Chunking Using Conditional Random Fields in Korean Texts

Yong-Hun Lee, Mi-Young Kim, and Jong-Hyeok Lee

Div. of Electrical and Computer Engineering POSTECH and AITrc,
San 31, Hyoja-dong, Nam-gu, Pohang, 790-784, R. of Korea
{yhlee95, colorful, jhlee}@postech.ac.kr

Abstract. We present a method of chunking in Korean texts using conditional random fields (CRFs), a recently introduced probabilistic model for labeling and segmenting sequence of data. In agglutinative languages such as Korean and Japanese, a rule-based chunking method is predominantly used for its simplicity and efficiency. A hybrid of a rule-based and machine learning method was also proposed to handle exceptional cases of the rules. In this paper, we present how CRFs can be applied to the task of chunking in Korean texts. Experiments using the STEP 2000 dataset show that the proposed method significantly improves the performance as well as outperforms previous systems.

1 Introduction

Text chunking is a process to identify non-recursive cores of various phrase types without conducting deep parsing of text [3]. Abney first proposed it as an intermediate step toward full parsing [1]. Since Ramshaw and Marcus approached NP chunking using a machine learning method, many researchers have used various machine learning techniques [2,4,5,6,10,11,13,14]. The chunking task was extended to the CoNLL-2000 shared task with standard datasets and evaluation metrics, which is now a standard evaluation task for text chunking [3].

Most previous works with relatively high performance in English used machine learning methods for chunking [4,13]. Machine learning methods are mainly divided into the generative approach and conditional approach. The generative approach relies on generative probabilistic models that assign a joint probability $p(X,Y)$ of paired input sequence and label sequence, X and Y respectively. It provides straightforward understanding of underlying distribution. However, this approach is intractable in most domains without strong independence assumptions that each input element is independent from the other elements in input sequence, and is also difficult to use multiple interacting features and long-range dependencies between input elements. The conditional approach views the chunking task as a sequence of classification problems, and defines a conditional probability $p(Y|X)$ over label sequence given input sequence. A number of conditional models recently have been developed for use. They showed better performance than generative models as they can handle many arbitrary and overlapping features of input sequence [12].

A number of methods are applied to chunking in Korean texts. Unlike English, a rule-based chunking method [7,8] is predominantly used in Korean because of its well-developed function words, which contain information such as grammatical

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 155 – 164, 2005.

relation, case, tense, modal, etc. Chunking in Korean texts with only simple heuristic rules obtained through observation on the text shows a good performance similar to other machine learning methods [6]. Park et al. proposed a hybrid of rule-based and machine learning method to handle exceptional cases of the rules, to improve the performance of chunking in Korean texts [5,6].

In this paper, we present how CRFs, a recently introduced probabilistic model for labeling and segmenting sequence of data [12], can be applied to the task of chunking in Korean texts. CRFs are undirected graphical models trained to maximize conditional probabilities of label sequence given input sequence. It takes advantage of generative and conditional models. CRFs can include many correlated, overlapping features, and they are trained discriminatively like conditional model. Since CRFs have single exponential model for the conditional probability of entire label sequence given input sequence, they also guarantee to obtain globally optimal label sequence. CRFs successfully have been applied in many NLP problems such as part-of-speech tagging [12], text chunking [13,15] and table extraction from government reports [19].

The rest of this paper is organized as follows. Section 2 gives a simple introduction to CRFs. Section 3 explains how CRFs is applied to the task of chunking in Korean texts. Finally, we present experimental results and draw conclusions.

2 Conditional Random Fields

Conditional Random Fields (CRFs) are conditional probabilistic sequence models first introduced by Lefferty et al [12]. CRFs are undirected graphical models, which can be used to define the joint probability distribution over label sequence given the entire input sequence to be labeled, rather than being directed graphical models such as Maximum Entropy Markov Models (MEMMs) [11]. It relaxes the strong independence assumption of Hidden Markov Models (HMMs), as well as resolves the label bias problem exhibited by MEMMs and other non-generative directed graphical models such as discriminative Markov models [12].

