Chinese Spoken Language Understanding with Pre-trained Language Models

Ziyin Zhang* and **Sizhe Zhou†**
{daenerystargaryen, sizhezhou}@sjtu.edu.cn

Abstract

Spoken Language Understanding (SLU) is one of the core technologies in building dialogue systems. It extracts semantic concepts from audio transcriptions by recognizing and filling action slots pre-defined for the system's application domain. In this project, we experiment with Chinese SLU by formulating it as a sequence tagging problem. We apply LSTM, RoBERTa, and XLM-R to the task, achieving 78.66 validation accuracy and 82.89 validation F_1 score. We propose a novel dual-channel decoder architecture for SLU that utilizes manually corrected input text to increase model's denosing capability, obtaining a notable performance gain over the LSTM baseline. We also formulate the task a a sequence generation problem and train an mT5, scoring 78.66 accuracy and 82.85 F_1 on the validation set.

1 Introduction

Spoken Language Understanding (SLU) is a core component in dialogue systems. It takes the Automatic Speech Recognition (ASR) transcriptions of users' audio as input, and converts them to structured semantic information that can be processed by the downstream dialogue management system (Figure 1). Most existing approaches in the literature towards SLU divide it into two sub-problems: intent classification and slot filling. Intent classification focuses on predicting a single intent label from the user query (eg. `Navigation` or `FindMovie`), while slot filling extracts more detailed semantic concepts related to the intent, such as destination, time and date, or the preferred genre of movies (Chen et al., 2019; Qin et al., 2021).

Historically, intent classification and slot filling were considered as two independent tasks and processed by separate modules (Yao et al., 2014; Ravuri and Stolcke, 2015). However, since these two tasks are inherently related, the recent trends in the literature have been to model them jointly, especially after the advent of pre-trained language models (Chen et al., 2019; Castellucci et al., 2019). However, since the intent of a user query can be easily inferred from semantic slots extracted by the slot filling module, in this work we take an even further step, completely discard the intent classification task, and regard SLU solely as a slot filling problem. Following the general practice in the literature, we formulate slot filling as a sequence tagging task (Figure 2), and apply various models to solve it in section 3. In section 4, however, we also explore the possibility of refomulating sequence tagging as a sequence to sequence generation task, which has demonstrated amazing potential when combined with pre-trained language models in recent years (Raffel et al., 2020; Xue et al., 2021).

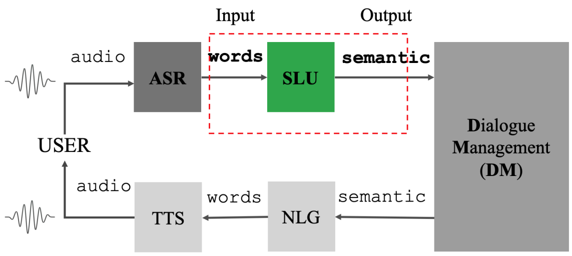


Figure 1: The general architecture of a dialogue system.

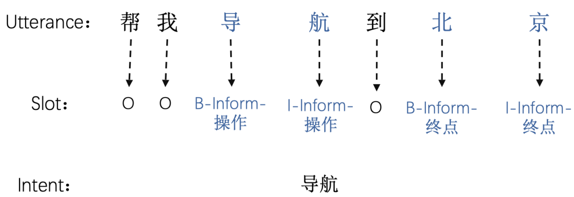


Figure 2: An illustration of SLU formulated as intent classification and sequence tagging.

* Department of Computer Science and Engineering, SEIEE

† Department of Electrical and Computer Engineering, UM-SJTU Joint Institute

```json
{
    "utt_id": 1,
    "manual_transcript": "导航到和田市公安局",
    "asr_1best": "导航到和田市公安局",
    "semantic": [
        [
            "inform",
            "终点名称",
            "和田市公安局"
        ],
        [
            "inform",
            "操作",
            "导航"
        ]
    ]
}
```

```json
{
    "utt_id": 1,
    "asr_1best": "朝阳县农机销售有限公司导航",
    "semantic": [],
    "pred": []
}
```

Figure 3: An example of annotated (left) and test (right) data instance.

2 Problem Formulation

2.1 Dataset

Our dataset consists of ASR transcriptions of 6014 navigation queries, partitioned into 5119 training samples and 895 validation samples. Each of these transcription is annotated with one or more semantic concept, represented by an action-slot-value triple, as shown in Figure 3, while at test time each new transcription contains an empty field of semantic information, and a `pred` field to be filled by the model's prediction. The annotated dataset also contains a field of manually corrected audio transcription, the, the application of which we will explore in section 3.2.

