Enhanced Chinese Spoken Language Understanding

1* Zhennan Shen
SEIEE
Shanghai Jiao Tong University
Shanghai, China
1641225799szn@gmail.com

1* Zhan Su
SEIEE
Shanghai Jiao Tong University
Shanghai, China
ililaoban@sjtu.edu.cn

1* Bo Huang
SEIEE
Shanghai Jiao Tong University
Shanghai, China
huang2003bo@sjtu.edu.cn

***Abstract*—Despite the remarkable success of spoken language understanding (SLU) in dialog system, it' s generally studied based on English. This time, we consider this task in Chinese field. Besides a basic rnn-based model provided, we have also explored and implemented several enhancement based on it. Generally saying, we've**

***Index Terms*—Chinese SLU, Bert, CRF, pinyin, data augmentation**

I. Introduction

Spoken language understanding (SLU) is the task of inferring the semantic meaning of spoken utterances. SLU is an essential component in speech assistants, social robots, and smart home devices. In these tasks, it converts natural language into structured semantic representations, thereby helping machines understand human intentions.

In this teamwork, we mainly focus on how to improve the final accuracy. Besides fine-tuning on the given baseline model structure, we tried transformer model: the most popular architecture in recent years in NLP field. Then we explored the usage of Chinese word segmentation and bert pre-trained model. To support word segmentation, we rewrote the whole pipeline (data preprocessing, testing, validation and main function). We then explore CRF..., . After that, we tried data agumentation. Beyond all these, we got inspired from the dataset, we built a pinyin correction model with ASR, which partially solved the ASR recognition error problem and improved the accuracy of the model.

Our contributions are:

- We finetuned all the baseline and proposed models and provided detailed experiment settings and hyper-parameters in our paper.
- We evaluate the performance of baseline models with different RNN encoders(LSTM, GRU, RNN), compare and analyze their results.
- We propose Conditional Random Fields(CRFs) to decoders, which captures the joint distribution of label sequences, significantly enhancing the the accuracy of labeling.
- We propose data augmentation which greatly enhanced the model's generalization capability.
- We propose pinyin correction to partly solve the ASR recognition error problem and improved the accuracy of the model from 1% to 5%.
- We propose Bert enhanced model, a simple yet effective framework for SLU tasks, and justify their performance with solid experiments, whose results surpass all the baselines' accuracy by nearly 10%.

*All authors contributed equally.

II. Problem Formulation

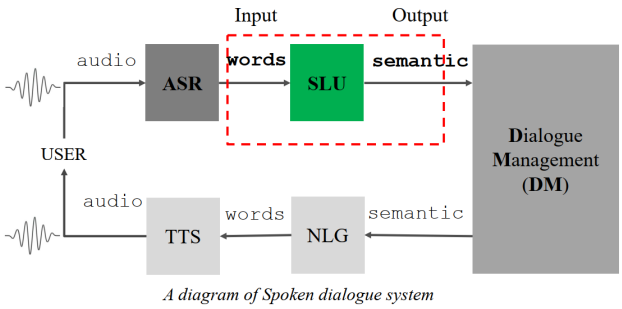

Fig. 1. Fig.1(a)

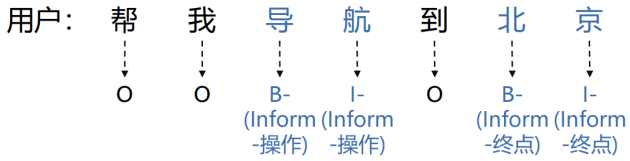

Fig. 2. Fig.1(b)

Fig. 1(a) shows a diagram of Spoken dialogue system, which includes five parts. Our task focuses on the SLU part, which takes words as input and output their semantics. In general, users' s intention is defined with some semantic units, each consists of a (act, slot, value) triplets. In the next few sections, we will introduce different algorithms, model architectures, and original improvements. Next is the introduction of datasets, experimental settings and result analysis.

