Improved Joint Kazakh POS Tagging and Chunking

Hao Wu [1, 2], Gulila Altenbek [1, 2]

[1] College of Information Science and Engineering, Xinjiang University , Urumqi, China

[2] The Base of Kazakh and Kirghiz Language of National Language Resource Monitoring and Research Centre Minority Languages, Urumqi, China

{18299151954,glaxd2014}@163.com

Abstract. This paper describes a mixing model of joint POS tagging and chunking for Kazakh where partial optimal solution provide feature information for joint model. A improved beam-search algorithm use dynamic beam instead of unified beam to obtain search space of small-but-excellent during both training and decoding phases of the model. Moreover we can statistical induction the information of chunk to dis-ambiguation of multi-category words and experiment shows the precision is improved from 81.6% to 87.7% by information of chunk.

Keywords: mixing model; joint model; dynamic beam; multi-category words

1 Introduction

The tasks of Kazakh part-of-speech (POS) tagging and chunking have been widely investigated since the early stages of NLP research. Meanwhile, this task is a the key techniques needed for automatic text categorization, topic extraction patent information retrieval and other text un-derstanding tasks and is very important for to the practice on information retrieval, machine tr-anslation, dictionary compiling, and other natural language processing tasks. At the present sta-ge, the Kazakh part-of-speech (POS) tagging and chunking is needed before Kazakh corpus an-notation and further processing at a later stage and is an important part of Kazakh Language Processing. The realization of the tasks will defnitely promote the research of linguistics and translation theory of the languages of minorties.Typically, POS tagging and chunking are mo-deled in a pipelined way. However, the piplined method is prone to error propagation. The s-tate-of-the-art accuracy of Kazakh POS tagging is about 89.3% [1], which is much lower than that of English (about 97%)[2] and Chinese(about 93.5). Moreover POS tagging errors cannot be corrected by tagging and structure of chunk.

In order to avoid error propagation and make use of chunk information for POS tagging, POS tagging and chunking can be viewed as a single task: given a raw Kazakh input sentence, the joint POS tagger considers all possible tagged and sequences, and chooses the overall best output. A major challenge for such a joint system is the large search space faced by the decoder. To deal with the increased search space, we adopt a recently proposed beam-search extension to shift-reduce joint model[5-8], which enables the model to pack equivalent tagger states, improving both speed and accuracy.

In this paper, we propose the mixing model which is acquire partial optimal solution to provide some feature in order to the joint model. For example, we obtain the unlabeled chunk to provide the coarse

structure of sentence firstly. Moreover for further improving both speed and accuracy of decoder, we propose the search space dynamic beam to obtain the small-but-excellent as far as possible. This provides the rich feature information, leading to large improvements over using either the generative model or the discriminative model in term of POS tagging (the mixing model achieves 89.1% precision on the Xingjiang Daily Tagging set, compared to figures of 81.4% and 86.1% for the HMM molde and ME Models). In term of chunking, the precision is large improvements over using either CRF model or ME model. The remaining part of the paper is organized as follows. Section 2 gives a brief introduction to the foregoing method of POS tagging and chunk and propose the mixing model. Section 3 describes the perceptron algorithm and decode process. Section 4 describes the disambiguation of multi-category words and the setting of feature. Section 5 presents experimental results and empirical analyses. Section 6 conclude the paper.

2 The Analyze Models

2.1 The Pipelined Method

Given an input sentence x = w1...wn, we denote its POS tagging sequence by t = $t_1...t_n$, where ti $\in$ T, $1 \le i \le n$, and T is the POS tagging set. A chunk sequence is denoted by d = {(b,e,ct) : $0 \le h \le n$, $0 < m \le n$,ct $\in$ CT}, where CT is the chunk tags set, (b,e) represents a chunk (wb...we) whose first word of chunk is wb and last word is we. The pipelined method treats POS tagging and chunking as two cascaded problems. First, an optimal POS tagging sequence $\hat{t}$ is determined.

