m)$, which is relatively "hard". By incorporating accent-specific similarities between initials or finals into Eq. (6), the resulted ARoBERT model would be more robust to accents, which is an open research topic in this field.

VI. Conclusion and Future Work

In this paper, we present an ASR robust PLM named ARoBERT, which fuses various phonetic knowledge into the BERT pre-training process to support SLU tasks. In ARoBERT, two novel pre-training tasks are proposed to learn ASR robust language representations, namely Phonetically-aware Masked Language Modeling (PMLM) and ASR Model-adaptive Masked Language Modeling (AMMLM). The PMLM fuses word phonetic similarities into ARoBERT, and the AMMLM mines typical ASR errors to help the model. Experiments on multiple open datasets prove the effectiveness of ARoBERT. We have also deployed ARoBERT in real-world e-commerce applications, and observed significant improvements. Future works include i) applying the ARoBERT model to other languages and tasks, ii) fusing rich audio features into ARoBERT to further improve the model performance by using a cross-modal neural architecture; and iii) extending ARoBERT to other ASR errors.

References

[1] Y. Kobayashi, T. Yoshida, K. Iwata, and H. Fujimura, "Slot filling with weighted multi-encoders for out-of-domain values," *Interspeech*, 2019, pp. 854–858.

[2] A. R. Mittal, S. Bharadwaj, S. Khare, S. A. Chemmengath, K. Sankaranarayanan, and B. Kingsbury, "Representation based meta-learning for few-shot spoken intent recognition," in *Proc. Interspeech*, 2020, pp. 4283–4287.

[3] W. Wei, H. Zhu, E. Benetos, and Y. Wang, "A-CRNN: A domain adaptation model for sound event detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 276–280.

[4] A. V. Haridas, R. Marimuthu, and V. G. Sivakumar, "A critical review and analysis on techniques of speech recognition: The road ahead," *Int. J. Knowl. Based Intell. Eng. Syst.*, vol. 22, no. 1, pp. 39–57, 2018.

[5] M. Alam, M. D. Samad, L. Vidyaratne, A. Glandon, and K. M. Iftekharuddin, "Survey on deep neural networks in speech and vision systems," *Neurocomputing*, vol. 417, pp. 302–321, 2020.

[6] D. Amodei *et al.*, "Deep speech 2: End-to-end speech recognition in English and Mandarin," in *Proc. Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 173–182.

[7] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2018, pp. 5884–5888.

[8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 12 449–12 460, 2020.

[9] E. Simonnet, S. Ghannay, N. Camelin, Y. Estève, and R. D. Mori, "Automatic speech recognition error management for improving spoken language understanding," in *Proc. Interspeech*, 2017, pp. 3329–3333.

[10] S. Zhu, O. Lan, and K. Yu, "Robust spoken language understanding with unsupervised ASR-error adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 6179–6183.

[11] S. Ghannay, Y. Estève, and N. Camelin, "Task specific sentence embeddings for automatic speech recognition error detection," in *Proc. Interspeech*, 2018, pp. 1288–1292.

[12] R. Errattahi, A. E. Hannani, T. Hain, and H. Ouahmane, "System-independent automatic speech recognition error detection and classification using recurrent neural network," *Comput. Speech Lang.*, vol. 55, pp. 187–199, 2019.

[13] S. Ghannay, Y. Estève, and N. Camelin, "A study of continuous space word and sentence representations applied to automatic speech recognition error detection," *Speech Commun.*, vol. 120, pp. 31–41, 2020.

[14] S. Toshniwal, H. Tang, L. Lu, and K. Livescu, "Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition," *Interspeech*, 2017, pp. 3532–3536.

[15] Z. Chen, M. Jain, Y. Wang, M. L. Seltzer, and C. Fuegen, "End-to-end contextual speech recognition using class language models and a token passing decoder," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2019, pp. 6186–6190.

[16] W. Zhou, R. Schlüter, and H. Ney, "Robust beam search for encoder-decoder attention based speech recognition without length bias," *Interspeech*, 2020, pp. 1768–1772.

[17] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technol. Sci.*, vol. 63, pp. 1872–1897, 2020.

[18] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[19] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Representations*, 2020.

[20] T. B. Brown *et al.*, "Language models are few-shot learners," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 1877–1901, 2020.

[21] X. Yang and J. Liu, "Using word confusion networks for slot filling in spoken language understanding," in *Proc. Interspeech*, 2015, pp. 1353–1357.

[22] W. Ruan, Y. Nechaev, L. Chen, C. Su, and I. Kiss, "Towards an automatic speech recognition error robust spoken language understanding system," in *Proc. Interspeech*, 2020, pp. 901–905.

[23] C. Huang and Y. Chen, "Learning asr-robust contextualized embeddings for spoken language understanding," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2020, pp. 8009–8013.

[24] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5998–6008.

