CHINERS: A Chinese Named Entity Recognition System for the Sports Domain

Tianfang Yao Wei Ding Gregor Erbach
Department of Computational Linguistics
Saarland University
Germany
yao@coli.uni-sb.de wding@mpi-sb.mpg.de
gor@acm.org

Abstract

In the investigation for Chinese named entity (NE) recognition, we are confronted with *two principal challenges*. *One* is how to ensure the quality of word segmentation and Part-of-Speech (POS) tagging, because its consequence has an adverse impact on the performance of NE recognition. *Another* is how to flexibly, reliably and accurately recognize NEs. In order to cope with the challenges, we propose a system architecture which is divided into two phases. In the *first* phase, we should *reduce* word segmentation and POS tagging *errors* leading to the second phase *as much as possible*. For this purpose, we utilize *machine learning techniques* to repair such errors. In the *second* phase, we design *Finite State Cascades* (*FSC*) which can be *automatically* constructed depending on the recognition rule sets as a shallow parser for the recognition of NEs. The advantages of that are *reliable*, *accurate* and *easy to do maintenance* for FSC. Additionally, to recognize *special NEs*, we work out the corresponding strategies to enhance the correctness of the recognition. The experimental evaluation of the system has shown that the total average recall and precision for six types of NEs are *83%* and *85%* respectively. Therefore, the system architecture is *reasonable* and *effective*.

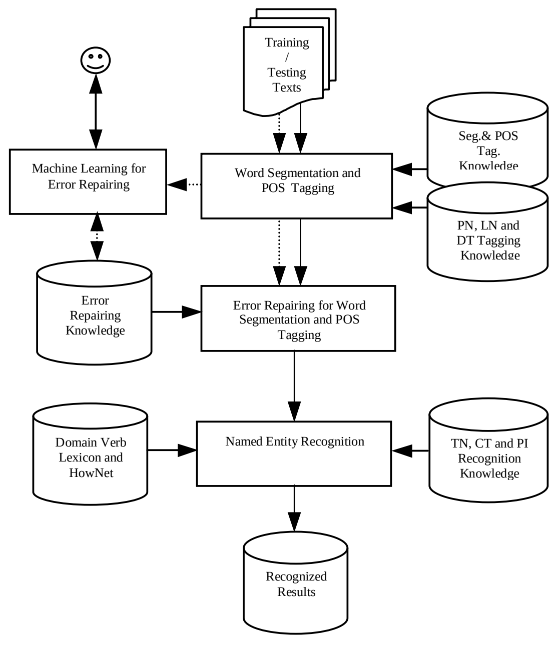
1 Introduction

The research for Chinese information extraction is one of the topics in the project COLLATE[1] (*Computational Linguistics and Language Technology for Real World Applications*). The main motivation is to investigate the strategies for information extraction for such language, *especially in some special linguistic phenomena*, to build a reasonable information extraction model and to implement an application system. *Chinese Named Entity Recognition System* (*CHINERS*) is a component of Chinese information extraction system which is being developed. CHINERS is mainly based on machine learning and shallow parsing techniques. We adopt *football competition news* as our corpus, because there exist a variety of named entities (NEs) and relations in the news. Among the NEs we select six of them as the recognized objects, that is, *personal name* (*PN*), *date or time* (*DT*), *location name* (*LN*), *team name* (*TN*), *competition title* (*CT*) and *personal identity* (*PI*). e.g. 莫晨月(Mo Chenyue), 卡洛斯 (Carlos); ９月１９日 (Sept. 19), 本周五 (this Friday), 前７０分钟 (former 70 minutes); 上海 (Shanghai), 柏林 (Berlin); 中国队 (China Team), 四川队 (Sichuan Team), 上海申花队

[1] COLLATE is a project dedicated to building up a German competence center for language technology in Saarbrücken, Germany.

(Shanghai Shenhua Team), 拜仁慕尼黑队 (Bayern München); 全国女足超级联赛 (National Woman Football Super League Matches), 泰王杯国际足球邀请赛 (Thailand King’s Cup International Football Tournament); 门将 (goalkeeper), 前锋 (forward), 外援 (foreign player), 主教练 (chief coach), 裁判员 (judge), 记者 (correspondent), etc.

