Product Named Entity Recognition Based on Hierarchical Hidden Markov Model*

Feifan Liu, Jun Zhao, Bibo Lv, Bo Xu
National Laboratory of Pattern Recognition
Institute of Automation Chinese Academy of Sciences
Beijing P.O. Box 2728, 100080
{ffliu,jzhao,bblv,xubo}@nlpr.ia.ac.cn

Hao Yu
FUJITSU R&D
Xiao Yun Road No.26
Chao Yang District, Beijing, 100016
yu@frdc.fujitsu.com

Abstract

A hierarchical hidden Markov model (HHMM) based approach of product named entity recognition (NER) from Chinese free text is presented in this paper. Characteristics and challenges in product NER is also investigated and analyzed deliberately compared with general NER. Within a unified statistical framework, the approach we proposed is able to make probabilistically reasonable decisions to a global optimization by leveraging diverse range of linguistic features and knowledge sources. Experimental results show that our approach performs quite well in two different domains.

1 Introduction

Named entity recognition(NER) plays a significantly important role in information extraction(IE) and many other applications. Previous study on NER is mainly focused either on the proper name identification of person(*PER*), location(*LOC*), organization(*ORG*), time(*TIM*) and numeral(*NUM*) expressions almost in news domain, which can be viewed as general NER, or other named entity (NE) recognition in specific domain such as biology.

As far as we know, however, there is little prior research work conducted by far on product named entity recognition which can be crucial and valuable in many business IE applications, especially with the increasing research interest in Market Intelligence Management(MIM), Enterprise Content Management (ECM) [Pierre 2002] and *etc.*

This paper describes a prototype system for product named entity recognition, ProNER, in which a HHMM-based approach is employed. Within a unified statistical framework, the approach based on a mixture model is able to make probabilistically reasonable decisions to a global optimization by exploiting diverse range of linguistic features and knowledge sources. Experimental results show that ProNER performs quite well in two different domains.

2 Related Work

Up to now not much work has been done on product named entity recognition, nor systematic analysis of characteristics for this task. [Pierre 2002] developed an English NER system capable of identifying product names in product views. It employed a simple Boolean classifier for identifying product name, which was constructed from the list of product names. The method is similar to token matching and has a limitation for product NER applications. [Bick *et al.* 2004] recognized named entities including product names based on constraint grammar based parser for Danish. This rule-based approach is highly dependent on the performance of Danish parser and suffers from its weakness in system portability. [C. Niu *et al.* 2003] presented a bootstrapping approach for English named entity recognition using successive learners of parsing-based decision

[0] This work was supported by the Natural Sciences Foundation of China(60372016,60272041) and the Natural Science Foundation of Beijing(4052027).

System	*Statistical Model*	*Linguistic Feature*	*Combinative Points*
[Zhang *et al.* 2003]	HMM	semantic role, tokens	pattern rules
[Sun *et al.* 2002]	class-based LM	word form, NE category	cue words list
[Tsai *et al.* 2004]	ME model	tokens	knowledge representation

Table 1: Comparison between several Chinese NER systems[1]

list and HMM, and promising experiment results (F-measure: 69.8%) on product NE (corresponding to our *PRO*) were obtained. Its main advantage lies in that manual annotation of a sizable training corpus can be avoided, but it suffers from two problems, one is that it is difficult to find sufficient concept-based seeds needed in bootstrapping for the coverage of the variations of *PRO* subcategories, another it is highly dependent on parser performance as well.

Research on product NER is still at its early stage, especially in Chinese free text collections. However, considerable amount of work has been done in the last decade on the general NER task and biological NER task. The typical machine learning approaches for English NE are transformation-based learning[Aberdeen *et al.* 1995], hidden Markov model[Bikel *et al.* 1997], maximum entropy model[Borthwick, 1999], support vector machine learning[Eunji Yi *et al.* 2004], unsupervised model[Collins *et al.* 1999]and *etc.*

For Chinese NER, the prevailing methodology applied recently also lie in machine learning combining other knowledge base or heuristic rules, which can be compared on the whole in three aspects showed in Table 1.

In short, the trend in NER is to adopt a statistical framework which try to exploit some knowledge base as well as different level of text features within and outside NEs. Further those ideas, we present a hybrid approach based on HHMM [S. Fine *et al.* 1998] which will be described in detail.

