A Joint Segmenting and Labeling Approach for Chinese Lexical Analysis

Xinhao Wang, Jiazhong Nie, Dingsheng Luo, and Xihong Wu*

Speech and Hearing Research Center,
Key Laboratory of Machine Perception (Ministry of Education),
School of Electronics Engineering and Computer Science,
Peking University,
100871, Beijing, China
{wangxh,niejz,dsluo,wxh}@cis.pku.edu.cn

Abstract. This paper introduces an approach which jointly performs a cascade of segmentation and labeling subtasks for Chinese lexical analysis, including word segmentation, named entity recognition and part-of-speech tagging. Unlike the traditional pipeline manner, the cascaded subtasks are conducted in a single step simultaneously, therefore error propagation could be avoided and the information could be shared among multi-level subtasks. In this approach, Weighted Finite State Transducers (WFSTs) are adopted. Within the unified framework of WFSTs, the models for each subtask are represented and then combined into a single one. Thereby, through one-pass decoding the joint optimal outputs for multi-level processes will be reached. The experimental results show the effectiveness of the presented joint processing approach, which significantly outperforms the traditional method in pipeline style.

Keywords: WFSTs, Chinese lexical analysis, joint segmentation and labeling.

1 Introduction

The Chinese lexical analysis involves solving a cascade of well-defined segmentation and labeling subtasks, such as word segmentation, named entity recognition and part-of-speech (POS) tagging. Like many problems in natural language processing, the cascade is traditionally processed in a pipeline manner. However, it has the disadvantage that errors introduced by earlier subtasks propagate through the pipeline and will never be recovered in downstream subtasks. Moreover, this manner prevents information sharing among multi-level processes. For example, the POS information is helpful to make better prediction in word segmentation and named entity recognition, while this is prohibited in pipeline processing.

To tackle these problems, several techniques were proposed recently. Reranking method has been widely applied in a number of different natural language

* Corresponding author.

W. Daelemans et al. (Eds.): ECML PKDD 2008, Part II, LNAI 5212, pp. 538–549, 2008.

processing problems, such as parsing [1,2], machine translation [3] and so on. In handling the cascaded tasks, a k-best list is preserved at each level firstly, and then reranked in the following subtasks [4,5,6]. Nevertheless, as an approximation of joint processing, reranking may miss the true result, which usually lies out of the k-best list. Another intuitive approach is to take multiple subtasks as a single one [7,8]. Such as in [9], the constituent labels of the Penn TreeBank are augmented with semantic role labels (SRL), thus parsing the trees also serves as a SRL system. Similarly in [10,11], word segmentation and POS tagging are performed simultaneously by marking each Chinese character with a word level POS tag. But an obvious obstacle of these label transformations is the requirement of corpus annotated with multi-level information, which is usually unavailable in many situations. Unlike the strategies mentioned above, some unified probabilistic models are proposed to process a cascade jointly. Sutton et al. [12] proposed the Dynamic Conditional Random Fields (DCRFs), which are trained jointly and performs the subtasks in one step, but it is expensive in training and exact inference. Moreover in [13], the Factorial Hidden Markov Model (FHMM) was also introduced to the joint labeling tasks of POS tagging and noun phrase chunking. Compared with DCRFs, FHMM has the computational advantage as a generative model, and the exact inference can be achieved easier. However, both DCRFs and FHMM also suffer from the absence of multi-level annotated corpus.

In this paper, based on Weighted Finite State Transducers (WFSTs), an integrated Chinese lexical analyzer is presented to jointly perform the cascade of segmentation and labeling tasks, including word segmentation, named entity recognition and part-of-speech tagging. Traditionally, WFSTs have already been successfully used in various fields of natural language processing, such as partial parsing [14], named entity recognition [15], semantic interpretation [16], as well as Chinese word segmentation [17,18]. However, being different from those applications, this study employs WFSTs to jointly conduct segmentation and labeling tasks. WFSTs turn to be an ideal choice for our purpose due to two following remarkable features: On one hand, most of the widely used models, like lexical constraints, n-gram language model and Hidden Markov Models (HMMs), can be encoded into WFSTs, and thus a unified transducer representation for these models is able to be achieved. On the other hand, since there exist mathematically well-defined operations to integrate multiple transducers into a single composed one, the optimal candidate can be extracted by one-pass decoding with multi-level knowledge sources represented by each transducer. In contrast to the joint processing techniques mentioned above, the presented approach has the following advantages. Firstly, rather than reranking the k-best candidates preserved at each level, it holds the full search space and chooses the optimal results based on the multi-level sources. Secondly, similar to the strategy of [19], the models for each level subtask are trained separately, while the decoding is conducted jointly. Accordingly, it avoids the necessary of corpus annotated with multi-level information. Other than [19], in this study the used generative models bring the benefit in computation, which is important in a joint processing