2.2 Task Definition

As explained in section 1, we formulate SLU as a sequence tagging task. Given an input token sequence $\mathbf{x}_{1:n}$, we aim to find a label y_i for each input token, such that the conditional probability of the label sequence $y_{1:n}$ is maximized:

$$\hat{y}_{1:n} = \max_{y_{1:n}} p(y_{1:n}|\mathbf{x}_{1:n}). \tag{1}$$

In Equation (1), each y_i can be one of `B`, `I`, `O`, indicating the corresponding token to be at the beginning of, inside, or outside a semantic concept. Moreover, the classical BIO tagging is extended to include action and slot information in the tagging labels. Specifically, we build a dictionary of semantic concepts from the training data that includes two types of actions - `inform` and `deny` - and 18 slot types (such as destination, travel method, route preference), and assume that any new query at test time contains only actions and slots within this dictionary. We than take the combination of each action-slot pair with either `B` or `I` to obtain a total of 73 tagging labels, each in the form of a triple `B-action-slot` or `I-action-slot` except the special tag `O`, thereby formulating SLU as a standard sequence tagging problem. In the inference stage, the slot values are simply extracted as substrings of the ASR transcription according to the predicted tagging labels.

2.3 Evaluation Metrics

In this work, we use two metrics to evaluate the performance of different SLU systems - accuracy and F_1 score. Accuracy is defined as the percentage of queries for which the system correctly predicts all the action-slot values, while F_1 score is the harmonic mean of precision and recall computed on the set of action-slot values of all queries. For each model in the following experiments, we report the best metrics on the validation set during training.

3 SLU with Sequence Tagging

3.1 LSTM

Recurrent Neural Network (RNN), due to its recurrent nature, has been widely adopted to process sequences with varied lengths, such as text or audio signals. And Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber, 1997) has demonstrated the ability to effectively cope with the vanishing gradient problem in RNN and model long-distance dependency within a sentence.

As a baseline, we train a two-layer bidirectional LSTM with 256 hidden units to encode the user query obtained by ASR. The last layer's forward and backward hidden states of each token is concatenated and projected by an output layer to the label space, as demonstrated in Figure 4. We initialize the model with pre-trained 768-dimensional word vectors with a vocabulary size of 9600. Each Chinese character in the user query is mapped to a word vector, and any token out of vocabulary is mapped to the vector of special token `<unk>`. Also, since sequential inputs must be padded to the same length to facilitate parallel computation in LSTM, we add another special tag `<pad>` to the label set apart from the 73 tags mentioned in section 2.2 to mark the labels for padded positions during training.

3.2 Dual-channel Decoder with Pre-training

When utilizing the vanilla sequence tagging for SLU, one can easily realize that in the original experimental setup, only the noisy input text is used and our task is ignorant of model's capability to de-noise from the noisy texts. This will lead to

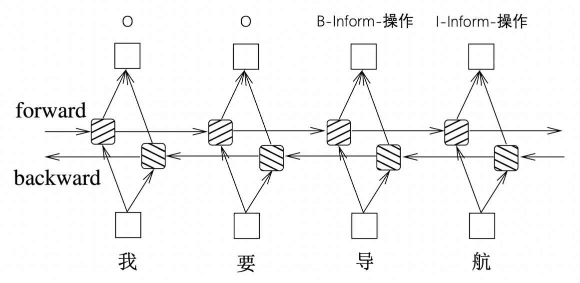


Figure 4: An illustration of SLU as sequence tagging using Bi-LSTM, adopted from Huang et al. (2015).

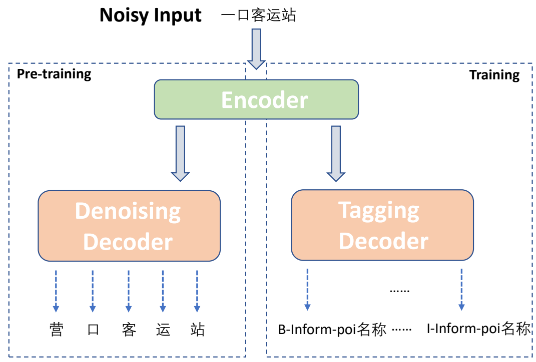


Figure 5: Overview of dual-channel decoder model with pre-training and training.

a deficiency in model's comprehension capability. Additionally, it can be seen that the dataset's domain is mainly related to the navigation system, which makes it possible to incorporate certain rules for model to learn so that it can de-noise from the noisy texts.