2.1 Fundamentals of CRFs

CRFs may be viewed as an undirected graphical model globally conditioned on input sequence [14]. Let $X=x_1x_2x_3\ldots x_n$ be an input sequence and $Y=y_1y_2y_3\ldots y_n$ a label sequence. In the chunking task, X is associated with a sequence of words and Y is associated with a sequence of chunk types. If we assume that the structure of a graph forms a simple first-order chain, as illustrated in Figure 1, CRFs define the conditional probability of a label sequence Y given an input sequence X by the Hammersley-Clifford theorem [16] as follows:

$$p(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_i \sum_k \lambda_k f_k(y_{i-1}, y_i, X, i) \right) \tag{1}$$

where $Z(X)$ is a normalization factor; $f_k(y_{i-1}, y_i, X, i)$ is a feature function at positions i and $i-1$ in the label sequence; λ_k is a weight associated with feature f_k.

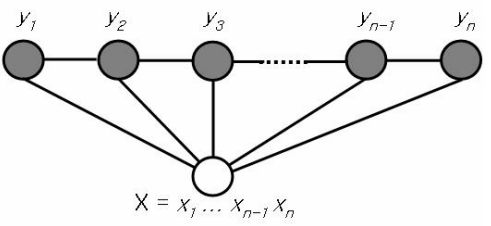

Fig. 1. Graphical structure of chain-structured CRFs

Equitation 1, the general form of a graph structure for modeling sequential data, can be expanded to Equation 2,

$$p(Y \mid X) = \frac{1}{Z(X)} \exp\left(\sum_i \sum_k \lambda_k t_k(y_{i-1}, y_i, X, i) + \sum_i \sum_k \mu_k s_k(y_i, X, i) \right) \quad (2)$$

where $t_k(y_{i-1}, y_i, X, i)$ is a transition feature function of the entire input sequence and the labels at positions i and $i-1$ in the label sequence; $s_k(y_i, X, i)$ is a state feature function of the label at position i and the observed input sequence; and λ_k and μ_k are parameters to be estimated from training data. The parameters λ_k and μ_k play similar roles to the transition and emission probabilities in HMMs [12]. Therefore, the most probable label sequence for input sequence X is Y^* which maximizes a posterior probability.

$$Y^* = \arg\max_Y P_\lambda(Y \mid X) \quad (3)$$

We can find Y^* with dynamic programming using the Viterbi algorithm.

2.2 Parameter Estimation for CRFs

Assuming the training data $\{(X^{(n)}, Y^{(n)})\}$ are independently and identically distributed, the product of Equation 1 over all training sequences is a likelihood function of the parameter λ. Maximum likelihood training chooses parameter values such that the log-likelihood is maximized [10]. For CRFs, the log-likelihood $L(\lambda)$ is given by

$$\begin{aligned} L(\lambda) &= \sum_n \log P_\lambda(Y^{(n)} \mid X^{(n)}) \\ &= \sum_n \left[\sum_i \sum_k \lambda_k f_k(y_{i-1}^{(n)}, y_i^{(n)}, X^{(n)}, i) - \log Z(X^{(n)}) \right] \end{aligned} \quad (4)$$

It is not possible to analytically determine the parameter values that maximize the log-likelihood. Instead, maximum likelihood parameters must be identified using an iterative technique such as iterative scaling [12] or gradient-based methods [13,14].

Lafferty et al. proposed two iterative scaling algorithms to find parameters for CRFs. However, these methods converge into a global maximum very slowly. To

overcome this problem of slow convergence, several researchers adopted modern optimization algorithms such as the conjugate-gradient method or the limited-memory BFGS(L-BFGS) method [17].

3 Chunking Using Conditional Random Fields in Korean Texts

We now describe how CRFs are applied to the task of chunking in Korean texts. Firstly, we explore characteristics and chunk types of Korean. Then we explain the features for the model of chunking in Korean texts using CRFs. The ultimate goal of a chunker is to output appropriate chunk tags of a sequence of words with part-of-speech tags.

3.1 Characteristics of Korean

Korean is an agglutinative language, in which a word unit (called an eojeol) is a composition of a content word and function word(s). Function words – postpositions and endings – give much information such as grammatical relation, case, tense, modal, etc. Well-developed function words in Korean help with chunking, especially NP and VP chunking. For example, when the part-of-speech of current word is one of determiner, pronoun and noun, the following seven rules for NP chunking in Table 1 can find most NP chunks in text, with about 89% accuracy [6].