task, especially as the scale of subtasks increasing. Thirdly, in the case when a segmentation task precedes a labeling task, the consistency restriction imposed by the segmentation task must be maintained in the successive labeling task. For instance, the POS tags assigned to each character in a segmented word must be the same. While for the methods taking the smallest characters in the segmentation task as modeling units, such as Chinese Characters in Chinese word segmentation, this restriction is not naturally satisfied. The WFSTs based approach ensures this restriction by the composition operation, i.e., the input sequence of one transducer and the output sequence of the other transducer must be identical. In addition, the unified framework of WFSTs provides the opportunity to easily apply the presented analyzer in other natural language related applications which are also based on WFSTs, such as speech recognition [20] and machine translation [21]. Since more linguistic knowledge in multi-level is modeled by the analyzer, performance improvements possibly can be achieved for those applications.

The remainder of this paper is structured as follows. Section 2 introduces the formal definition and notation of WFST. In section 3, by describing the integrated Chinese lexical analyzer in detail, the joint segmenting and labeling approach is presented. Then simulations are performed to evaluate the new analyzer in section 4. Finally, section 5 draws the conclusion and discusses the future work.

2 Weighted Finite State Transducers

The Weighted Finite State Transducer (WFST) is the generalization of the finite state automata. In weighted transducer, besides of an input label, an output label and a weight are also placed on each transition. With these labels, the transducer is capable of realizing a weighted relation between strings. In the most general case, the definition of WFST depends on the algebraic structure of a semiring,$(\mathbb{K}, \oplus, \otimes, \overline{0}, \overline{1})$ [22,23,24]. In a semiring, two operation $\oplus$ and $\otimes$ are associative and closed over the set $\mathbb{K}$. $\overline{0}, \overline{1}$ are their identities respectively. Unlike a ring, a semiring may not have negation. For example, $(\mathbb{N}, +, *, 0, 1)$ is a semiring.

2.1 Definition

In this paper, a weighted transducer W over the semiring $\mathbb{K}$ is formally defined as a 6-tuple $W = (Q, i, F, \Sigma, \Delta, T)$, where

- Q is the set of its states,
- $i \in Q$ is an initial state,
- $F \in Q$ is the set of final states,
- Σ is the input alphabet,
- Δ is the output alphabet,
- $T \subset Q \times Q \times \Sigma \cup \{\epsilon\} \times \Delta \cup \{\epsilon\} \times \mathbb{K}$ is the set of transitions. Each $t \in T$ consists of a source state $src(t) \in Q$, a destination state $des(t) \in Q$, an input label $in(t) \in \Sigma \cup \{\epsilon\}$, an output label $out(t) \in \Delta \cup \{\epsilon\}$ and a weight $wght(t) \in \mathbb{K}$.

A successful path π in W, is a chain of successive transitions: $p = t_1 t_2 ... t_n$, satisfying $src(t_1) = i$, $des(t_n) \in F$ and $des(t_i) = src(t_{i+1}) \quad 1 \leq i \leq n-1$. The input and output strings mapped by this path are the concatenation of transitions' input and output labels: $in(p) = in(t_1)in(t_2)...in(t_n)$, $out(p) = out(t1)out(t_2)...out(t_n)$, where the symbol ϵ represents the empty string. The weight of π is the $\otimes$-product of its transitions' weights: $wght(\pi) = wght(t_1) \otimes wght(t_2) \otimes ... \otimes wght(t_n)$. For a given input string s and an output string r, the set $path_W(s, r)$ consists of all the successful paths in W, whose input and output strings match s and r respectively. The weight associated by W to the (s, r) is the $\oplus$-sum of the paths' weights in $path_W(s, r)$ and $\overline{0}$:

$$W(s, r) = (\sum_{p \in path_W(s,r)} wght(p)) \oplus \overline{0}$$

In our system, probabilities are adopted as weights and each string pair is associated with a weight indicating the probability of the mapping between them. Due to the numerical stability, log probabilities are used in implementation instead of probabilities. The appropriate semiring for the finite-state representation of log probabilites and operations is the tropic semiring $(\mathbb{R}_+ \cup \{\infty\}, min, +, \infty, 0)$ [22].

