A Two-Stage Graph-Based Method for Chinese AMR Parsing with Explicit Word Alignment

Liang Chen, Bofei Gao, Baobao Chang
Key Laboratory of Computational Linguistics, Peking University, MOE, China
leo.liang.chen@outlook.com
gaobofei@163.com
chbb@pku.edu.cn

Abstract

In this report, we provide a detailed description of our system at CAMRP-2022 evaluation. We firstly propose a two-stage method to conduct Chinese AMR Parsing with alignment generation, which includes Concept-Prediction and Relation-Prediction stages. Our model achieves 0.7756 and 0.7074 Align-Smatch F1 scores on the CAMR 2.0 test set and the blind-test set of CAMRP-2022 individually. We also analyze the result and the limitation such as the error propagation and class imbalance problem we conclude in the current method. Code and the trained models are released at https://github.com/PKUnlp-icler/Two-Stage-CAMRP for reproduction.

1 Introduction

Abstract Meaning Representation (AMR) (Banarescu et al., 2013) parsing targets to transform a sentence into a directed acyclic graph, which represents the relations among different concepts. The original AMR does not provide concept-to-word alignment information, which hinders the trace-back from concept to input word and brings difficulties to AMR parsing. To solve the problem, based on Chinese AMR (Li et al., 2016) , Li et al. (2019) further propose to add concept and relation alignment to the structure of Chinese AMR.

Currently, while a majority of work is focusing on improving the performance of English AMR Parsing (Xu et al., 2020; Bevilacqua et al., 2021; Wang et al., 2022; Bai et al., 2022; Chen et al., 2022), those methods or models can not be directly applied to Chinese AMR Parsing since English AMR does not provide alignment information itself. To better reflect the full structure of Chinese AMR, CAMRP-2022 evaluation[1] firstly requires the AMR parser to generate explicit word alignment including concept and relation alignment which calls for novel models and algorithms.

We propose a two-stage method to conduct Chinese AMR Parsing with alignment generation[2]. In a nutshell, the method includes the Concept-Prediction and Relation-Prediction stages, which can be regarded as the process of graph formation. In the Concept-Prediction stage, we develop a hierarchical sequence tagging framework to deal with the concept generation and the complex multi-type concept alignment problem. In the Relation-Prediction stage, we utilize the biaffine network to predict relations and Relation-alignment simultaneously among predicted concepts. Our model ranks 2nd in the closed-track of the evaluation, achieving 0.7756 and 0.7074 Align-Smatch (Xiao et al., 2022) F1 scores on the CAMR 2.0 test set and the blind-test set of CAMRP-2022 individually.

2 Method

In this section, we will introduce our two-stage Chinese AMR parsing method, which includes the Concept-Prediction stage and Relation-Prediction stage.

2.1 Concept-Prediction

Different from English AMR where nodes or concepts can have arbitrary variable names, a large portion of concepts of Chinese AMR have standard variable names which denote the alignment to input words.

[1] https://github.com/GoThereGit/Chinese-AMR

[2] We participate in the closed-track evaluation where we can only use HIT-roberta(Cui et al., 2020) as the pretrained language model

Moreover, there are different alignment rules which make generating the right alignment a complex problem. We'll first introduce the alignment rules we conclude in CAMR 2.0 and our method and model to handle the problem.