III. Methodology

A. The Baseline Model

To solve the problem of spoken language understanding, one easy way is to transform it into a sequence labeling problem, which is used by our baseline method. The standard way to do this is the "BIO" encoding[2], where we label each word by "B-X", "I-X" or "O". Here, "B-X" means "begin a phrase of type X", "I-X" means "continue a phrase of type X" and "O" means "not of any type." There are 2 types of action and 18 types of slots in our dataset, so X has 36 types totally. Then we' re faced with the problem of assigning the label to each word in a sequence, which can then be seen as a multi-class classification task. All of the single words in the sequences will be embedded as vectors, and denoted by indexes. Since all the types of X can be the beginning or middle of the phrase, and we add a type of ' PAD' to represent words that should be ignored, there' re totally 74 tags denoted by their indexes as well. Then we feed the input data into RNN encoders to extract their features. The encoders we choose are bi-LSTM, bi-GRU and bi-RNN.

Bi-RNN. A multi-layer Elman RNN with tanh can be applied to the model input as encoder layer. For each word in the input sequence, each layer computes the following function:

$$h_t = tanh(w_{ih}x_t + b_{ih} + w_{hh}h_{t-1} + b_{hh})$$

where h_t is the hidden state at time t, x_t is the input at time t, and $h(t_1)$ is the hidden state of the previous layer at time $t-1$ or the initial hidden state at time o.

Bi-GRU. For each word in our model input, each layer computes the following function:

$$r_t = sigmoid(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr})$$

$$i_t = sigmoid(W_{ii}x_t + b_{ii} + W_{hi}h_{(t-1)} + b_{hi})$$

$$n_t = tanh(W_{in}x_t + b_{in} + r_t(W_{hn}h_{(t-1)} + b_{hn}))$$

$$h_t = (1-i_t)n_t + i_t * h_{(t-1)}$$

where h_t is the hidden state at time t, x_t is the input at time t, and $h(t_1)$ is the hidden state of the previous layer at time $t-1$ or the initial hidden state at time o. And r_t, z_t, n_t are the reset, update, and new gates, respectively. $*$ is the Hadamard product.

Bi-LSTM. A multi-layer long short-term memory (LSTM) RNN has been chosen as one of the encoders. For each word in the input sequence, each layer computes the following function:

$$i_t = sigmoid(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi})$$

$$f_t = sigmoid(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf})$$

$$o_t = sigmoid(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho})$$

$$g_t = tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg})$$

$$c_t = f_t c_{t-1} + i_t g_t$$

$$h_t = o_t * tanh(c_t)$$

(3) where h_t is the hidden state at time t, ctis the cell state at timet, xtis the input at time t, h_{t-1} is the hidden state of the layer at time $t-1$ or the initial hidden state at time o. i_t, f_t, g_t, o_t are the input, forget, cell, and output gates, respectively.

B. CRF

Applying Conditional Random Fields (CRFs) to decoders presents significant advantages. As a potent tool for sequence modeling, CRFs effectively address sequence labeling tasks such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging by considering the dependencies between elements in a sequence. Compared to models that only consider individual elements, CRFs capture the joint distribution of label sequences, significantly enhancing the accuracy of labeling.

C. Data Augment

To enhance the model's generalization capability, we incorporate various phrases from "ontology.json" into the training set. The data is mainly divided into three major parts:

1. "poi 名称", " 操作", "序列号" : These are directly used as manual_transcript and asr_best, with semantic as [inform, data type, data].

Example: For the "poi 名称" ——"锦州南站", we should add the following entry to the training set.

```
"utt_id": 1,
"manual_transcript": "锦州南站",
"asr_1best": "锦州南站",
"semantic": [
    [
        "inform",
        "poi名称",
        "锦州南站"
    ]
]
```

2. " 终点名称": This type of data mainly appears in the training set in forms such as " 导航到 xxx," " 导到 xxx," etc., and there may be slight variations in the natural language expressions of the "navigate" semantics in the dataset. To mitigate bias, we will include "navigate to xxx" in the training set.