2.2 Decoding and Composition

Given an input string s and a WFST W, the goal of decoding is to find the best output string r^* maximizing $W(s, r)$. Similarly, when multiple WFSTs are involved, a joint decoding is desired to find the optimal final output string r^*, which maximizes the $\otimes$ -product of each mapping $W_1(s, m_1) \otimes W_i(m_{i-1}, m_i) \otimes ... \otimes W_n(m_{n-1}, r)$, where m_i is an arbitrary string on W_i's output alphabet. An efficient way to implement this desired decoding is using the composition algorithm to combine multiple WFSTs into a single one [22,23,24].

For two WFSTs E and F satisfying the input alphabet of F and output alphabet of E are the same, the composition $G = E \circ F$ represents the composition of the weighted relations realized by E and F. As in the classical finite state automata intersection, the states in G are pairs of states in E and F. G's initial state is the pair of initial states of E and F, and final states are pairs of a final state in E and a final state in F. For each pair of transition t_E from e to e' in E and transition t_F from f to f' in F, there exists exactly one transition t in G from (e, f) to (e', f'). The input label of t is taken from t_E and output label from t_F. Wght(t) is the $\otimes$ -product of wght(t_E) and wght(t_F), when the weights correspond to probabilities. For transducers have ϵ transitions, special treatments are needed as in [25]. Figure 1 shows two simple transducer Figure 1(a) and Figure 1(b), and the result of their composition, Figure 1(c). All of them are defined on the tropic semiring.

Apparently, for a path in E mapping s to v and a path in F mapping v to r, G has exactly one path mapping s to r directly and its weight is the $\otimes$ -product of the corresponding paths' weights in E and F. This property enables us to find

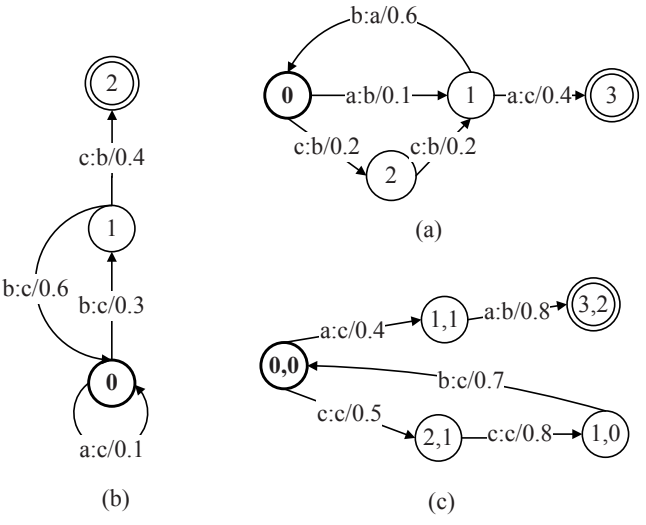

Fig. 1. Example of WFSTs composition. Two simple WFSTs are showed in (a) and (b), in which states are represented by circles and labeled with their unique numbers. The bold circles represent initial states and double circles for final states. The input and output labels as well as weight of transition t are marked as in(t):out(t)/wght(t). In (c), the composition of (a) and (b) is illustrated

the optimal final output string among multiple WFSTs by finding the optimal output string in the combination of these WFSTs $W = ((W_1 \circ W_2)...\circ W_n)$,which can be easily realized by a standard Viterbi search.

3 Joint Chinese Lexical Analysis

Word segmentation is the first stage of Chinese text processing since Chinese is typically written without blanks, and POS tagging is to assign the part-of-speech to each segmented word in a sentence. Both tasks face the challenge of tackling unknown words that are out of the dictionary. In this study, two kinds of typical unknown words, person names and location names, are also focused.

In this section, within the unified framework of WFSTs, the models for three level subtasks, i.e. words segmentation, POS tagging and named identity recognition, are presented, and then they are combined to reach an integrated Chinese lexical analyzer.