2.1.1 Multi-Type Concept Alignment Rule

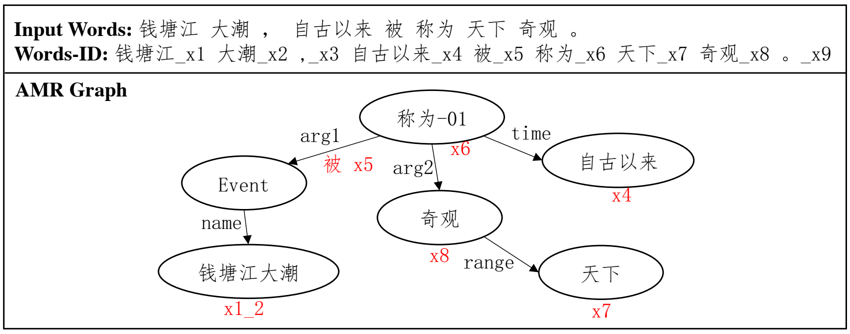
We mainly summarize 6 different alignment rules for concepts, which are ***Direct Alignment***, ***Normalization Alignment***, ***Continued Multi-word Alignment***, ***Uncontinued Multi-word Alignment***, ***Split Alignment*** and ***Non-Aligned Concepts***. The difference among alignment rules lies in how one abstract concept corresponds to the input words. We pick three cases as shown in Figure 2 as examples.

Direct Alignment is the easiest alignment where a concept directly corresponds to a certain word in the input without the need for any modification. **Normalization Alignment** exists when a concept still corresponds to one word in the input however needs to be "normalized" into the final concept. The normalization includes different situations like word sense disambiguation for predicate and Arabic numerals transformation for numerals in other languages. For example, as shown in Figure 2, in case (a) the word "称为" corresponds to the concept "称为-01" after word sense disambiguation. In case (c), Chinese numeral "一" would be mapped to concept "1" since all numeral concepts in CAMR are Arabic. **Continued Multi-word Alignment** exists when multiple continued words in the input sentence are concatenated into the final concept, which usually happens for named entities. **Uncontinued Multi-word Alignment** means multiple discontinued words in the input sentence are joined into the final concept or preposition patterns like "在...上" . **Split Alignment** denotes one word that could correspond to multiple concepts, which usually suggests the word corresponds to a sub-graph in the final AMR graph. At last, there are also **Non-Aligned Concepts** that do not have alignment and could have arbitrary variable names. These concepts usually abstract away from syntactic features, making them harder for the model to predict. In fact, according to our experiment, our system could reach a 0.91 f1 score for with-alignment concepts' prediction but only a 0.70 f1 score for Non-Aligned concepts' prediction.

As shown in Figure 3, we further collect more statistics about the alignment between concepts and input words with the training set of the CAMR 2.0 dataset. From the perspective of input sentences, about 75% of words in the input sentences are associated with certain concepts under one alignment rule. From the perspective of concepts, there are 83% of concepts with alignment. For all concepts with alignment, a majority of them belong to Direct(56%) and Normalization(33%) Alignments.

2.1.2 Hierarchical Sequence Tagging Framework

In spite of the complex word alignment rules, we can see that a large portion of concepts are directly or indirectly aligned with a single word of input and one word can only correspond to one concept at most, which inspires us to adopt sequence tagging method. It can deal with concept prediction and Direct Alignment prediction simultaneously. Considering different alignment rules, we develop three sequence tagging rules to cover all possible situations.



Input Words: 钱塘江 大潮 ， 自古以来 被 称为 天下 奇观 。
Words-ID: 钱塘江_x1 大潮_x2 ，_x3 自古以来_x4 被_x5 称为_x6 天下_x7 奇观_x8 。_x9

Figure 1: An example of Chinese AMR. Red word-IDs under concepts denote concept alignment. Red words and word-IDs under relation denote relation alignment.

(a)

Input Words: 钱塘江 大潮 ， 自古以来 被 称为 天下 奇观 。
Words-ID: 钱塘江_x1 大潮_x2 ，_x3 自古以来_x4 被_x5 称为_x6 天下_x7 奇观_x8 。_x9

Gold Concepts: 钱塘江大潮，自古以来，称为-01，天下， 奇观， event
Direct Alignment: 自古以来:x4，天下:x7，奇观:x8
Normalization Alignment: 称为-01: x6
Continued Multi-word Alignment: 钱塘江大潮: x1_x2
Non-Aligned Concept: event

(b)

