# A Dual-Iterative Method for Concept-Word Acquisition from Large-Scale Chinese Corpora

Guogang Tian
Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences
Graduate University of the Chinese Academy of Sciences
Beijing, China 100080

naitgg@hotmail.com

Cungen Cao
Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences
Graduate University of the Chinese Academy of Sciences
Beijing, China 100080

cgcao@ict.ac.cn

**Abstract - This paper proposes a dual-iterative method, a hierarchical inner and outer iteration method (HIO), to acquire concept words from a large-scale, un-segmented Chinese corpus. It has two levels of iteration: the EM-CLS algorithm and the Viterbi-C/S algorithm constitute the inner iteration for generating concept words, and the concept word validation constitutes the outer iteration together with the concept word generation. Through multiple iterations, it integrates the concept word generation and validation into a uniform acquisition process. In the process of acquisition, the HIO method can cope with the problem of over-segmentation, over-combination and data sparseness. The experimental result shows that the HIO method is valid for concept word acquisition that can simultaneously increase the precision and recall rate of concept word acquisition.**

## 1. Introduction

Concept word acquisition is an important research in knowledge acquisition from text (KAT) (Cao and Sui, 2003), and it is also the foundation of ontology learning (Maedche, 2002). Its main purpose is to acquire plentiful concept words from text corpora. It is very similar to unknown word recognition (Chen and Bai, 1998), (Feng, Chen, et al., 2004) and term extraction (Bourigault and Jacquemin, 1999). However, there are subtle distinctions among these three researches. Generally, concept word can be classified into three types: proper name, compound word and derived word. Except for these three word types, unknown word recognition also identifies numeric-type compounds, and it does not concern known words listed in a dictionary. Term extraction (Bourigault and Jacquemin, 1999) mainly processes domain texts, and often extracts commonly used professional terms from a specific domain text corpus.

Fu and Luke (2003) proposed a two-stage Chinese segmentation system. At the first stage, it segmented the input text according to known words on the basis of 2-gram statistical model, and then identified unknown words at the second stage using a hybrid method which consisted of word context, word composition and word juncture model.

Yang and Li (2003) proposed a heuristic method that it generated five rules using mutual information and significance estimation to extract unknown word.

Peng and Schuurmans (2001) proposed an unsupervised training method to build probability models that accurately segmented Chinese character sequences into words. It used successive EM phases to learn a good probability model over character strings, and then prunes the model with a mutual information selection criterion to obtain a more accurate word lexicon.

Lai and Wu (2000, 2002) proposed a likelihood ratio method to extract possible unknown words or phrases defined by PLUs (phrase-like-unit). The final PLU was decided by two principles of overlap competition and inclusion competition.

Nagao and Mori (1994) proposed a rapid n-gram extraction method to extract adjacent substrings with same prefix in an ordered prefix table. It was noted that it was an affix method intrinsically.

From these above works, we can summarize that there are two kinds of method to identify or acquire unknown words, that is, the non-iterative statistical method and the affix method. The non-iterative unknown word recognition (Fu and Luke, 2003), (Yang and Li, 2003) , (Peng and Schuurmans, 2001) , (Lai and Wu, 2000, 2002), (Zhang, Lv, et al., 2003 ) usually adopts n-gram statistical model that is combined with segmentation and combination operation to identify unknown words. It can deal with over-segmentation, but can not tackle over-combination. In addition, the length of unknown word must be restricted in order to ensure system performance. The acquired unknown words are often 2-grams, 3-grams and 4-grams. The affix method (Nagao and Mori, 1994) has even more limits. For example, it can not deal with unknown words that have not obvious affix features, and it can not use contextual information of unknown words, either.

This paper, motivated by the work of Chang and Su (1997) and Liu, Zhang, et al. (2004) presents a hierarchical inner and outer iteration method to acquire concept words from a large-scale, un-segmented Chinese text corpus. It has two levels of iteration which involves concept word generation and validation. It makes some extension on EM algorithm and Viterbi algorithm which make up the concept word generation. The concept word validation combines mutual information and context entropy into a validation criterion. These two levels of iteration can simultaneously increase the precision and recall rate of concept word acquisition.

