Chinese Spoken Named Entity Recognition in Real-world Scenarios: Dataset and Approaches

Anonymous ACL submission

Abstract

Spoken Named Entity Recognition (NER) aims to extract entities from speech. The extracted entities can help voice assistants better understand user's questions and instructions. However, current Chinese Spoken NER datasets are laboratory-controlled data that collected by reading existing texts in quiet environments, rather than natural spoken data, and the texts used for reading are also limited in topics. These limitations obstruct the development of Spoken NER in more natural and common real-world scenarios. To address this gap, we introduce a real-world Chinese Spoken NER dataset (RWCS-NER), encompassing open-domain daily conversations and task-oriented intelligent cockpit instructions. We compare several mainstream pipeline approaches on RWCS-NER. The results indicate that the current methods, affected by Automatic Speech Recognition (ASR) errors, do not perform satisfactorily in real settings. Aiming to enhance Spoken NER in real-world scenarios, we propose two approaches: self-training-asr and mapping then distilling (MDistilling). Experiments show that both approaches can achieve significant improvements, particularly MDistilling. Even compared with GPT4.0, MDistilling still reaches better results. We believe that our work will advance the field of Spoken NER in real-world settings.

1 Introduction

As one of the core tasks in Spoken Language Understanding, Spoken Named Entity Recognition (NER) aims to extract entities like person names (PER), locations (LOC), and organizations (ORG) from speeches (Tur and De Mori, 2011). The extracted entities can help voice assistants better grasp the intent behind users' questions or instructions, thereby benefiting various downstream natural language processing (NLP) tasks such as information retrieval (Weston et al., 2019) and question answering (Chen et al., 2017). While significant advancements have been made in Chinese text-based NER (Yu et al., 2020; Shen et al., 2023), Chinese Spoken NER still faces substantial challenges. The main issue lies in the disparity between current research in Chinese Spoken NER and its applications in real-world scenarios. This gap hinders the application and development of Spoken NER.

Firstly, a key challenge is that current Chinese Spoken NER datasets do not closely match real-world scenarios. For Chinese Spoken NER, the only significant dataset currently available is Aishell-NER, introduced by Chen et al. (2022). This dataset was built upon the Automatic Speech Recognition (ASR) dataset Aishell-1, which already included paired speech-text data. They utilized the MSRA (Levow, 2006) guidelines to annotate entities within the text. However, Aishell-NER does not accurately reflect real-life conditions. On the one hand, speeches in Aishell-NER are reading speeches recorded in quiet environments. In contrast, in applications like using Siri or other voice assistants, human speech is natural and often includes various background noises and informal expressions like interjections, stutters, and grammatical errors. On the other hand, the Aishell-NER dataset primarily encompasses topics related to finance and news, whereas conversations in real life cover a broader range of subjects. These differences lead to a gap between the research on Aishell-NER and its applications in real-world scenarios.

In this paper, we address the issue of the lack of annotated Spoken NER datasets for real-world scenarios by introducing a real-world Chinese Spoken NER dataset, RWCS-NER. It covers two scenarios: open-domain daily conversation (DC) and task-oriented intelligent cockpit instructions (ICI). RWCS-NER facilitates the evaluation of Spoken NER models in real-world contexts, thereby bridging the gap between Spoken NER researches and its applications.

Secondly, beyond the data shortage, current Spoken NER models are not specifically tailored for real-world scenarios and often underperform in such environments. Spoken NER typically follows a pipeline process where speech is first converted into text by an ASR model, followed by the use of an NER model to identify entities within the transcribed text. Considering the scarcity of annotated Spoken NER data, researchers have sought to leverage unlabeled data to help Spoken NER. Pasad et al. (2022) successfully used the self-training with unlabeled text data to help the NER model in pipeline. However, their approach does not fully align with real-world conditions. In practice, NER is executed on ASR outputs, which inevitably contain transcription errors from ASR. In contrast, in the self-training approach by Pasad et al. (2022), the NER models within the pipeline are trained on clean texts, devoid of any ASR inaccuracies. The discrepancy, known as exposure bias (Ranzato et al., 2015; Zhang et al., 2019), will lead the model trained on clean text to struggle with the error-prone text from ASR in real-world scenarios. Furthermore, in the pipeline workflow, transcription errors from ASR can propagate to the NER stage, disrupting the NER model's ability to accurately identify entities. This issue is also not considered in Pasad et al. (2022).

To help Spoken NER on texts with ASR errors in real-world scenarios, we first introduce self-training-asr by conducting self-training on ASR outputs to mitigate the exposure bias issue. Furthermore, to reduce the impact of ASR errors on the NER model in the pipeline process, we propose a novel mapping then distilling (MDistilling) approach. We then evaluate various approaches on RWCS-NER. Experiments demonstrate that both our self-training-asr and MDistilling approaches achieve significant improvements. Notably, MDistilling effectively alleviates the impact of ASR errors on NER in the pipeline workflow. Finally, we also evaluate ChatGPT (OpenAI, 2023) on our RWCS-NER. Results show that GPT4.0 has reached performance comparable to that of supervised models on clean texts. But our MDistilling is more competitive than GPT4.0 in real-world scenarios where texts come from the ASR model. In summary, our work makes the following contributions:

- We introduce a Chinese Spoken NER dataset, RWCS-NER, tailored for two real-world scenarios: open-domain daily conversation and task-oriented intelligent driving. It can be used for evaluating Spoken NER models in actual scenarios. We will release our dataset and codes for free at `github.com`.
- We present benchmark results for several mainstream approaches on RWCS-NER, alongside an in-depth analysis of how different ASR errors affect NER. These results show that, in real-world settings, the performance of Spoken NER is far from satisfactory, highlighting the necessity of drawing more attention to this field.
- To help the Spoken NER models in real-world scenarios, we propose two approaches, i.e., self-training-asr and MDistilling. Results indicate both our approaches achieve significant improvements, especially MDistilling. Moreover, even in comparison to GPT4.0, MDistilling maintains its edge in real-world scenarios.