Based on the above analysis and inspired by recent successful applications of denoising auto-encoder in pre-trained language models such as BART (Lewis et al., 2020), we introduce our dual-channnel decoder model with pre-training. As shown by Figure 5 , it consists of a single encoder and two distinct decoders. The encoder is in charge of encoding as well as comprehending the input text. The tagging decoder produces the regular tagging sequence for SLU, while the de-noising decoder reconstructs the noise-free input text to improve the encoder's de-noising capability. The de-noising decoder's mission is mainly fulfilled in the pre-training stage.

Pre-training We conduct pre-training by the following steps: (1) Construct pre-training data: from 5119 training samples, we extract 3186 samples that have the same noisy and de-noised input texts and 420 samples with different noisy and de-noised input texts of the same token length [1]; (2) Conduct pre-training: the task for pre-training is to let the encoder and the de-noising decoder to reconstruct the de-noised text (pre-training output) from the noisy texts (pre-training input).

Setup The pre-training loss is set to be the cross-entropy loss for sequence tagging. For the de-noising decoder, we use a 2-layer feed-forward network with hidden size twice of the encoder's output hidden size and a ReLU activation function after the intermediate layer. It's expected that during pre-training, the de-noising reconstruction objective will enable the encoder to learn from this domain's data to handle the rules for simple de-noising. Then, with the encoder pre-trained, the next step - regular sequence tagging training for SLU - will start from a more learned point.

3.3 RoBERTa and XLM-R

While LSTM and its variant Gated Recurrent Unit (GRU) have historically achieved promising performance on Machine Translation (MT) and ushered the NLP community into neural age (Sutskever et al., 2014; Cho et al., 2014), they still suffer from deficiency in modeling long-distance dependencies and inherent incompatibility with parallelization. Bahdanau et al. (2015) applied attention mechanism on top of GRU to address the first issue, and Vaswani et al. (2017) groundbreakingly introduced Transformer, completely replacing recurrent units with self-attention modules, setting new records on machine translation at the cost only a fraction of previously state-of-the-art model's training time. More recently, BERT (Devlin et al., 2019) combined the idea of self-supervised pre-training with Transformer encoder architecture, breaking records on practically all natural language understanding (NLU) tasks (Wang et al., 2018), including Named Entity Recognition (NER), a sequence tagging task that is similar to our SLU formulation.

Therefore, we follow the history trend in language representation, and replace the LSTM in section 3.1 with a pre-trained language model. More specifically, we use Chinese RoBERTa with whole word masking (Cui et al., 2020). RoBERTa (Liu et al., 2019) is a variant of BERT pre-trained on a much larger corpus without Next Sentence Predic-

[1]The reason is: (1) To maintain the consistency between pre-training and training since in training, the input length is the same as the output length; (2) For convenience. If the input and output lengths could be different, then our tagging decoder would fail to reconstruct the input and a generative decoder would be required.

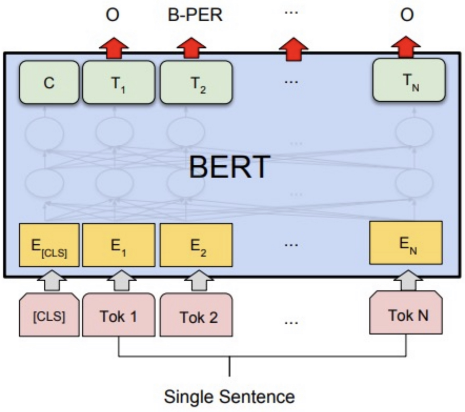


Figure 6: An illustration of sequence tagging using BERT or BERT-like Transformer encoders, adopted from Devlin et al. (2019).

tion (NSP) objective, and has demonstrated better performance on downstream NLU tasks. The modeling details of RoBERTa for sequence tagging, demonstrated in Figure 6, is essentially the same as LSTM, except that recurrent units are replaced by Transformer blocks and no extra word vectors are required (Chinese RoBERTa released by (Cui et al., 2020) has a vocabulary size of around 21 thousand).