Figure 1 shows the system architecture. The system is principally composed of *three components*. The *first* one is *Modern Chinese Word Segmentation and POS Tagging System* from Shan Xi University, China (Liu, 2000), which is our baseline system. The *second* one is an *error repairer* which is used to repair the word segmentation and POS tagging errors from the above system. The *third* one is a shallow parser which consists of *Finite State Cascades* (*FSC*) with three recognition levels. The *dotted line* shows the flow process for the *training texts*; while the *solid line* is the one for the *testing texts*. When training, the texts are segmented and tagged, then the error repairing candidate rules are produced and some of them are selected as the regular rules under the appropriate conditions. Thereafter, the errors caused during word segmentation and POS tagging in testing texts can be automatically repaired through such regular rules. Among the six types of NEs, *PN, DT and LN are tagged by the first component and repaired by the second component*. They are *immediately recognized after error repairing*; while TN, CT and PI are recognized and then tagged by the third component.



Training / Testing Texts
Seg.& POS Tag. Knowledge
Machine Learning for Error Repairing
Word Segmentation and POS Tagging
PN, LN and DT Tagging Knowledge
Error Repairing Knowledge
Error Repairing for Word Segmentation and POS Tagging
Domain Verb Lexicon and HowNet
Named Entity Recognition
TN, CT and PI Recognition Knowledge
Recognized Results

Figure 1: System architecture

In Section 2, an effective repairing approach for Chinese word segmentation and POS tagging errors will be presented. Next, Section 3 will aim to illustrate the principle of an automatically constructed FSC and NE recognition procedure. On the basis of that, Section 4 will show the three experimental conditions and results. Finally, Section 5 will draw some conclusions and introduce future work.

2 Repairing the Errors for Word Segmentation and POS Tagging

For the purpose of *ensuring good quality* in segmenting word and tagging POS, we compared different existing Chinese word segmentation and POS tagging systems and introduced one of them (Liu, 2000) as the first component in our system. Unfortunately, we found there still are considerable errors of word segmentation and POS tagging when we use this system to process our texts on sports domain. For example:

Error$_1$: 大(large)|A|连队(company)|N
Correct$_1$: 大连(Dalian)|N5|队(team)|N
Error$_2$: ‘|W|上三路(Shang Shan Road)|N5|’|W
Correct$_2$: ‘|W| 上三路 (upper three paths)|N|’|W

In the above examples, A, N, N5, and W represent an adjective, a general noun, a Chinese LN, and a punctuation respectively. According to the domain knowledge, the word “大连” is a city name as a constituent of TN, which should not be segmented; while the word “上三路” is an attack strategy of the football match, it should not be tagged as a Chinese LN. Obviously, these errors will have an unfavorable effect on the consequent recognition for NEs.

In order to improve the quality for word segmentation and POS tagging, there may be two ways to achieve such goal:

- *Develop a novel general Chinese word segmentation and POS tagging system*, which will have higher performance than the current systems of the same kind or
- *Utilize a baseline system* with good quality and further improve its performance on a specific domain, so that it can be suitable to real-world application.

We have chosen the *second* way in our investigation. First, the research of word segmentation and POS tagging is a *secondary* task for us in the project. In order to ensure the overall quality of the system, we have to enhance basic quality. Second, it is more effective to improve the quality for word segmentation and POS tagging on a *specific* domain.

The *transformation based error-driven machine learning* approach (Brill, 1995) is adopted to repair word segmentation and POS tagging errors, because it is suitable for fixing Chinese word segmentation and POS tagging errors as well as producing effective repairing rules automatically. Following (Hockenmaier and Brew, 1998; Palmer, 1997) we divide the error repairing operations of word segmentation into three types, that is, *concat*, *split* and *slide*. In addition, we add *context-sensitive* or *context-free* constraints in the rules to repair the errors of word segmentation and POS tagging. It is important that the *context constraints* can help us *distinguish different sentence environments*. The error repairing rules for word segmentation and POS tagging are defined as follows:

rectify_segmentation_error (operator, old_word(s)_and_tag(s), repairing_mode, new_tag(s), preceding_context_constraint, following_context_constraint)

rectify_tag_error (old_word_and_tag, new_tag, preceding_context_constraint, following_context_constraint)

Using these rules, we can move the word segmentation position newly and replace an error tag with a correct tag. e.g. 国|N|足|N|抵|V|沪|N (The national football team arrived in Shanghai). The word “国足” (the national football team) is an abbreviated TN that should not be segmented; while the word “沪” (Hu) is an abbreviated Chinese LN for Shanghai. We can use the following two rules to repair such errors:

rectify_segmentation_error (*concat*, 国|N|足|N, 1, J, _|_, 抵|V)

rectify_tag_error (沪|N, J, 抵|V, _|_)

Here, the digit 1 means the *operation number* of *concat*. J is a POS tag for the *abbreviated word*. After the errors are repaired, the correct result is 国足|J|抵|V|沪|J .