3.1 Multiple Subtasks Modeling

For word segmentation, the class based n-gram technique is adopted. Given an input character sequence, it is encoded by a finite state acceptor FSA_{input}. For example, the input "合成分子时"(while synthesizing molecule) is represented as Figure 2(a). Then a dictionary can be represented by a transducer with empty

Table 1. Toy dictionary

Chinese Words	English Words
合	together
合成	synthesize
成分	element
分子	molecule
子时	the period of the day from 11 p.m.to l a.m.
时	present

Table 2. Definition of word classes

Classes	Description
w_i	Each word w_i listed in dictionary
CNAME	Chinese person names as one class
TNAME	Translated person names as one class
LOC	Location names
NUM	Number expressions
LETTER	Letter strings
NON	Other non Chinese character strings
BEGIN	Beginning of sentence as one class
END	End of sentence as one class

weights, denoted as FST_{dict}. Figure 2(b) illustrates a toy dictionary listed in Table 1, in which a successful path encodes a mapping from a Chinese character sequence to some word in dictionary. Afterwards a class based n-gram language model is used to weight the candidate segmentations. In Figure 2(c), a toy bigram with three words is depicted by $WFST_{n-gram}$, and the word classes are defined in Table 2.

For both POS tagging and named entity recognition, the Hidden Markov Model (HMM) is used. Each HMM is represented with two WFSTs. Taking the POS tagging as an example, Figure 3(a) models the generation of words by POS ($P(word/pos)$), and similar to the word n-gram, Figure 3(b) models the transitions between POS tags. For named entity recognition, the HMM states correspond to 30 named entity role tags, such as surname, the first character of a given name with two characters, the first character of a location name, and so on.

Besides the primary WFSTs described above, there are also some other finite state transducers, which are used to represent various rules for recognizing the number strings and letter strings, or to be responsible for the transformation from name roles to word classes.

3.2 Integration of Multiple Models

Based on the WFSTs built above, an integrated model is obtained by combining them into a single one using the composition algorithm as describe in section 2.

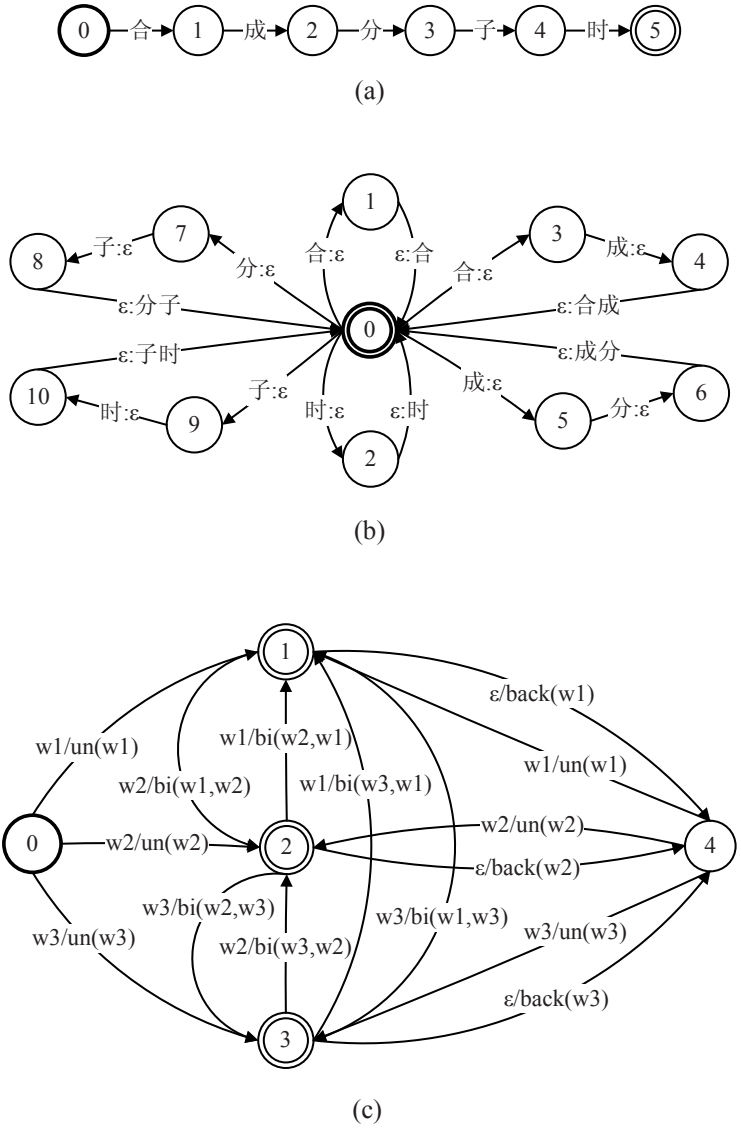

Fig. 2. Word WFSTs. (a) is the FSA representing an input example; (b) is the FST representing a toy dictionary; and (c) is the WFSA representing a toy bigram language model, where $un(w1)$ denotes the unigram of $w1$; $bi(w1, w2)$ and $back(w1)$ respectively denotes the bigram of $w2$ and the backoff weight given the word history $w1$.