However, recent works in the literature have also found that multilingual pre-training can improve the quality of language representation for low-resource languages (at the cost of lower performance on high-resource languages) compared with monolingual models, in the case of both NLU and MT (Conneau et al., 2020; Arivazhagan et al., 2019). Since Chinese is usually considered to be an intermediately sized language (Xue et al., 2021), we also train a model with XLM-R (Conneau et al., 2020), the multilingual counterpart of RoBERTa with a vocabulary of 250 thousand Sentence Piece (Kudo and Richardson, 2018) tokens. For the purpose of comparison, we train XLM-R using strictly the same set of hyper-parameters and Chinese RoBERTa.

For these pre-trained language models, we also apply the dual-channel decoder architecture described in section 3.2, but reduce the denosing decoder to only one forward layer so that we introduce only a minimal number of extra parameters on top of the models pre-trained by MLM.

3.4 Training Details

Our LSTM model is trained by Adam optimizer (Kingma and Ba, 2015) with learning rate 1×10^{-3} and batch size 32 for 20 epochs. The pre-trained language models, on the other hand, are optimzied by AdamW (Loshchilov and Hutter, 2019) with learning rate 1×10^{-5}, batch size 32, and also for 20 epochs. We use the base version of both RoBERTa and XLM-R with 12 Transformer layers. For our dual-channel decoder model, we pre-train for 10 epochs and then proceed with the same set of hyper-parameters as corresponding baselines. All the training is conducted on an RTX 3090, and takes only a matter of minutes.

Model	Dev Acc	Dev F_1
LSTM	71.40	77.35
RoBERTa	78.10	82.31
XLM-R	78.44	**82.89**
Denoising LSTM	73.85↑↑	78.93↑↑
Denoising RoBERTa	77.99↓	82.63↑
Denoising XLM-R	**78.66**↑	82.50↓

Table 1: Validation performance of sequence tagging models.

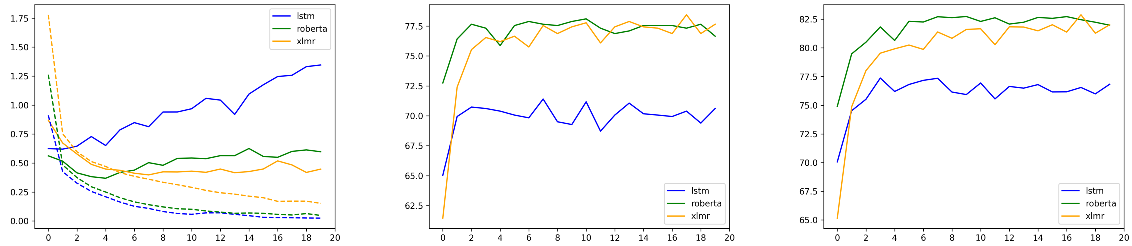
3.5 Results and Analysis

3.5.1 Vanilla Sequence Tagging

The training curves of vanilla LSTM, RoBERTa, and XLM-R are plotted in Figure 7, and their validation performance is recorded in Table 1. The best result is achieved by XLM-R, with 78.44 accuracy and a 82.89 F_1 score. LSTM, as expected, underperforms by 5-7 points when compared with the pre-trained models.

An intriguing observation from Figure 7 is that the training loss does not reflect the representation power of the models. While LSTM is obviously the least expressive model among the three, its training loss is the lowest, even though the three models' losses are calculated in the same label space. A similar phenomenon is that the validation loss seems to be uncorrelated with our evaluation metrics. The first subfigure clearly shows that LSTM and RoBERTa start to overfit on the training set after 2 and 5 epochs respectively, while XLM-R, being the most powerful model[2], is more robust to overfitting. The evaluation metrics of all three models, however, reach a plateau after several epochs of training and do not demonstrate any downward trend.

[2]XLM-R's Transformer layers are the same as RoBERTa, but it has a much larger embedding layer.



(a) Training (dashed) and validation loss. (b) Validation accuracy. (c) Validation F_1 score.

Figure 7: Training curves of LSTM, Chinese RoBERTa, and XLM-R.

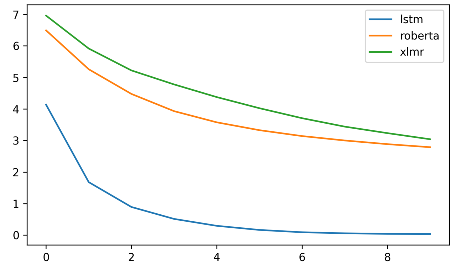


Figure 8: Reconstruction loss during pre-training.

