Annotation Guideline to Character-level Dependencies of Chinese

Version 0.98

Zhao Hai

Department of Computer Science and Engineering,

Shanghai Jiao Tong University, Shanghai, China

zhaohai@cs.sjtu.edu.cn

Masao Utiyama and **Eiichiro Sumita**

NICT, Kyoto, Japan

mutiyama@nict.go.jp, eiichiro.sumita@nict.go.jp

2012.04.23

This corpus was annotated by the joint support of natural science foundation of China, No. 60903119, and NICT, Japan.

Abstract

The guideline is give principles to annotate the relations between characters inside a Chinese word. Using a modifier-head view, all Chinese words can be easily annotated in an iterative analysis and finally a dependency tree structure will be built for the given word. The annotation is based on the absolute part-of-speech (POS) of the word and POS tag for each characters inside it which is determined by the annotator as well. Dependency labels can be directly derived by all the POS tags.

1. Annotation Goal

This guideline is to show how to correctly annotate character-level dependencies that were proposed in (Zhao, 2009).

We assume that the annotators have the basic knowledge about Chinese languages, both writing and speaking.

This guideline is translated and significantly revised from an early version of the guideline in Chinese.

Most words are extracted from Chinese treebank by Penn, version 7.0. During the annotating, annotators perhaps have to retrieve the original text of CTB to fully understand the meaning of the annotated word.

The goal is to annotate all character-level dependencies with character-level part-of-speech and dependency labels for all these words.

2. Annotation Format

As we have a three-character Chinese word,

天安门

Each character in the word is indexed from 1 to 3. So we have indices for

1 天

2 安

3 门

Annotation will be done through giving the head for each character. Assume that 天's head is 安, 安's head is 门 and 门 is the root of this word. We will have the output character annotation like this,

1 天 2

2 安 3

3 门 0

Here , index 0 means the root character.

For some annotation that dependency label is required, we add the labels after the numbers for head.

1 天 2 *n-n-v*

2 安 3 *nn*

3 门 0 *root*

There are three types of dependency labels in the above word, *n-n-v, nn* and *root*.

For simplicity, we will write the annotation results in the following description as

天安门 2 3 0

Or

天安门 2 3 0 *n v n n-n-v nn root*

where n or v are part-of-speech of characters.

Blanks should be inserted between words, numbers and letters.

3. Word Types

We will distinguish different word types for annotation

(1) Single character word

No annotation is required.

(2) Words that are completely consist of numbers or letters.

一百万 2 3 0 *g g root*

100,000 2 3 4 5 6 7 0 *g g g g g g root*

NATO 2 3 4 0 *l l l root*

Annotation for each character should be done in an incremental order: each character takes its right neighbor character as its head and the last character is the root one. For number words, we use dependency label *g*, and *l* for letter words.

(3) Named entity

It is referred to person name, location, organization name or other translated name.

胡锦涛 2 3 0 *r r root*

上海 2 3 0 r r root.

纽约 2 0 *r root*

奥巴马 2 3 *0 r r root*

Annotation for each character should be done in an incremental order: each character takes its right neighbor character as its head and the last character is the root one. *r* is the dependency label of name entity.

Note for annotation of (2) and (3): annotators may simply mark their types as *_number, _letter,* or *_person, _loc, and _org* so that the annotation for number, letter and named entity can be skipped, where _person, _loc and _org represent three different named entities, person name, location name and organization name, respectively.

(4) Mixed names

A mixed name may typically include a foreign (or other) name part and other Chinese characters as follows,

纽约城 2 3 0 *r nn root*

Note for (3) and (4): for some named entities, especially those from China, Japan, Korea and Viet Nam, characters have become a meaningful combination such as 天安门, 安定县, its internal structure should be annotated rather than simply mark it as a *_loc*.

4. Generic Annotation

This is about how to annotate basic dependencies between characters if they have some sort of syntactic or semantic connection.

We recommend **<u>an iterative annotation strategy</u>**. For example, we consider a four-character word,

abcd

abc is first identified as an adjective constituent, and then *d* as a noun constituent, we may mark *abc* as the modifier, and *d* is its head. Inside *abc*, *bc* is identified as a verb constituent, and *a* as ad adverbial constituent, so that *bc* being the head for *abc*. Let's turn to the last multiple character part, *bc*, *c* is taken as the head for it. At last, we have the annotation,

abcd 2 3 4 0

There are no absolute rules to determine which character should be always the head, but we propose basic rules to direct the annotation.

At first, we need part-of-speech (POS) tag for both the annotated word and all characters inside it.

Note the annotation will be done without considering the context of the word, so we will give a simplified POS set for words.

We define a group of **absolute POS tags for word** as follows.

1. *p*: Pronoun 这个(this one) 他们(they) …
2. *n*: Noun 大门(main door) 汽车(automobile) …
3. *v*: Verb 书写(write) 玩耍(play) 举行(held) …
4. *a*: Adjective 热闹(alive) 邪恶(evil) 平坦(plain)…
5. *d*: Adverbial 最近(recently) 很难(difficultly) 大概(about) …

Note that Chinese is a language whose words may hold quite different POS tags as they are really adopted in the text. Nearly no Chinese word only keeps one possible POS.

So as we say it is an absolute POS tag for a word is to choose a possible POS from a POS priority list for the word. The following is to give rules to make the right choice to determine the absolute POS tag for a given word.

(1) The most popular case for multiple POS is words that can be both noun and verb. The rule is that if a word can be both verb and noun, then its absolute POS is verb.