$$\hat{t} = \arg\max t \ \mathrm{Score}_{pos}(x, t) \qquad (1)$$

In a perceptron, the score of a tag sequence is:

$$\mathrm{Score}_{pos}(x, t) = w_{pos} \cdot f_{pos}(x, t) \qquad (2)$$

Where $f_{pos}(x, t)$ refers to the feature vector and w_{pos} is the corresponding weight vector. Then, an optimal chunking $\widehat{ct}$ is determined based on x and $\hat{t}$.

$$\widehat{ct} = \arg\max_{ct} \ \mathrm{Score}_{cnk}(x, ct) \qquad (3)$$

Similar to POS tagging, the score a chunking sequence is

$$\mathrm{Score}_{cnk}(x, \hat{t}, ct) = w_{cnk} \cdot f_{cnk}(x, \hat{t}, ct) \qquad (4)$$

2.2 The Joint Models

In the joint method[5-8], we aim to simultaneously solve the two problems.

$$(\hat{t}, \widehat{ct}) = \arg\max t, ct \ \mathrm{Score}joint(x, t, ct) \qquad (5)$$

Under the linear model, the score of a tagged sentence sequence is:

$$\mathrm{Score}_{joint}(x, t, ct) = \mathrm{Score}_{pos}(x, t) + \mathrm{Score}_{cnk}(x, t, ct) = w_{pos\oplus cnk} \cdot f_{pos\oplus cnk}(x, t, d) \qquad (6)$$

For simplicity, we denote

$$w_{pos} \cdot f_{pos}(x, t) + w_{cnk} \cdot f_{cnk}(x, \hat{t}, ct) = w_{pos\oplus cnk} \cdot f_{pos\oplus cnk}(x, t, d)$$

where $w_{pos\oplus cnk}$ means the concatenation of weights of feature.

Under the joint model, the weights of POS and chunk features, $w_{pos\oplus cnk}$ are simultaneously learned. We expect that POS and chunk features can interact each other to determine an optimal joint result.

2.3 The Mixing Model

In section 2.1, we know that POS tagging and chunking are modeled in a pipelined way. However, the pipelined method is prone to error propagation. The state-of-the-art accuracy of Kazakh POS tagging is about 89.3%, which is much lower than that of English and Chinese. Simultaneously, Joint approach is that since the shift-reduce model processes an input sentence in a right-to-left manner[9], it cannot exploit look-ahead POS tags, which a pipeline shift-reduce parser can consider, to determine the next action. In experiment, the ablation of the features including look-ahead POS results about 1.1% decrease in POS tagging performance on the development set, suggesting that the look-ahead POS information is indispensable to achieve the state-of-the-art performance.

In order to make up for shortcomings pipelined method and joint method, we combine the two approaches. First, before we analyze the sentence, we use the method of segmentation with perceptron algorithm (Yue Zhang and Stephen Clark,2007) to identify sentence chunks(not give a label). We follow the convention of word-based chunking, and define the set of chunks tags as {B, E, M, S}. The tags B, E, M represent the character being the beginning, end, and middle of a multiple-word chunk, respectively, and the tags S represents the character being a single-word chunk.

For example: ورىندىق/Eارقالى/M مەن /Mۇستەل/B (table and chair).

Then we have improved shift-reduce process[9], and show in **Fig. 1**. Note that the transfer system is composed of four states and four actions. Four states, respectively, comprise one queue, two linked list and a stack, the input is assumed to sentence, and the word waiting to be processed are stored in first queue, the first linked list holds the partial POS tagging that are built during the labeling process, the chunk of none tag waiting to be processed are stored in second linked list, the s tack holds the partial POS tagging and chunk that are built during the labeling process.

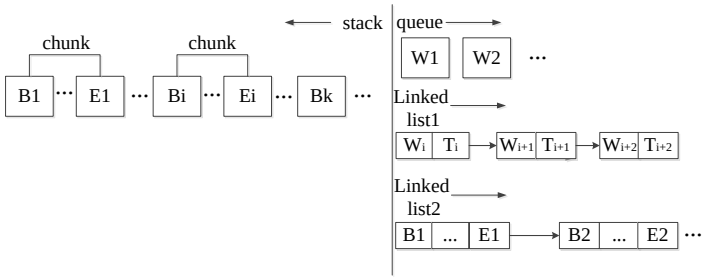

Fig. 1. The improved shift-reduce process

The main shift-reduce actions are:

- SHIFT1, which choose the next word from queue to POS tagging,then pushes the word-POS pair in the queue onto the first linked list;
- SHIFT2, which pushes the first word-POS pair in the linked list onto the stack;
- REDUCE, which predict the position of word in chunk(start,middle and end). If the word is start of chunk, predict tag of chunk according the feature. Otherwise the word merge with top element of stack.
- TERMINATE, which check the linked list 2 is empty.

