

# CroAno : A Crowd Annotation Platform for Improving Label Consistency of Chinese NER Dataset

Baoli Zhang<sup>\*1</sup>, Zhucong Li<sup>\*1,2</sup>, Zhen Gan<sup>3</sup>, Yubo Chen<sup>1,2</sup>, Jing Wan<sup>3</sup>, Kang Liu<sup>1,2</sup>, Jun Zhao<sup>1,2</sup>, Shengping Liu<sup>4</sup>, Yafei Shi<sup>4</sup>

<sup>1</sup> National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

<sup>2</sup> School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China

<sup>3</sup> Beijing University of Chemical Technology, Beijing, China

<sup>4</sup> Unisound Ai Technology Co.,Ltd, Beijing, China

{baoli.zhang, zhucong.li, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn, {ganzhen,wanj}@mail.buct.edu.cn, {liushengping, shiyafei}@unisound.com

## Abstract

In this paper, we introduce CroAno, a web-based crowd annotation platform for the Chinese named entity recognition (NER). Besides some basic features for crowd annotation like fast tagging and data management, CroAno provides a systematic solution for improving label consistency of Chinese NER dataset. **1) Disagreement Adjudicator:** CroAno uses a multi-dimensional highlight mode to visualize instance-level inconsistent entities and makes the revision process user-friendly. **2) Inconsistency Detector:** CroAno employs a detector to locate corpus-level label inconsistency and provides users an interface to correct inconsistent entities in batches. **3) Prediction Error Analyzer:** We deconstruct the entity prediction error of the model to six fine-grained entity error types. Users can employ this error system to detect corpus-level inconsistency from a model perspective. To validate the effectiveness of our platform, we use CroAno to revise two public datasets. In the two revised datasets, we get an improvement of +1.96% and +2.57% F1 respectively in model performance.

## 1 Introduction

Named entity recognition (NER), the task of detecting and classifying named entities in texts, has made significant progress relying on data-driven methods (Lample et al., 2016). Existing supervised approaches to NER require massive high-quality annotated data. Since hiring annotation experts is costly and time-consuming, crowd annotation and non-experts annotators are more generally used, the drawback is a higher proportion of inconsistency in the annotation.

<sup>\*</sup> co-first authors, they contributed equally to this work.

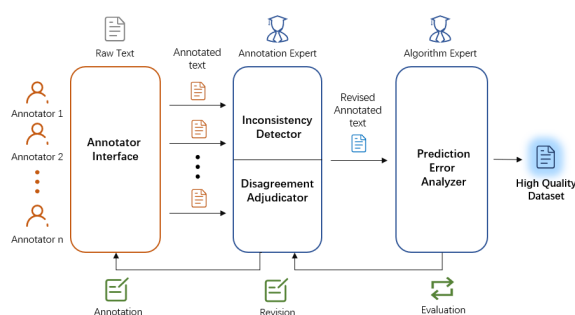


Figure 1: The whole architecture of CroAno.

However, existing NER annotation tools for crowd annotation (Ogren Philip, 2006; Chen and Styler, 2013; Manning et al., 2014; Samih et al., 2016) mainly aim to improve annotating efficiency and rarely consider the dataset’s consistency. Label inconsistency is ubiquitous in NER datasets. For example, *OntoNotes 4.0* (Weischedel et al., 2011), which is a classical Chinese NER benchmark, the proportion of label inconsistency is up to 10% according to our estimation. In this dataset, the mention of “中国人民” (Chinese People) appears 36 times, of which the whole mention is marked as entity 23 times, and “中国” (China) is separately labeled as entity 13 times. Such inconsistency may confuse the NER model and cause disastrous results.

In this paper, we propose a web-based **crowd annotation platform** named CroAno. As shown in Figure 1, CroAno contains three modules to improve label consistency of the Chinese NER dataset.

**Disagreement Adjudicator:** Crowd annotation tools usually distribute the same instance to different annotators, which would cause disagreement. We call this phenomenon instance-level label incon-



Figure 2: The screenshot of Annotator Interface.

sistency, because this inconsistency occurs in the same instance. Instance-level label inconsistency is easy to locate but difficult to display and correct. YEDDA (Yang et al., 2018) employs a comparison report to show these inconsistencies to annotation experts. But it failed to display detailed information of inconsistent entities, and annotation experts cannot directly correct these entities through the comparison report. To solve these display and correction issues, CroAno uses a multi-dimensional display mode to show inconsistent instances and employs the "click to correct" method to facilitate correction.