The main contribution of this paper is that it proposes a HIO method for concept word acquisition. The HIO method unifies concept word generation and validation into a consecutively iterative process so that it can increase precision and recall simultaneously. The rest of this paper is organized as follows: Section 2 presents the HIO method. Concept word generation is discussed in section 2.1, and concept word validation is discussed in section 2.2. The whole HIO algorithm is presented in Section 2.3. The experiment result and error analysis are provided in section 3. Section 4 concludes this paper and outlines the future work.

## 2. The HIO Method

The HIO method (a Hierarchical Inner and Outer iteration method) has two levels of iteration, that is, the inner iteration and the outer iteration. The alternation of EM-CLS and Viterbi-C/S algorithm constitutes the inner iteration of the HIO – concept word generation, and concept word validation constitutes the outer iteration of the HIO. The basic structure of the HIO method is illustrated in Fig. 1.

The HIO method can cope with the two primary problems in concept word acquisition: over-segmentation and over-combination. Data sparseness is one of common problems in statistical language processing. Concept word acquisition is not the exception. In the acquisition process, it may produce the sparse data. Katz smoothing is applied in the HIO method to smooth sparse data and reduce their effect on concept word acquisition.

### 2.1 Concept Word Generation

### 2.1.1 EM-CLS algorithm

The EM-CLS algorithm, which is based on EM (expectation maximization) algorithm, estimates generated terms' probability distribution and identifies their types in a large corpus.

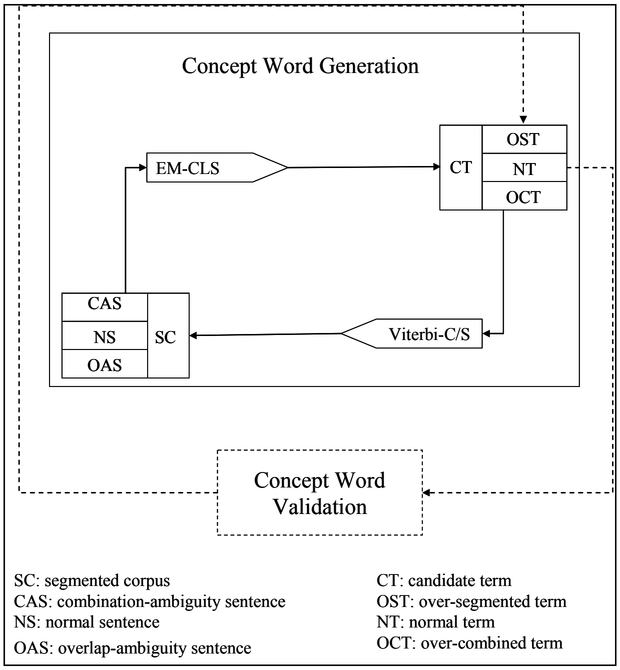

Fig.1. The Structure of HIO Method

EM algorithm (Figueiredo, 2004), (Prescher, 2003) is a common method for estimating maximum-likelihood when missing data are present. It has two steps: *E-step* (expectation step) and *M-step* (maximization step). Given the observed data x and the current parameter estimation $\hat{\theta}^{(t)}$, E-step computes the conditional expectation (with respect to the missing data y) of the logarithm of a complete *posteriori* probability function, *logp(y,θ|x)*. Usually E-step is called as Q function, as illustrated in (1). Equation (2) shows the M-step of EM algorithm. M-step chooses the parameters which can maximize Q function as the estimated parameters. Through consecutive iterations of E-Step and M-Step, EM algorithm can get stabilized parameters.

E-Step:

$$\begin{aligned} Q(\theta \mid \hat{\theta}^{(t)}) &\equiv E[\log p(y,\theta \mid x) \mid x, \hat{\theta}^{(t)}] \\ &\propto \log p(\theta) + E[\log p(y,x \mid \theta) \mid x, \hat{\theta}^{(t)}] \qquad (1) \\ &= \log p(\theta) + \int p(y \mid x, \hat{\theta}^{(t)}) \log p(y,x \mid \theta) dy \end{aligned}$$

M-Step:

$$\hat{\theta}^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \hat{\theta}^{(t)}) \, (2)$$

An un-segmented corpus is denoted as C={$C_1$, $C_2$, … ,$C_n$} where $C_i(1 \leqslant i \leqslant n)$ represents an un-segmented sentence. After segmentation, C is converted into the segmented corpus denoted as S={$S_1$, $S_2$, …, $S_n$} where $S_i$ *(*$1 \leqslant i \leqslant n$) is a segmentation of $C_i$. The generated candidate terms[1] are grouped into a set denoted as T={$t_1$, $t_2$, …, $t_m$}, where $t_j$ ($1 \leqslant j \leqslant m$) is the generated candidate term.