Input: 共计 有 30余 份 。
Words-ID: 共计_x1 有_x2 30余_x3 份_x4 。_x5

Gold Concepts: 共计-01，有，30，余，份，thing
Direct Alignment: 有:x2，份:x4
Normalization Alignment: 共计-01: x1
Split Alignment: 30:x3_1_2，余:x3_3
Non-Aligned Concept: thing

(c)

Input: 在 这 一 凭证 上 。
Words-ID: 在_x1 这_x2 一_x3 凭证_x4 上_x5 。_x6

Gold Concepts: 在...上，这，1，凭证
Direct Alignment: 这:x2，凭证:x3
Normalization Alignment: 1: x_3
Discontinued Multi-word Alignment: 在...上: x1_x5

Figure 2: Example of different Concept Alignment cases. Gold concepts denote all concepts in the gold AMR graph of the input sentence. The concepts can be divided into different categories according to the alignment rules. We use words in color to represent the unique alignments in each example.

Model Structure As depicted in Figure 4, we add a linear layer on the top of the Chinese RoBERTa model as a Tag Classifier. Adapting to character-based Chinese pretrained language model, Tag classification is conducted on the first character's hidden state for a word with multiple characters.

Surface Tagging We design an 8-classes BIO tagging rule as the first step to process the input sentence. The eight classes are O, B-Single, B-Continued-Multiword, I-Continued-Multiword, B-Discontinued-Multiword, I-Discontinued-Multiword, B-Split, and B-Virtual. This tagging rule can cover 4 out of 6 alignment rules, which are Direct Alignment, Continued Multi-word Alignment, Discontinued Multi-word Alignment, and Split Alignment. Note that the B-Single tag is for both Direct Alignment and Normalization Alignment because they both correspond to one input word. As for B-Split, we use manually curated rules to split the word with Split Alignment. Note that B-Virtual is also added to label the virtual word for the later relation classification task. The F1-score of the Surface Tagging step can reach 91% on the development set in our experiment.

Normalization Alignment Tagging Previous Surface Tagging can not recognize words that need normalization like word sense disambiguation so we introduce a 2-class Tagging rule to identify whether a word from the input sentence needs normalization before becoming a concept in the AMR graph. The labels can be collected directly from the gold AMR graph. If one concept is aligned to one identical word from the input sentence, then the word's label is negative. If the concept is aligned to a word different from itself, then the word's label is positive. The F1-score of Normalization Alignment Tagging can reach 0.95% on the development set. After recognizing words needing normalization, we run a statistical