**Inconsistency Detector:** This module is designed to solve corpus-level label inconsistency, especially corpus-level pre&suffix inconsistency. Corpus-level pre&suffix inconsistency refers to the inconsistency of whether a descriptive string is included in the entity string. In *OntoNotes4.0*, the string “超过”(over) is included as a prefix of *Money* entity in some instances while in other instances the string is excluded as an external prefix. Locating these inconsistent entities is hard because a global perspective is required. CroAno uses a detector to locate potential inconsistency and provides an interface to support users correct these inconsistent entities in batches.

**Prediction Error Analyzer:** As mentioned above, disagreement adjudicator and inconsistency detector are responsible to solve label inconsistency from a data perspective, while prediction error analyzer can detect label inconsistency from a model perspective. To analyze inconsistency, we deconstruct entity prediction error of the model to a novel error system, which gives entity prediction error more abundant information with six fine-grained error types. CroAno provides a search API, which supports users to employ the error system to locate specific entities and their context. Moreover, CroAno employs an elaborately designed interface to differentiate the model prediction and the annotation.

In summary, the contributions of this paper are



Figure 3: The screenshot of Disagreement Adjudicator.

as follows:

- We propose a crowd annotation platform, which can promote label consistency of the Chinese NER dataset. The site can be accessed by <http://116.62.20.198:3000>, and instruction video is provided at <https://www.youtube.com/watch?v=wt2ma9FU540>. To the best of our knowledge, CroAno is the first crowd annotation platform that aims at promoting label consistency of NER dataset.
- We introduce three novel modules to promote label consistency of the NER dataset. Disagreement Adjudicator provides a multi-dimensional highlight mode to visualize and an interface function to revise instance-level inconsistency. Inconsistency Detector employs locating and revising strategies to correct corpus-level label inconsistency. Prediction Error Analyzer can help algorithm experts detect corpus-level inconsistency from a model perspective.
- To validate the effectiveness of our platform, we use CroAno to revise two public datasets. In the two revised datasets, we get an improvement of +1.96% and +2.57% in F1 respectively. It is worth mentioning that the promotion of data can be inherited by any NER model.

## 2 User Roles in CroAno

CroAno defines three roles: the crowd annotator, the annotation expert and the algorithm expert. The crowd annotator completes basic annotation tasks. The annotation expert is responsible for guideline formulation, data management, task distribution, and label consistency optimization. The algorithm expert is responsible for providing analyses and



Figure 4: The screenshot of Inconsistency Detector, We provide English translations for inconsistent entities.

suggestions of the annotation result from the perspective of the model.

The crowd annotator uses Annotator Interface for annotation. After preliminary annotation, results are sent to the annotation expert.

The annotation expert uses Disagreement Adjudicator and Inconsistency Detector for revision. Specifically, the annotation expert uses Disagreement Adjudicator to modify instance-level inconsistency from multi-annotators, and then uses Inconsistency Detector to detect and modify corpus-level inconsistency. After that, the promoted dataset is sent to the algorithm expert.

The algorithm expert uses Prediction Error Analyzer for evaluation. The algorithm expert can use Prediction Error Analyzer to detect the difference between model prediction and annotators annotation to get promotion direction of annotation or just revise the annotation.

The following sections will introduce the main modules of CroAno.

### 3 Annotator Interface

This section describes the interface, which is designed for easily annotating. As shown in Figure 2, the interface uses different colors to distinguish the entity categories. Annotators can locate an entity and its span with the left mouse button. And They can choose an entity category by short-cut key or click the corresponding button in the entity label bar.

Except for basic annotating function, they can mark the current instance as "annotated" by clicking the first top-left button, filter "annotated instances" by clicking the second top-left button, and get the guideline by clicking the last top-left button.

## 4 Disagreement Adjudicator

This section describes Disagreement Adjudicator, which is designed for solving instance-level label inconsistency. Instance-level inconsistency means disagreement annotations between different annotators within the same instance. Apart from annotation errors, different understandings of the guideline can also lead to disagreement. This type of inconsistency needs to be revised by the annotation expert. Difficulties in implementing this operation are to visualize disagreement entities and revise.

CroAno uses a novel multi-dimensional highlight mode to visualize disagreement entities. As shown in Figure 3, underlines of text represent annotators annotation, the background represents agreement entities or approved entities. Figure 3 shows an instance annotated by two annotators. The “美国”(America) entity has two black underlines and black background, which means the two annotators consistently annotated this span as GPE.