In the *training* algorithm (Yao et al., 2002), the error positions are determined by comparing between manually error-repaired text and automatically processed text from the baseline system. Simultaneously, the error environments are recorded. Based on such information, the candidate transformation rules are generated and the final error repairing rules are selected depending on their *error repairing number* in the *training* set. In order to use these rules with *priority*, the rules in the final rule library are sorted.

Considering the requirements of context constraints for different rules, we manually divide the rule context constraints into three types: *whole POS context constraint*, *preceding POS context constraint* and *without context constraint*. Hence, each error repairing rule can be used in accordance with either common or individual cases of errors. In the *testing* algorithm (Yao et al., 2002), the usage of error repairing rules with context constraints is *prior* to those without context constraints, the employment of error repairing rules for word segmentation has *priority* over those for POS tagging. Thus, it ensures that the rules can repair *more* errors. At the same time, it prevents *new errors* occur during repairing existing errors.

3 Named Entity Recognition

3.1 An Automatically Constructed FSC

After error repairing, the text with repaired word segmentations and POS tags is used as the input text for NE recognition.

We make use of *Finite-State Cascades* (*FSC*) (Abney, 1996) as a shallow parser for NE recognition in our system. An FSC is *automatically* constructed by the NE recognition rule sets and consists of three recognition levels. Each level has a *NE recognizer*, that is, TN, CT and PI recognizer (Other three NEs, namely, PN, DT and LN are immediately recognized after error repairing).

In order to build a *flexible* and *reliable* FSC, we propose the following construction algorithm to *automatically* construct FSC by the recognition rule sets.

The NE recognition rule is defined as follows:

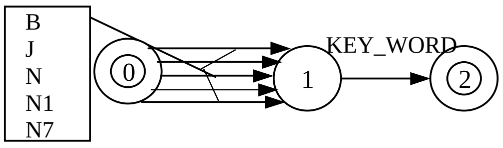
Recognition Category $\rightarrow$ POS Rule | Semantic Constraint$_1$ | Semantic Constraint$_2$ | … | Semantic Constraint$_n$

The NE recognition rule is composed of POS rule and its corresponding semantic constraints. The rule sets include 19 POS tags and 29 semantic constraint tags.

Four adjacent matrices, *POS matrix*, *POS index matrix*, *semantic index matrix* and *semantic constraint matrix*, are used in this algorithm as data structure. The POS matrix is used for the corresponding POS tags between two states. The POS index matrix provides the position of indexes related with POS tags between two states in the semantic index matrix. The semantic index matrix indicates the position of semantic constraints for each POS tag in semantic constraint matrix. The semantic constraint matrix saves the semantic constraint information for each POS tag in the POS matrix. We store the information for both the multi-POS tags between two states and the POS tags that have multi-semantic constraints in these matrices. As an example, Figure 2 shows that the following CT recognition rule set is used to build a *deterministic* finite automaton, that is, *CT recognizer*, using the above adjacent matrices. In the figure of the automaton, the semantic constraints in the rule set is *omitted*.

Rule$_1$: B + KEY_WORD | Rank + CompetitionTitleKeyword
Rule$_2$: J + KEY_WORD | AbbreviationName + CompetitionTitleKeyword
Rule$_3$: N + KEY_WORD | AbbreviationName + CompetitionTitleKeyword | CTOtherName + CompetitionTitleKeyword | Range + CompetitionTitleKeyword
Rule$_4$: N1 + KEY_WORD | CountryName + CompetitionTitleKeyword
Rule$_5$: N7 + KEY_WORD | CityName + CompetitionTitleKeyword | ContinentName + CompetitionTitleKeyword | CountryName + CompetitionTitleKeyword

In the POS tags, B, N1, and N7 represent a discrimination word, a proper noun and a transliterated noun separately. In the semantic constraint tags, Rank and Range mean the competition rank and range, such as super, woman etc.