For example, if the destination name is " 锦州南站", we will add the following entry to the dataset.

```
"utt_id": 1,
"manual_transcript": "锦州南站",
"asr_1best": "导航到锦州南站",
"semantic": [
    [
        "inform",
        "终点名称",
        "锦州南站"
    ]
        ,[
        "inform",
```

```
        "操作",
        "导航"
    ]
]
```

3. Other action: The processing method for this category is consistent with 1., but due to differences in data storage format, there are variations in code implementation. We have categorized it separately. We do not perform data augmentation on the ' 对象' (object) category of data."The reason is that " 对象" data alone lacks practical meaning and must be considered in context to be meaningful. Adding it without context could lead to data overfitting, resulting in a decrease in model performance.

D. Pinyin Correction

Data about the knowledge structure and dictionary of the current navigation domain is provided. In this task, we utilize these structures to construct a pinyin-based correction system to enhance the accuracy of SLU.

For each slot in the ontology, we build a map from the pinyin of the value in the ontology to the corresponding value of the same slot in the ontology. Then we can correct the value of the slot in the ontology by finding the most pinyin-similar value in the ontology.

Pinyin correction can enhance the accuracy of SLU from two aspects:

1. pinyin correction can map the error value caused by ASR recognition error to the correct value, thereby enhancing the accuracy of SLU

A typical exmaple is

```
    "manual_transcript": "导航去合一映像",
    "asr_1best": "导航去合一印象",
```

According to the ontology, we can map " 合一印象" to " 合一映像" and thus corrected the value.

This type of correction has great benefits, because under the BIO method, no model can overcome the error of ASR, but through the correction of pinyin, we can correct the error of ASR, thereby improving the effect of SLU.

2. pinyin correction can map the error value caused by the model BIO error to the correct value, thereby enhancing the accuracy of SLU

A typical example is

```
    "manual_transcript": "导航奉贤区振华路六百
        九十九弄",
    "asr_1best": "导航奉贤区振华路六百九十九弄
        ",
    "semantic":
    ...
        [
            "inform",
            "终点名称",
            "奉贤区振华路六百九十九弄"
        ]
```

The baseline model predicted the value as '区振华路六百九十九弄 ', which is a wrong answer, but we can map it to '奉贤区振华路六百九十九弄 ' according to the dictionary of poi_name by the pinyin.

This shows the ability of pinyin correction to enhance the accuracy of SLU, this especially works for those models that have a poor BIO performance.

E. Bert

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a pre-trained natural language processing (NLP) model introduced by Google in 2018. It is part of the Transformer architecture family, which has become widely adopted in various NLP tasks due to its effectiveness.

In this task, BERT could be utilized to seize the bi-direction relationship between characters in order to assist the act-slot-value prediction. We choose '**bert-base-chinese**', a model which has been pre-trained for Chinese. We encode the text in training data with and without noise using a tokenizer to obtain the bert model's input. After encoding, we get the output tensor from the model and extract the vector representations for text with and without noise. Then we shall use the processed vector in the following part.

In this task, utilizing BERT could augment the accuracy of SLU by approximately 11%.

IV. Experiments

A. Experiment Setup

1) Data: We use the augmented training and original validation datasets in our experiment.

- **"utt_id"**: Represents a unique identifier for the speech or dialogue.
- **"manual_transcript"**: Contains the manually transcribed text.
- **"asr_1best"**: Contains the result of the Automatic Speech Recognition (ASR) system.
- **"semantic"**: A list containing semantic understanding information. Each element is a list with three values, representing information about a semantic slot. Specifically:
  - The first value is a string representing the role or type of the semantic slot (e.g., "inform").
  - The second value is a string representing a specific operation or information related to the semantic slot (e.g., " 操作", " 序列号").
  - The third value is a string representing the specific value associated with the semantic slot (e.g., " 导航到凯里大十字" " 第二个").

2) pinyin correction: ontology.json explains the knowledge structure of the navigation domain, data/lexicon/operaion_verb.txt, ordinal_number.txt, poi_name.txt contain the dictionaries of operation, ordinal_number, poi_name.

- extract the possible value from the data, and thus to build a standard pinyin set and a dictionary from the pinyin set to these words for each slot

- Use a trained model to predict and get the predicted slot and value
- find the most similar pinyin in the slot's pinyin set of the predicted value's pinyin
- use the corresponding value of the most similar pinyin as the value after pinyin correction

3) bert:

- Use the tokenizer to tokenize both the text with noise (`text_with_noise`) and the text without noise (`text_without_noise`).
- Convert the tokenized texts into the model input format.
- Input the tokenized texts into a pre-trained language model. Then obtain the model outputs as vectors, serving as semantic representations.
- Include vectors and corresponding tokenized results for both text with noise and text without noise in the dictionary for following use.