2 Related Work

2.1 Chinese Spoken NER Datasets

Compared to text-based NER, there are fewer datasets for Chinese Spoken NER. In fact, both Sui et al. (2021) and Chen et al. (2022) have annotated Chinese datasets based on Aishell-1 (Bu et al., 2017). The difference lies in the scope of annotation: Sui et al. (2021) annotated entities only for a subset of Aishell-1, whereas Chen et al. (2022) annotated the entire dataset. In this paper, we opt for the larger-scale annotation from Chen et al. (2022) as the representative Chinese Spoken NER dataset, Aishell-NER. However, both the two datasets are based on the Aishell-1 ASR dataset, where the audios are reading speeches recorded in quiet environment. Additionally, the content in Aishell-1 mainly comes from the news and finance domain. These factors prevent Aishell-NER from accurately representing the performance of Spoken NER in real-world settings. Therefore, our focus is on constructing datasets tailored for real-world scenarios.

2.2 Pipeline vs. End-to-end

To mitigate the impact of ASR errors on NER in the pipeline process, recent studies have proposed end-to-end approaches that simultaneously generate text and entity labels from speech (Ghannay et al., 2018; Yadav et al., 2020). End-to-end approach integrates special entity markers into the token vocabulary of ASR model to denote the bound-

ary of entities. For instance, in Chen et al. (2022), brackets are used to denote LOC entities. The training and inference target of the raw ASR model is transformed from "I come from New York" to "I come from (New York)". Then, the ASR model becomes an end-to-end model, which can generate transcribed text from the input speech and identify entities within it.

Although end-to-end model can aviod the error propagation problem, its performance is still inferior to that of the pipeline model (Shon et al., 2022). This is because pipeline approaches can leverage the powerful representational capabilities of pretrained large language models (LLMs) when performing NER on texts. Recently, Pasad et al. (2022) propose to use unlabeled data to help the end-to-end model, which surpasses the pipeline model. However, this requires the prior development of an enhanced pipeline model using unlabeled data. Then they use the pipeline model to predict pseudo labels. The end-to-end model then learning from these labels to achieve superior performance.

In summary, while both pipeline and end-to-end have their pros and cons, the ability of the pipeline approach to leverage LLMs generally allows it to achieve commendable performance, making it the predominant method in practice. In this paper, we focus on investigating the pipeline approach.

3 Construction of RWCS-NER

In this section, we detail the construction process of our dataset. We consider two typical real-world scenarios: open-domain daily conversation (DC) and task-oriented intelligent cockpit instruction (ICI). DC typically involves dialogues between individuals about various topics. In contrast, ICI involve interactions between humans and machines, where humans interact with machines to accomplish certain tasks during driving.

Our selection of DC is driven by its prevalence in the conversation scenarios, encompassing a range of spoken expressions like interjections, stuttering, and grammatical errors, which are absent in formal written text. ICI is chosen due to the rising trend of intelligent driving and the increasing reliance on speech instructions for controlling in-car devices. Effective entity recognition in these speech instructions is pivotal for in-car systems to assist users. In addition, ICI data is relatively scarce and, to our knowledge, no publicly accessible NER dataset for intelligent cockpit instruction exists.

3.1 Data Selection

DC. DC is built upon MagicData-RAMC (Yang et al., 2022), an open-source ASR dataset comprising daily conversation speeches recorded from native speakers. This dataset preserves a multitude of spoken phenomena in both speech and transcribed text, such as interjections and stutterings. Since entities appear less frequently in some dialogues , we focus on selecting dialogues that have a higher occurrence of entities.

Specifically, we employ a Roberta-CRF NER model, which was trained on the MSRA-NER (Levow, 2006), to identify entities in all dialogues. We then filter out dialogues with fewer entities. After this, we carefully choose dialogues from the remaining set to ensure that the selected conversations cover a variety of topics. Finally, we choose 13 dialogues covering 8 distinct topics, totaling 1,559 utterances for annotation[1].

ICI. The ICI dataset is built upon the ICSRC dataset (Zhang et al., 2022), which was recently released for ASR competitions in the intelligent driving. The dataset comprises instructions relevant to intelligent driving, such as "turn on the air conditioner", "navigate to a location", and "call someone".

Unlike DC, the ICI dataset, being task-driven, exhibits a higher frequency of entities in its instructions, like location names in navigation instructions or personal names in "call someone" instructions. Therefore, we randomly select 3000 utterances from ICSRC for annotation.