POS adjacent matrix

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & & \text{B/J/N/N1/N7} & \\ 1 & & & \text{KEY\_WORD} \\ 2 & & & \end{array}$$

POS index adjacent matrix

$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline 0 & & 0 & \\ 1 & & & 1 \\ 2 & & & \end{array}$$

Semantic index adjacent matrix

$$\begin{array}{c|cccccc} & \text{B} & \text{J} & \text{N} & \text{N1} & \text{N7} & \text{KEY\_WORD} \\ \hline 0 & 0 & 1 & 2 & 3 & 4 & \\ 1 & & & & & & 5 \end{array}$$

Semantic constraint adjacent matrix

$$\begin{array}{c|cccccccc} & \text{AbbreviationName} & \text{CityName} & \text{CompetitionTitleKeyword} & \text{ContinentName} & \text{CTOtherName} & \text{CountryName} & \text{Range} & \text{Rank} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 4 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$$

Figure 2: An example to build a deterministic finite automaton using four adjacent matrices

The construction algorithm is summarized as follows:

- Input a NE recognition rule set and initialize four adjacent matrices.
- Get the first POS tag of a POS rule, start from the *initial state* of the NE recognizer, add its corresponding edge into the POS adjacent matrix under correct construction

condition (see *below explanation*). At the same time, add its corresponding semantic constraints into the semantic constraint adjacent matrix by the POS and semantic index adjacent matrices.

- If a tag's edge is successfully added, but it doesn't arrive in the final state, *temporarily*, push its POS tag, semantic constraints and related states (edge information) into a stack. If the next tag's edge isn't successfully added, pop the last tag's edge information from the stack. If the added edge arrives in the final state, pop all tag's edge information of the POS rule and add them into the POS and the semantic constraints adjacent matrix.
- If *all* existing states in the NE recognizer are tried, but the current edge can not be added, add a *new* state to the NE recognizer. In the following adding edge procedure, *share* the existing edge with tag's edge to be added *as much as possible*.
- If *all* the POS rules are processed, the construction for certain NE recognition level of FSC is completed.

It is important that the *correct construction condition* in the procedure of adding POS tag's edge must be met. For example, whether its corresponding semantic constraints *conflict* with the existing edge's semantic constraints between two states or the *in-degree of starting state* and the *out-degree of arriving state* must be less than or equal to 1, etc in the NE recognizer. Otherwise *give up* adding this tag's edge. Figure 3 is a part of the constructed recognition level of FSC for CT.

The construction algorithm is a rule-driven algorithm. It only depends on the format of rules. Therefore, it is *easy* to *change* the *size* of POS and semantic constraint tags or *easy* to *add*, *modify* or *delete* the rules. Additionally, the algorithm can be applied to build all the recognition levels, it is also *easy* to *expand* the NE recognizers in FSC for *new NEs*.

3.2 Recognition Procedure

When FSC has been constructed, we use it to recognize *TN*, *CT* and *PI*. First of all, input the text to be processed and different resources, such as *country name*, *club name*, *product name library* and *HowNet knowledge lexicon* (Dong and Dong, 2000). Then attach *keyword*, *potential personal identity*, *digital* or *alphabetic string*, and *other semantic constraint tags* to the corresponding constituents in a sentence. Thirdly, match the words one by one with a NE recognizer. Start from the *initial state*, match the POS tag's edge in the POS adjacent matrix, then match its corresponding semantic constraint tags in the semantic constraint adjacent matrix through two index adjacent matrices. If it is successful, push the related information into the stack. Otherwise find *another edge* that can be matched. Until arriving in the *final state*, pop the recognized named entity from the stack. Fourthly, if the current word is not successfully matched and the stack is not empty, pop the information of the *last* word and *go on* matching this word with other edges. If some words are successfully matched, the following words will be matched until *all* of the words in the sentence are tried to match. Finally, if there is still a sentence which is not processed in the text, *continue*. Otherwise finish the NE recognition procedure.