[25] M. E. Peters *et al.*, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2018, pp. 2227–2237.
[26] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
[27] W. Wang *et al.*, "StructBERT: Incorporating language structures into pre-training for deep language understanding," in *Proc. Int. Conf. Learn. Representations*, 2020.
[28] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2978–2988.
[29] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, pp. 140:1–140:67, 2020.
[30] X. Wang *et al.*, "Transfer learning methods for spoken language understanding," in *Proc. Int. Conf. Multimodal Interact.*, 2019, pp. 510–515.
[31] C. Tan and Z. Ling, "Multi-classification model for spoken language understanding," in *Proc. Int. Conf. Multimodal Interact.*, 2019, pp. 526–530.
[32] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2018, pp. 5754–5758.
[33] N. A. Tomashenko, A. Caubrière, and Y. Estève, "Investigating adaptation and transfer learning for end-to-end spoken language understanding from speech," in *Proc. Interspeech*, 2019, pp. 824–828.
[34] E. Palogiannidi, I. Gkinis, G. Mastrapas, P. Mizera, and T. Stafylakis, "End-to-end architectures for asr-free spoken language understanding," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2020, pp. 7974–7978.
[35] Y. Chen *et al.*, "Top-down attention in end-to-end spoken language understanding," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, 2021, pp. 6199–6203.
[36] X. Cheng *et al.*, "SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 871–881.
[37] A. Fang, S. Filice, N. Limsopatham, and O. Rokhlenko, "Using phoneme representations to build predictive models robust to automatic speech recognition errors," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 699–708.
[38] P. G. Shivakumar and P. G. Georgiou, "Confusion2vec: Towards enriching vector space word representations with representational ambiguities," *Peer J. Comput. Sci.*, vol. 5, 2019, Art. no. e195.
[39] S. Zhu, Z. Zhao, T. Zhao, C. Zong, and K. Yu, "CATSLU: The 1st Chinese audio-textual spoken language understanding challenge," in *Proc. Int. Conf. Multimodal Interact.*, 2019, pp. 521–525.
[40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[41] M. Qiu *et al.*, "Easytransfer: A simple and scalable deep transfer learning platform for natural language processing applications," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 4075–4084.
[42] H. Li, C. Liu, S. Zhu, and K. Yu, "Robust spoken language understanding with acoustic and domain knowledge," *Int. Conf. Multimodal Interact.*, 2019, pp. 531–535.
[43] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2249–2255.
[44] M. Qiu *et al.*, "Transfer learning for context-aware question matching in information-seeking conversations in e-commerce," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 208–213.
[45] Y. Kim, "Convolutional neural networks for sentence classification," *Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1746–1751.
[46] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, "Convolutional recurrent neural networks for text classification," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–6.
[47] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1243–1252.
[48] M. Cheng, C. Wang, X. Hu, J. Huang, and X. Wang, "Weakly supervised construction of automatic speech recognition systems with massive video data," in *Proc. Interspeech*, 2021, pp. 4533–4537.
[49] X. Song, A. Salcianu, Y. Song, D. Dopson, and D. Zhou, "Fast wordpiece tokenization," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 2089–2103.
[50] R. Sharma, K. Dhawan, and B. Pailla, "Phonetic word embeddings," 2021, *arXiv:2109.14796*.
[51] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7871–7880.

Chengyu Wang received the Ph.D. degree from East China Normal University, Shanghai, China, in 2020. He is currently working on deep learning algorithms on various topics with Alibaba Cloud Machine Learning Platform of AI. He has authored or coauthored more than 60 research papers in international conferences and journals, such as ACL, KDD, WWW, AAAI, TKDE, CIKM, EMNLP, ICASSP, and Interspeech. His research interests include natural language processing, human speech understanding, transfer learning, and few-shot learning.

Suyang Dai received the master's degree from Fudan University, Shanghai, China, in 2019. He is currently an Algorithm Engineer with Alibaba Damo Academy. His research interests include natural language processing, spoken language understanding, and human-computer interaction.

Yipeng Wang received the master's degree from the University of Southern California, Los Angeles, CA, USA, in 2019. He is currently an Algorithm Engineer with Alibaba Damo Academy. His research interests include natural language understanding and spoken language understanding.

Fei Yang received the Ph.D degree from the Eindhoven University of Technology, Eindhoven, The Netherlands, majored in concurrency theory and theoretical computer science. He is currently an Advanced Research Specialist with Zhejiang Lab. His main research interests with Zhejiang Lab include deep learning framework, deep learning algorithms, and distributed learning techniques. He is also the Technical Director of the AI open-source platform "Dubhe".

Minghui Qiu received the Ph.D. degree from the School of Information Systems, Singapore Management University, Singapore, in 2015. From 2013 to 2014, he was a Visiting Scholar with Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Senior Algorithm Expert in Platform of AI with Alibaba Cloud, working on deep learning and transfer learning for many NLP and IR tasks. He has led a team to develop and open-source the NLP and transfer learning toolkit named EasyTransfer.

Kehan Chen received the master's degree from Zhejiang University, Hangzhou, China, in 2013. He is currently in charge of the speech-based conversation AI with Alibaba DAMO Academy, specifically the hotline Alime-bot. He has authored or coauthored several papers in conferences, including Interspeech and KDD. His research interests include speech language understanding, human-computer interaction, and dialogue systems.

Wei Zhou received the master's degree from Nanjing University, Nanjing, China, in 2013. He is currently working on deep learning algorithms with Alibaba DAMO Academy. He has authored or coauthored more than ten papers in international conference, such as ACL, KDD, SIGIR, and Interspeech. His research interests include natural language processing, speech language recognition, multimedia computing, and information retrieval.

Jun Huang received the Ph.D. degree in modern physics from the University of Science and Technology of China, Hefei, China, in 2008. He was an Associate Research Fellow with the China Academy of Engineering Physics, Mianyang, China. He currently leads a team for developing AI algorithms on the Platform of AI with Alibaba Group, responsible for developing innovative algorithms and platforms, such as deep learning, transfer learning, and federal learning. He also serve for important internal and external business of Alibaba. His research focuses on high performance distributed implementation of AI algorithms and applying them to real applications.