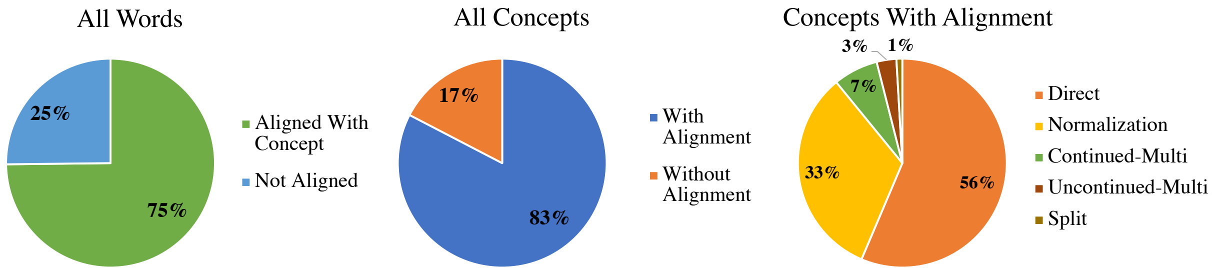
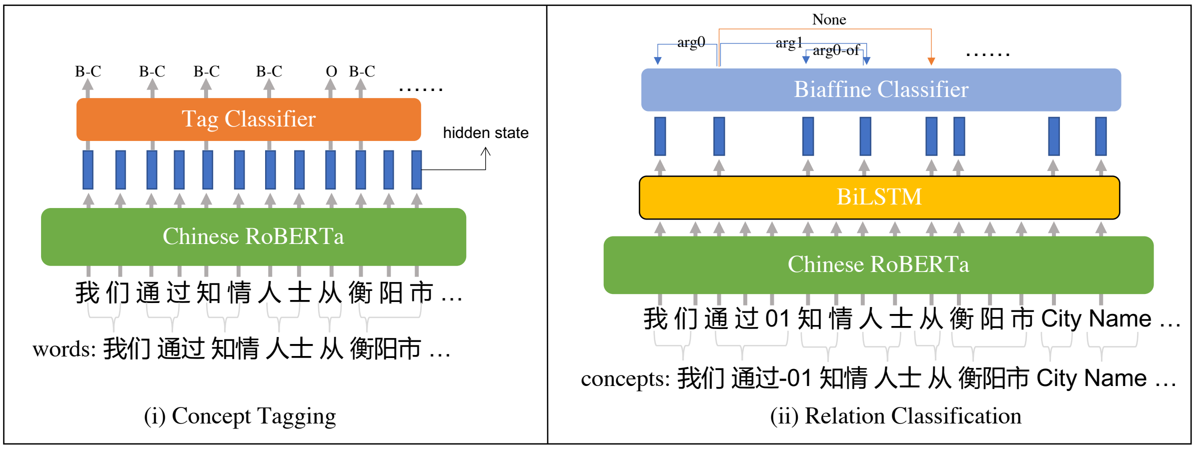


Figure 3: Statistics about the alignment between concepts and input words in CAMR 2.0.


B-C
B-C
B-C
B-C
O
B-C
......
Tag Classifier
hidden state
Chinese RoBERTa
我 们 通 过 知 情 人 士 从 衡 阳 市 …
words: 我们 通过 知情 人士 从 衡阳市 …
None
arg0
arg1
arg0-of
......
Biaffine Classifier
BiLSTM
Chinese RoBERTa
我 们 通 过 01 知 情 人 士 从 衡 阳 市 City Name …
concepts: 我们 通过-01 知情 人士 从 衡阳市 City Name …

(i) Concept Tagging

(ii) Relation Classification

Figure 4: The Two-Stage Parsing Model. We use the same Concept Tagging model structure to conduct three hierarchical sequence tagging tasks, each with a different Tag Classifier. Relation Classification takes concepts as input and the Biaffine Classifier outputs the relation between every two concepts. In both models, words or concepts are first splitted into characters before feeding into the pretrained language model and we use the hidden state of the first character to represent the word in the last classifier layers.

normalization method as described in Appendix A. This step can cover and predict the Normalization Alignment.

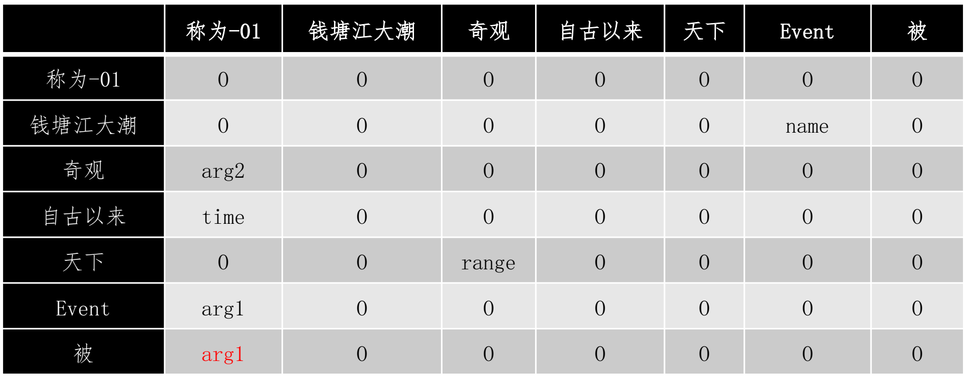
Non-Aligned Concept Tagging For concepts that do not have alignment with input words, we define trigger words for those concepts and also use sequence tagging method. To be more specific, we first collect the dictionary of all Non-Aligned concepts in the training set and there are 184 at all. The label of the input word is the class of concept it triggers, or “None” if it triggers nothing.

