3 Experiment

3.1 Experimental setting

Our system consists of three parts; first, the SVM-based tagger extracts the neighboring attachment relations of the input sentence. Second, the parser analyzes further dependency attachments. If a new dependency attachment is generated, the MaxEnt based tagger estimates the label of the relation. The three parts of our parser are trained on the available data of the languages.

In our experiment, we used the full information of each token (FORM, LEMMA, CPOSTAG, POSTAG, FEATS) when we train and test the model. **Fig. 2** describes the features of each token. Some languages do not include all columns; such that the Chinese data does not include LEMMA and FEATURES, these empty columns are shown by the symbol "-" in **Fig. 2**. The features for the neighboring dependency tagging are the information of the focused word, two preceding words and two succeeding words. **Fig. 2** shows the window size of our features for estimating the word dependency in the main procedures. These features include the focused words (*n*, *t*), two preceding words and two succeeding words and their children. The features for estimating the relation label are the same as the features used for word dependency analysis. For example, if the machine learner estimates the operation of this situation as "**Left**" or "**Right**" by using the features in **Fig. 2**, the parser uses the same features in **Fig. 2** and the dependency relation to estimate the label of this relation.

For training the models efficiently, we divided the training instances of all languages at the CPOSTAG of the focused word *n* in **Fig .2.** In our preceding work, we found this procedure can get better performance than training with all the instances at once. However, only the instances in Czech are divided at the CPOSTAG of the focused word-pair *t*-*n*[3]. The performance of this procedure is worse than using the CPOSTAG of the focused word *n*, because the training instances of each CPOSTAG-pair will become scarce. However, the data size of Czech is much larger than other languages; we couldn't finish the training of Czech using the CPOSTAG of the focused word *n*, before the deadline for submitting. Therefore we used this procedure only for the experiment of Czech.

All our experiments were run on a Linux machine with XEON 2.4GHz and 4.0GB memory. The program is implemented in JAVA.

3.2 Results

Table 1 shows the results of our parser. We do not take into consideration the problem of cross relation. Although these cross relations are few in training data, they would make our performance worse in some languages. We expect that this is one reason that the result of Dutch is not good. The average length of sentences and the size of training data may have affected the performance of our parser. Sentences of Arabic are longer and training data size of Arabic is smaller than other languages; therefore our parser is worse in Arabic. Similarly, our result in Turkish is also not good because the data size is small.

We compare the result of Chinese with our preceding work. The score of this shared task is better than our preceding work. It is expected that we selected the FORM and CPOSTAG of each nodes as features in the preceding work. However, the POSTAG is also a useful feature for Chinese, and we grouped the original POS tags of Sinica Treebank from 303 to 54 in our preceding work. The number of CPOSTAG(54) in our preceding work is more than the number of CPOSTAG(22) in this shared task, the training data of each CPOSTAG in our preceding work is smaller than in this work. Therefore the performance of our preceding work in Sinica Treebank is worse than this task.

The last column of the **Table 1** shows the unlabeled scores of our parser without the preprocessing. Because our parser estimates the label after the dependency relation is generated. We only consider whether the preprocessing can improve the unlabeled scores. Although the preprocessing can not improve some languages (such as Chinese, Spanish and Swedish), the average score shows that using preprocessing is better than parsing without preprocessing.

Comparing the gold standard data and the system output of Chinese, we find the CPOSTAG with lowest accuracy is "P (preposition)", the accuracy that both dependency and head are correct is 71%. As we described in our preceding work and Section 2.3, we found that boundaries of prepositional phrases are ambiguous for Chinese. The bottom-up algorithm usually wrongly parses the prepositional phrase short. The parser does not capture the correct information of the children of the preposition. According to the results, this problem does not cause the accuracy of head of

[3] For example, we have 15 SVM-models for Arabic according to the CPOSTAG of Arabic (A, C, D, F, G…etc.). However, we have 139 SVM-models for Czech according the CPOSTAG pair of focused words (A-A, A-C, A-D…etc.)