3 Problem Statements and Analysis

3.1 Task Definition

3.1.1 Definition of Product Named Entity

In our study, only three kinds of product named entities are considered, namely *Brand Name(BRA)*, *Product Type(TYP)*, *Product Name(PRO)*, and *BRA* and *TYP* are often embedded in *PRO*. In the following two examples, there are two BRA NEs, one TYP NE and one PRO NE all of which belong to the family of product named entities.

Exam 1: 明基*(Benq)/**BRA*** 品牌(brand)市场占有率(market shares)稳步(steadily)上升(ascend)。

Exam 2: 公司(corporation)即将(will)推出(deliver) *[Canon/**BRA** 334万(ten thousand)*像素*(pixels)*数码*(digital)*相机*(camera) Pro90IS/**TYP**]/**PRO*** 。

Brand Name refer to proper name of product trademark such as “明基(Benq)” in Exam 1.

Product Type is a kind of product named entities indicating version or series information of product, which can consist of numbers, English characters, or other symbols such as “+” and “-” *etc.*In our study, two principles should be followed.

(1) Chinese characters are not considered to be *TYP*, nor subpart of *TYP* although some of them can contain version or series information. For instance, in “2005新年贺岁(happy new year)版(version)手机(cell phone)”, here “新年贺岁(happy new year)版(version)”should not be considered as a *TYP*.

(2) Numbers are essential elements in product type entity. For instance, in “PowerShot 系列(series)数码(digital)相机(camera)”, “PowerShot” is not considered as a *TYP*, however, in “PowerShot S10 数码(digital)相机(camera)”, “PowerShot S10” can make up of a *TYP*.

Product Name, as showed above in Exam 2, is a kind of product named entities expressing self-contained proper name for some specified product in real world compared to *BRA* and *TYP* which only express one attribute of product. *i.e.* a *PRO* NE must be assigned with distinctly discriminative information which can not shared with other general product-related expressions.

[1] Note: LM(language model); ME(maximum entropy).

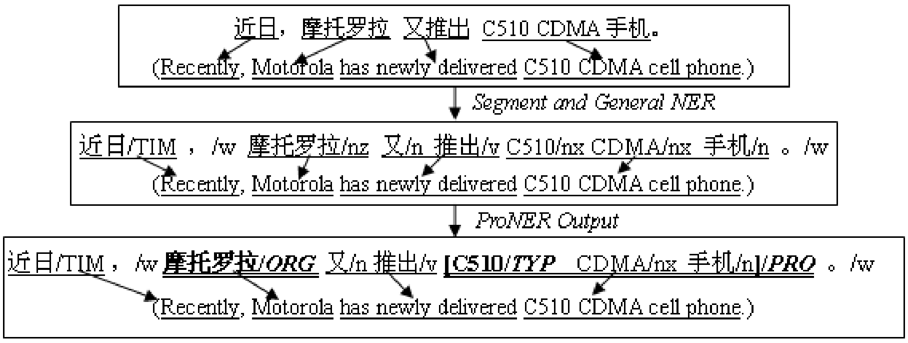
近日，摩托罗拉 又推出 C510 CDMA 手机。
(Recently, Motorola has newly delivered C510 CDMA cell phone.)
Segment and General NER
近日/TIM ， /w 摩托罗拉/nz 又/n 推出/v C510/nx CDMA/nx 手机/n 。/w
(Recently, Motorola has newly delivered C510 CDMA cell phone.)
ProNER Output
近日/TIM ， /w 摩托罗拉/ORG 又/n 推出/v [C510/TYP CDMA/nx 手机/n]/PRO 。/w
(Recently, Motorola has newly delivered C510 CDMA cell phone.)

Figure 1 An Example of Product Named Entity Recognition Task

(1) Product-related expressions which are embedded with either *BRA* or *TYP* can be qualified to be a *PRO* entity. *e.g.* "BenQ闪存(flash)盘(disk)" is a *PRO* entity, but the general product-related expression "闪存(flash)市场(market)调查(investigation)" cannot make up of a *PRO* entity.