For all Non-Aligned concepts, we define the concepts that it has a direct relation to as its trigger concepts and the aligned word of the trigger concept as the trigger word. For example, as shown in Figure 1, the Non-Aligned concept “Event” has direct relation to concept “钱塘江大潮”. According to the alignment information of the concept, the trigger words of the concept “Event” are x1 and x2. Since a Non-Aligned concept could have multiple concepts it has a direct relation to, we tried using the first and the last of the concepts. The experimental result shows that using the last concept is more effective with a 0.03 F1 improvement.

There are nearly 5% cases where the trigger concepts are all Non-Aligned concepts. Under such circumstances, we keep tracing back from the trigger concept until we reach the first concept with alignment and we regard this concept as the trigger concept.

2.2 Relation-Prediction

As shown in Figure 4, we design a RoBERTa-BiLSTM-Biaffine network to conduct relation prediction given the predicted concepts. All concepts are first split into characters before feeding into the RoBERTa model to extract hidden representations. After the RoBERTa model, all hidden states are fed into a one-layer BiLSTM network to better encode sequential information to the hidden states. At last, the **first hidden states of every two concepts** are fed into the biaffine network to get the relation between the two concepts. During training, we use Cross-Entropy as the loss function and use the average loss of $N \times N$ relations as the final loss, as described in Equation 1,



	称为-01	钱塘江大潮	奇观	自古以来	天下	Event	被
称为-01	0	0	0	0	0	0	0
钱塘江大潮	0	0	0	0	0	name	0
奇观	arg2	0	0	0	0	0	0
自古以来	time	0	0	0	0	0	0
天下	0	0	range	0	0	0	0
Event	arg1	0	0	0	0	0	0
被	arg1	0	0	0	0	0	0

Figure 5: An example of Relation Classification label matrix. The inputs are from the gold concepts of sentence “钱塘江大潮被称为天下奇观”. Each column denotes the start node of a relation while each row denotes end node of a relation. “O” denotes there is no relation between the two nodes. Relation in red denotes the relation alignment for functional words.

$$
\begin{aligned}
\text{Biaffine}(\mathbf{a},\mathbf{b}) &= [\mathbf{a};\mathbf{1}]\mathbf{W}[\mathbf{b};\mathbf{1}]^T, (\mathbf{a} \in \mathbb{R}^{1\times d}, \mathbf{b} \in \mathbb{R}^{1\times d}, \mathbf{W} \in \mathbb{R}^{(d+1)\times c\times(d+1)}) \\
\text{Loss} &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} CrossEntropy(\text{Biaffine}(\mathbf{h_i},\mathbf{h_j}), \hat{r}(\mathbf{h_i},\mathbf{h_j})) \qquad (1) \\
\text{Relation}_{a,b} &= \arg\max \text{Biaffine}(\mathbf{a},\mathbf{b})
\end{aligned}
$$

where d denotes the hidden size, c denotes the number of relations, N denotes the number of input concepts and $\hat{r}$ denotes the gold relation.

Relation-Alignment Prediction On top of relation between concepts, another important feature of Chinese AMR is the relation alignment, which takes Chinese functional words’ semantics into consideration in AMR graph. For example, as shown in Figure 1, functional word “被” is aligned to the “arg1” relation between concepts “称为-01” and “Event”. In fact, in the input Chinese sentence, the word “被” implies “钱塘江大潮” or the “Event” is the object of concept “称为-01”.