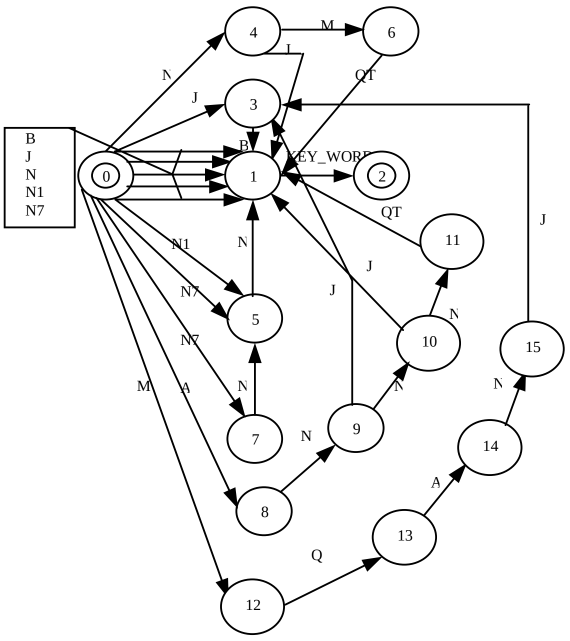


Figure 3: A part of the constructed recognition level of FSC for CT

The matching algorithm guarantees that any NE match is a match with *maximum length*. Because the finite automaton in FSC is *deterministic* and has *semantic constraints*, it can process ambiguous linguistic cases. Therefore, it has *reliability* and *accuracy* during the NE recognition procedure.

The following is an example to give the NE recognition procedure with FSC. L_1 to L_3 represent three *NE recognition levels* of FSC, namely, *TN*, *CT* and *PI* recognition levels. Every level has its NE recognition rule set.

上海|N5|申花|N|队|N|在|P|百事可乐|N|甲|N|Ａ|QT|联赛|N|中|F|击败|V|对手|N|吉林|N5|敖东|N|队|N|。|W1|

Shanghai Shenhua Team defeated the opponent – Jilin Aodong Team in the Pepsi First A League Matches.

```
L3  ----TN P ------CT F V PI----TN W1
L2  ----TN P ------CT F V N ----TN W1
L1  ----TN P N N QT N F V N ----TN W1
L0  N5 N N P N N QT N F V N N5 N N W1
```

3.3 Some Special Named Entities

Sometimes there are TNs or CTs *without keyword* in sentences. For instance, TNs without keyword:

Ex.1: 上视队一球小胜大连。(Shanghai Television Team feebly won one ball against Dalian.)

Ex.2: 中国取胜瑞典并不容易。(It is not easy that China wants to win Sweden.)

Ex.3: 沪连之战。(the fight between Shanghai and Dalian.)

For such special cases, we propose the following strategy for recognizing TN without keyword:

Step 1: Collect *domain verbs* and their *valence constraints* (Lin et al., 1994)

We organize domain verb lexicon and collect domain verbs, such as 胜 (win), 负于(lose), 对 (vs.), 进攻(attack), 防守 (guard), 迎战(take on), 瓦解(disintegrate), and their corresponding valence constraints. For instance, the valence constituents for the verb "胜" in our domain are defined as follows:

Basic Format: *Essiv* win *Object* ($Team_1$ win_1 $Team_2$; $Person_1$ win_2 $Person_2$; Personal $Identity_1$ win_3 Personal $Identity_2$; …)

Extended Format: *Link* + Basic Format; *Accompaniment* + Basic Format; …

In the basic format, *Essiv* is a subject that represents a *non-spontaneous* action and state in an event. *Object* is a direct object that deals with an non-spontaneous action. In the extended format, *Link* indicates a type, identity or role of the subject. In general, it begins with the word "作为…" (As …). *Accompaniment* expresses an indirect object that is accompanied or excluded. It often begins with the word "除了…" (Except …).

Step 2 : Keep the *equity* of domain verbs and analyze the constituents of TN candidates

According to the valence constraints, we examine whether the constituents in both sides of domain verbs are identical with the valence basic format or extended format, e.g. in Ex. 1 the team $name_1$ should be balanced with the team $name_2$ in the light of the valence basic format of domain verb 胜 (win_1). Besides, the candidate of team $name_2$ is checked, its constituent is a city name (Dalian) that can be as a constituent of team name.