To perform word segmentation, a WFST embracing all the possible candidates is obtained as below:

$$WFST_{words} = FSA_{input} \circ FST_{dict} \circ WFST_{ne} \circ WFSA_{n-gram} \tag{1}$$

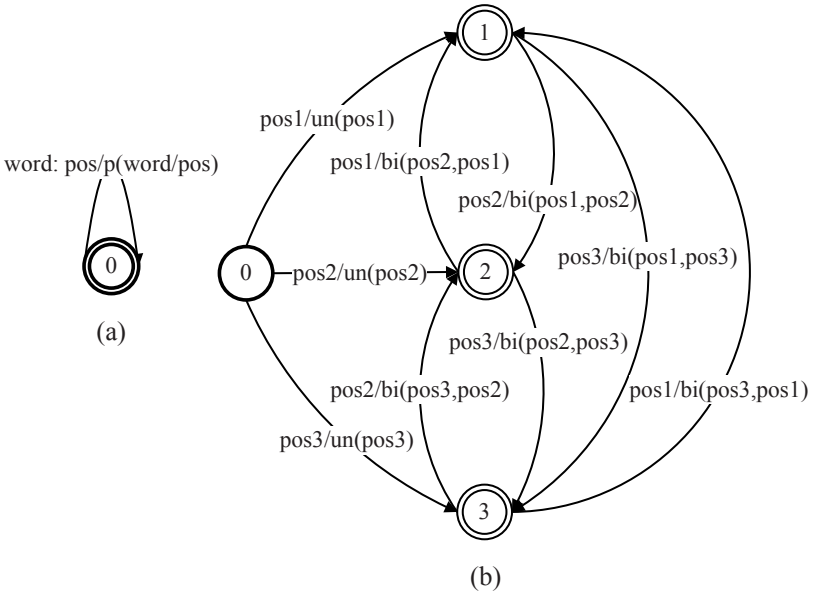

Fig. 3. POS WFSTs. (a) is the WFST representing the relationship between the word and the pos; (b) is the WFSA representing a toy bigram of POS.

As the class based n-gram is adopted, the named entity recognition is conducted along with word segmentation. Taking the POS tagging into account, the decoding, which aims to extract the joint optimal results according to multi-level information, is performed on the WFST composed as following, where α is a weight for combining different level subtasks.

$$WFST_{analyzer} = (\alpha * WFST_{words}) \circ WFST_{POS}. \tag{2}$$

4 Simulation

To evaluate the presented analyzer, two systems are constructed, as illustrated in Figure 4, where the system based on the pipeline style method is taken as the baseline. The experimental corpus comes from the People's Daily of China in 1998 from January to June, annotated by the Institute of Computational Linguistics of Peking University[1]. The January to May data are taken as the training set. The first two thousand sentences of June data are extracted as the develop set, which is used to fix the composition weight α in equation 2, and the remains are taken as the test set. A dictionary including about 113,000 words is extracted from the training data. The models for different level subtasks are trained separately, where the class based language model is trained with the