We use the same model as relation prediction to align functional words with relations. To be more specific, concepts and functional words are both fed into the RoBERTa-BiLSTM-Biaffine network. For any relation triples(concept1, concept2, relation), if the relation is aligned with functional word w, we create another triple(concept1, w, relation) for the model to predict. In this way, we can predict the relation and relation alignment simultaneously. After predicting all relations, if one concept is linked with two different concepts[3] with the same relation, then the functional word in the two concepts will be aligned to the relation. Here we first collect all functional words in the training set as the dictionary to identify whether a word is a functional word.

2.3 Teacher Forcing in Training

Since our method has two stages, during inference the Relation-Prediction model takes the output of Concept-Prediction as input. To stabilize and prevent error propagation during training, we adopt the Teacher Forcing method, where we use the gold concepts and relations as the input of the relation prediction model. However, error propagation still exists in the inference phase. We will discuss the error propagation situation of our system in section 4.1.

Dataset Split	Sentences	Tokens
Train	16576	386234
Development	1789	41822
Test	1713	39228
Blind Test	1999	36940

Table 1: The dataset description of CAMRP-2022. Note that the train, development and test splits are directly from CAMR 2.0 dataset while the blind test split is from this evaluation.

3 Experiment

3.1 Dataset

The CAMRP-2022 evaluation uses the training, development, and test splits of the CAMR 2.0 dataset as its dataset and also involves an out-of-domain blind test set to measure the generalization performance of parsers. The statistics of the dataset are shown in Table 4. We further collect some detailed figures, that there are 31941 different concepts in the training set while only 11293 concepts appear more than 5 times. Among all concepts, there are 8443 different predicates that need to conduct word sense disambiguation and 185 Non-Aligned concepts that are never aligned with input words. As for relations, there are 142 different relations and 841 relation alignment words. The top 5 most frequent relation alignment words are “的”,“是”,“和”,“在” and “对”.

3.2 Model

Both the Concept Tagging and Relation Classification model adopt the HIT-roberta-large(Cui et al., 2020) pretrained model downloaded from HuggingFace model hub[4]. For Concept Tagging models, the output size of the tag classifier is 8, 2, 185 for Surface Tagging, Normalization Alignment Tagging, and Non-Aligned Concept Tagging individually and the dropout rate of the classifiers is 0.1 in all experiments. For the Relation Classification model, there is one BiLSTM layer and the hidden size is 4096. the dimension of Biaffine matrix is $4097 \times 142 \times 4097$.

3.3 Training Details

We use Adam as the optimizer and conduct hyper-parameter searches on batch-size (from 10 to 100) and learning rate (from 1e-5 to 1e-4) in all models. The optimal hyper-parameters for each model are listed in Appendix B. We train all models for 100 epochs with 1% warmup steps and select the one with the best result on the development set as the final model.

3.4 Results

As shown in Table 2, we list the results of our trained 2-stage AMR Parser on the development, test, and blind-test set of CAMRP-2022. For the development set, we list the concrete results of all different sub-tasks in two stages along with the overall and fine-grained AlignSmatch scores.

Sub-task Results For the three sub-tasks of the Concept-Prediction stage, we can tell from Table 2 that our model performs better in the Surface Tagging task with a 0.931 F1 score and Normalization Alignment Tagging task (0.878 F1) than in Non-Aligned Concept Tagging task (0.693 F1). It suggests that the model can better recognize concepts with alignment and there is a big performance drop when predicting concepts without alignment under the same sequence tagging framework. For the Relation Classification task, our model can reach 0.744 F1 when given gold concepts while only 0.583 F1 in inference when the concepts are generated by the Concept-Prediction stage instead of gold concepts. It reveals a train-inference discrepancy existing in the current method since the model might generate wrong concepts during the Concept Prediction stage in inference which would bias the Relation Prediction stage.

[3] Precisely, one is concept and the other is functional word.