(2) Product-related expressions indicating some specific version or series information which is unique for a *BRA* can also be considered as a *PRO* entity. *e.g.* "DIGITAL IXUS系列(series)数码(digital)相机(camera)" is a *PRO* because "DIGITAL IXUS" series is unique for Canon product, but "智能(intelligent)型(version)手机(cell phone)" is not a *PRO* because the attribute of "intelligent version" can be assigned to any cell phone product.

3.1.2 Product Named Entity Recognition

Product named entity recognition involves the identification of product-related proper names in free text and their classification into different kinds of product named entities, referring to *PRO, TYP* and *BRA* in this paper.In comparison with general NER, nested product NEs should be tagged separately rather than being tagged just as a single item, shown as Figure 1.

3.2 Challenges for Product Named Entity Recognition

①For general named entities, there are some cues which are very useful for entity recognition, such as "市"(city), "公司"(Inc.), and *etc.* In comparison, product named entities have no such named conventions and cues, resulting in higher boundary ambiguities and more complex NE candidate triggering difficulties.

②In comparison with general NER, more challenges in product NER result from miscellaneous classification ambiguities. Many entities with identical form can be a kind of general named entity, a kind of product named entity, or just common words.

③In comparison with general named entities, product named entities show more flexible variant forms. The same entity can be expressed in several different forms due to spelling variation, word permutation and *etc.* This also compounds the difficulties in product named entity recognition.

④In comparison with general named entities, it is more frequent that product named entities are nested as *Figure* 1 illustrates. More efforts have to be made to identify such named entities separately.

3.3 Our Solutions

We adopt the following strategies in triggering and disambiguating process respectively.

(1) As to product NER, it's pivotal to control the triggering candidates efficiently for the balance between precision and recall. Here we use the knowledge base such as brand word list, and other heuristic information which can be easily acquired.

(2)After triggering candidates, we try to employ a statistical model to make the most of multi-level context information mentioned above in disambiguation. We choose hierarchical hidden Markov model (HHMM) [S. Fine et al. 1998] for its more powerful ability to model the multiplicity of length scales and recursive nature of sequences.

4 Hybrid Approach for Product NE Recognition

4.1 Overall Workflow of ProNER

①***Preprocessing:*** Segment, POS tagging and general NER is primarily conducted using our off-shelf SegNer2.0 toolkit on input text.

②***Generating Product NE Candidates:*** First, *BRA* or *ORG* and *TYP* are triggered by brand word list and some word features respectively. Here we categorize the triggering word features into six classes: alphabet string, alphanumeric string, digits, alphabet string with fullwidth, digits with fullwidth and other symbols except Chinese characters. Then *PRO* are triggered by *BRA* and *TYP* candidates as well as some clue words indicating type information to some extent such as “版”(version), “系列”(series). In this step the model structure(topology) of HHMM[S. Fine *et al.* 1998] is dynamically constructed, and some conjunction words or punctuations and specified maximum length of product NE are used to control it.

③***Disambiguating Candidates:*** In this module, boundary and classification ambiguities between candidates are resolved simultaneously. And Viterbi algorithm is applied for most-likely state sequences based on the HHMM topology.

4.2 Integration with Heuristic Information

To get more efficient control in triggering process above, we try to integrate some heuristic information. The heuristic rules we used are as domain-independent as possible in order that they can be integrated with statistical model systematically rather than just some tricks on it.

(1) Stop Word List:

Common English words, English brand word, and some punctuations are extracted automatically from training set to make up of stop word list for *TYP*; by co-occurrence statistics between *ORG* and its contexts, some words are extracted from the contexts to make up of stop word list for *PRO* in order to overcome the case that brand word is prone to bind its surroundings to be a *PRO*.

(2) Constrain Rules:

Rule 1: For the highly frequent pattern “数字+英文量词”(number + English quantifier word), all the corresponding *TYP* candidates triggered by categorized word features(CWF) should be removed.

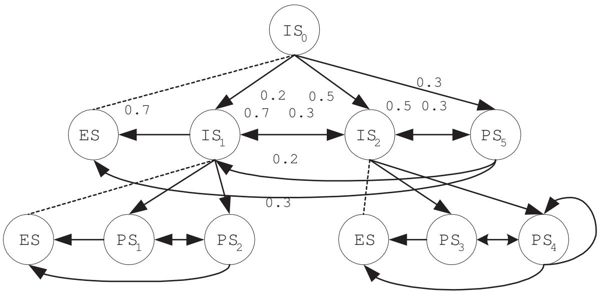

Figure 2 Structure of Hierarchical Hidden Markov Model (HHMM)

Rule 2: Product NE candidates in which some binate symbols don’t match each other should be removed.

