In: *Proceedings of CoNLL-2000 and LLL-2000*, pages 163–165, Lisbon, Portugal, 2000.

# Hybrid Text Chunking

**GuoDong Zhou** and **Jian Su** and **TongGuan Tey**
Kent Ridge Digital Labs
21 Heng Mui Keng Terrace
Singapore 119613
{zhougd, sujian, tongguan}@krdl.org.sg

## Abstract

This paper proposes an error-driven HMM-based text chunk tagger with context-dependent lexicon. Compared with standard HMM-based tagger, this tagger incorporates more contextual information into a lexical entry. Moreover, an error-driven learning approach is adopted to decrease the memory requirement by keeping only positive lexical entries and makes it possible to further incorporate more context-dependent lexical entries. Finally, memory-based learning is adopted to further improve the performance of the chunk tagger.

## 1 Introduction

The idea of using statistics for chunking goes back to Church(1988), who used corpus frequencies to determine the boundaries of simple non-recursive noun phrases. Skut and Brants(1998) modified Church's approach in a way permitting efficient and reliable recognition of structures of limited depth and encoded the structure in such a way that it can be recognised by a Viterbi tagger. Our approach follows Skut and Brants' way by employing HMM-based tagging method to model the chunking process.

## 2 HMM-based Chunk Tagger with Context-dependent Lexicon

Given a token sequence $G_1^n = g_1 g_2 \cdots g_n$, the goal is to find an optimal tag sequence $T_1^n = t_1 t_2 \cdots t_n$ which maximizes $log\, P(T_1^n|G_1^n)$:

$$log\ P(T_1^n|G_1^n) = log\ P(T_1^n) + log \frac{P(T_1^n, G_1^n)}{P(T_1^n)P(G_1^n)}$$

The second item in the above equation is the mutual information between the tag sequence $T_1^n$ and the given token sequence $G_1^n$. By assuming that *the mutual information between $G_1^n$ and $T_1^n$ is equal to the summation of mutual information between $G_1^n$ and the individual tag $t_i$ ($1 \le i \le n$)*:

$$log \frac{P(T_1^n, G_1^n)}{P(T_1^n)P(G_1^n)} = \sum_{i=1}^{n} log \frac{P(t_i, G_1^n)}{P(t_i)P(G_1^n)}$$

$$MI(T_1^n, G_1^n) = \sum_{i=1}^{n} MI(t_i, G_1^n),$$

we have:

$$log\ P(T_1^n|G_1^n) = log\ P(T_1^n) + \sum_{i=1}^{n} log \frac{P(t_i, G_1^n)}{P(t_i)P(G_1^n)}$$

$$= log\ P(T_1^n) - \sum_{i=1}^{n} log\ P(t_i) + \sum_{i=1}^{n} log\ P(t_i|G_1^n)$$

The first item of above equation can be solved by chain rules. Normally, each tag is assumed to be probabilistic dependent on the N-1 previous tags. Here, backoff bigram(N=2) model is used. The second item is the summation of log probabilities of all the tags. Both the first item and second item constitute the language model component while the third item constitutes the lexicon component. Ideally the third item can be estimated by the forward-backward algorithm(Rabiner 1989) recursively for the first-order(Rabiner 1989) or second-order HMMs. However, several approximations on it will be attempted later in this paper instead. The stochastic optimal tag sequence can be found by maximizing the above equation over all the possible tag sequences using the Viterbi algorithm.

The main difference between our tagger and the standard taggers lies in our tagger has a context-dependent lexicon while others use a context-independent lexicon.

For chunk tagger, we have $g_1 = p_i w_i$ where $W_1^n = w_1 w_2 \cdots w_n$ is the word sequence and $p_1^n = p_1 p_2 \cdots p_n$ is the part-of-speech(POS) sequence. Here, we use structural tags to representing chunking(bracketing and labeling) structure. The basic idea of representing the structural tags is similar to Skut and Brants(1998) and the structural tag consists of three parts:

1) Structural relation. The basic idea is simple: structures of limited depth are encoded using a finite number of flags. Given a sequence of input tokens(here, the word and POS pairs), we consider the structural relation between the previous input token and the current one. For the recognition of chunks, it is sufficient to distinguish the following four different structural relations which uniquely identify the sub-structures of depth 1(Skut and Brants used seven different structural relations to identify the sub-structures of depth 2).

- 00: the current input token and the previous one have the same parent
- 90: one ancestor of the current input token and the previous input token have the same parent
- 09: the current input token and one ancestor of the previous input token have the same parent
- 99 one ancestor of the current input token and one ancestor of the previous input token have the same parent

Compared with the B-Chunk and I-Chunk used in Ramshaw and Marcus(1995), structural relations 99 and 90 correspond to B-Chunk which represents the first word of the chunk, and structural relations 00 and 09 correspond to I-Chunk which represents each other in the chunk while 90 also means the beginning of the sentence and 09 means the end of the sentence.

2)Phrase category. This is used to identify the phrase categories of input tokens.

3)Part-of-speech. Because of the limited number of structural relations and phrase categories, the POS is added into the structural tag to represent more accurate models.

Principally, the current chunk is dependent on all the context words and their POSs. However, in order to decrease memory requirement and computational complexity, our baseline HMM-based chunk tagger only considers previous POS, current POS and their word tokens whose POSs are of certain kinds, such as preposition and determiner etc. The overall precision, recall and $F_{\beta=1}$ rates of our baseline tagger on the test data of the shared task are 89.58%, 89.56% and 89.57%.