3.5.2 Dual-channel Denoising Decoder

The experimental results of SLU models pre-trained with denoising decoder are also recorded in Table 1. With dual-decoder pre-training, the performance of LSTM baseline increases from 71.40/77.35 to 73.85/78.93, by 2.5 and 1.6 points respectively. RoBERTa and XLM-R, however, do not benefit from this pre-training scheme.

To explore the reasons behind these phenomena, we plot the denoising reconstruction loss of pre-training in Figure 8. All three curves in the figure are cross-entropy loss calculated on a vocabulary of 1928 tokens, constructed from the combined set of ASR transcriptions and manual corrections. The loss of LSTM is significantly lower than the pre-trained language models and close to zero, both indicating overfitting. We hypothesize that this is a direct result of the limited size of our dataset. RoBERTa and XLM-R, on the other hand, are orders of magnitude larger than the LSTM baseline, and thus require much more training data to fit on the task. Also, the knowledge learned during their self-supervised pre-training may prevent them from overfitting on a small amount of data. This is also corroborated by LSTM's lower training loss in Figure 7.

Another possible explanation for the pre-trained language models' insensitivity toward our denoising pre-training is that they have already acquired some denoising capability from MLM pre-training. Since the starting point of distributed representation (Mikolov et al., 2013) and contextual representation (Devlin et al., 2019) is to "represent a word by the companies it keeps", models thus trained should be able to adapt each token's representation to its context and implicitly correct the errors in ASR transcriptions to a certain extent.

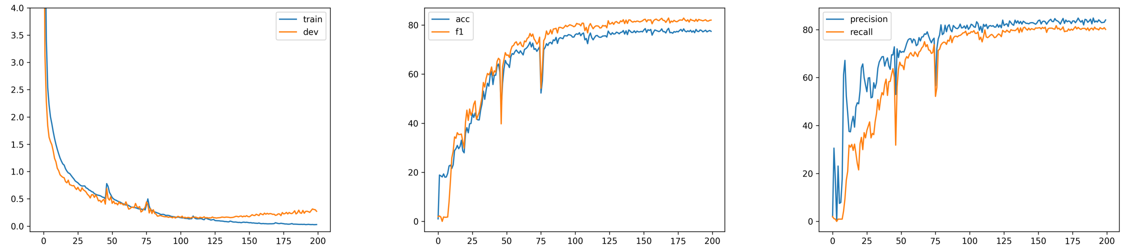
4 SLU with Sequence to Sequence Generation

4.1 Generative Sequence Tagging

While RoBERTa and other variants of BERT have pushed language model's performance in natural language understanding to a new height, an inherent shortcoming of these models is that they can only be used for discriminative tasks, but not for text generation. Another problem is the inconsistency between their pre-training and fine-tuning objectives. Fine-tuning these models is mostly achieved by registering a task-specific classification head, which not only introduces new parameters and does not utilize the model's MLM pre-training objective, but also renders models fine-tuned for different downstream tasks incompatible with each other.

In response to these issues, Raffel et al. (2020) proposed T5, a Transformer architecture that unifies all text-based tasks into a sequence to sequence framework, obviating the need of different fine-tuning schemes for each task and in most cases surpassing previously state-of-the-art models. Since sequence tagging, by its definition, is in the middle ground between classification and generation, we reformulate our SLU objective in the generative fashion.

Formally, the probability in Equation (1) is fac-



(a) Training and validation loss. (b) Validation accuracy and F_1 score. (c) Validation precision and recall.

Figure 9: Training curves of mT5.

torized it into a product of conditional probability of y_i given the input and all previous labels $y_{1:i-1}$:

$$p_\theta(y_{1:n}|\mathbf{x}_{1:n}) = \prod_{i=1}^{n} p_\theta(y_i|y_{1:i-1}, \mathbf{x}_{1:n}), \quad (2)$$

where each tag is conditioned explicitly on the previous tags (i.e. teacher forcing during training and autoregression during inference) as opposed to being conditioned only on the contextualized hidden state extracted by the encoder as in all models introduced in section 3.

4.2 Training Details

In practice, we add the 72 labels that start with `B` or `I` as special tokens into the model's vocabulary, and register a randomly initialized embedding vector for each of them in the model's output layer. We train the model with token-wise maximum likelihood on pairs of token lists where the input sequence (query text) and output sequence (labels) have the same length. At inference time, we set the maximum output length to the input length, and disable the generation of any tokens other than the special tokens (including `</s>`, `<pad>`, and the 72 labels) and `O` to ensure that the output is a legit tag sequence.