Table 1. Rules for NP chunking in Korean texts

No	Previous eojeol	Chunk tag of current word
1	determiner	I-NP
2	pronoun	I-NP
3	noun	I-NP
4	noun + possessive postposition	I-NP
5	noun + relative postfix	I-NP
6	adjective + relative ending	I-NP
7	others	B-NP

For this reason, boundaries of chunks are easily found in Korean, compared to other languages such as English or Chinese. This is why a rule-based chunking method is predominantly used. However, with sophisticated rules, the rule-based chunking method has limitations when handling exceptional cases. Park et al. proposed a hybrid of the rule-based and the machine learning method to resolve this problem [5,6]. Many recent machine learning techniques can capture hidden characteristics for classification. Despite its simplicity and efficiency, the rule-based method has recently been outdone by the machine learning method in various classification problems.

3.2 Chunk Types of Korean

Abney was the first to use the term ‘chunk’ to represent a non-recursive core of an intra-clausal constituent, extending from the beginning of constituent to its head. In

Korean, there are four basic phrases: noun phrase (NP), verb phrase (VP), adverb phrase (ADVP), and independent phrase (IP) [6]. As function words such as postposition or ending are well-developed, the number of chunk types is small compared to other languages such as English or Chinese. Table 2 lists the Korean chunk types, a simple explanation and examples of each chunk type.

Table 2. The Korean chunk types

No	Category	Explanation	Example
1	NP	Noun Phrase	**[NP**저 아름다운 여인을**]** [보세요]. ([the beautiful woman] [look])
2	VP	Verb Phrase	[지붕이] [몽땅] **[VP**내려앉아 있다**]**. ([the roof] [completely] [has fallen in])
3	ADVP	Adverb Phrase	[새가] **[ADVP** 매우 높이**]** [날고 있다]. ([a bird] [very high] [is flying])
4	IP	Independent Phrase	**[IP** 와**]**, [이거] [정말] [맛있다]. ([wow] [this] [very] [is delicious])

Like the CoNLL-2000 dataset, we use three types of chunk border tags, indicating whether a word is outside a chunk (O), starts a chunk (B), or continues a chunk (I). Each chunk type XP has two border tags: B-XP and I-XP. XP should be one of NP, VP, ADVP and IP. There exist nine chunk tags in Korean.

3.3 Feature Set of CRFs

One advantage of CRFs is that they can use many arbitrary, overlapping features. So we take advantage of all context information of a current word. We use words, part-of-speech tags of context and combinations of part-of-speech tags to determine the chunk tag of the current word,. The window size of context is 5; from left two words to right two words. Table 3 summarizes the feature set for chunking in Korean texts.

Table 3. Feature set for the chunking in Korean texts

Word	POS tag	Bi-gram of tags	Tri-gram of tags
w_{i-2}= w	t_{i-2}= t	t_{i-2}= t', t_{i-1}= t	t_{i-2}= t'', t_{i-1}= t', t_i= t
w_{i-1}= w	t_{i-1}= t	t_{i-1}= t', t_i= t	t_{i-1}= t'', t_i= t', t_{i+1}= t
w_i= w	t_i= t	t_i= t', t_{i+1}= t	t_i= t'', t_{i+1}= t', t_{i+2}= t
w_{i+1}= w	t_{i+1}= t	t_{i+1}= t',t_{i+2}= t	
w_{i+2}= w	t_{i+2}= t		

4 Experiments

In this section, we present experimental results of chunking using CRFs in Korean texts and compare the performance with previous systems of Park et al [6]. To make a fare comparison, we use the same dataset as Park et al [6].

4.1 Data Preparation

For evaluation of our proposed method, we use the STEP 2000 Korean chunking dataset (STEP 2000 dataset)[1], which is converted from the parsed KAIST Corpus [9].

Table 4. Simple statistics on the STEP 2000 dataset

Information	Value
POS tags	52
Words	321,328
Sentences	12,092
Chunk tags	9
Chunks	112,658

그	npp	B-NP	his
의	jcm	I-NP	postposition: possessive
책	ncn	I-NP	book
은	jxt	I-NP	postposition: topic
파기	ncpa	B-VP	destructed
되	xsv	I-VP	be
었	ep	I-VP	pre-final ending : past
다	ef	I-VP	ending : declarative
.	sf	O	

Fig. 2. An example of the STEP 2000 dataset

The STEP 2000 dataset consists of 12,092 sentences. We divide this corpus into training data and test data. Training data has 10,883 sentences and test data has 1,209 sentences, 90% and 10% respectively. Table 4 summarizes characteristics of the STEP 2000 dataset. Figure 2 shows an example sentence of the STEP 2000 dataset and its format is equal to that of CoNLL-2000 dataset. Each line is composed of a word, its part-of-speech (POS) tag and a chunk tag.