If $C_i$ is taken as the observed data, $S_i$ as the missing data, we can estimate the maximum-likelihood of term $t_j$ with the EM algorithm which is deemed as its probability distribution in the corpus C. Equation (3) shows the probability estimation of term $t_j$. In (3), $S_i^*$ denotes the optimal segmentation of sentence $C_i$, which can be achieved by the Viterbi-C/S algorithm (to be discussed in the next section). f($t_j$, $S_i$) denotes the frequency of term $t_j$ in sentence $S_i$. $\hat{t}_j = p(t_j), \hat{T} = \{p(t_j) \,|\, t_j \in T\}$ .

After estimating the probability of term $t_j$, we still have to judge to which type it belongs. The candidate term has three types that are *normal term*, *over-segmented term* and *over-combined term*.

$$\hat{t}_j^{(t+1)} = \frac{\sum_{i=1}^{n} f(t_j, S_i) \times p(S_i^* \,|\, C_i, \hat{T}^{(t)})}{\sum_{j=1}^{m} \sum_{i=1}^{n} f(t_j, S_i) \times p(S_i^* \,|\, C_i, \hat{T}^{(t)})} = \frac{\sum_{i=1}^{n} f(t_j, S_i) \times p(S_i^*, C_i \,|\, \hat{T}^{(t)})}{\sum_{j=1}^{m} \sum_{i=1}^{n} f(t_j, S_i) \times p(S_i^*, C_i \,|\, \hat{T}^{(t)})} \quad (3)$$

If a concept word (or meaningful word) is segmented into several components, it is called *over-segmented term.* For example, *高血糖 (hyperglycemia)* is possibly spitted into *高 (high)* and *血糖 (blood sugar)*.

If a word is combined with another word, but their combination is not a concept word (or meaningful word), it is called *over-combined term*, such as *但也 (but also)*.

Equation (4) can assign a type label to $t_j$, which is denoted by CLS($t_j$).

$$CLS(t_j) = \arg\max_{H^{(i)}} \frac{|\{S_k \,|\, S_k \in Sen(t_j) \wedge S_k \in H^{(i)}\}|}{|\, Sen(t_j) \,|} \quad (4)$$

In (4), Sen($t_j$)={$S_i$| $t_j \in S_i$}, $H^{(i)}$ denotes the sentence type label.

### 2.1.2 Viterbi-C/S algorithm

The Viterbi-C/S algorithm dynamically segments a corpus using the estimated probability of candidate terms and executes combination and segmentation operations on ambiguous terms in order to achieve the optimal segmentation. After completing corpus segmentation, it judges if a sentence contains overlap ambiguity or combination ambiguity.

Given a segmented sentence $S_i$, $S_i = t_1 t_2 \ldots t_k$ ($1 \leqslant j \leqslant k$, $t_j \in T$), it is assumed that terms are independent each other, *the likelihood of sentence $S_i$* is defined as:

$$p(S_i) = \prod_{j=1}^{k} p(t_j) \quad (5)$$

*Definition 1*: It is *the optimal segmentation* that its likelihood is maximal among all segmentations of a sentence. The optimal segmentation is denoted as $S_i^*$

$$S_i^* = \arg\max_{S_i} p(S_i \,|\, C_i, \hat{T}) = \arg\max_{S_i} p(S_i, C_i | \hat{T}) \quad (6)$$

Like candidate terms, segmented sentences are also classified into three types: normal sentence (*N-Sen*), overlap-ambiguity sentence (*OA-Sen*), and combination-ambiguity sentence (*CA-Sen*).

If a segmented sentence contains over-combined terms, it is considered as an *OA-Sen*.

If a segmented sentence contains over-segmented terms, it is considered as a *CA-Sen*.

It is observed that there is a direct correspondence between the type of candidate term and segmented sentence: *normal term – N-Sen*, *over-segmented term – CA-Sen* and *over-combined term – OA-Sen*.

*Definition 2: Segmented Density* is defined as the number of segmented term in each length unit.