3.2 Annotation Process

During the annotation process, we adopt a double-blind procedure, in which each utterance is independently annotated by two annotators. A third annotator is tasked with comparing the two annotations[2]. If the two annotations are consistent, the annotation is adopted as the final answer. Otherwise, the third annotator have to make a final decision. During the annotation process, utterances containing personal information are discarded to protect privacy. We pay the annotators based on the quality and quantity of their annotations.

DC. When anotating DC, as in previous studies (Sui et al., 2021; Chen et al., 2022), we focus

[1]Detailed statistics of the selected topics are shown in Appendix A.

[2]We build our annotation platform using Doccano.

on three types of entities: PER, LOC, and ORG. We follow the widely recognized MSRA (Levow, 2006) guidelines for Chinese NER.

ICI. For ICI, due to its unique smart driving scenario, we notice that besides the typical entities like LOC, PER, and ORG, other entities such as device names and song names are also crucial. Recognizing these entities can aid the smart driving system in accurately executing user's commands. Additionally, we notice that certain entities tend to appear in specific types of instructions. For instance, song names are commonly found in instructions for playing music. Consequently, discerning the type of an instruction can substantially assist annotators in accurately identifying entities. So, we design a new two-step annotation guideline for ICI.

In the first step, annotators determine the label of each instruction based on its function and purpose. We define five instruction labels: Navigation (NAV), Air Conditioning Instructions (AIC), Calls (CAL), Music Instructions (MUS), and Others (OTH). For instance, NAV includes user instructions related to directions, such as "navigate to a location" or "avoid highways". OTH refers to content that does not belong to the other four labels.

After determining the label of each instruction, the second step involves annotators identifying entity labels. We define seven entity labels in total. In addition to the general labels like PER, LOC, and ORG, we introduce labels tailored to the intelligent driving context: Device Name (DEV), Song Name (SON), Music Attribute (M-att), and Air Conditioning Attribute (A-att). For example, the most commonly used devices in speech instructions, i.e., conditioner and music/speaker will be labeled as DEV. Detailed descriptions for instruction and entity labels are shown in Appendix E.

Finally, in the DC and ICI datasets, we manually annotate 2862 and 2291 entities, respectively.

3.3 Data Analysis

Inter-annotator consistency. To assess inter-annotator consistency, we employ Cohen's Kappa κ (Cohen, 1960) as our metric. The value of κ, ranging from 0 to 1, indicates the level of agreement, with higher values signifying greater consistency.

In DC, the κ score of entity labels is 0.82, and in ICI, it reaches 0.88, demonstrating relatively high inter-annotator agreement. Although it might seem

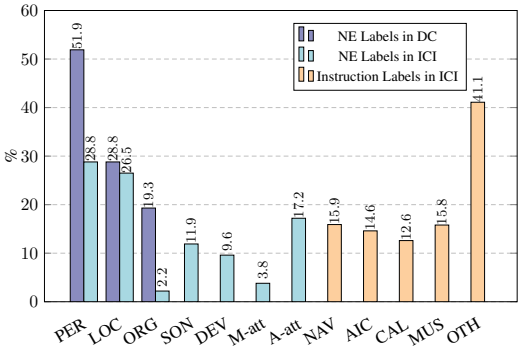


Figure 1: The distribution of labels in DC and ICI.

that the DC, with only three types of entity labels, should have a higher kappa compared to the ICI, which has seven types of entity labels, the reality is actually the opposite. On one hand, the complexity of utterances in DC exceeds that in ICI due to the inclusion of various colloquial expressions. On the other hand, the length of utterances in DC is significantly longer than in ICI. Together, these aspects make annotating DC more challenging than ICI, consequently leading to a lower κ score among annotators.

Label distribution. Figure 1 shows the distribution of labels in DC and ICI. Notably, the PER label is especially dominant in DC, constituting 51.9% of the labels. This prevalence is attributed to the more unstructured nature of daily conversations as opposed to ICI. During some discussions, it is common for individuals to pause and repeat names when mentioning others, leading to the elevated occurrence of the PER label in DC. We also observe that the ORG label is less frequent in ICI because organizations are seldom mentioned in driving instructions. Additionally, we can see that the four new entity labels defined for ICI make up a significant part, accounting for 42.5% of the labels, with A-att being the most prevalent at 17.2%.

In ICI, each instruction is assigned an instruction label. As shown in Figure 1, the four main instruction labels are distributed relatively evenly, collectively constituting 58.9% of all labels. We note that the OTH label comprises 41.1% of the dataset. This prevalence is attributed to the ICI dataset being a subset of the ICSRC dataset (Zhang et al., 2022), which was originally designed for evaluating the speech recognition accuracy of ASR models. The dataset encompasses a substantial amount of non-instructional content, like reading news, all categorized under the OTH label.

4 Our Approach

4.1 Self-training-asr

Nowadays, Chinese NER models have made great progress on formal text, such as News, thanks to the abundant annotated training data. However, when it comes to the spoken text, the NER performance drops drastically due to the lack of NER annotations for spoken text in real-world scenarios. Considering the abundance of unlabeled spoken speech-text data, we follow Pasad et al. (2022) and propose self-training-asr on these data to help Spoken NER in real-world scenarios.