B. Main Results (Q1)

TABLE I
BEST PREDICTION RESULTS OF BASELINE MODELS

Model	Accuracy	Precision	Recall
Baseline-RNN	69.77	78.17	72.87
Baseline-LSTM	70.79	79.12	74.58
Baseline-GRU	70.91	80.17	74.52

TABLE II
BEST PREDICTION RESULTS OF UPGRADED BASELINE MODELS

Model	Accuracy	Precision	Recall
Baseline-GRU-Pinyin	75.20	86.83	77.69
Baseline-GRU-Crf	78.43	84.50	80.71
Baseline-GRU-Pinyin-Crf	83.80	89.08	86.76

TABLE III
BEST PREDICTION RESULTS OF **Bert**_ENHANCED MODELS

Model	Accuracy	Precision	Recall
Bert-RNN	75.41	81.64	79.34
Bert-LSTM	79.44	84.93	85.32
Bert-GRU	79.89	83.86	87.20
Bert-RNN-Pinyin	78.32	83.78	84.27
Bert-LSTM-Pinyin	80.34	85.98	86.38
Bert-GRU-Pinyin	82.01	84.94	90.26

C. Further Study (Q2)

After completing the above tasks, we are still not satisfied with the accuracy of only little higher than 0.8. After analysis on model considering the dev dataset, we found that from the perspective of the BIO taggging method alone, our model has been very successful, and the current major problem that really limits the model to be applied to specific scenarios is ASR recognition error.

Therefore, in further study, we analyzed the impact of ASR error on the model effect, and analyzed the improvement effect of our pinyin correction method on ASR error, thus analyzing the feasibility of industrial deployment.

The related data of further study is provided in the `further_study_data`.

1) ASR error Limit: ASR error widely exists in the dataset.

For example, in the dataset, " 导航到哈尔滨医科大学附属第一" is recognized as " 导航到哈尔滨医科大学附属" by ASR, while one of its correct semantics is ["inform", "destination", " 哈尔滨医科大学附属第一"], such ASR error cannot be solved by using BIO to perform POS tagging.

We define the proportion of data that cannot be solved by BIO tagging as ASR error limit.

The specific case is that if the value of a semantic triple can't be found in the asr_1best, then this triple is considered to be unsolvable through performing BIO tagging by BIO.

Based on the analysis of development.json, it is found that there are 853 triples that are BIO-solvable, the other 110 triples are BIO-unsolvable, that is, the ASR error limit is 110 / (853 + 110) = 0.0.1142263759086189.

Thus any BIO tagging method, it's dev-acc is definitely lower than 0.8857736240913812.

2) ASR error + pinyin Limit: ASR error can be roughly divided into two categories, one is mainly the ASR error of place names, and the other is the ASR error of non-place names.

Our pinyin correction method theoretically works for both types of ASR error, but in fact, pinyin correction hardld work for non-place names ASR error.

The reason is that the place name is longer, which makes the modification of the place name more feasible, while the value of the non-place name is shorter, and the modification is more difficult.

We can roughly assume that for non-place name ASR error, our pinyin correction method is basically invalid.

Through the analysis of the data, we found that among the 110 ASR errors, 85 are place name ASR errors and 25 are non-place name ASR errors.

Thus we can conclude that the upper bound of dev-acc for BIO-tagging+pinyin correction(used for place name) method is 0.9740394600207685.

Although our current model cannot reach this limit because the accuracy of pinyin correction on place names is low, this limit provides an upper bound for all BIO-tagging+correction on place name methods.

And correction on place name is a relatively easy method to implement, for example, by introducing maps, so this upper bound of 0.9740394600207685 is instructive.

3) how to handle ASR error with non-place value: Based on the above analysis, although non-place name value only accounts for a small proportion of the data, but in fact it is of great significance to improve the customer experience.