Step 3 : Utilize *context clue* of TN candidates

We find whether there is a TN that is equal to current TN candidate with the keyword through the context of the TN candidate in the text, in order to enhance the correctness of TN recognition. As an example, in Ex. 2, depending on Step 2, a team name can occur on both sides of the domain verb 取胜 (win victory). A country name can be a constituent of team name. At the same time, the context of two TN candidates will be examined. Finally, if there is such context clue, the candidates are determined. Otherwise, continue to recognize the candidates by the next step.

Step 4: Distinguish *team* name from *location* name

Because a LN can be a constituent of a TN, we should distinguish TNs without keywords from LNs. With the help of other constituents (e.g. *nouns*, *propositions* etc.) in a sentence, the differences of both NEs can be distinguished to a certain extent. In Ex. 3 the noun 战 (fight) is an analogy for the match in sports domain. Therefore, here "沪" (Shanghai) and "连" (Dalian) represent two TNs separately. But it is still difficult to further improve the precision of TN recognition. (see the third experimental result.)

4 System Implementation and Experimental Results

CHINERS has been implemented with Java 2 (ver.1.4.0) under Windows 2000. The user interface displays the result of word segmentation and POS tagging from the baseline system, the error repairing result, the recognized result for six types of NEs and the statistic report for error repairing and NE recognition. The recognized text can be entered from a disk or directly downloaded from WWW (http://www.jfdaily.com/). HowNet knowledge lexicon is used to provide English and concept explanation for Chinese words in the recognized results. Except the error repairing rule library (for the most part) and HowNet knowledge lexicon, other resources have been manually built.

To evaluate this system, we have completed *three different* experiments. The *first* one is *only* for the performance of error repairing component. The *second* one is about comparison for NE recognition performance *with* or *without* error repairing. The *third* one is to test the recognition performance for TNs and CTs *without keyword*. The training set consists of 94 texts including 3473 sentences (roughly 37077 characters) from *Jiefang Daily* in 2001. The texts come from the *football sports news*. After machine learning, we obtain 4304 transformation rules. Among them 2491 rules are for word segmentation error repairing, 1813 rules are for POS tagging error repairing. There are 1730 rules as *concat* rules, 554 rules as *split* rules and 207 rules as *slide* rules in the word segmentation error repairing rules. Subsequently, we distinguish above rules into *context-sensitive* or *context-free* category *manually*. In the error repairing rules for word segmentation, 790, 315 and 77 rules are as *concat*, *split* and *slide* context-sensitive rules respectively; while 940, 239 and 130 rules are as *concat*, *split* and *slide* context-free rules separately. In the error repairing rules for POS tagging, 1052 rules are context-sensitive rules and 761 are context-free rules. The testing set is a separate set, which contains 20 texts including 658 sentences (roughly 8340 characters). The texts in the testing set have been randomly chosen from *Jiefang Daily* in May 2002. The texts also come from *football sports news*.

Table 1 and 2 show the first experimental results for the performance in different cases. These results indicate that the average F-measure of *word segmentation* has increased by *5.11%*; while one of *POS tagging* has even increased by *12.54%*.

In addition, using *same* testing set, we give the second and third experimental results in Figure 4, 5 and Table 3. In Figure 4 and 5, the performance of

	Average Recall	Average Precision	Average F-measure
Without Error Repairing	80.41	74.73	77.47
With Error Repairing	92.39	87.75	90.01

Table 1: Performance for word segmentation

	Average Recall	Average Precision	Average F-measure
Without Error Repairing	91.07	84.67	87.75
With Error Repairing	95.08	90.74	92.86

Table 2: Performance for POS tagging

six types of NEs has *manifestly* been improved. The *total average recall* is increased from 58% to *83%*, and the *total average precision* has also increased from 65% to *85%*. In Table 3, the average recall for TN without keyword has exceeded the average recall of TN in Figure 4; the average recall and precision of CT without keyword have also exceeded the average recall and precision of CT in Figure 4 and 5. But the average precision of TN *only* reaches 66%. We analyze the error reasons for the recognition of TN without keyword. Among 19 errors there are 17 errors from the wrong recognition for LN and 2 errors from imperfect recognition for TN. That is to say, the Step 4 of the recognition strategy in section 3.3 should be *further* improved.