Rule 3: Unreasonable symbols such as “-” or “:” should not occur in the beginning or end of product NE candidates.

4.3 HHMM for product NER application

By HHMM [S. Fine *et al.* 1998] the product NER can be formulated as a tagging problem using Viterbi algorithm. Unlike traditional HMM in POS tagging, here the topology of HHMM is not fixed and internal states can be also a similar stochastic model on themselves, called internal states compared to production states which will emit only observations.

Our HHMM structure actually consists of three level approximately illustrated as figure 2 in which *IS* denotes internal state, *PS* denotes production state and *ES* denote end state at every level. For our application, an input sequence from our SegNer2.0 toolkit can be formalized as $w_1/t_1w_2/t_2 \dots w_i/t_i \dots w_n/t_n$, among which w_i and t_i is the *i*th word and its part-of-speech, n is the number of words. The POS tag set here is the combination of tag set from Peking University(***PKU-POS***) and our general NE categories(***GNEC***) including *PER*(person), *LOC*(location), *ORG*(organization), *TIM*(time expression), *NUM*(numeric expression). Therefore we can construct our HHMM model by the state set {S} consisting of {***GNEC***}, {*BRA*, *PRO*, *TYP*}, and {***V***} as well as the observation set {O} consisting of {V} which is the word set from training data. That is to say, the word forms in $\{$V$\}$ which are not included in NEs are also viewed as production states.

In our model, only *PRO* are internal state which may activate other production states such as *BRA* and *TYP* resulting in recursive HMM. In consistence with S. Fine's work, q_i^d (1$\leq d \leq D$) is used to indicate the *i*th state in the *d*th level of hierarchy. So, the product NER problem is to find the most-likely state activation sequence Q*, a multiscale list of states, based on the dynamic topology of HHMM given a observation sequence W = $w_1 w_2 \dots w_i \dots w_n$, formulated as follows based on Bayes rule ($P(W)$=1).

$$Q^{*}= \arg\max_{Q} P(Q|W)= \arg\max_{Q} P(Q)P(W|Q) \tag{1}$$

From the root node of HHMM, activity flows to all other nodes at different levels according to their transition probability. For description convenience, we take the *k*th level as example(activated by the *m*th state at the *k-1*th level).

$$P(Q) \cong \underbrace{p(q_1^k|q_m^{k-1})}_{vertical\ transition} \overbrace{p(q_2^k|q_1^k)\prod_{j=3}^{|q^k|}p(q_j^k|q_{j-1}^k, q_{j-2}^k)}^{horizontal\ transition} \tag{2}$$

$$P(W|Q)=\begin{cases} \cong \prod\limits_{j=1}^{|q_{PS}^k|} p([w_{q_j^k-begin} \cdots w_{q_j^k-end}]|q_j^k) \\ \qquad\qquad if\ q_j^k \notin \{IS\} \\ activate\ other\ states\ recursively \\ \qquad\qquad if\ q_j^k \in \{IS\} \end{cases} \tag{3}$$

Where $|q^k|$ is the number of all states and $|q_{PS}^k|$ is the number of production states in the kth level; $w_{q_j^k-begin} \cdots w_{q_j^k-end}$ indicates the word sequence corresponding to the state q_j^k.

(1) In equation (3), if $q_j^k \in \{\{$***GNEC***$\}$,$\{$***V***$\}\}$, $p([w_{q_j^k-begin} \cdots w_{q_j^k-end}]|q_j^k)$=1, because we assume that the general NER results from the preceding toolkit are correct;

(2) If $q_j^k = PRO$, production states in the (k+1)th level will be activated by this internal state through equation (2),(3) and go back when arriving at an end state, thus hierarchical computation is implemented;

(3) If $q_j^k = BRA$, we assign equation (3) a constant value in that *BRA* candidates consist of only a single brand word in our method. In addition brand word can also generate *ORG* candidates, thus we can assign equation (3) as follows.