4.2 Evaluation Metric

The standard evaluation metrics for chunking performance are precision, recall and F-score ($F_{\beta=1}$) [3]. F-score is used for comparisons with other reported results. Each equation is defined as follows.

[1] STEP is an abbreviation of *Software Technology Enhancement Program*. We download this dataset from http://bi.snu.ac.kr/~sbpark/Step2000. If you want to know the part-of-speech tags used in the STEP 2000 dataset, you can reference KAIST tagset [9].

$$precision = \frac{\#\ of\ \ correct\ \ chunks}{\#\ of\ \ chunks\ \ in\ \ output} \tag{5}$$

$$recall = \frac{\#\ of\ \ correct\ \ chunks}{\#\ of\ \ chunks\ \ in\ test\ \ data} \tag{6}$$

$$F_{\beta=1} = \frac{2 \times recall\ \times precision}{recall\ +\ precision} \tag{7}$$

4.3 Experimental Results

Experiments were performed with C++ implementation of CRFs (FlexCRFs) on Linux with 2.4 GHz Pentium IV dual processors and 2.0Gbyte of main memory [18]. We use L-BFGS to train the parameters and use a Gaussian prior regularization in order to avoid overfitting [20].

Table 5. The performance of proposed method

Chunk tag	Precision	Recall	F-score
NP	94.23	94.30	94.27
VP	96.71	96.28	96.49
ADVP	96.90	97.02	96.96
IP	99.53	99.07	99.30
All	95.42	95.31	95.36

Total number of CRF features is 83,264. As shown in Table 5, the performances of most chunk type are 96~100%, very high performance. However, the performance of NP chunk type is lowest, 94.27% because the border of NP chunk type is very ambiguous in case of consecutive nouns. Using more features such as previous chunk tag should be able to improve the performance of NP chunk type.

Table 6. The experimental results of various chunking methods[2]

	HMMs	DT	MBL	Rule	SVMs	Hybrid	CRFs
Precision	73.75	92.29	91.41	91.28	93.63	94.47	95.42
Recall	76.06	90.45	91.43	92.47	91.48	93.96	95.31
F-score	74.89	91.36	91.38	91.87	92.54	94.21	95.36

Park et al. reported the performance of various chunking methods [6]. We add the experimental results of the chunking methods using HMMs-bigram and CRFs. In Table 6, F-score of chunking using CRFs in Korean texts is 97.19%, the highest

[2] Performances of all methods except HMMs and CRFs are cited from the experiment of Park et al [6]. They also use the STEP 2000 dataset and similar feature set. Therefore, the comparison of performance is reasonable.

performance of all. It significantly outperforms all others, including machine learning methods, rule-based methods and hybrid methods. It is because CRFs have a global optimum solution hence overcoming the label bias problem. They also can use many arbitrary, overlapping features.

Figure 3 shows the performance curve on the same test set in terms of the precision, recall and F-score with respect to the size of training data. In this figure, we can see that the performance slowly increases in proportion to the size of training data.

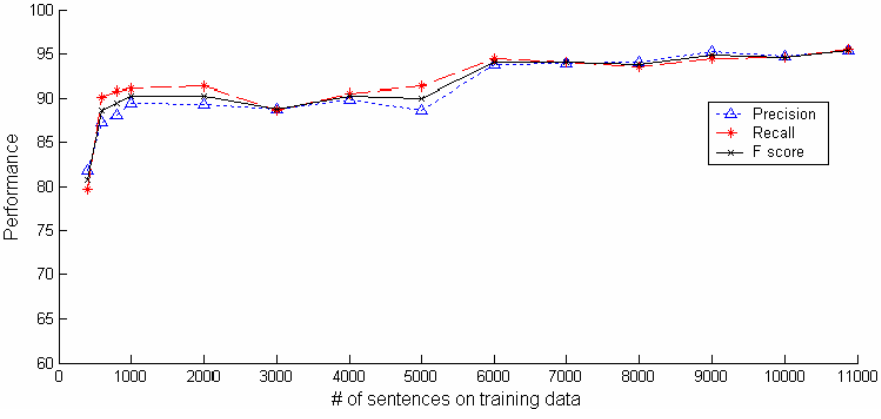

Fig. 3. The performance curve respect to the size of training data

In the experiment, we can see that CRFs can help improve the performance of chunking in Korean texts. CRFs have many promising properties except for the slow convergence speed compared to other models. In the next experiment, we have tried to analyze the importance of each feature and to make an additional experiment with various window sizes and any other useful features.