3 Learning and Decoding Algorithm

3.1 Linear Models for NLP

We follow the framework outlined in Collins[2-3]. The task is to learn a mapping from inputs $x \in \chi$ to outputs $y \in y$. For example, χ might be a set of sentences, with being a set of possible POS tagging. We assume:

- Training examples （xi,yi） i = 1...n
- A function GEN which enumerates a set of candidates GEN(x) for an input x.
- A representation Φ mapping each $(x,y) \in \chi \times y$ to a feature vector $\Phi(x,y) \in R^d$.
- A parameter vector $\bar{\alpha} \in R^d$.
- The components GEN, Φ and $\bar{\alpha}$ define a mapping from an input x to an output F(x) through

$$F(x) = \arg \max_{y \in GEN(x)} \Phi(x,y) \cdot \bar{\alpha} \qquad (7)$$

Where $\Phi(x,y) \cdot \bar{\alpha}$ is the inner product $\sum_s \alpha_s \Phi_s(x,y)$,The learning task is to set the parameter values $\bar{\alpha}$ using the training examples as evidence. The decoding algorithm is a method for searching for the arg max in Eq.7.

3.2 The Perceptron Algorithm

In the 3.1 section,we assume the $\bar{\alpha}$ is the parameter vector in the model. Each element in $\bar{\alpha}$ gives a weight to its corresponding element in Φ(x,y), which is the count of a particular feature over the whole sentence y. We calculate the $\bar{\alpha}$ value by supervised learning, using the averaged perceptron algorithm[10-12], given in Algorithm 1.

After review the averaged perceptron algorithm, due to its convergence properties have a full description, we now only consider the problem of decode.

Inputs: Training examples (xi; yi)
Initialization: Set $\bar{\alpha} = 0$
Output: Parameters $\bar{\alpha}$
Algorithm:
For $t = 1 \ldots T, i = 1 \ldots n$
Calculate $z_i = \arg\max_{z \in GEN(x_i)} \Phi(x_i, z) \cdot \bar{\alpha}$
if $(z_i \neq y_i)$ then $\bar{\alpha} = \bar{\alpha} + \Phi(x_i, y_i) - \Phi(x_i, z_i)$

Algorithm 1. Averaged perceptron algorithm

```
Variables: state item item = (S,Q), where
     S is stack and Q is incoming queue;
     the agenda agenda;
     list of state items next;
Algorithm:
for item ∈ agenda:
   if item.score = agenda.bestScore and
  item.isFinished:
    rval = item
        break
    next = []
    for move ∈ item.legalMoves:
      next.push(item.TakeAction(move))
agenda = next.getBBest()
Outputs: rval
```

Algorithm 2. Beam-search algorithm

3.3 Decode Algorithm

In section 3.2, we introduced the averaged perceptron algorithm. Note that the most complex step of the method is finding zi = $\arg\max_{y \in GEN(x)} \Phi(x,y) \cdot \bar{\alpha}$ and this is precisely the decodin

g problem[14-15]. In this section, we will introduce an improved beam-search algorithm to imp rove decode performance.

Beam Search Algorithm

First of all, we apply beam-search[9-10], keeping the B highest scoring state items in an agen da during the shift-reduce process. The agenda is initialized with a state item containing the st arting state, i.e. an empty stack and three queue consisting of all word, partial word-POS pairs a-nd all chunk of unlabeled from the sentence.

At each stage in the decoding process, existing items from the agenda are progressed by appl-ying legal shift-reduce actions. From all newly generated state items, the B highest scoring are put back on the agenda. The decoding process is terminated when the highest scored state it em in the agenda reaches the final state. If multiple state items have the same highest score, shift-reduce process terminates if any of them are finished. The algorithm is shown in Algori-thm 2.