(2) For a word that can be either adjective or noun in sentence, its absolute POS is set to adjective. To distinguish those difficultly disambiguating words, we set a semantic criterion to make the decision, noun is to give a name for an entity in real word, and adjective is to give a property description.

(3) For a word that can be either adjective or adverbial in sentence, the absolute POS for it is adverbial.

Character-level POS tag set

Character-level POS tagging will be dynamic or contextual for this annotation like word's POS tag in a real sentence. Note that's the big difference from the above absolute POS assignment for the annotated word.

We set the following character-level POS definition:

(1) *p*: Pronoun 这(this) 那(that) 我(I) 一(one)…

(2) *n*: Noun 门(door) 体(body) 名(name)

(3) *i*: Number, foreign letter transliteration and other non-Chinese characters

(4) *v*: Verb 写(write) 跑(run) …

(5) *a*: Adjective 红(red) 苦(difficult) …

(6) *d*: Adverbial 很(very) 最(most)…

(7) *f*: Functional character 的(of) 们(-es) 在(at) …

Note the question about character-level POS tagging is to resolve the POS ambiguity.

书 is most considered as a noun character as in 读书(read book), but for 书写(write, writing) 书 is actually a verb character, that is because in ancient Chinese, 书 is mostly being a verb.

Head rules

Assume that the absolute POS for the given word and all character-level POS have been determined, we may finish the annotation by following the below head rules.

The following head rules follow the general principle defined by traditional dependency grammar.

(1) For noun-character and its modifier characters in a noun, the former should be the head.

红花 2 0 *ad root*

蓝天 2 0 *ad root*

一本书 2 3 0 *ad ad root*

(2) For a given word, the kernel verb character should be the root for the word or the verb part at least.

吃掉 0 1 *root cd*

快跑 2 0 *vv root*

(3) For a coordinate structure, the right character will be always be the head.

玩耍 2 0 *vc root*

神仙 2 0 *nc root*

(4) If a functional character is involved in a constituent, then it should be always the head.

好的 2 0 *af root*

我们 2 0 *pf root*

Dependency label set

For a dependency relation, its dependency label is generally determined by POS tags of three constituents, i.e., those of dependant and head and their concatenation. For a dependency that *a* is the dependant and *b* is the head, the corresponding character subsequence should be like *a…b*. It is assumes that there are some other characters between *a* and *b*. If *p1*, *p2* and *p* are POS tags of *a*, *b* and *a…b*, respectively, then the dependency label for *a* <- *b* should be *p-p1-p2*. For the case that a should be included in a inseparable constituent such as named entity from transliteration, *p1* should be the POS tag of the constituent that a is being its head.

(1) For convenience, we define a group of abbreviations for the dependency labels.

ad n-a-n

af a-a-f

pf p-p-f

vd v-d-v

cd v-v-d

nn n-n-n

vv v-v-v

If no abbreviations are defined for a specific case, then please use the original format such as d-d-f as the dependency label during annotating.

(2) For a coordinate relation of noun, verb, adjective and adverbial, we correspondingly define four types of labels, *nc*, *vc, ac* and *dc*. The following are two examples.

凤凰 2 0 *nc root*

书写 2 0 *vc root*

So, *nc* and *vc* can be regarded as

nc n-n-n

vc v-v-v

Note these are different from *nn* and *vv* that is to represent a modification relation between two noun characters.

(3) For different word types which include number, letter and inseparable named entity, we use three labels to identify them, respectively.

g for number

l for letter

r for named entity

For mixed names, *g*/*l*/*r* will be seen as a type of *n*. For example, we have

ad n-a-g

and so on.

5. Examples for Full Annotation

We give couples of samples for character-level dependency annotation.

The green columns are supposed to be the input and the red all the output given by the annotators.

Note we require the absolute POS tag for the word should be given after the root in the format of *root-xx*.

Word 鲟鱼(sturgeon)

index	char	Pos-of-char	Head-index	Dep-label
1	鲟	*n*	2	*nn*
2	鱼	*n*	0	*root-n*

Word 天安门(Tian’anmen)

index	char	Pos-of-char	Head-index	Dep-label
1	天	*n*	2	*n-n-v*
2	安	*v*	3	*nn*
3	门	*n*	0	*root-n*

Word 普天同庆(the whole world is celebrating)

index	char	Pos-of-char	Head-index	Dep-label
1	普	*a*	2	*ad*
2	天	*n*	4	*n-n-v*
3	同	*d*	4	*vd*
4	庆	*v*	0	*root-n*

Word ５０余(more than fifty)

index	Char	Pos-of-char	Head-index	Dep-label
1	５	*i*	2	*g*
2	０	*i*	3	*a-g-a*
3	余	*a*	0	*root-a*

Word 拉美裔(Hispanic)

index	Char	Pos-of-char	Head-index	Dep -label
1	拉	*n*	2	*r*
2	美	*n*	3	*nn*
3	裔	*n*	0	*root-n*

Word 空对地(air to ground)

index	Char	Pos-of-char	Head-index	Dep-label
1	空	*n*	3	*nn*
2	对	*v*	0	*root-n*
3	地	*n*	2	*v-v-n*

Word 好人好事(good man and good deed)

index	char	Pos-of-char	Head-index	Dep-label
1	好	*a*	2	*ad*
2	人	*n*	4	*nc*

3	好	*a*	4	*ad*
4	事	*n*	0	*root-n*

References

Hai Zhao, Character-Level Dependencies in Chinese: Usefulness and Learning, EACL-2009, Athens, Greece, April, 2009, http://www.aclweb.org/anthology-new/E/E09/E09-1100.pdf