$$p([w_{q_j^k-begin} \cdots w_{q_j^k-end}]|q_j^k = BRA) = 0.5 \tag{4}$$

(4) If $q_j^k = TYP$, categorized word features(CWFs) defined in section 4.1 are applied, *i.e.* the words associated with the current state are replaced with their CWFs (WC) acting as observations. Then we can compute the emission probability of this *TYP* production state as the following equation, among which $|q_j^k|$ is the length of observation sequence associated with the current state.

$$p([w_{q_j^k-begin} \cdots w_{q_j^k-end}]|q_j^k = TYP)$$
$$\cong p(wc_1|begin)p(end|wc_{|q_j^k|})\prod_{m=2}^{|q_j^k|} p(wc_m|wc_{m-1})$$

All the parameters in every level of HHMM can be acquired using maximum likelihood method with smoothing from training data.

4.4 Mixture of Two Hierarchical Hidden Markov Models

Now we have implemented a simple HHMM for product NER. Note that in the above model(HHMM-1), we exploit both internal and external features of product NEs only at levels of simply semantic classification and just word form. To achieve our motivation in section 3.3, we construct another HHMM(HHMM-2) for exploiting multi-level contexts by mixing with HHMM-1.

In HHMM-2, the difference from HHMM-1 lies in the state set S$_{II}$ and observation set O$_{II}$. Because the input text will be processed by segment, POS tagging and general NER, as a alternative, we can also take T=$t_1 t_2 \dots t_i \dots t_n$ as observation sequence, *i.e.* O$_{II}$=$\{$PKU-POS$\}$. Accordingly, S$_{II}$= $\{\{$PKU-POS$\}$, $\{$GNEC$\}$, BRA, TYP,

Data Sets	*PRO*	*BRA*	*TYP*	*PER*	*LOC*	*ORG*
DataSetPRO1.2	12,432	5,047	10,606	424	1,733	4,798
OpenTestSet	1800	803	1364	39	207	614
CloseTestSet	1553	513	1296	55	248	619

Table 2: Overview of Data Sets

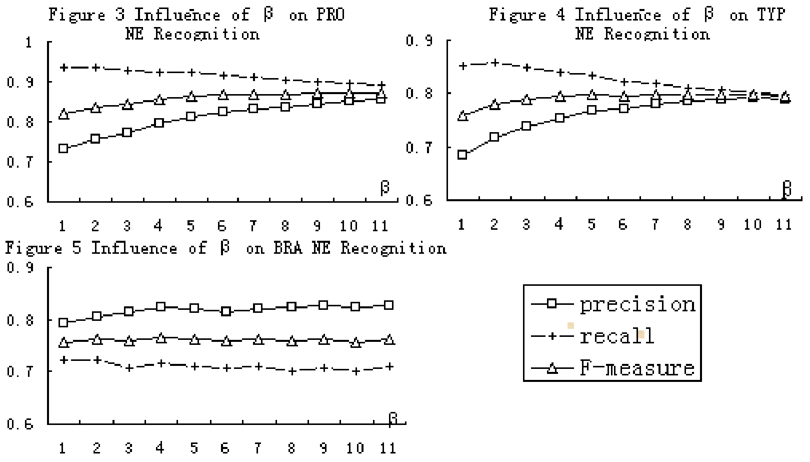

Figure 3 Influence of β on PRO NE Recognition

Figure 4 Influence of β on TYP NE Recognition

Figure 5 Influence of β on BRA NE Recognition

PRO}, among which *PRO* is internal state. Similarly, the problem is formulated as follows with HHMM-2.

$$
\begin{aligned}
Q^*_{II} &= \arg\max_{Q_{II}} P(Q_{II}|T) \\
&= \arg\max_{Q_{II}} P(Q_{II})P(T|Q_{II}) \qquad (5)
\end{aligned}
$$

The description and computation of HHMM-2 is similar to HHMM-1 and is omitted here.

We can see that besides making use of semantic classification of NEs in common, HHMM-1 and HHMM-2 exploit word form and part-of-speech (POS) features respectively. Word form features make the model more discriminative, while POS features result in robustness. Intuitively, the mixture of these two models is desirable for higher performance in product NER by balancing the robustness and discrimination which can be formulated in logarithmic form as follows.

$$
\begin{aligned}
&(Q^*, Q^*_{II}) \\
&= \arg\max_{Q,Q_{II}} \{\log(P(Q)) + \log(P(W|Q)) \\
&+ \beta[\log(P(Q_{II})) + \log(P(T|Q_{II}))]\} \qquad (6)
\end{aligned}
$$

Where β is a tuning parameter for adjusting the weight of two models.