[4] https://huggingface.co/hfl/chinese-roberta-wwm-ext-large

Task(Dev)	Precision	Recall	F1
Surface Tagging	0.918	0.944	0.931
Normalization Aligment Tagging	0.878	0.878	0.878
Non-Aligned Concept Tagging	0.708	0.679	0.693
Relation Classification (With Gold Concepts)	0.751	0.737	0.744
AlignSmatch	0.778	0.766	0.768
- Only Instance	0.830	0.833	0.832
- Only Attribute	0.928	0.954	0.941
- Only Relation	0.614	0.556	0.583

Task(Test)	Precision	Recall	F1
AlignSmatch	0.786	0.765	0.776
- Only Instance	0.834	0.840	0.837
- Only Attribute	0.932	0.959	0.945
- Only Relation	0.628	0.570	0.598

Task(Blind Test)	Precision	Recall	F1
AlignSmatch	0.715	0.696	0.705
- Only Instance	0.768	0.775	0.772
- Only Attribute	0.866	0.901	0.883
- Only Relation	0.549	0.492	0.519

Table 2: The fine-grained results of our model in CAMRP-2022. We report the overall and fine-grained AlignSmatch scores of our model on the development, test and blind test sets. We also report the results of each sub-task in the two-stage method on the development set.

AlignSmatch Results As for the overall AlignSmatch scores, we can tell from the results of three dataset splits that there exists a domain shift comparing the development and test set of CAMR 2.0 to the blind test set. When looking at the fine-grained scores, the trend is consistent among three splits that the performance of attribute or alignment prediction is better than instance prediction and far better than relation prediction. The trend indicates that the model generally outperforms in the first stage than in the second stage.

Moreover, for relation prediction, we can see that the recall is about 5 points lower than the precision in all experiments and the gap is much bigger than instance or attribute prediction. The reason is that in the relation classification model there exists a performance gap in relation prediction and relation alignment prediction. Compared to relations, a lot more relation alignments are not predicted while the relation-only score in AlignSmatch takes both relation and relation alignment into account, which makes the recall score lower. In fact, if we preclude relation alignment prediction in the relation-only score, the gap between precision and recall will be reduced to 2 points. It hints to us that we need to pay more attention to the relation alignment prediction to improve the overall performance.

4 Discussion

In this section, we summarize some problems that need to be addressed to improve the performance of the Chinese AMR parser.

4.1 Error Propagation in the Two-Stage Model

As pointed out in Section 3.4, there exists error propagation in the two-stage model. The direct evidence is that while the relation prediction could reach 0.744 F1 with gold concepts, this score drops to 0.583 when giving it the model predicted concepts. Error propagation also exists in the Concept-Prediction stage since Normalization Alignment needs both the correct result from Surface Tagging and Normalization Alignment Tagging.

4.2 Class Imbalance Problem

As pointed out in section 3.1, there exist severe class imbalance problems in both stages of the parsing task. As for the Concept-Prediction stage, the problem reflects the great differences in the distribution of different tags in the three tagging tasks, especially for the Non-Aligned Concept Tagging tasks. For the Relation-Prediction stage, a large portion of labels is "None Relation" as shown in Figure 5.

We have tried some techniques like using weighted loss that assigns greater weight to the minority classes to handle the class imbalance problem. While this can greatly reduce the time required for the model to converge, it does not improve the final performance when all epochs are finished.

4.3 Improving the Non-Aligned Concept Prediction Performance

In our model, we use a trigger-based method to predict concepts without alignment. This method could cover nearly 95% cases while the rest 5% is neglected because they are mostly triggered by another Non-Aligned concept. Though we design methods to overcome the drawback by tracing back to the first aligned concept, the overall result of Non-Aligned Concept Prediction is still the lowest in the Concept-Prediction stage, which could lead to great bias for the next stage. A more natural method to predict those concepts might greatly improve this task.

5 Conclusion

In this report, we provide a detailed description of the proposed two-stage Chinese AMR Parsing model which is the first to deal with the explicit word alignment problem for CAMRP-2022 evaluation. We also analyze the result and point out the limitation of the current method and some potential roads that might lead to improvement. Though straightforward, the method is far from perfect that it still calls for future exploration to reach a better result in the Chinese AMR Parsing task.