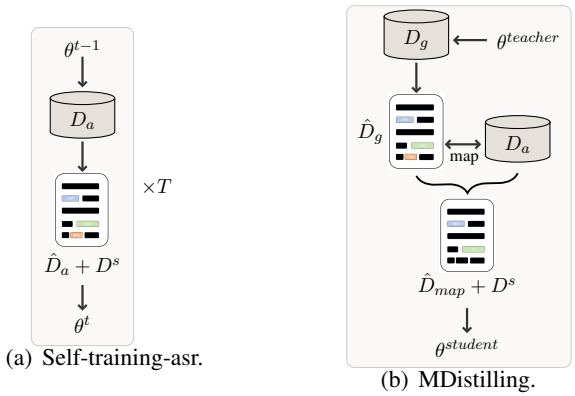
Unlike Pasad et al. (2022), who perform self-training using clean, gold text (referred to as self-training-gold), we engage in self-training on ASR output text. This choice is driven by the fact that, in real-world situations, NER models predict on ASR outputs, which might have transcription errors. Models trained solely on gold text would struggle to handle texts with ASR-induced noise in real-world scenarios. This challenge is known as exposure bias (Zhang et al., 2019).

The core principle of self-training involves using the model itself to predict unlabeled data, then treating these predictions as pseudo labels. The pseudo labels are incorporated back into the labeled data, and the model is retrained with this augmented dataset. This process can be iterated multiple times until model convergence (Yarowsky, 1995; Rotman and Reichart, 2019; Xu et al., 2021).

Specifically, in our self-training-asr we convert spoken speeches into transcribed texts using an ASR model, and then use the transcribed texts as unlabeled data D_a. As depicted in Figure 2(a), during the t^{th} iteration, we use the NER model θ^{t-1}, obtained from the last iteration, to predict on D_a, resulting in pseudo labels $\hat{D}_a$. These pseudo labels are then merged with the source labeled data D^s. Then, the NER model is retrained on the merged dataset, yielding a new model θ^t. In the first iteration, we train an NER model on the D^s as θ^0.

4.2 MDistilling

While self-training-asr improves the model's robustness in real-world scenarios by training directly on transcribed text, it doesn't explicitly consider the impact of errors in ASR outputs on NER predictions. In self-training-asr, the NER model predicts labels from transcribed text and uses these predicted labels to train the next round's model. However, errors in the transcribed text can lead to incorrect entities predicted by NER, and learning from these flawed entity labels can further impair the NER model.



(a) Self-training-asr. (b) MDistilling.

Figure 2: Frameworks of our approaches.

To analyze the impact of ASR errors on NER, we broadly categorize ASR errors into two types: entity errors and non-entity errors. Entity errors occur within the text of the entities themselves (e.g., the LOC entity "北京 (Beijing)" being transcribed as "北极 (north pole)"), while non-entity errors occur in other parts of the text. Entity errors directly result in the NER model failing to recall the original entity, which we believe cannot be rescued by the NER model in the pipeline. Therefore, our core idea is mitigating the impact of non-entity errors on NER, aiming to ensure that the NER model can still accurately identify entities as more as possible amidst the existence of non-entity errors. This guides us to propose the MDistilling. As illustrated in Figure 2(b), MDistilling has three stages:

Stage 1: Producing Silver Labels. Due to the lack of labeled data for Spoken NER in real-world scenarios, inspired by traditional distillation (Hinton et al., 2015), we use a model trained on error-free text as the teacher model $\theta^{teacher}$ to predict pseudo labels $\hat{D}_g$ from the gold Spoken text data D_g. Since $\theta^{teacher}$ is trained without ASR errors and $\hat{D}_g$ is predicted from clean text, we regard $\hat{D}_g$ as silver labels unaffected by ASR errors.

Stage 2: Mapping. To enable the model to learn correct labels from transcribed error-containing text, we map the silver labels $\hat{D}_g$ to the ASR outputs to get $\hat{D}_{map}$. We employ the maximum segmentation matching method to map labels. For instance, as depicted in Figure 3, given a gold text x_g and its ASR output x_a, $\hat{D}_g$ has two LOC entities "北京" and "中国" in x_g. We then find identical

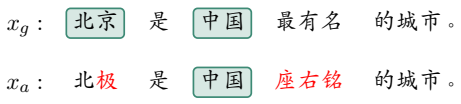


Figure 3: An example of mapping entities between the gold text"北京是中国最有名的城市 (Beijing is the most famous city in China.)" and the wrongly transcribed text. Green boxes denote entities. Red characters represent errors in transcription.

text strings in x_a and label the matched spans with corresponding entity labels. Incorrectly transcribed entity text like "北极" is disregarded. And the wrongly transcribed non-entity text "座右铭" has no effect on entity labels. As a result, $\hat{D}_{map}$ removes entity labels that are irrecoverable due to entity errors, retaining only those stemming from the silver labels and unaffected by non-entity errors.

Stage 3: Distilling. Merge D^s with $\hat{D}_{map}$ to create the augmented labeled dataset. The student model $\theta^{student}$ is then trained using this dataset. Through this process, not only is the gold knowledge from the teacher model distilled into the student model, but the student model also gains the ability to predict correct labels from noisy text, reducing the impact of non-entity errors on NER.

5 Experiments

5.1 Experimental settings

Model Settings. We follow the pipeline workflow in this work. The pipeline involves two parts, i.e., ASR model and NER model.