5 Experiments and analysis

5.1 Data Set Preparation

A large number of web pages in mobile phone and digital domain are compiled into text collections, DataSetPRO, on which multi-level processing were performed. Our final version, DataSetPRO1.2, consists of 1500 web pages, roughly 1,000,000 Chinese characters. Randomly selected 140 texts (digital 70, mobile phone 70) are separated from DataSetPRO1.2 as our OpenTestSet, the rest as TrainingSet, from which 160 texts are extracted as CloseTestSet. Table 2 illustrates the overview of them.

5.2 Experiments

Due to various and flexible forms of product NEs, though some boundaries of recognized NEs are inconsistent with manual annotation, they are also reasonable. So soft evaluation is also applied in our experiments to make the evaluation more reasonable. The main idea is that a discount score will be given to recognized NEs with wrong boundary but correct detection and classification. However, strict evaluation only score completely correct ones.

All the results is conducted on OpenTestSet unless it is particularly specified. Also, the evaluation scores used below are obtained mainly by

Digital Domain (β=8)						
Product NEs	Close Test			Open Test		
	Precision	Recall	F-measure	Precision	Recall	F-measure
PRO	0.864	0.799	0.830	0.762	0.744	0.753
TYP	0.903	0.906	0.905	0.828	0.944	0.882
BRA	0.824	0.702	0.758	0.723	0.705	0.714
Mobile Phone Domain (β=8)						
Product NEs	Close Test			Open Test		
	Precision	Recall	F-measure	Precision	Recall	F-measure
PRO	0.917	0.935	0.926	0.799	0.856	0.827
TYP	0.959	0.976	0.967	0.842	0.886	0.864
BRA	0.911	0.741	0.818	0.893	0.701	0.785

Table 3: Experimental Results in Digital and Mobile Phone Domain

soft metrics, and strict scores are also given for comparison in experiment 3.

1. Evaluation on the Influence of β in the Mixture Model.

In the mixture model denoted as equation (6), the β value reflects the different contribution of two individual models to the overall system performance. The larger β, the more contribution made by HHMM-2. Figure 3, 4, 5 illustrate the varying curves of recognition performance with the β value on *PRO*, *TYP*, *BRA* respectively.

Note that, if β equal to 1 then two models are mixed with equivalent weight. We can see that, as β goes up, the F-measures of *PRO* and *TYP* increase obviously firstly, and begin to go down slightly after a period of growing flat. It can be explained that HHMM-2 mainly exploits part-of-speech and general NER features which can relieve the sparseness problem to some extent, which is more serious in HHMM-1 due to using lower level of contextual information such as word form. However, as β becomes larger, the problem of imprecise modeling in HHMM-2 will be more salient and begin to illustrate a side-effect in the mixture model. Whereas, the influence of β on *BRA* is negligible because its candidates are triggered by the relatively reliable knowledge base and its sub-model in HHMM is assigned a constant as shown in equation(4).

Summings-up:

(1) Mixture with HHMM-2 can make up the weakness of HHMM-1.

(2) HHMM-2 can make more contributions to the mixture model under the conditions that limited annotated data is available at present. In our system, β is assigned to 8 based on above experimental results.

2. Evaluation on the portability of ProNER in two domains.

First, we can see from Table 3 that ProNER have achieved fairly high performance in both digital and mobile phone domain. This can validate to some extent the portability of our system，which is consistent with our initial motivation.

Second, the results also show that our system performs slightly better in mobile phone domain for both close test and open test. This can be explained that there are more challenging ambiguities in digital domain due to more complex product taxonomy and more flexible variants of product NEs.

Summings-up: *The results provide promising evidence on the portability of our system to different domains though there are some differences between them.*

3. Evaluation on the efficiency of the mixture model and the improvement of the triggering control with heuristics.

In table 4, “1” denotes HHMM-1; “2” denotes HHMM-2; “+” means the mixture model; “*” means integrating with heuristics mentioned in section 4.2.