For a sentence $S_i$,

$$SD(S_i) = \frac{p(S_i) \times NT(S_i)}{length(S_i)} \quad (7).$$

For a corpus S,

$$SD(S) = \frac{\sum_{S_i \in S} p(S_i) \times NT(S_i)}{length(S)} = \frac{\sum_{S_i \in S} p(S_i) \times NT(S_i)}{\sum_{S_i \in S} length(S_i)} \quad (8).$$

In (7)-(8), *NT(X)* denotes the number of terms in sentence *X*, and *length(Y)* denotes the length of sentence *Y*.

The type of segmented sentence is measured by (9). Setting a threshold range [$r_1$, $r_2$] ($r_1 < r_2$), if CLS($S_i$)<$r_1$, $S_i$ is a *OA-Sen*, if CLS($S_i$)>$r_2$, $S_i$ is a *CA-Sen*, if $r_1 \leqslant$ CLS($S_i$) $\leqslant r_2$, $S_i$ is a *N-Sen*.

$$CLS(S_i) = \frac{\mathrm{SD}(S_i)}{\mathrm{SD}(S)} \quad (9)$$

[1] The meaning of “term” here is different from the meaning of “term” in term extraction. Here term refers to ordinary word.

We make an extension to the classical Viterbi algorithm (Rabiner, 1989), thus get the Viterbi-C/S algorithm as illustrated in Fig. 2.

When segmenting a corpus, Viterbi-C/S binds combination and segmentation operations (C/S operation) on the selected terms according to their types. If over-segmented term is selected, combination operation is performed, if over-combined term is selected, segmentation operation is performed. These combination or segmentation operations on candidate term possibly causes data sparseness problem. So we use Katz Smoothing method (Goodman, 2001) as the smoothing strategy to eliminate sparse data.

$$p(t_{i-n+1}...t_{i-1}t_i) = \alpha(t_{i-n+1}...t_{i-1})p(t_{i-n+2}...t_{i-1}t_i)\ (10)$$

In (10), $t'=t_{i-n+1}\ldots t_{i-1}t_i$ doesn't exist in candidate term set T, $\alpha$ is a normalization constant.

## 2.2 Concept Word Validation

The generated candidate terms need to be further validated to filter out ambiguous terms. The validation takes into considerations candidate term's composition and local context. The former is considered as a cohesion validation which adopts mutual information method (Sproat and Shih, 1990). The latter is considered as an independence validation which adopts context entropy method (Tung and Lee, 1994). So the concept word validation is the combination of mutual information and context entropy method.

```
Viterbi-C/S Algorithm
Input: un-segmented corpus C, candidate terms' probability estimation
Output: the optimal segmentation and its type
1. selecting a sentence $C_i$ from corpus C;
2. selecting all possible candidate terms at the current position of sentence $C_i$, which constitute a set denoted as $T^p=\{t_1^p, t_2^p, t_3^p, \ldots\}$;
3. selecting a candidate term which has maximum-likelihood from the set $T^p$ as a possible segmented term of sentence $C_i$, denoted as st;
4. performing the corresponding operation according to the type of term st
   a. if an over-segmented term, performing segmenting operation on it and re-estimating the likelihood of new term, goto (3);
   b. if an over-combined term, performing combining operation on it and re-estimating the likelihood of new term, goto (3);
   c. if a normal term, segmenting the sentence, computing the likelihood of current segmentation and moving the position pointer forwardly
5. repeating step 1-4, until all sentences in corpus are segmented
6. computing segmented density for corpus and sentences, and determining the type label of sentence
```

Fig. 2. The Viterbi-C/S Algorithm

The basic assumptions of concept word validation are that:

If a term is an over-combined term, it contains at least a division point where its cohesion degree must be low.

If a term is an over-segmented term, its local context features in corpus must be weak.

### 2.2.1 Mutual Information

It is assumed that there is at most two division points in a validating term $t_v= c_1c_2\ldots c_n$.

If $t_v^l=c_1c_2\ldots c_l\in T$ $(1\leqslant l<n)$ and $t'= c_1c_2\ldots c_lc_{l+1}\notin T$, $t_v^l$ is called the *maximal left substring* of $t_v$, and *l* is the *left division point* of *$t_v$*.

If $t_v^r=c_rc_{r+1}\ldots c_n\in T$ $(1<r\leqslant n)$ and $t'= c_{r-1}c_r\ldots c_n\notin T$, $t_v^r$ is called the *maximal right substring* of $t_v$, and *r* is the *right division point* of $t_v$.