We use the popular conformer-based (Gulati et al., 2020) CTC/AED (Hori et al., 2017) model as the ASR model. It contains a encoder with 6 conformer layers and a decoder with 4 transformer layers. A 6-gram language model is used during decoding to get better hypothesis [3].

NER has been extensively researched with various approaches (Panchendrarajan and Amaresan, 2018; Yu et al., 2020; Shen et al., 2023). Among them, the most prevalent approach is treating NER as a sequence labeling task. So, we implement a BERT-based (102M) (Devlin et al., 2019) CRF model and a stronger RoBERTa-based (325M) (Cui et al., 2021) model as our baseline NER models. In self-training and MDistilling, we use the RoBERTa CRF model as the backbone model.

[3]Due to space constraints, we have included the detailed experimental parameter settings in Appendix C.

Datasets. In training of ASR model, in addition to the speech-text data from our RWCS-NER training sets, we incorporate about 1300 hours of open-source Chinese speech-text data to enhance its performance. These extra datasets are merged with the RWCS-NER training sets for ASR model training. Details of the additional open-source ASR data used are provided in Table 6.

As for the training of NER model, due to the lack of real-world Spoken NER training sets, most current NER models are trained on well-structured written text. Therefore, we employ the commonly used Chinese NER dataset MSRA (Levow, 2006) from the news domain. It is utilized for fine-tuning BERT and RoBERTa models as NER baseline models. Additionally, it serves as the annotated data from the source domain in the self-training and MDistilling approaches.

To evaluate on our RWCS-NER, we split our annotated datasets into development and test sets. The remaining unannotated speech-text data is used as unlabeled training set. Specific details are shown in Table 2. Additionally, we also assess our NER baseline on the MSRA test and Aishell-NER test.

Comparison Approaches. In addition to comparing with BERT and RoBERTa baselines trained only on written news text, we also compare with the self-training-gold method used by Pasad et al. (2022). Furthermore, we conduct comparisons with LLMs, i.e., GPT3.5 and GPT4.0 (Ouyang et al., 2022; OpenAI, 2023). The specific settings for GPT experiments can be found in Appendix D. To ensure a fair comparison, we utilize the same ASR model to obtain transcribed text for all methods.

Metrics. For NER evaluation, in line with Pasad et al. (2022), we employ Precision (P), Recall (R), and F1 Score as our metrics.

For ASR, besides the most commonly used CER, we also apply the Named Entity Accuracy (NEA) metric to to analyze the performance of ASR on entity texts. NEA is defined as the percentage of correctly transcribed named entity phrases. An entity phrase is considered correctly transcribed only if all its characters are accurately converted in the right order.

5.2 Main Results

Results of Baselines. We first compare the results of BERT and RoBERTa models. To gain a clear perception of the current performance of Spoken NER on our RWCS-NER datasets, we also

Model	$DC_{.Dev}$			$DC_{.Test}$			$ICI_{.Dev}$			$ICI_{.Test}$		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
On Transcribed Text												
BERT	63.82	54.46	58.77	68.63	58.16	62.96	42.02	35.29	38.36	41.42	34.08	37.39
RoBERTa	65.80	55.19	60.03	69.15	58.45	63.35	41.01	34.35	37.39	41.98	35.20	38.29
self-training-gold	67.51	56.19	61.33	71.20	59.69	64.94	40.68	36.47	38.46	42.34	36.88	39.42
self-training-asr	70.96	55.19	62.09	73.04	59.75	65.73	42.22	35.76	38.73	42.52	35.99	38.98
MDistilling	70.48	**56.74**	**62.87**	73.73	**60.94**	**66.73**	**45.96**	34.82	**39.63**	**48.07**	34.87	**40.42**
GPT3.5	61.95	46.27	52.97	63.06	47.90	54.45	26.65	34.12	29.93	28.49	35.20	31.49
GPT3.5 few shot	63.49	46.72	53.83	65.08	47.34	54.81	27.91	**38.35**	32.31	29.89	38.23	33.55
GPT4.0	69.32	47.54	56.40	72.00	52.32	60.60	36.07	31.06	33.38	37.48	33.74	35.52
GPT4.0 few shot	**71.52**	50.55	59.23	**74.39**	55.67	63.68	37.15	37.41	37.28	38.52	**39.69**	39.09
On Gold Text												
BERT	81.10	83.24	82.16	82.84	82.37	82.61	79.44	73.65	76.43	80.82	72.76	76.58
RoBERTa	86.24	86.79	86.52	86.61	83.48	84.53	80.98	74.12	77.40	78.56	71.08	74.63
self-training-gold	**88.32**	**88.16**	**88.24**	87.32	**84.89**	**86.09**	**84.54**	79.76	**82.08**	79.67	76.46	78.03
GPT3.5	73.61	76.23	74.90	72.55	71.49	72.02	55.54	74.35	63.58	59.16	75.67	66.40
GPT3.5 few shot	78.27	71.86	74.93	78.73	71.15	74.75	60.00	82.59	69.50	63.64	83.18	72.11
GPT4.0	86.12	79.69	82.78	85.54	77.15	81.13	77.41	71.76	74.48	80.18	71.19	75.42
GPT4.0 few shot	86.47	83.24	84.83	**87.42**	82.37	84.82	79.50	**83.06**	81.24	**81.38**	**85.76**	**83.52**

Table 1: Spoken NER results on our RWCS-NER.