## 3 Error-driven Learning

After analysing the chunking results, we find many errors are caused by a limited number of words. In order to overcome such errors, we include such words in the chunk dependence context by using error-driven learning. First, the above HMM-based chunk tagger is used to chunk the training data. Secondly, the chunk tags determined by the chunk tagger are compared with the given chunk tags in the training data. For each word, its chunking error number is summed. Finally, those words whose chunking error numbers are equal to or above a given threshold(i.e. 3) are kept. The HMM-based chunk tagger is re-trained with those words considered in the chunk dependence context.

The overall precision, recall and $F_{\beta=1}$ rates of our error-driven HMM-based chunk tagger on the test data of the shared task are 91.53%, 92.02% and 91.77

## 4 Memory based Learning

Memory-based learning has been widely used in NLP tasks in the last decade. Principally, it falls into two paradigms. First paradigm represents examples as sets of features and carries out induction by finding the most similar cases. Such works include Daelemans et al.(1996) for POS tagging and Cardie(1993) for syntactic and semantic tagging. Second paradigm makes use of raw sequential data and generalises by reconstructing test examples from different pieces of the training data. Such works include Bod(1992) for parsing, Argamon et al.(1998) for shallow natural language patterns and Daelemans et al.(1999) for shallow parsing.

The memory-based method presented here follows the second paradigm and makes use of raw sequential data. Here, generalization is performed online at recognition time by comparing the new pattern to the ones in the training corpus.

Given one of the N most probable chunk sequences extracted by the error-driven HMM-based chunk tagger, we can extract a set of chunk patterns, each of them with the format: $XP = p_0 r_0^1 p_1^n r_n^{n+1} p_{n+1}$, where $r_i^{i+1}$ is the structural relation between $p_i$ and $p_{i+1}$.

As an example, from the bracketed and labeled sentence:

> [NP He/PRP ] [VP reckons/VBZ ] [NP the/DT current/JJ account/NN deficit/NN ] [VP will/MD narrow/VB ] [ PP to/TO ] [NP only/RB #/# 1.8/CD billion/CD ] [PP in/IN ] [NP September/NNP ] [O ./. ]

we can extract following chunk patterns:

NP=NULL 90 PRP 99 VBZ
VP=PRP 99 VBZ 99 DT
NP=VBZ 99 DT JJ NN NN 99 MD
PP=VB 99 TO 99 RB
NP=TO 99 RB # CD CD 99 IN
PP=CD 99 IN 99 NNP
NP=IN 99 NNP 99 .
O=NNP 99 . 09 NULL

For every chunk pattern, we estimate its probability by using memory-based learning. If the chunk pattern exists in the training corpus, its probability is computed by the probability of such pattern among all the chunk patterns. Otherwise, its probability is estimated by the multiply of its overlapped sub-patterns. Then the probability of each of the N most probable chunk sequences is adjusted by multiplying the probabilities of its extracted chunk patterns.

Table 1 shows the performance of error-driven HMM-based chunk tagger with memory-based learning.

## 5 Conclusion

It is found that the performance with the help of error-driven learning is improved by 2.20% and integration of memory-based learning further improves the performance by 0.35% to 92.12%.

For future work, the experimentation on large scale task will be speculated in the near future. Finally, a closer integration of memory-based method with HMM-based chunk tagger will also be conducted.

| test data | precision | recall | $F_{\beta=1}$ |
|---|---|---|---|
| ADJP | 76.17% | 70.78% | 73.37 |
| ADVP | 78.25% | 78.52% | 78.39 |
| CONJP | 46.67% | 77.78% | 58.33 |
| INTJ | 20.00% | 50.00% | 28.57 |
| LST | 00.00% | 00.00% | 00.00 |
| NP | 92.19% | 92.59% | 92.39 |
| PP | 96.09% | 96.94% | 96.51 |
| PRT | 72.36% | 83.96% | 77.73 |
| SBAR | 83.56% | 79.81% | 81.64 |
| VP | 92.77% | 92.85% | 92.81 |
| all | 91.99% | 92.25% | 92.12 |

Table 1: performance of chunking

## References


S. Argamon, I. Dagan, and Y. Krymolowski. 1998. A memory-based approach to learning shallow natural language patterns. In *COLING/ACL-1998*, pages 67–73. Montreal, Canada.

R. Bod. 1992. A computational model of language performance: Data-oriented parsing. In *COLING-1992*, pages 855–859. Nantes, France.

C. Cardie. 1993. A case-based approach to knowledge acquisition for domain-specific sentence analysis. In *Proceeding of the 11th National Conference on Artificial Intelligence*, pages 798–803. Menlo Park, CA, USA. AAAI Press.

K.W. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of Second Conference on Applied Natural Language Processing*, pages 136–143. Austin, Texas, USA.

W. Daelemans, J. Zavrel, P. Berck, and S. Gillis. 1996. Mbt: A memory-based part-of-speech tagger generator. In *Proceeding of the Fourth Workshop on Large Scale Corpora*, pages 14–27. ACL SIGDAT.

W. Daelemans, S. Buchholz, and J. Veenstra. 1999. Memory-based shallow parsing. In *CoNLL-1999*, pages 53–60. Bergen, Norway.

L.R. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286.

Lance A. Ramshaw and Mitchell P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third ACL Workshop on Very Large Corpora*. Cambridge, Massachusetts, USA.

W. Skut and T. Brants. 1998. Chunk tagger: statistical recognition of noun phrases. In *ESSLLI-1998 Workshop on Automated Acquisition of Syntax and Parsing*. Saarbruucken, Germany.