(1) If $l<r-1$, $t_v$ has two division points, which is denoted as $t_v=t_v^2=t_v^lt_v^mt_v^r$;

(2) If $l=r-1$, $t_v$ has a division point, which is denoted as $t_v=t_v^1=t_v^lt_v^r$;

(3) If $l\geqslant r$, $t_v$ has two possible divisions which are denoted as $t_v=t_v^{1-L}= t_v^lt_v^{-l}$ and $t_v^{1-R}= t_v=t_v^{-r}t_v^r$ respectively.

To case (1)

$$MI(t_v) = MI(t_v^2)$$

$$= \log\frac{p(t_v^lt_v^mt_v^r)}{p(t_v^l)p(t_v^m)p(t_v^r)+p(t_v^l)p(t_v^mt_v^r)+p(t_v^lt_v^m)p(t_v^r)} \quad (11)$$

To case (2),

$$MI(t_v) = MI(t_v^1) = \log\frac{p(t_v^lt_v^r)}{p(t_v^l)p(t_v^r)}\ (12)$$

To case (3),

$$MI(t_v^{1-L}) = \log\frac{p(t_v^lt_v^{-l})}{p(t_v^l)p(t_v^{-l})};$$

$$MI(t_v^{1-R}) = \log\frac{p(t_v^{-r}t_v^r)}{p(t_v^{-r})p(t_v^r)}; \quad (13)$$

$$MI(t_v) = \min\{MI(t_v^{1-L}), MI(t_v^{1-R})\}$$

Before computing the mutual information of the validating term, we above all identify to which type it belongs among the above case (1) to (3) and then adopt the corresponding equation (11-13). Similarly, we still apply equation (10) to deal with data sparseness problem.

### 2.2.2 Context Entropy

It is assumed that $t_v$ is a validating term. Its left context is denoted as α={$\alpha_1$, $\alpha_2$, … ,$\alpha_l$}, and its right context is denoted as β={$\beta_1$, $\beta_2$, … , $\beta_r$}. The left context entropy, right context entropy and context entropy of the validating term $t_v$ is defined in (14).

$$\begin{aligned} Entr_L(t_v) &= -\sum_{\alpha_i \in \alpha} p(\alpha_i t_v) \log\, p(\alpha_i t_v); \\ Entr_R(t_v) &= -\sum_{\beta_i \in \beta} p(t_v \beta_i) \log\, p(t_v \beta_i); \quad (14) \\ Entr\,(t_v) &= \min\{\, Entr_L(t_v), Entr_R(t_v)\} \end{aligned}$$

Table 1 lists the joint validation rules combining mutual information and context entropy criterions. $th_{mi}$ and $th_{entr}$ are thresholds we designate to mutual information and context entropy, respectively.

The wrong candidates are removed from the dictionary. The other three types of terms are saved into the candidate dictionary again. After validating terms, concept word generation is restarted again and the concept word acquisition goes into the next iteration.

## 2.3 The HIO Algorithm

The HIO Algorithm is illustrated in Fig.3.

```
HIO Algorithm
Input: un-segmented corpus C, system dictionary, the
    predefined iteration number (including outer iteration and
    inner iteration)
Output: concept words set
1.  Initialization
    a.  identifying numeric-type string and English string;
    b.  extracting concept words matching with specific
        pattern;
    c.  segmenting corpus C with dictionary and Forward
        Maximum Matching algorithm, and estimating initial
        probability of terms
2.  outer iteration
    a.  concept word generation
        I.   estimating term probability and identifying
             term's type using EM-CLS algorithm;
        II.  achieving the optimal segmentation of sentence
             and identifying their type using Viterbi-C/S
             algorithm;
        III. if the number of iteration < the predefined
             number, goto ( I );
    b.  concept word validation
        I.   validating normal candidate terms using mutual
             information and context entropy, identifying their
             type and precluding wrong results
        II.  if the number of iteration < the predefine number,
             goto (2.a)
3.  outputting all normal candidate terms as concept
    words set
```

Fig.3. The HIO Algorithm

## 3. Experimental Result and Error Analysis

Table 1. The Joint Validation of MI and Entropy

| MI($t_v$) | Entr($t_v$) | Term type($t_v$) |
|---|---|---|
| ≥$th_{mi}$ | ≥$th_{entr}$ | correct candidate |
| ≥$th_{mi}$ | <$th_{entr}$ | over-segmented |
| <$th_{mi}$ | ≥$th_{entr}$ | over-combined |
| <$th_{mi}$ | <$th_{entr}$ | wrong candidate |

We adopt a *400M* Chinese corpus extracted from web pages as the experimental corpus. Before running the HIO method, a series of preprocessing operations are performed, which involve recognizing special unknown words such as numeric-type words, English words, etc., acquiring concept words matching with specific context patterns, using forward maximum matching method to initially segment the corpus and estimating the initial probability of terms.