RWCS-NER		Labeled	Utterance	Entity
DC	Train	✗	93,580	-
	Dev	✓	559	1,098
	Test	✓	1,000	1,764
ICI	Train	✗	14,878	-
	Dev	✓	1,000	752
	Test	✓	2,000	1,539

Table 2: Split of DC and ICI.

Metric	$Aishell_{.Test}$	$DC_{.Dev}$	$DC_{.Test}$	$ICI_{.Dev}$	$ICI_{.Test}$
CER	4.25	13.86	12.98	9.60	10.50
NEA	83.47	63.11	67.97	46.35	46.08

Table 3: CER ↓ and NEA ↑ on Aishell and our datasets.

evaluate on the Aishell-NER, which is collected by reading texts in ideal scenarios. On Aishell-NER, BERT and RoBERTa achieve F1 scores of 72.22 and 72.89 respectively. However, as we can see from the upper part of Table 1, both models have significant drops on the DC and ICI domains. Specifically, on DC, there is a decline of about 10 in F1 score, and on ICI, the decline was even more pronounced, exceeding 30. This shows that current models underperform in real-world scenarios.

Regarding the self-training-gold, compared to the RoBERTa baseline, we observe improvements of F1 scores on the DC and ICI test sets by 1.59 and 1.13 respectively. However, overall, the performance of current spoken NER models in real-world scenarios remains unsatisfactory. This underscores the challenging nature of our datasets and indicates substantial room for improvement in Spoken NER, particularly in real-world scenarios.

Looking into Table 3, we can see that the decrease in real-world scenarios could be mainly attributed to the ASR model's difficulty in transcribing entity text. Although, the ASR model maintains a relatively low error rate across the overall text. However, when it comes to entity text, the ASR model's accuracy is notably lower, with NEA reaching only 67.97% and 46.08% on DC and ICI, respectively. As mentioned in section 4.2, errors in entity text directly lead to the loss of these entities in NER, and such errors are not correctable within the pipeline workflow.

Results of Self-training-asr. To alleviate the exposure bias caused by discrepancies between training and prediction, we introduce the self-training-asr approach. From Table 1, we observe that self-training-asr shows improvements over self-training-gold on most datasets. Notably, on the DC test set, there's an increase of 1 point in the F1. This suggests that conducting self-training on transcribed text can effectively make the NER model more adaptable to transcribed text.

However, we notice a slight lag in self-training-asr compared to self-training-gold on the ICI test. We attribute this to the poor performance of ASR on ICI, where the impact of ASR errors is particularly severe.

Results of MDistilling. To minimize the impact of ASR errors, especially non-entity errors, we in-

Gold or ASR	Gold NonNE		ASR NonNE	
	STA	MD	STA	MD
Gold NE	86.76	87.33	84.99	86.89
ASR NE	62.69	62.99	62.09	62.87

Table 4: Effect of different types of errors on NER F1. STA: self-training-asr; MD: MDistilling.

troduce MDistilling, which trains the model to predict correct entities from the transcribed noisy text. From Table 1 we can see that, MDistilling demonstrates significant improvements than self-training-asr. On the DC and ICI test sets, MDistilling's F1 scores are higher than those of self-training-asr by 1 and 1.44 points, respectively.

On the DC dataset, we notice considerable improvements in both Precision (P) and Recall (R) for NER. On the ICI dataset, MDistilling significantly enhances the Precision value (by approximately 4.5), effectively reducing incorrect predictions by the model. Although this also results in a slight decrease in the Recall value (by about 1), the overall F1 score still demonstrates the effectiveness of MDistilling in mitigating the impact of ASR errors.

5.3 Analysis

Effect of different error types. As discussed in Section 4.2, MDistilling is designed to reduce the interference of non-entity errors in transcribed texts on NER. Therefore, we conduct a more detailed analysis of the effects of non-entity and entity errors. In Table 4, we categorize the text into entity and non-entity sections. Subsequently, we delve into understanding how gold entity/non-entity text and ASR entity/non-entity text impact NER.

Comparing the first and second rows of the table, we observe a significant drop in NER performance when ASR-predicted entity text is used. This indicates that errors in entity text are the primary contributors to the decline. Next, we examine the effect of non-entity text. Taking self-training-asr as an example, the F1 is 86.76 with gold entity and non-entity text but drops by 1.77 with ASR-predicted non-entity text. In contrast, MDistilling shows only a 0.44 of decrease. A similar phenomenon is also observed when using ASR-predicted entity text. This suggests that non-entity errors also affect NER, and the impact on MDistilling is less than that on self-training-asr.

Through the analysis, we find that MDistilling effectively mitigates the impact of non-entity errors. This validates our motivation for proposing MDistilling.

Using gold text vs. transcribed text. Apart from ASR errors, there is also an effect from cross-domain issue arising due to differences between the training and testing domains. Therefore, we conduct experiments in an ideal scenario using gold text to analyze the impact of cross-domain issues. It also shows the upper bound of Spoken NER in the absence of ASR errors.