[1] http://icl.pku.edu.cn/

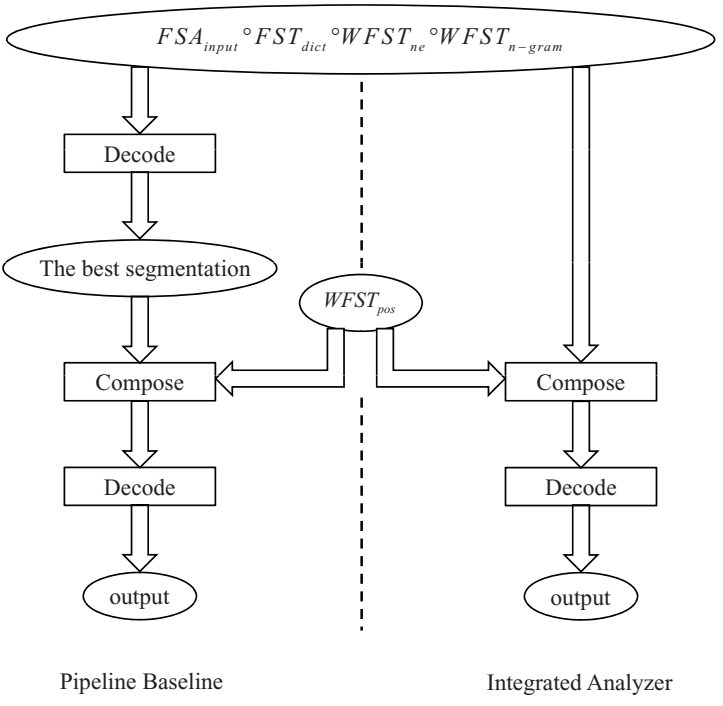

Fig. 4. The pipeline system vs The joint processing system

SRI Language Modeling Toolkit[2]. While the decoding is implemented through one-pass Viterbi search on the combined WFST.

In Table 3 the performances of the pipeline baseline and the integrated analyzer are compared. Due to the joint decoding, the integrated analyzer outperforms the pipeline baseline on all the tasks in F1-score metric, especially for the case of person name recognition. It is as expected that person names are greatly improved when incorporating POS information during recognition, where the POS appears effective in preventing to segment a person name into pieces (possibly into characters).

In addition, to further investigate the significance of performance improvement, a statistical test using approximate randomization approach [26] is performed on the word segmentation results. In this approach, the responses for each sentence produced by two systems are shuffled and equally resigned to each system, and then the significance level is computed as to whether shuffles bring differences not smaller than the difference produced when running two systems on the test data. In general, given n test sentences, the shuffle times s is fixed as in equation 3, i.e., when n is small, no larger than 20, the exact randomization

[2] http://www.speech.sri.com/projects/srilm/

Table 3. Performance comparison between the pipeline baseline and the integrated analyzer. The system performances are measured with F1-score in the tasks of word segmentation (WS), POS tagging, as well as the person and location name recognition.

	Pipeline Baseline	Integrated Analyzer
Word Segmentation	95.94%	96.77%
POS Tagging	91.06%	91.81%
Person Name Recognition	83.31%	88.51%
Location Name Recognition	89.90%	90.91%

is performed, otherwise, only the approximate randomization is performed since the number of different shuffle ways, 2^n, is too large to be exhaustively evaluated.

$$s = \begin{cases} 2^n, & n \leqslant 20 \\ 2^{20} = 1048576, & n > 20 \end{cases} \tag{3}$$

However, in our test set there are more than 21,000 sentences, the use of 2^{20} shuffles to approximate 2^{21000} shuffles as in formula 3 turns unreasonable any more. Thus, here ten sets (500 sentences for each) are randomly selected from the test corpus. For each set, we run 1048576 shuffles twice and calculate the significance level $p-value$ according to the shuffled results. Statistical test shows that all $p - values$ are less than 0.001 on the ten sets, which reveals that the performance improvement introduced by the integrated analyzer is statistically significant.

5 Conclusion and Future Work

In this research, within the unified framework of WFSTs, a joint processing approach is presented to perform a cascade of segmentation and labeling subtasks. It has been demonstrated that the joint processing is superior to the traditional pipeline manner. The finding suggests two directions for future research: Firstly, more linguistic knowledge will be integrated in the analyzer, such as organization name recognition and shallow parsing. For some tough tasks in related areas, such as large vocabulary continuous speech recognition and machine translation, rich linguistic knowledge will play an important role, thus incorporating our integrated lexical analyzer may lead to a promising performance improvement, and these attempts will be another future work.

Acknowledgments

The authors would like to thank three anonymous reviewers for their helpful comments. This work was supported in part by the National Natural Science Foundation of China (60435010; 60535030; 60605016), the National High Technology Research and Development Program of China (2006AA01Z196; 2006AA010103), the National Key Basic Research Program of China (2004CB318005), and the

New-Century Training Program Foundation for the Talents by the Ministry of Education of China.