5 Conclusion

In this paper, we proposed a chunking method for Korean texts using CRFs. We observed that the proposed method outperforms other approaches. Experiments on the STEP 2000 dataset showed that the proposed method yields an F-score of 95.36%. This performance is 2.82% higher than that of SVMs and 1.15% higher than that of the hybrid method. CRFs use a number of correlated features and overcome the label bias problem. We obtained a very high performance using only small features. Thus, if we use more features such as semantic information or collocation, we can obtain a better performance.

From the experiment, we know that the proposed method using CRFs can significantly improve the performance of chunking in Korean texts. CRFs are a good framework for labeling an input sequence. In our future work, we will investigate how CRFs can be applied to other NLP problems: parsing, semantic analysis and spam filtering. Finally, we hope that this work can contribute to the body of research in this field.

Acknowledgements

This work was supported by the KOSEF through the Advanced Information Technology Research Center (AITrc) and by the BK21 Project.

References

1. S. Abney: Parsing by chunks. In R. Berwick, S. Abney, and C. Tenny, editors, Principle-based Parsing. Kluwer Academic Publishers (1991).
2. L. A. Ramashaw and M. P. Marcus: Text chunking using transformation-based learning. Proceedings of the Thired ACL Workshop on Very Large Corpora (1995).
3. E. F. Tjong Kim Sang and S. Buchholz: Introduction to the CoNLL-2000 shared task: Chunking. Proceedings of CoNLL-2000 (2000) 127-132.
4. T. Kudo and Y. Matsumoto: Chunking with support vector machines. Proceedings of NAACL2001, ACL (2001).
5. Park, S.-B. and Zhang, B.-T.: Combining a Rule-based Method and a *k*-NN for Chunking Korean Text. Proceedings of the 19th International Conference on Computer Processing of Oriental Languages (2001) 225-230.
6. Park, S.-B. and Zhang, B.-T.: Text Chunking by Combining Hand-Crafted Rules and Memory-Based Learning. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (2003) 497-504.
7. H.-P. Shin: Maximally Efficient Syntactic Parsing with Minimal Resources. Proceedings of the Conference on Hangul and Korean Language Information Processing (1999) 242-244.
8. M.-Y. Kim, S.-J. Kang and J.-H. Lee: Dependency Parsing by Chunks. Proceedings of the 27th KISS Spring Conference (1999) 327-329.
9. J.-T. Yoon and K.-S. Choi: Study on KAIST Corpus, CS-TR-99-139, KAIST CS (1999).
10. A. L. Berger, S. A. Della Pietra and V. J. Della Pietra: A maximum entropy approach to natural language processing. Computational Linguistics, 22(1) (1996) 39-71.
11. Andrew McCallum, D. Freitag and F. Pereira: Maximum entropy Markov models for information extraction and segmentation. Proceedings of International Conference on Machine Learning , Stanford, California (2000) 591-598.
12. John Lafferty, Andrew McCallum and Fernando Pereira: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the 18th International Conference on Machine Learning (2001) 282-289.
13. Fei Sha and Fernando Pereira: Shallow Parsing with Conditional Random Fields. Proceedings of Human Language Technology-NAACL, Edmonton, Canada (2003).
14. Hanna Wallach: Efficient Training of Conditional Random Fields. Thesis. Master of Science School of Cognitive Science, Division of Informatics. University of Edinburgh (2002).
15. Yongmei Tan, Tianshun Yao, Qing Chen and Jingbo Zhu: Applying Conditional Random Fields to Chinese Shallow Parsing. The 6th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2005) . LNCS, Vol.3406, Springer, Mexico City, Mexico (2005) 167-176.
16. J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. Unpublished manuscript (1971).

17. D. C. Liu and J. Nocedal: On the limited memory bfgs method for large-scale optimization. Mathematic Programming, 45 (1989) 503-528.
18. Hieu Xuan Phan and Minh Le Nguyen: FlexCRFs: A Flexible Conditional Random Fields Toolkit. http:://www.jaist.ac.jp/~hieuxuan/flexcrfs/flexcrfs.html (2004).
19. D. Pinto, A. McCallum, X. Wei and W. B. Croft: Table extraction using conditional random fields. Proceedings of the ACM SIGIR (2003).
20. S. F. Chen and R. Rosenfeld: A Gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, Carnegie Mellon University (1999).