The revision stage is easy to execute. The annotation expert can click any disagreement entity to get a revision dialog. The dialog will display all corresponding disagreement entities, and the annotation expert can select a proper one.

## 5 Inconsistency Detector

This section describes Inconsistency Detector, which is designed for reducing corpus-level pre&suffix inconsistency.

### 5.1 Overview

In NER dataset, some entity types have descriptive words. It is difficult to reach an agreement on a descriptive string that should be included in the entity

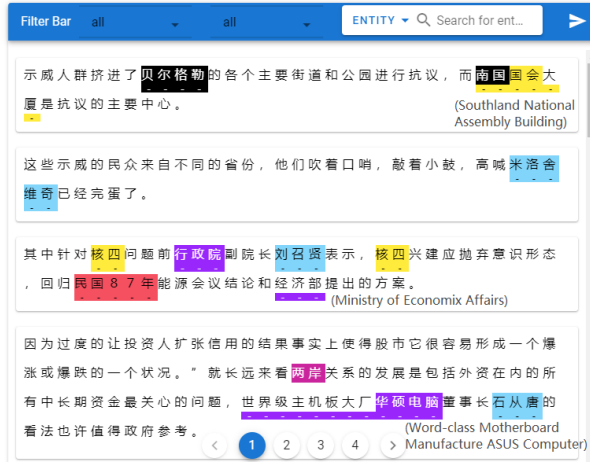


Figure 5: The screenshot of Prediction Error Analyzer. We provide English translations for mistakenly predicted entities in this figure.

string. For example, the *MONEY* entity type has a descriptive word "more than". In some instances the word "more than" is contained inside the entity string as a prefix, while in other instances the word is excluded as an external prefix. Appendix A displays inconsistent pre&suffixes of two NER dataset and some example instances.

Our proposed framework contains a detector algorithm to detect potential inconsistent pre&suffixes and an interface to help users to revise these inconsistencies.

## 5.2 Pre&Suffix Inconsistency Detector

We denote the dataset as  $D$ , the entity set extracted from the dataset as  $S$ . We use  $entity$  denotes a specific entity object from entity set. An entity object has at least four attributes, including *string*, *sentence*, *start*, *end*. *String* denotes the entity string, *sentence* represents the instance text string that contains this entity. Given a specific entity, the expression  $sentence[start: end] == string$  is always true.

We use  $p$  denotes entity prefix,  $p_i$  denotes prefix with length  $i$ .

$$p_i = \begin{cases} string[:i], & \text{if } len(string) \leq i \\ "", & \text{otherwise} \end{cases} \quad (1)$$

We use  $e$  denotes external descriptive word before entity string,  $e_i$  denotes external prefix with length  $i$ .

$$e_i = \begin{cases} sentence[start-i: start], & \text{if } start-i \geq 0 \\ "", & \text{otherwise} \end{cases} \quad (2)$$

Take the inconsistent prefix detection algorithm as an example as shown in Algorithm 1. The basic

## Algorithm 1 Inconsistent Prefix Detection Algorithm

**Input:** Entity set  $S$ , Max prefix length:  $l$

**Output:** Inconsistent prefixes

$I \leftarrow \emptyset$

$E \leftarrow \emptyset$

**for** entity in  $S$  **do**

**for**  $i = 1$  to  $l$  **do**

    Obtain  $p_i$ . (Eq.1)

    Obtain  $e_i$ . (Eq.2)

$I = I \cup p_i$

$E = E \cup e_i$

**end for**

**end for**

**return**  $I \cap E$

idea of inconsistency detector is that if a descriptive string appears both as a prefix or an external prefix, this string is considered an inconsistent prefix.

We first construct two empty string sets called *prefix\_set* and *external\_prefix\_set*. We then traverse the entity set to add entity prefix and external\_prefix respectively to the two string set.

After the traverse, the intersection of two string sets are strings that appearing in both entity prefix or entity external prefix. These strings are potential inconsistent prefixes.

## 6 Prediction Error Analyzer

This section describes Prediction Error Analyzer, which is designed for detecting label consistency from the model perspective.

## 6.1 Overview

From an intuitive perspective, The basic idea of this module is that the NER model trained with trainset can learn regulations of this annotation. When an entity gets a wrong prediction compared to annotation result in testset, it has a high proportion that the model has learnt an inconsistent regulation from trainset.