First, we test on Aishell-NER. BERT and RoBERTa achieve F1 scores of 86.58 and 87.59, respectively. However, as seen in the lower part of Table 1, even in the absence of ASR errors, both models underperform on our DC and ICI datasets. The F1 scores decrease by 4 and 10 points on the DC and ICI test sets, respectively. Hence, even in an ideal scenario without ASR errors, the cross-domain issue also affects the performance of Spoken NER in real-world scenarios.

5.4 Comparison with LLMs

In experiments with GPT, we task GPT with identifying entities and their types in input sentences. From the results in Table 1, it is evident that GPT3.5 falls behind compared to supervised models. GPT4.0, in ideal scenarios, achieves comparable results to supervised models, and it even surpasses them on the ICI test. However, in real-world scenarios, GPT4.0 still lags behind our MDistilling, indicating that ASR errors also significantly impact GPT models. Considering the expensive cost of GPT4.0, our MDistilling proves to be a better choice in real-world scenarios.

6 Conclusion

In this paper, we present RWCS-NER, a Chinese Spoken NER dataset designed for real-world scenarios, focusing on daily conversation and intelligent driving. To boost the performance of NER in the pipeline, especially when dealing with ASR noise, we introduce two novel approaches: self-training-asr and MDistilling. We compared our approaches with mainstream pipeline models, including ChatGPT, on RWCS-NER. Our findings indicate that both our approaches, particularly MDistilling, significantly enhance performance. However, on the whole view, these approaches still leaves room for improvement on RWCS-NER. Our work contributes to the advancement of Chinese Spoken NER by providing new data and approaches, aiming to spotlight the importance of Spoken NER in real-world applications.

Limitations

In this paper, we have only annotated development and test sets to evaluate the performance of Chinese Spoken NER in real-world scenarios. Given that Chinese Spoken NER data is still in short supply, we plan to release more higher-quality dataset in the future.

On the other hand, while the pipeline approach currently shows clear advantages in Spoken NER, end-to-end approach also possess potential, such as easier deployment. Therefore, we will also explore improving the performance of end-to-end approach in real-world scenarios in our future work.

References

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5.

Boli Chen, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Meishan Zhang, and Fei Huang. 2022. Aishell-ner: Named entity recognition from chinese speech. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8352–8356.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE Transactions on Audio, Speech and Language Processing*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

Sahar Ghannay, Antoine Caubriere, Yannick Esteve, Antoine Laurent, and Emmanuel Morin. 2018. End-to-end named entity extraction from speech. *arXiv preprint arXiv:1805.12045*.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Takaaki Hori, Shinji Watanabe, and John Hershey. 2017. Joint CTC/attention decoding for end-to-end speech recognition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 518–529.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. Bidirectional LSTM-CRF for named entity recognition. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

Ankita Pasad, Felix Wu, Suwon Shon, Karen Livescu, and Kyu Han. 2022. On the use of external data for spoken named entity recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 724–737.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics*, 7:695–713.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. DiffusionNER: Boundary diffusion for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3875–3890, Toronto, Canada. Association for Computational Linguistics.

Suwon Shon, Ankita Pasad, Felix Wu, Pablo Brusco, Yoav Artzi, Karen Livescu, and Kyu J Han. 2022. Slue: New benchmark tasks for spoken language understanding evaluation on natural speech. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7927–7931. IEEE.

Dianbo Sui, Zhengkun Tian, Yubo Chen, Kang Liu, and Jun Zhao. 2021. A large-scale Chinese multimodal NER dataset with speech clues. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2807–2818, Online. Association for Computational Linguistics.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Leigh Weston, Vahe Tshitoyan, John Dagdelen, Olga Kononova, Amalie Trewartha, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. 2019. Named entity recognition and normalization applied to large-scale information extraction from the materials science literature. *Journal of chemical information and modeling*, 59(9):3692–3702.

Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034.

Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. 2020. End-to-end named entity recognition from english speech. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 4268–4272. ISCA.

Zehui Yang, Yifan Chen, Lei Luo, Runyan Yang, Lingxuan Ye, Gaofeng Cheng, Ji Xu, Yaohui Jin, Qingqing Zhang, Pengyuan Zhang, et al. 2022. Open source magicdata-ramc: A rich annotated mandarin conversational (ramc) speech dataset. *arXiv preprint arXiv:2203.16844*.

Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei. 2021. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. In *Proc. Interspeech*, Brno, Czech Republic. IEEE.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Ao Zhang, Fan Yu, Kaixun Huang, Lei Xie, Longbiao Wang, Eng Siong Chng, Hui Bu, Binbin Zhang, Wei Chen, and Xin Xu. 2022. The iscslp 2022 intelligent cockpit speech recognition challenge (icsrc): Dataset, tracks, baseline and results. In *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 507–511. IEEE.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Appendices

A Statistics of the Topics in DC

When selecting dialogues from MagicData-RAMC, to ensure that the dataset covers a broader range of topics, we also take into consideration the topics of the dialogues. As a result, the DC dataset encompasses the following eight topics.

Topic	Utterance	Token	Duration (min)
Family Life	71	2,698	11.5
Military	94	6,956	27.5
Humanities	154	18,169	73.2
Digital Devices	98	6,370	23.8
Sports	312	13,912	50.3
Art	546	19,752	77.6
Entertainment	271	7,317	26.1
Politics & Law	13	7,150	27.7
Total	1,559	82,324	317.7

Table 5: Statistics of the DC dataset.