The results reveal that the mixture model outperforms each individual model with both soft and strict metrics. Also, the results show that heuristic information can increase the F-measure of *PRO* and *TYP* by 10 points or so for both indi-

HHMM	*BRA*		*TYP*		*PRO*	
	strict score	soft score	strict score	soft score	strict score	soft score
1	0.68	0.72	0.57	0.66	0.52	0.61
1*	0.70	0.74	0.70	0.80	0.63	0.72
2	0.67	0.73	0.66	0.74	0.61	0.68
2*	0.70	0.74	0.76	0.85	0.70	0.76
1+2	0.70	0.75	0.67	0.77	0.67	0.72
1+2*	0.72	0.76	0.76	0.87	0.75	0.80

Table 4: Improvement results (F-measure) with heuristics and model mixture

vidual model and the mixture model. Additionally we can see that HHMM-2 performs better on the whole than HHMM-1, which is consistent with experiment 1 that heavier weights should be assigned to HHMM-2 in the mixture model.

Summings-up:

(1) Either HHMM-1 or HHMM-2 can not perform quite well independently, but systematical integration of them can achieve obvious performance improvement due to the leverage of diverse levels of linguistic features by their efficient interaction.

(2) Heuristic information can highly enhance the performance for both individual model and the mixture model.

6 Conclusions and Future Work

This paper presented a hierarchical HMM (hidden Markov model) based approach of product named entity recognition from Chinese free text. By unifying some heuristic rules into a statistical framework based on a mixture model of HHMM, the approach we proposed can leverage diverse range of linguistic features and knowledge sources to make probabilistically reasonable decisions for a global optimization. The prototype system we built achieved the overall F-measure of 79.7%, 86.9%, 75.8% corresponding to *PRO*, *TYP*, *BRA* respectively, which also provide experimental evidence to some extent on its portability to different domains.

Our future work will focus on the following:

(1) Using long dependency information;

(2) Integrating segment, POS tagging, general NER and product NER to avoid error spread.

References

John M. Pierre. (2002) Mining Knowledge from Text Collections Using Automatically Generated Metadata. In: *Procs of Fourth International Conference on Practical Aspects of Knowledge Management.*

Michael Collins and Yoram Singer. (1999) Unsupervised Models for Named Entity Classification. In: *Proc. of EMNLP/VLC-99.*

Eunji Yi, Gary Geunbae Lee, and Soo-Jun Park. (2004) SVM-based Biological Named Entity Recognition using Minimum Edit-Distance Feature Boosted by Virtual Examples. In: *Proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04).*

Bick, Eckhard (2004) A Named Entity Recognizer for Danish. In: *Proc. of 4th International Conf. on Language Resources and Evaluation*,pp:305-308.

Jian Sun, Jianfeng Gao, Lei Zhang, Ming Zhou, Changning Huang. (2002) Chinese Named Entity Identification Using Class-based Language Model. In: *COLING 2002*. Taipei, Taiwan.

Huaping Zhang, Qun Liu, Hongkui Yu, Xueqi Cheng, Shuo Bai. Chinese Named Entity Recognition Using Role Model. *Special Iissue "Word Formation and Chinese Language processing" of the International Journal of Computational Linguistics and Chinese Language Processing*, 8(2),2003, pp:29-60

Aberdeen, John *et al.* (1995)MITRE: Description of the ALEMBIC System Used for MUC-6. *Proc. of MUC-6*, pp. 141-155

D.M. Bikel, S. Miller, R. Schwartz, R. Weischedel. (1997) Nymble: a High-Performance Learning Name-finder. In: *Fifth Conference on Applied Natural Language Processing*, pp 194-201.

Borthwick. A. (1999) A Maximum Entropy Approach to Named Entity Recognition. PhD Dissertation.

Tzong-Han Tsai, S.H. Wu, C.W. Lee, Cheng-Wei Shih, and Wen-Lian Hsu. (2004) Mencius: A Chinese Named Entity Recognizer Using the Maximum Entropy-based Hybrid Model. *International Journal of Computational Linguistics and Chinese Language Processing*, Vol. 9, No 1.

Cheng Niu, W. Li, J.h. Ding and R.K. Srihari. (2003) A Bootstrapping Approach to Named Entity Classification Using Successive Learners. In: *Proceedings of the 41st ACL*, Sapporo, Japan, pp:335-342.

S. Fine, Y. Singer, N. Tishby. (1998) The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*. 32(1), pp:41-62