To support the need to detect inconsistent entities with the NER model, CroAno provides a will-designed error system, a search API based on the error system, and a visualization interface for users to compare the differences between model prediction and annotations from the annotator.

## 6.2 Error System

Existing entity error only measures if an entity is correctly predicted or not, which misses much useful information. CroAno deconstruct entity prediction error to a well-designed error system.

The error system consists of six different error types. **Extra Error** and **Missing Error** means that the NER model predicts an extra entity or misses an entity compared with annotators annotation. **Long Error** and **Short Error** means that the NER model predicts an entity that contains or is within an annotated entity. **Tag Error** and **Intersect Error** means different tags and intersect boundary between model prediction and annotation.

## 6.3 Search and Visualize Instances

The search API is used to filter entities and their context by the error system and other features. For example, if the algorithm expert finds that the *PERSON* entity type has a high proportion to be mistakenly missing in model prediction, they can use the search API to extract these instances.

The visualization interface is designed to make users directly perceive the difference between model prediction and annotations from the annotator. As shown in Figure 5, annotations from the annotator are marked by the background, while the highlighted underline exhibits annotations predicted by the model.

Except for detecting corpus-level inconsistency of the NER dataset, Prediction Error Analyzer can be used to detect annotation errors or catch the model’s weakness to decide the improvement direction of the model itself.

| Dataset     | Type     | Train  | Dev    | Test   |
|-------------|----------|--------|--------|--------|
| 2*OntoNotes | Sentence | 15.7K  | 4.3K   | 4.3K   |
|             | Char     | 491.9K | 200.5K | 208.1K |
| 2*CCKS      | Sentence | 1K     | -      | 0.4K   |
|             | Char     | 418.4K | -      | 132.7K |

Table 1: Statistics of dataset.

## 7 Technical Details

This section introduces some necessary technical details. CroAno has a web-based front-end server built on Vue in JavaScript and a back-end server built on Django in Python. CroAno has an environment decoupled from the operating system, which means it is deployment-free. CroAno implement the interface design based on an open-source Django framework named doccano<sup>1</sup>.

## 8 Experiments

In this section, we conduct experiments on a standard benchmark and a medical benchmark to verify the effectiveness of Inconsistency Detector.

### 8.1 Experimental Setting

**Dataset.** Two Chinese NER datasets are used in this paper, which include *OntoNotes 4.0* (Weischedel et al., 2011) and *CCKS 2019* (Han et al., 2020). *OntoNotes 4.0* is collected from the news domain, while *CCKS 2019* is collected from the medical domain. For *OntoNotes 4.0*, we use the same data split as (Zhang and Yang, 2018). Since the *CCKS 2019* dataset does not have a development set, we randomly select 20% samples from the training set as the development. Statistics of dataset is shown in Table 1.

**Dataset Correction.** We invite two volunteers to use Inconsistency Detector to correct the two datasets. The correction is for the entire dataset including training set, development set, and test set. It is worth mentioning that to count the learning and mastering time of the function, modify both datasets took less than half an hour.

**Model Settings.** We apply two most widely recognized NER baseline models, denoted as BiLSTM-CRF (Ma and Hovy, 2016) and BERT-Tagger (Devlin et al., 2018), respectively. We uniformly use AdamW (Loshchilov and Hutter, 2018) as an optimizer. For (Ma and Hovy, 2016), we use the same character embeddings as (Zhang and Yang, 2018) and set the initial learning rate to 0.01.

<sup>1</sup><http://doccano.herokuapp.com/>



For (Devlin et al., 2018), we set the initial learning rate to 0.00005.

## 8.2 Results

The model performance before/after correcting is shown in Table 2 and the statistics of inconsistent entities correction is shown in Table 3.

**OntoNotes 4.0.** We correct 730 entities in total, accounting for 1.4% of the total number of entities. After correction, BiLSTM-CRF reaches a 69.38% F1-score, an increase of 1.48%. The improvement of BERT-Tagger is even more significant, an increase of 1.96%, reaching 76.52% F1-score.

**CCKS 2019.** We correct 700 entities in total, accounting for 4.3% of the total number of entities. After correction, BiLSTM-CRF reached a 82.88% F1-score, an increase of 2.09%. Consistent with Ontonotes, the improvement of BERT-Tagger is more significant, an increase of 2.57%, reaching 85.88% F1-score.