References

Xuefeng Bai, Yulong Chen, and Yue Zhang. 2022. Graph pre-training for AMR parsing and generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6001–6015, Dublin, Ireland, May. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*.

Liang Chen, Peiyi Wang, Runxin Xu, Tianyu Liu, Zhifang Sui, and Baobao Chang. 2022. ATP: AMRize then parse! enhancing AMR parsing with PseudoAMRs. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2482–2496, Seattle, United States, July. 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, November. Association for Computational Linguistics.

Bin Li, Yuan Wen, Weiguang Qu, Lijun Bu, and Nianwen Xue. 2016. Annotating the little prince with Chinese AMRs. In *Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016)*, pages 7–15, Berlin, Germany, August. Association for Computational Linguistics.

Bin Li, Yuan Wen, Li Song, Weiguang Qu, and Nianwen Xue. 2019. Building a chinese amr bank with concept and relation alignments. 18, Jun.

Peiyi Wang, Liang Chen, Tianyu Liu, Damai Dai, Yunbo Cao, Baobao Chang, and Zhifang Sui. 2022. Hierarchical curriculum learning for amr parsing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Liming Xiao, Bin Li, Zhixing Xu, Kairui Huo, Minxuan Feng, Junsheng Zhou, and Weiguang Qu. 2022. Align-smatch: A novel evaluation method for chinese abstract meaning representation parsing based on alignment of concept and relation. In *Proceedings of the Language Resources and Evaluation Conference*, pages 5938–5945, Marseille, France, June. European Language Resources Association.

Dongqin Xu, Junhui Li, Muhua Zhu, Min Zhang, and Guodong Zhou. 2020. Improving amr parsing with sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2501–2511.

Appendix A: Statistical Word Normalization Method

Case Description	Examples
Word Sense Disambiguation	合作: 合作-01, 成: 成-01, 大: 大-01
Special Concept Transform	不: - , 在: be-located-at-91
Number Normalization	第一: 1, 一万: 10000
Error Correction	JY: 精英, 暴光: 曝光-01, 惊荒: 惊慌-01

Table 3: Main cases that need word normalization.

As shown in Table 3, we mainly summarize 4 cases that require word normalization. The Word Sense Disambiguation case denotes that the concept is required to select a word sense in terms of semantic. In Table 3, ”合作” is interpreted as the No.1 meaning in the dictionary, so ”合作” should become ”合作-01”. The Special Concept Transform case is that many concepts with similar semantics in the corpus are replaced by special token. For example, ”不” and ”否” are replaced by ”-”. The Number Normalization case is that all concepts containing numbers need to be converted to Arabic numerals. The Error Correction case denotes that there may be errors in the sentence and we need to correct it according to the dictionary.

We design a set of word sense disambiguation rules to realize word normalization. As listed in Table 3, there are 4 categories of concepts that need to be normalized. For the 3rd case, namely, concepts that need to be converted into numbers, we extract the numbers in a sentence using regular expressions and then convert the numbers into Arabic numerals. For the 4th case, we retrieve the most similar word in the dictionary which are provided based on the phonological and calligraphical code of Chinese characters to replace the wrong one. For the other cases, we directly use the concept that appears most frequently in the training set.

When we simply use the concepts that appear most frequently in the training set for all cases, the accuracy on the development set is 85.1. When we incorporated the rules mentioned above, the accuracy increase to 90.6, proving that the rules we designed can effectively handle a number of normalized concepts.

Appendix B: Optimal Hyper-Parameters for different models

Model	Batch Size	Learning Rate
Surface Tagging	10	2e-5
Normalization Tagging	40	3e-5
Non-Aligned Concept Tagging	30	3e-5
Relation Classification	50	7e-5

Table 4: The optimal hyper-parameters in each model.