We can roughly put the idea that ASR error about non-place name value is BIO-unsolvable.

But owing to non-place name value is closely related to the context, we can solve this problem by combining the way of context semantic understanding instead of tagging. This will be a direction of our future study.

V. Conclusion and Outlook

A. Conclusion

In the overall enhancement process of our SLU (Spoken Language Understanding) model, we have made significant strides through various avenues of exploration and optimization. We employed diverse techniques, including but not limited to data augmentation, pinyin correction, integrating BERT for data preprocessing, and enhancing the decoder through CRF. These strategies underwent thorough validation during implementation, confirming their effectiveness.

Specifically, the introduction of data augmentation enriched the model's training data, thereby improving its generalization capabilities. Simultaneously, pinyin correction enhanced the model's ability to process speech input more accurately, facilitating a better understanding and recognition of spoken information. The incorporation of BERT for data preprocessing further strengthened the model's semantic understanding, resulting in an overall performance boost.

On the decoder front, the adoption of CRF as an improvement strategy yielded significant achievements in model performance. This enhancement not only increased accuracy but also contributed to better capturing sequential information in speech tasks, enhancing the model's robustness and precision.

In summary, after multiple rounds of experimentation and validation, our model, with comprehensive improvements in pinyin correction, CRF, and data augmentation, achieved the optimal performance with an accuracy rate of 83.8%. This establishes a solid foundation for attaining superior performance in SLU tasks.

B. Outlook

During this task, we've explored lots of measures to enhance the capability of model in SLU task. However, there still remains a lot of valuable and promising ideas to be considered and implemented. Here we shall list them and provide our consideration about them:

1) **Assistive Knowledge Retrieval**
   During this task, we found several 'miss' act-slot result due to the rareness of the corresponding value. Considering the capacity of the model, it is quite impossible to rely on the diversity of the training data to handle this. Thus, the assistive knowledge retrieval should be able to eliminate such limit, like connecting to a pre-processed database or even a search engine through agent.
2) **Preprocess of Input**
   In the course of our experiments, we have frequently observed interference in the provided Automatic Speech Recognition (ASR) data, leading to the loss of objective information and consequently resulting in suboptimal model performance. In response to this, we deem it necessary to preprocess the input data. Specifically for Chinese, we contemplate replacing or eliminating undesirable inputs by considering factors such as pinyin, general grammar rules, and even commonly used vocabulary. We believe that this, in practice, introduces prior knowledge but still falls within the realm of improving the performance of Spoken Language Understanding (SLU) tasks.
3) **Dynamic Contextual Adaptation**
   Dynamic contextual adaptation is also an important factor in real-world application of SLU related job. In certain scenarios, the context surrounding a user's input may significantly impact the interpretation. Implementing a mechanism for the model to dynamically adapt its understanding based on contextual cues could further improve its robustness and accuracy. This involves continuously analyzing and adjusting the model's comprehension in real-time, allowing it to adapt to the evolving context of the conversation.

Acknowledgment

We would like to express our sincere gratitude to our teacher and teaching assistants for their guidance, support, and invaluable feedback throughout the duration of this project.

Additionally, we extend our appreciation to everyone in the group for their collaborative efforts and contributions to the project. Their dedication and teamwork played a crucial role in the success of the exploration and experimentation with various models.

Furthermore, we would like to acknowledge and thank the authors of the referenced articles, whose research and insights provided the foundation for our survey and informed the decisions in our project.

In summary, we are grateful to everyone who played a part in the development and completion of this project, contributing to a rewarding and enriching learning experience.

References

[1] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[2] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging[J]. arXiv preprint arXiv:1508.01991, 2015.

[3] Xu P, Hu Q. An end-to-end approach for handling unknown slot values in dialogue state tracking[J]. arXiv preprint arXiv:1805.01555, 2018.

[4] Kawakami K. Supervised sequence labelling with recurrent neural networks[D]. Technical University of Munich, 2008.

[5] Kim K, Jha R, Williams K, et al. Slot Tagging for Task Oriented Spoken Language Understanding in Human-to-human Conversation Scenarios[C]//Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL). 2019: 757-767.