The experiments prove that the annotator expert can use CROANO to promote dataset, and this promotion can be inherited by widely recognized BiLSTM-CRF model and BERT-Tagger model.

## 9 Related Works

Most of the existing crowd annotation tools for NER are dedicated to basic functions such as interface friendliness, the convenience of operation, and annotation prompts, which are all designed for annotators. Next, this section will compare the features of CROANO with the following related works:

BRAT (Stenetorp et al., 2012) is a web-based general annotation tool that can handle various annotation tasks, including span annotations and the relationship between spans. As an early annotation tool, BRAT has great influence. However, compared with crowd annotation platforms such as CROANO, it cannot manage crowdsourcing or reduce the label inconsistency.

GATE (Bontcheva et al., 2013) is a web-based collaborative text annotation framework. It enables users to perform complex corpus annotation projects, which involve a distributed team of annotators. Same as brat, GATE also lacks the ability to reduce the label inconsistency.

SLATE (Kummerfeld, 2019) is a lightweight annotation tool with a terminal-based workflow. It is designed for annotation experts, focusing on fast labeling. SLATE has a certain visual disagreement adjudication capability. However, due to the lim-

| 2*Model     | OntoNotes 4.0 |              | CCKS 2019 |              |
|-------------|---------------|--------------|-----------|--------------|
|             | Before        | After        | Before    | After        |
| BiLSTM-CRF  | 67.9          | <b>69.38</b> | 80.79     | <b>82.88</b> |
| BERT-Tagger | 74.56         | <b>76.52</b> | 83.31     | <b>85.88</b> |

Table 2: Model performance before/after correcting.

| Dataset   | Total Entity | Correction | Correction Ratio |
|-----------|--------------|------------|------------------|
| OntoNotes | 49262        | 730        | 1.48%            |
| CCKS      | 18294        | 700        | 3.83%            |

Table 3: Statistics of inconsistent entities correcting.

ited visualization ability of the terminal-based approach, it can only simply prompt disagreement samples. Besides, the disagreement adjudication cannot be applied at the entity-level.

AlpacaTag (Lin et al., 2019) applies the model ensemble mechanism to merge the results of different annotators, thereby realizing disagreement adjudication without relying on annotating experts. It is worth noting that the results of such a black box model are not interpretable for humans. Especially for fields such as clinical diagnosis and drug discovery, black box models often mean potential risks and poor persuasiveness.

YEDDA (Yang et al., 2018) provides a systematic solution for text span annotations such as collaborative user annotations, administrator evaluation. YEDDA can locate the position of disagreement and display it by generating a Latex file, but it cannot directly adjudicate disagreement like CROANO.

To the best of our knowledge, CROANO is the first crowd annotation platform providing the tool that can locate and fix inconsistent labels. CROANO can also make the full use of the ability of the algorithm expert and the annotation expert.

## 10 Conclusion and Future Directions

In this paper, we propose a web-based crowd annotation platform, which provides a systematic solution for improving the label consistency of the Chinese NER dataset. To solve instance-level inconsistency, we propose Disagreement Adjudicator. To solve corpus-level inconsistency, we propose Inconsistency Detector and Prediction Error Analyzer from statistic and model perspectives respectively.

The future directions are to extend CROANO to a cross-task and multi-language version and implement more prosperous model analysis functions.

## Acknowledge

This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106400). This work is supported by the National Natural Science Foundation of China (No.61976211, No.61806201). This work is supported by Beijing Academy of Artificial Intelligence (BAAI2019QN0301), the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006) and the Youth Innovation Promotion Association CAS.

## References

- Kalina Bontcheva, Hamish Cunningham, Ian Roberts, Angus Roberts, Valentin Tablan, Niraj Aswani, and Genevieve Gorrell. 2013. Gate teamware: a web-based, collaborative text annotation framework. *Language Resources and Evaluation*, 47(4):1007–1029.
- Wei-Te Chen and Will Styler. 2013. Anafora: A web-based general purpose annotation tool. In *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 14–19.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Xianpei Han, Zhichun Wang, Jiangtao Zhang, Qinghua Wen, Wenqi Li, Buzhou Tang, Qi Wang, Zhifan Feng, Yang Zhang, Yajuan Lu, et al. 2020. Overview of the ccks 2019 knowledge graph evaluation track: Entity, relation, event and qa. *arXiv preprint arXiv:2003.03875*.
- Jonathan K Kummerfeld. 2019. Slate: A super-lightweight annotation tool for experts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 7–12.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270.
- Bill Yuchen Lin, Dong-Ho Lee, Frank F Xu, Ouyi Lan, and Xiang Ren. 2019. Alpacatag: An active learning-based crowd annotation framework for sequence tagging. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 58–63.
- Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.
- Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.
- V Knowtator Ogren Philip. 2006. A protégé plug-in for annotated corpus construction. In *Human language technology conference of the North American chapter of the association of computational linguistics (HLT-NAACL)*.
- Younes Samih, Wolfgang Maier, and Laura Kallmeyer. 2016. Sawt: Sequence annotation web tool. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 65–70.
- Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun’ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.
- Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.
- Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. 2018. Yedda: A lightweight collaborative text span annotation tool. In *Proceedings of ACL 2018, System Demonstrations*, pages 31–36.
- Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564.