We set the inner iteration to 10 times and the outer iteration to 5 times. When completing the HIO operations, we randomly select 2000 sentences from this corpus. After filtering out many common words such as auxiliary words, adjectives and adverbs, we get many concept words which have higher precision and recall rates as listed in Table 2.

Table 2. The Experimental Result

| length | Count | P (%) | R (%) |
|---|---|---|---|
| 2 | 7782 | 92.2 | 83.4 |
| 3 | 2234 | 86.1 | 69.0 |
| 4 | 1627 | 89.3 | 70.4 |
| 5 | 1893 | 94.7 | 60.3 |
| ≥6 | 856 | 91.6 | 51.1 |
| Sum. | 14392 | 91.2 | 73.1 |

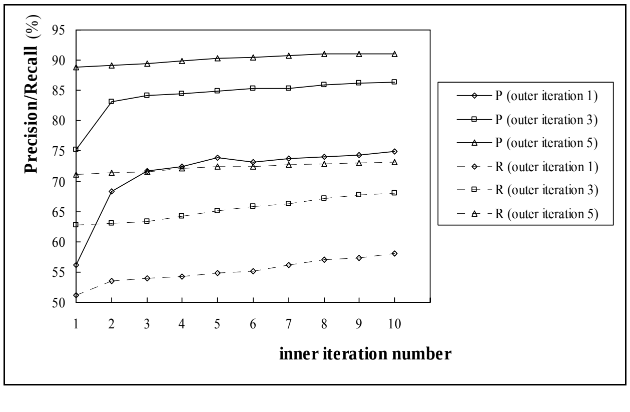

Fig. 4. The Precision and Recall *w.r.t.* the Inner and Outer Iteration

We get 5387 unknown words in total 14392 terms, among which there are 833 bi-gram words, 1593 tri-gram words, 1049 four-gram words, 1196 five-gram words, 716 six- and over-six-gram words.

Fig. 4 shows the effect of the inner and outer iteration on the precision and recall rate of concept word acquisition. It is observed that the precision and recall rate are both increased with the increase of iteration times.

There are two types of errors produced in HIO: commission error and omission error (Yang and Li, 2004). A commission error is that an acquired term is actually not a concept word, but the HIO considers it as a concept word. The reason is that every component of this term is common words and often occurs simultaneously. An omission error is that the HIO misses a concept word in the corpus. The reason is that one component of this word is more commonly used than the rest and the statistical feature of their combination is not prominent. However, the error distribution we get is contrary to the result of Yang and Li (2004). The number of omission errors exceeds that of commission errors, especially in tri-gram concept word.

## 4. Conclusions and Future Work

This paper proposes a hierarchical inner and outer iteration method (HIO) for concept word acquisition. It can deal with the problem of over-segmentation, over-combination and data sparseness produced in the process of acquisition. Its prominent features involve:

(1) The HIO method is the combination of the inner and outer iteration, which can increase the precision and recall rate of concept words acquisition simultaneously.
(2) Concept word generation and validation are uniform and consistent in the HIO method.
(3) The EM-CLS algorithm can classify candidate terms as well as estimate their probability distribution.
(4) The Viterbi-C/S can perform segmenting and combining operations on terms while segmenting corpus.
(5) HIO uses Katz smoothing to lessen data sparseness effect on concept word acquisition.

Now we are going on a series of research on knowledge acquisition from text. The acquired knowledge types include concepts and their relations. Concept word acquisition is fundamental, which can provide essential support for other work in KAT research. We are also developing methods for acquiring relations, including isa, part-of and co-title.

## Acknowledgement

This work is supported by the Natural Science Foundation (grant no. 60273019 and 60496326), and the National 973 Programme (grant no. 2003CB317008).