B Datasets Used for Training ASR Models

When training the ASR model, in addition to utilizing the unlabeled training sets corresponding to DC and ICI datasets, we have also incorporated additional open-source ASR data. The datasets we used are listed in Table 6.

Dataset	Duration (h)	Address
Aishell	178	openslr.org/33
ST-CMDS	110	openslr.org/38
Primewords	99	openslr.org/47
aidatatang	200	openslr.org/62
MagicData	755	openslr.org/68
Total	1,342	

Table 6: Datasets used for training ASR models.

ASR		NER	
encoder blocks	6	encoder dropout	0.33
decoder blocks	4	scorer dropout	0.2
hidden size	512	scorer hidden size	800
learning rate	2e-3	learning rate	5e-4
batch size	16	batch size	64
warmup steps	25000	warmup steps	1500
epoch	80	epoch	20
ctc weight	0.3	self-training T	5
beam size	15		
length Penalty	-3		
log Mel	80		
window size	25ms		
window shift	10ms		
gram of LM	6		

Table 7: Parameter settings.

C Parameter Settings

For ASR models, we follow the most popular settings, including 80-dimensional log Mel filterbank features as input, 6-gram language model for decoding among others. We build the ASR model based on the WeNet toolkit (Yao et al., 2021). We set the random seed to 777 to train the ASR model only once.

For the BERT and RoBERTa NER models, two linear layers are added on top of the encoder to compute the emission scores in CRF. We choose seeds randomly to run NER models for 2 times and report the best results. The detailed parameter settings are shown in Table 7.

D Details about GPT3.5 and GPT4.0

When evaluating the performance of GPT-3.5 and GPT-4.0, we utilized OpenAI's API. The prompt we used is formed as follows: *You are an excellent linguist. Your task is to identify person entities, location entities, and organization entities in the given sentences. Return the answers in the following format: person: A B, location: C, organization: D. Separate each entity type with a space. If there are no entities of a particular type, return an empty string*. And the specific versions of GPTs we used are gpt-3.5-turbo-1106 and gpt-4-1106-preview.

k-shot	P	R	F1
0-shot	73.61	**76.23**	74.90
1-shot	**78.27**	71.86	**74.93**
2-shot	78.22	68.67	73.13
3-shot	76.30	62.75	68.87

Table 8: Effect of k in ICL few-shot.

In conducting few-shot experiments on GPT, we employ an in-context learning (ICL) approach (Dong et al., 2022). Initially, we utilize the NER model obtained through self-training-gold to make predictions on unlabeled data, resulting in $\hat{D}_e$. During testing, for a given input sentence x, we compute the similarity between x and each sentence $\hat{E}_i$ in $\hat{E}$ based on OpenAI's embedding model. When selecting k-shot ICL examples, we consider not only the semantic similarity of sentences but also the quantity of entities contained in $\hat{E}_i$. Specifically, the criterion for selecting examples, denoted as $v_{x,i}$, is defined as follows:

$$v_{xi} = s_{x,i} + 0.1 \times N_i \quad (1)$$

Here, $s_{x,i}$ represents the semantic similarity between x and $\hat{E}_i$, and N_i denotes the number of entities in $\hat{E}_i$. Finally, we select the top k examples based on the highest $v_{x,i}$ values for use in the ICL.

To determine the value of k, we conduct experiments using GPT-3.5 on the dev set of DC. The experimental results are presented in Table 8. We observe that a larger value of k does not necessarily yield better results. We attribute this to the fact that when selecting more examples in ICL, those containing entities constitute a minority. GPT is taught to predict fewer entities, resulting in a decrease in recall. We speculate that choosing examples with more entities but relatively lower semantic similarity might alleviate this issue. However, this is not the primary focus of this work, and we plan to explore it further in future research. In the end, we set $k = 1$ for all few-shot experiments.

E Definition of Labels in ICI

Due to the specific task-oriented nature of ICI, we devise a new two-step guideline. In the first step, we assign instruction labels to each utterance based on the objective of the instruction. In the second step, we identify entities within the instructions. We define a total of four instruction labels, and in addition to PER, ORG, and LOC, we introduce

Labels	Meaning	Example
	Instruction Labels	
NAV	Instructions related to the point of interest to the user.	帮我导航到图书馆 Navigate me to the library
AIC	Instructions related to the air conditioner.	把温度调到十度 Set the temperature to ten degrees
CAL	Instructions related to contacting someone.	给李先生发条短信说... Send a message to Mr. Li saying...
MUS	Instructions related to music or speaker settings.	播放周杰伦的歌 Play Jay Chou's song
OTH	Instructions that do not belong to above four categories.	告诉我明天的天气 Tell me tomorrow's weather
	Entity Labels	
DEV	Device names.	把（空调）的调到十度 Set the (air conditioner) to ten degrees
SON	Song names.	播放（我心永恒） Play (My Heart Will Go On)
M-att	Attributes related to music or speakers.	把（音量）调到最大 Set the (volume) to maximum
A-att	Attributes related to air conditioner.	打开（制热模式） Turn on the (heating mode)

Table 9: The definition of labels in ICI. Entities corresponding to the entity labels are enclosed in parentheses.

four new entity labels. The specific meanings and examples of these labels are outlined in Table 9.