Optimized Beam-Search Algorithm

In the averaged beam-search algorithm, choose the B highest scoring state items in an agenda during the shift-reduce process. For uniform B, we select the appropriate value of B is difficu-lt. If we choose the value of B is too large, while decoding accuracy has improved, but the decoding speed is greatly reduced. The value of B is too small, the opposite effect. Therefore we designed a dynamic value B to solve the above problems.

At each step in the decoding process, outputs a set of prediction $\{<(t_1(x), r_1), \ldots, (t_1(x), r_k)>, r_i \geq r_{i+1}\}$, where $t_i(x)$ is a possible outcome of each shift action, r_i is a score of result. Sup-pose that $(< r_1/r_i = Br>, Br \geq 1)$,where Br is a relative score. Note that with a larger value of Br, I have a less possibility to as a gold prediction. Therefore we filter out the result of sli-m chance by setting a reasonable threshold. Moreover, the closer that the value of Br is to 1, the word have greater possibility is a multi-category words in the word tagging process. He-nce, we can set a threshold and mark each words of relative value within the threshold for th-e disambiguation of multi-category words. For chunking, we summary rule to constrain the se-arch space of our models due to their high complexity.

4 A full description of the Tagging Approach

This section gives a full description of the tagging approach. We first describe the disambiguat ion of multi-category words, and then move on to the baseline feature set for the mixing mo-del.

4.1 Disambiguation of Multi-Category Words

In the section 3.3,we introduce the relative score Br and mark each words of relative value w-ithin the threshold. For mark words, we regard it as a multi-category words. Next, we will in troduce the main process disambiguation of multi-category words by using the information of chunking[13].

First of all, we extract the structure of basic chunk from the corpus as a set of rule.

For example, we statistical induction the strucure of noun phrase shown in following:

(1)ەر وقىتۇشىلار (Male teachers, n + n)

(2)ۇستەل جانە ورىندىق (table and chair, n + conj + n)
(3)مەن جانە ونىڭ (he and me, pron + conj + pron)
(4)مەنىڭ كىتاپ (my book, pron + n)
(5)گۈل ادەمى (beautiful flower, adj + n)
(6)قارتتار قۇرمەت ەتۈ (respect senior, v + n)
(7)العان نارسەلەرىنە ٴراسىميات تا گۈلدەۋ(pleased with hold the flower, adj + adv + n)
(8) قول اربا (push cart, v + n)
(9)5 تال گۈل (five flower, num + n)

We now assume w_i is a marked word,a_1、a_2 is ambiguity tag of w_i and t_k is a tag of current chunk that contains the word w_i. Moreover the C_1, C_2 represent the corresponding structure of a_1 and a_2. Then find t_k corresponding rule sets, traverse the set to check C_1 and C_2 whether or not it was in the set.Next we operate in three steps:

1. If there is only one in the set,we think that it is POS tagging of w_i.
2. If neither of them not in the set, we mark the word and the chunk as wrong recognition, then to further identify them by ME(Maximum Entropy) model. For the word, we can use the information of chunk from the context.
3. If all of them in the set, the word will tagged by ME model again. For the two score of each part-of-speech, you might multiply them together. Choose the part-of-speech of the biggest product as tagging of the word.

4.2 Features

For this paper, we wanted to compare the results of a perceptron model with a generative model for a comparable feature set. Unlike in (Roark, 2001a; 2004), there is no look-ahead statistic, so we modified the feature set from those papers to explicitly include the next unlabeled chunk and POS tag of the next word. But for the neighboring chunk, we think that it have less correlation between the last word of first chunk and first word of second chunk, therefore we have not choose it as feature. Otherwise the features are basically the same as in those papers. To concisely present the baseline feature set, let us establish a notation. All of the labels that we will include in our feature sets.Note that we recognize POS and chunk to choose different feature. In terms of feature extraction, we according feature template to choose features set. For POS tagging:

(1) If Wi is first word of the chunk and the chunk and previous chunk is more than one word, the POS tagging feature template shown in **Fig. 2.**(a).
(2) If Wi is last word of the chunk and the chunk and next chunk is more than one word, the POS tagging feature template shown in **Fig. 2.**(b).
(3) If the chunk is only a word. the POS tagging feature template shown in **Fig. 2.**(c).