## References

Christopher C. Yang and K.W. Li. 2003. Segmenting Chinese Unknown Words by Heuristic Method, *ICADL 2003*, LNCS 2911, pp 510–520

Christopher C. Yang and K.W. Li. 2004. Error Analysis of Chinese Text Segmentation using Statistical Approach. *Proceedings of 2004 Joint ACM/IEEE Conference on Digital Library (JCDL' 04)*, Tucson, Arizona, USA, pp256-257

Cungen Cao and Yuefei Sui. 2003. Constructing Ontology and Knowledge Bases from Text. *20th International Conference on Computer Processing of Oriental Languages*, Shenyang, China, pp 34-42

Detlef Prescher. 2003. A Tutorial on the Expectation-Maximization Algorithm Including Maximum-Likelihood Estimation and EM Training of Probabilistic Context-Free Grammars, Presented at the 15th European Summer School in Logic, Language and Information (ESSLLI 2003), Vienna, Austria, August 18-29, 2003

Didier Bourigault and Christian Jacquemin. 1999. TERM EXTRACTION + TERM CLUSTERING: An Integrated Platform for Computer-Aided Terminology. *Proceedings of EACL '99*, pp. 15-22

Fuchun Peng and Dale Schuurmans. 2001 Self-supervised Chinese Word Segmentation. *In Advances in Intelligent Data Analysis (Proceedings of IDA-01)*, pp 238-247

Goodman, J. T. 2001. A Bit of Progress in Language Modeling. *Computer Speech and Language*, 2001 (10), pp 403-434

Guhong Fu and K.K Luke. 2003. A two-stage statistical word segmentation system for Chinese, *Second SIGHAN Workshop on Chinese Language Processing*, Sapporo, Japan, pp. 156-159

Haodi Feng, Kang Chen, Xiaotie Deng, Weimin Zheng. 2004. Accessor Variety Criteria for Chinese Word Extraction, Computational Linguistics, 30 (1), pp. 75-93

Jing-Shin Chang and Keh-Yih Su. 1997. An Unsupervised Iterative Method for Chinese New Lexicon Extraction, *Computational Linguistics and Chinese Language Processing,* 2 (2), pp 97-148

Keh-Jiann Chen and Ming-Hong Bai. 1998. Unknown word detection for Chinese by a corpus-based learning method, *International Journal of Computational Linguistics and Chinese Language Processing*, 3(1):27–44

Lawrence R. Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in

Speech Recognition. *Proceedings of the IEEE*, 77 (2), pp 257-286

Liu Qun, Zhang Huaping, Yu HongKui, and Cheng Xueqi. 2004. Chinese Lexical Analysis Using Cascaded Hidden Markov Model, *Journal of Computer Research and Development*, 41(8), pp. 1421-1429

Maedche. 2002. Ontology Learning for the Semantic Web. Kluwer Academic Publishers

Mário A. T, Figueiredo. 2004. Lecture Notes on the EM Algorithm, http://www.lx.it.pt/~mtf/learning/aboutEM.pdf

Nagao, M and Mori,S. 1994. A New Method of N-gram Statistics for Large Number of N and Automatic Extraction of Words and Phrases from Large Text Data of Japanese. *COLING-94*

Sproat, R. and Shih, C. 1990. A Statistical Method for Finding Word Boundaries in Chinese Text, *Computer Processing of Chinese and Oriental Languages*, 1990 (4), pp.336–351

Tung, Cheng-Huang and Hsi-Jian Lee. 1994. Identification of Unknown Words from a Corpus, *Computer Processing of Chinese and Oriental Languages*, Vol. 8, pp. 131-145

Yusheng Lai and ChungHsien Wu. 2000. Unknown Word and Phrase Extraction Using a Phrase-Like-Unit-Based Likelihood Ratio. *International Journal of Computer Processing of Oriental Languages*, 13 (1), pp 83–95

Yu-sheng Lai and Chung-hsien Wu. 2002. Meaningful Term Extraction and Discriminative Term Selection in Text Categorization via Unknown-Word Methodology. *ACM Transactions on Asian Language Information Processing*, Vol. 1, No. 1, March 2002, pp 34-64

Zhang Le, Lv Xue-qiang, Shen Yan-na and Yao Tian-shun. A Statistical Approach to Extract Chinese Chunk Candidates from Large Corpora. *ICCPOL 2003*, pp109-117, 2003