We use the base version of mT5 (Xue et al., 2021), with a vocabulary of 250 thousand tokens and 12 Transformer encoder, 12 Transformer decoder layers respectively. We optimize the model using AdamW (Loshchilov and Hutter, 2019), with learning rate 1×10^{-4}, batch size 32, for 200 epochs. The training process takes about two hours on an RTX 3090.

4.3 Resutls and Analysis

The validation loss and metrics of mT5 during training are plotted in Figure 9. In terms of loss, the model starts overfitting on the training set after about 125 epochs, but only mildly when compared with the encoder-only models in Figure 7, most likely due to its larger capacity. A more interesting observation is that recall on the validation set is notably lower than precision at the beginning of training, especially in the first ten epochs. This is probably a result of the gap between teacher forcing during training and auto-regressive generation during evaluation, which causes the model to be only able to correctly generate the first few tokens and to veer off without turning back after the first wrong prediction.

Model	Dev Acc	Dev F_1
XLM-R	78.44	82.89
Denoising XLM-R	78.66	82.50
mT5	78.66	82.85

Table 2: Validation performance of mT5, in comparison with XLM-R and denoising XLM-R.

The best checkpoint of mT5 performs on par with XLM-R, with 78.66 accuracy and 82.85 F_1 score, as shown in Table 2.

5 Conclusion and Discussion

In this work, we formulated SLU as a sequence tagging problem, and applied vanilla LSTM, RoBERTa, and XLM-R to it, obtaining progressively better performance. Based on these models, we introduced a dual-channel decoder model with denoising pre-training, observing more than two points' performance gain in LSTM but negligible impact on the pre-trained language models. We also formulated SLU as a tag sequence generation task, and trained an mT5, yielding results comparable with the best discriminative models.

For further researches on SLU, an intriguing direction is the utilization of manually corrected transcriptions in the training data in ways other than denoising pre-training. As these transcriptions are unavailable at test time, they can be viewed as privileged information in the training stage, and models such as hallucination network (Hoffman et al., 2016) may be applied to improve the performance of SLU systems with this information. And within the denoising pre-training framework, a natural extension of our dual-channel decoder is to adopt more data augmentation methods or self-supervised training techniques to enlarge the pre-training gain, like replacing spans of text with text of similar tones. Additionally, we can also specifically collect pairs of phrases or words that are prone to be recognized mistakenly by the ASR system to enrich the pre-training dataset and improve the domain adaptation effect of denoising pre-training.

Also, the inconsistency between validation loss and validation metrics in Figure 7 suggests that sequence tagging may not be the optimal solution for SLU, and other paradigms, especially sequence to sequence generation, may be worth more attention. While we have tried using mT5 for SLU in this work, it is used to generate the tagging sequence and thus still falls into the sequence labeling framework. We do believe that the end-to-end approach of letting the model directly learn to map from query text to semantic concepts is worth researching in the future.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Giuseppe Castellucci, Valentina Bellomaria, Andrea Favalli, and Raniero Romagnoli. 2019. Multi-lingual intent detection and slot filling in a joint bert-based model. *CoRR*, abs/1907.02884.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. BERT for joint intent classification and slot filling. *CoRR*, abs/1902.10909.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Judy Hoffman, Saurabh Gupta, and Trevor Darrell. 2016. Learning with side information through modality hallucination. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 826–834. IEEE Computer Society.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium,* *October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Libo Qin, Tianbao Xie, Wanxiang Che, and Ting Liu. 2021. A survey on spoken language understanding: Recent advances and new frontiers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4577–4584. ijcai.org.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Suman V. Ravuri and Andreas Stolcke. 2015. Recurrent neural network and LSTM models for lexical utterance classification. In *INTERSPEECH 2015, 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015*, pages 135–139. ISCA.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pages 353–355. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.

Kaisheng Yao, Baolin Peng, Yu Zhang, Dong Yu, Geoffrey Zweig, and Yangyang Shi. 2014. Spoken language understanding using long short-term memory neural networks. In *2014 IEEE Spoken Language Technology Workshop, SLT 2014, South Lake Tahoe, NV, USA, December 7-10, 2014*, pages 189–194. IEEE.