For chunking: We set the feature template in **Fig. 2.**(d).

Unigrams	S_0w, S_0t, N_0w, N_0t, N_1w, N_1t, P_0t , P_0c, P_0w, P_1t, P_1c, L_1c, L_1w,L_2w.
Bigrams	S_0wN_0w, S_0wN_{0t}, N_0wN_1w, N_0tN_1w, N_0tN_1t, P_0tS_0w, P_0cS_0w, P_0wS_0w, P_0tL_1c, P_0tL_1w, P_1tL_1c, P_0wL_1c, P_0cL_1c
Trigrams	$P_0tS_0wN_0w$, $P_0wS_0wN_0t$, $P_0cS_0wN_0t$, $S_0wN_0tL_1c$, $P_0tN_0wN_1w$, $P_0cN_0wN_1w$, $N_0wN_1wL_1c$, $N_0wN_1wL_1w$

(a)

Unigrams	S_0w, S_0t, S_1w, S_1t, S_2w, S_2t, P_0t , P_0c, P_0w, P_1t, P_1c, L_1c, L_1w,L_2w.
Bigrams	S_0wS_1w, S_0wS_1t, S_1wS_2w, S_1tS_2w, S_1tS_2t, P_0tS_0w, P_0cS_0w, P_0wS_0w, P_0tL_1c, P_0tL_1w, P_1tL_1c, P_0wL_1c, P_0cL_1c
Trigrams	$P_0tS_0wS_1w$, $P_0wS_0wS_1t$, $P_0cS_0wS_1t$, $S_0wS_1tL_1c$, $P_0tS_1wN_1w$, $P_0cS_1wN_1w$, $S_1wS_2wL_1c$, $S_1wS_2wL_1w$

(b)

Unigrams	S_0w, S_0t, P_0t , P_0c, P_0w, P_1t, P_1c, L_1c, L_1w,L_2w.
Bigrams	S_0wS_0t, P_0tS_0t, P_0tS_0w, P_0cS_0w, P_0wS_0w, P_0tL_1c, P_0tL_1w, P_1tL_1c, P_0wL_1c
Trigrams	$P_0tS_0wS_0t$, $P_0wS_0wS_0t$, $P_0cS_0wS0_1t$ $S_0wS_1tL_1c$, $P_0tP_0wP_0c$, $P_0cL_1wL_1c$, $P_0tP_0cL_1c$, $P_0tP_0cL_1w$

(c)

Unigrams	P_0t , P_0c, P_0w, P_1t, P_1c, L_0c, L_0w, L_1c, L_1w,L_2w.
Bigrams	L_0wL_0c, P_0tL_0c, P_0tL_0w, P_0cL_0w, P_0cL_0c, P_0tL_1c, P_1tL_1c, P_0cL_1c
Trigrams	$P_0tL_0wL_0c$, $P_0cL_0wL_0c$, $P_0cP_0tL_0c$ $P_0tP_0cL_1c$, $P_0tP_1tL_0c$, $P_0cP_1tL_0c$, $P_0tP_1cL_1c$, $P_1tP_1cL_1c$

(d)

Fig. 2. Feature template

In those feature template, The current word the symbols S0 represent the current word, S1, S2, and S3 represent the top three nodes on the stack, the symbols N0, N1 and represent the first two words in the incoming linked list1, and symbols P1 and P1 represent the top two chunk on the stack.L0, L1, L2 represents following two unlabeled chunk of the current word. w represents word, c represents structure of chunk, t represents the tag for a word or a chunk.

5 Empirical Result

5.1 Experimental Settings

The experimental corpus from annotated Xingjiang Daily(Kazakh language version), including political, economic, cultural, sports, entertainment, military. The total corpus include 40233 sentences, 231208 words and 83026 chunks.