## A Inconsistent Pre&Suffix Example

The cause of the inconsistent prefixes and suffixes is divergent perceptions of different annotators on the annotation guideline. For example, some entity categories have specific descriptive strings. Because deleting these additive strings does not change the entity semantics, this leads to opposite annotation strategies.

Some of the inconsistent prefixes or suffixes detected by this algorithm in the Ontonotes dataset and the CCKS medical dataset are shown in Table4 and Table5. Some instances that contain inconsistent pre&suffix in the Ontonotes dataset are shown in Table6.

| Entity Type | Inconsistent Prefixes                              | Inconsistent Suffixes   |
|-------------|--|---|
| WORK_OF_ART | 《  | 》、节目(show)  |
| CARDINAL    | 上(more than)、超过(more than)、近(nearly)               | 辆、颗、位、个、家、种、项、名、所<br>(all of these suffixes are Chinese measure word) |
| DATA        | 不到(less than)、过去(over the past)、过去的(over the past) | 之内(within)、内(within)  |
| LOC         |  | 地区(area)  |
| NORP        |  | 民族(nation)、人民(people)、人士(person)、民众(the public)                       |
| GPE         | 驻  | 市区(city)、特区(special administrative region)、政府(government)             |

Table 4: Some inconsistent prefixes and suffixes in Ontonotes4.0.

| Entity Type            | Inconsistent Prefixes                            | Inconsistent Suffixes  |
|------------------------|--|--|
| Anatomic Site          | 近端(proximal)、左(left)、双侧(bilateral)、左侧(left side) | 上端(upper)、旁(side)、部  |
| Diseases and Diagnoses | 轻度(mild)   | 不能明确(not clear)、可能(possible)、术后(post operation)、化疗(chemotherapy) |
| Image Examination      |  | 增强(enhancement)、检查(examination)、平扫(plain CT scan)                |
| Operation              | 行(perform)                                       | 治疗(therapy)  |
| Medicine               |  | 片(tablet)  |
| Laboratory Inspection  | 血(blood)   | 百分比(percentage)、数(count)、浓度(concentration)、压积                    |

Table 5: Some inconsistent prefixes and suffixes in CCKS medical dataset.

| Entity Type | Prefix or Suffix | Prefix or Suffix in Entity Content  | Prefix or Suffix out of Entity Content  |
|-------------|------------------|---|---|
| WORK OF ART | Prefix: 《        | 他们想到了哈尔·荷尔布鲁克和电影 [《惊天大阴谋》] WORK_OF_ART<br>They thought of Hal Holbrook and the movie "Plot."                                | 张曼玉、梁朝伟主演的《[花样年华]WORK_OF_ART》<br>In the Mood for Love starring Maggie Cheung and Tony Leung Chiu-wai  |
| CARDINAL    | Suffix: 位        | 已有[172位]CARDINAL个人和14个机构获奖<br>172 individuals and 14 institutions have been awarded   | 参与评证的律师总计约[700]CARDINAL位, 评件总案数为7200余件<br>A total of about 700 lawyers participated in theevaluation, and the total number of cases was more than 7,200                                   |
| LOC         | Suffix: 地区       | 要求日本自卫队分担[亚太地区]LOC的安全责任<br>Asking Japan's Self-Defense Forces to share security responsibilities in the Asia-Pacific region | 英文[亚洲周刊]发布的[亚太]LOC地区城市排名中, 台北和大阪并列第二<br>Taipei and Osaka have tied for second place in a ranking of cities in the Asis-Pacific region released by the English-language magazine Asia Week |

Table 6: Pre&Suffix inconsistent instances.