In short, the experimental results have shown that the performance of whole system has been *significantly* improved after error repairing for

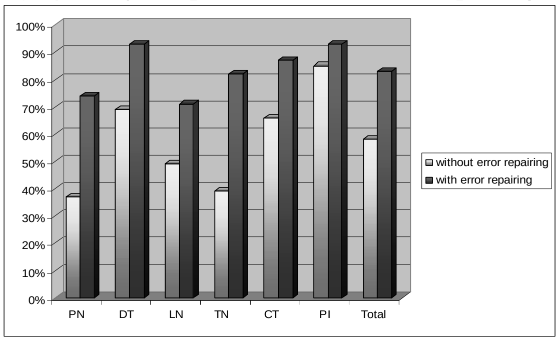


Figure 4: Recall comparison

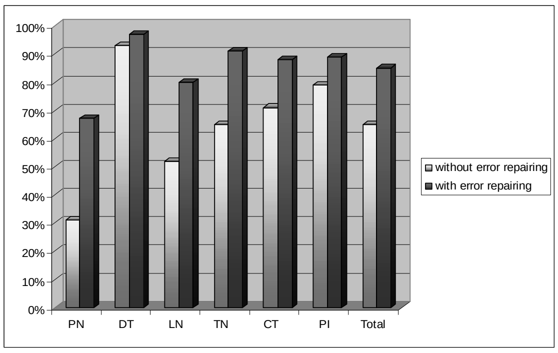


Figure 5: Precision comparison

	Total Number	Total Recognized Number / (Total Error Number)	Average Recall	Average Precision
TN without keyword	65	56 / (19)	86.15	66.07
CT without keyword	45	44 / (1)	97.78	97.73

Table 3: Recognition performance for TN and CT without keyword

word segmentation and POS tagging as well as the recognition for special NEs. It also proves that the system architecture is *reasonable* and *effective*.

5 Conclusions and Future Work

During the research for Chinese NE recognition, we noted that the errors from word segmentation and POS tagging have *adversely* affected the performance of NE recognition to a certain extent. We utilize transformation based error-driven machine learning to perform error repairing for word segmentation and POS tagging *simultaneously*. In the error repairing procedure, we add *context-sensitive* or *context-free* constraints in the rules. Thus, the introduction of *further errors* during error repairing can be *avoided*. In order to recognize NEs *flexibly*, *reliably* and *accurately*, we design and implement a FSC as a shallow parser for the NE recognition, which can be *automatically* constructed on basis of the recognition rule sets. In accordance with special NEs, additionally, we hit on the corresponding solutions for the recognition correctness.

The experimental results have shown that the performance of word segmentation and POS tagging has been improved, *leading* to an improved performance for NE recognition in our system. Such a *hybrid* approach used in our system synthesizes the advantages of *knowledge engineering* and *machine learning.*

For future work we will focus on *relation extraction*. On one hand, we will build an ontology including sports objects, movements and properties as a knowledge base to support corpus annotation. On the other hand, we utilize machine learning to automatically build relation pattern library for relation recognition.

Acknowledgement

This work is a part of the *COLLATE project* under contract no. 01INA01B, which is supported by the *German Ministry for Education and Research.*

References

S. Abney. 1996. *Partial Parsing via Finite-State Cascades*. In Proceedings of the ESSLLI ’96 Robust Parsing Workshop. Prague, Czech Republic.

E. Brill. 1995. *Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging*. Computational Linguistics. Vol. 21, No. 4:543-565.

Z. Dong and Q. Dong. 2000. *HowNet*. http://www.keenage.com/zhiwang/e_zhiwang.html.

J. Hockenmaier and C. Brew. 1998. *Error-Driven Learning of Chinese Word Segmentation*. Communications of COLIPS 8 (1):69-84. Singapore.

X. Lin et al. 1994. *Dictionary of Verbs in Contemporary Chinese*. Beijing Language and Culture University Press. Beijing China. (In Chinese)

K. Liu. 2000. *Automatic Segmentation and Tagging for Chinese Text*. The Commercial Press, Beijing, China. (In Chinese)

D. Palmer. 1997. *A Trainable Rule-Based Algorithm for Word Segmentation*. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL ’97): 321-328. Madrid, Spain.

T. Yao, W. Ding, and G. Erbach. 2002. *Repairing Errors for Chinese Word Segmentation and Part-of-Speech Tagging*. In Proc. of the First International Conference on Machine Learning and Cybernetics 2002 (ICMLC 2002) , Beijing, China.