Precision, recall and F which is used to identify the results of the evaluation. For the Kazak POS tagging result, we evaluate the different tag respectively. Then the evaluation function is defined as follows:

- a = number of correct tagging
- b = number of undiscerned tagging
- c = number of wrong tagging

Then: $P(\text{Precision}) = \frac{a}{a+c} \times 100\%$

$R(\text{Recall}) = \left(\frac{a}{a+b}\right) \times 100\%$

$\text{F-measure} = \left(\frac{2 \times P \times R}{P+R}\right) \times 100\%$

5.2 Pipelined Model and Joint Model Performance

First of all, we evaluate the performance of our pipelined model and joint model annotator de-

scribed in Section 2. The result of pipelined model shown in Table 1. For joint model, based on our preliminary experiments, we both set the beam size to 16 and 32 for the joint annotator, the result shown in Table 2. Compare to the Table 1 and Table 2, we can see that joint model leading to large improvements over pipelined model. Only observe the Table 2, 32 beam size compare to 16 beam size, while the precision improved about 0.6%, the speed decreased approximately 50%.

Table 1. Result of pipelined model

	Precision (%)	Recall (%)	F (%)
POS tagging	89.3	80.5	84.7
Chunking	78.6	71.2	74.7

Table 2. Result of beam size is 16 and 32 for the joint model

beam size	16		32	
	Precision (%)	Speed (s)	Precision (%)	Speed (s)
POS tagging	90.6	40.4 sentence	91.2	23.6 sentence
Chunking	81.6		82.4	

5.3 Development Results

In this paper, we propose the dynamic beam to control the search space. For dynamic beam, we should set the appropriate threshold to improving the efficiency of decoder. In this experiment, we set the threshold to 2, 3, 4, 5, 6, 7, 8, 9 for compare with the performance of annotator in different threshold. The dynamic beam curves of the mixing models are shown in **Fig. 3**. From the **Fig. 4.**,we integrated consideration the precision and the decode speed choose 6 as the threshold. In addition, we conduct a large number of experiment which discern multi-ca-tegory words to confirm the threshold of best performance. From the Figure, we obtain the best threshold is 3.5.

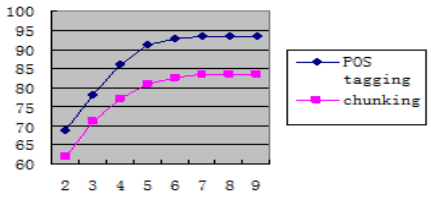

Fig. 3. Accuracies for joint POS tagging and chunking using dynamic beam threshold 2, 3, 4, 5, 6, 7, 8, 9 respectively.

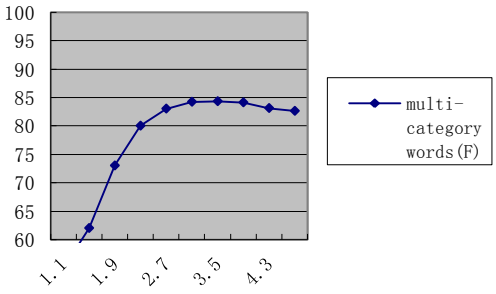

Fig. 4. F-measure for discern multi-category words using the threshold 1.1, 1.5, 1.9, 2.3, 2.7, 3.1, 3.5, 3.9, 4.3 respectively.

In this experiment, recognition rate of multi-category words is improved greatly in the POS tagging. Moreover for output word as error, we correct its tagging and push it into corpus. From the Table 3, we contrast it with previous work to know that chunk provides the rich information, leading to recognition rate of multi-category words large improvements whether in precision or F-measure.

Table 3. Contrast pipelined method with mixing model for recognition rate of multi-category words

	Precision (%)	Recall (%)	F (%)
pipelined method	81.6	75.8	78.6
Mixing model	87.7	81.3	84.3

Table 4. Compare result of mixing model with result of joint model

beam size	Joint method			Mixing model		
	Precision (%)	Recall (%)	F (%)	Precision (%)	Recall (%)	F (%)
POS tagging	90.6	83.4	86.9	92.1	88.5	90.3
Chunking	81.6	76.5	79	83.4	80.8	82.1

5.4 Final Results and Analysis

For show the final result, we contrast previous work with mixing model. First of all, we test the POS tagging and chunking which utilize joint model and mixing model respectively. The contrast result shown in the Table. For the chunking, our result contrast with theresult of shown in Table 4.