References

1. Charniak, E., Johnson, M.: Coarse-to-fine n-best parsing and maxent discriminative reranking. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Michigan, pp. 173–180 (June 2005)
2. Collins, M., Koo, T.: Discriminative reranking for natural language parsing. Computational Linguistics 31(1), 25–70 (2005)
3. Shen, L., Sarkar, A., Och, F.J.: Discriminative reranking for machine translation. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Boston, Massachusetts, pp. 177–184 (May 2004)
4. Shi, Y., Wang, M.: A dual-layer CRFs based joint decoding method for cascaded segmentation and labeling tasks. In: Proceedings of the International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 1707–1712 (January 2007)
5. Sutton, C., McCallum, A.: Joint parsing and semantic role labeling. In: Proceedings of the 9th Conference on Computational Natural Language Learning, Ann Arbor, Michigan, pp. 225–228 (June 2005)
6. Zhang, H., Yu, H., Xiong, D., Liu, Q.: Hhmm-based chinese lexical analyzer ictclas. In: Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing, Sapporo Japan, pp. 184–187 (July 2003)
7. Luo, X.: A maximum entropy chinese character based parser. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan, pp. 192–199 (July 2003)
8. Miller, S., Fox, H., Ramshaw, L., Weischedel, R.: A novel use of statistical parsing to extract information from text. In: Proceedings of the 6th Applied Natural Language Processing Conference, Seattle, Washington, pp. 226–233 (April 2000)
9. Yi, S.T., Palmer, M.: The integration of syntactic parsing and semantic role labeling. In: Proceedings of the 9th Conference on Computational Natural Language Learning, Ann Arbor, Michigan, pp. 237–240 (June 2005)
10. Nakagawa, T., Uchimoto, K.: A hybrid approach to word segmentation and pos tagging. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics: Demo and Poster Sessions, Prague, pp. 217–220 (June 2007)
11. Ng, H.T., Low, J.K.: Chinese part-of speech tagging: one-at-a-time or all-at-once? word-based or character-based? In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, pp. 277–284 (July 2004)
12. Sutton, C., Rohanimanesh, K., McCallum, A.: Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. In: Proceedings of The 21st International Conference on Machine Learning, Banff, Alberta, Canada, pp. 783–790 (July 2004)
13. Duh, K.: Jointly labeling multiple sequences: a factorial hmm approach. In: Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Ann Arbor, Michigan, pp. 19–24 (June 2005)
14. Abney, S.: Partial parsing via finite-state cascades. Natural Language Engineering 2(4), 337–344 (1996)

15. Friburger, N., Maurel, D.: Finite-state transducer cascades to extract named entities in texts. Theoretical Computer Science 313(1), 93–104 (2004)
16. Raymond, C., de ric Bechet, F., Mori, R.D., raldine Damnati, G.: On the use of finite state transducers for semantic interpretation. Speech Communication 48(3-4), 288–304 (2006)
17. Sproat, R., Shih, C., Gale, W., Chang, N.: A stochastic finite-state word-segmentation algorithm for chinese. In: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, pp. 66–73 (June 1994)
18. Sproat, R., Shih, C., Gale, W., Chang, N.: A stochastic finite-state word-segmentation algorithm for chinese. Computational Linguistics 22(3), 377–404 (1996)
19. Sutton, C., McCallum, A.: Composition of conditional random fields for transfer learning. In: Proceedings of the Joint Conference of Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing, Vancouver, pp. 748–754 (October 2005)
20. Mohri, M., Pereira, F., Riley, M.: Weighted finite-state transducers in speech recognition. Computer Speech and Language 16(1), 69–88 (2002)
21. Tsukada, H., Nagata, M.: Efficient decoding for statistical machine translation with a fully expanded wfst model. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain, pp. 427–433 (July 2004)
22. Mohri, M.: Finite-state transducers in language and speech processing. Computational Linguistics 23(2), 269–311 (1997)
23. Kuich, W., Salomaa, A.: Semirings, Automata, Languages. Monographs in Theoretical Computer Science. An EATCS, vol. 5. Springer, Heidelberg (1986)
24. Berstel, J., Reutenauer, C.: Rational Series and Their Languages. Springer, Berlin (1988)
25. Mohri, M., Pereira, F.C.N., Riley, M.: he design principles of a weighted finite-state transducer library. Theor. Comput. Sci. 231(1), 17–32 (2000)
26. Yeh, A.: More accurate tests for the statistical significance of result differences. In: Proceedings of the 18th International Conference on Computational Linguistics, Saarbrücken, pp. 947–953 (August 2000)