It is noteworthy that we obtained the first positive result that the mixing model does improve POS tagging about 2.8%, while, remove 6.1% improved in multi-category words, otherwise only have 1.8% improved. But for the chunking, this is our mixing model is considered to have improved the chunking accuracy over the pipelined tagger about 4.8. Therefore we should consider peculiarity of broader and more in-depth to perfect the feature of POS in order to obtain best performance.

6 Conclusion

We proposed a joint POS tagging and chunking mixing model, which achieved a considerable reduction in error rate compared to a baseline two stage system. We used a single linear model for combined POS tagging and chunking, and chose the generalized perceptron algorithm for joint training. And beam search for efficient decoding and propose the dynamic beam to improve the preformence of decode. Moreover the search space is reduction greatly and precision of multi-category words is improved by set the appropriate threshold.We statistical induction the structural relationship of between the chunk and the word to obtain broader and more in depth feature.

The joint system takes features only from partal context. There may be additional features that are particularly useful to the joint system. Open features, such as knowledge of numbers and relationships from semantic networks [12], have been reported to improve the accuracy of segmentation and POS tagging. Therefore, given the flexibility of the feature-based linear model, an obvious next step is the study of open features in the joint POS tagger and chunk recognition.

References

1. Altenbek, G., Wang, X., & Haisha, G.: Identification of Basic Phrases for Kazakh Language using Maximum Entropy Model. In: COLING, pp. 1007-1014. Dublin (2014)

2. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing, Volume 10 (pp. 1-8). Association for Computational Linguistics. Philadelphia (2-002, July)
3. Collins, M. : Parameter estimation for statistical parsing models: Theory and practice of distribution--free methods. In: New developments in parsing technology, pp. 19-55. Springer Netherlands (2004)
4. Zhang, Y., & Clark, S.: Syntactic processing using the generalized perceptron and beam search. Computational linguistics, 37(1), pp. 105-151. (2011)
5. Hatori, J., Matsuzaki, T., Miyao, Y., & Jun'ichi Tsujii.: Incremental Joint POS Tagging and Dependency Parsing in Chinese. In: IJCNLP, pp. 1216-1224. (2011)
6. Hatori, J., Matsuzaki, T., Miyao, Y., & Tsujii, J.: Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese. In: Meeting of the Association for Computational Linguistics: Long Papers, Vol.1, pp.1045-1053). Jeju (2012)
7. Saraclar, M., & Roark, B.: Joint Discriminative Language Modeling and Utterance Classification. In: CASSP (1), pp. 561-564. (2005)
8. Wang, Z., & Xue, N.. Joint POS Tagging and Transition-based Constituent Parsing in Chinese with Non-local Features. In: ACL (1), pp. 733-742. Maryland(2014)
9. Zhang, Y., & Clark, S.: Transition-based parsing of the Chinese treebank using a global discriminative model. In: International Conference on Parsing Technologies, pp.162-171. Association for Computational Linguistics, Paris (2009)
10. Zhang, Y., & Clark, S.: Chinese Segmentation with a Word-Based Perceptron Algorithm. In: ACL 2-007, Proceedings of the, Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic(2007)
11. Collins, M., & Roark, B.. Incremental Parsing with the Perceptron Algorithm. In: Meeting of the Association for Computational Linguistics, 21-26 July, 2004, pp.111—118. Barcelona (2004)
12. Freund, Y., & Schapire, R. E.: Large margin classification using the perceptron algorithm. Machine Learning, 37(3), pp. 277-296. (1999)
13. Collins, M., & Duffy, N.: New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In: Meeting on Association for Computational Linguistics, pp.263-270. Association for Computational Linguistics. Philadelphia (2002)
14. Daum&#, Iii, H., & Marcu, D.. Learning as search optimization: approximate large margin methods for structured prediction. In: Icml, pp. 169--176. Bonn(2009)
15. Shi, B. Y.: A dual-layer crf based joint decoding method for cascade segmentation and labelling tasks. Proceedings of Ijcai, pp. 1707-1712. (2012)