Error-driven HMM-based Chunk Tagger with Context-dependent Lexicon

GuoDong ZHOU
Kent Ridge Digital Labs
21 Heng Hui Keng Terrace
Singapore 119613
zhougd@krdl.org.sg

Jian SU
Kent Ridge Digital Labs
21 Heng Hui Keng Terrace
Singapore 119613
sujian@krdl.org.sg

Abstract

This paper proposes a new error-driven HMM-based text chunk tagger with context-dependent lexicon. Compared with standard HMM-based tagger, this tagger uses a new Hidden Markov Modelling approach which 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. Experiments show that this technique achieves overall precision and recall rates of 93.40% and 93.95% for all chunk types, 93.60% and 94.64% for noun phrases, and 94.64% and 94.75% for verb phrases when trained on PENN WSJ TreeBank section 00-19 and tested on section 20-24, while 25-fold validation experiments of PENN WSJ TreeBank show overall precision and recall rates of 96.40% and 96.47% for all chunk types, 96.49% and 96.99% for noun phrases, and 97.13% and 97.36% for verb phrases.

Introduction

Text chunking is to divide sentences into non-overlapping segments on the basis of fairly superficial analysis. Abney(1991) proposed this as a useful and relatively tractable precursor to full parsing, since it provides a foundation for further levels of analysis, while still allowing more complex attachment decisions to be postponed to a later phase.

Text chunking typically relies on fairly simple and efficient processing algorithms. Recently, many researchers have looked at text chunking in two different ways: Some researchers have applied rule-based methods, combining lexical data with finite state or other rule constraints, while others have worked on inducing statistical models either directly from the words and/or from automatically assigned part-of-speech classes. On the statistics-based approaches, Skut and Brants(1998) proposed a HMM-based approach to recognise the syntactic structures of limited length. Buchholz, Veenstra and Daelemans(1999), and Veenstra(1999) explored memory-based learning method to find labelled chunks. Ratnaparkhi(1998) used maximum entropy to recognise arbitrary chunk as part of a tagging task. On the rule-based approaches, Bourigaut(1992) used some heuristics and a grammar to extract "terminology noun phrases" from French text. Voutilainen(1993) used similar method to detect English noun phrases. Kupiec(1993) applied finite state transducer in his noun phrases recogniser for both English and French. Ramshaw and Marcus(1995) used transformation-based learning, an error-driven learning technique introduced by Eric Brill(1993), to locate chunks in the tagged corpus. Grefenstette(1996) applied finite state transducers to find noun phrases and verb phrases.

In this paper, we will focus on statistics-based methods. The structure of this paper is as follows : In section 1, we will briefly describe the new error-driven HMM-based chunk tagger with context-dependent lexicon in principle. In section 2, a baseline system which only includes the current part-of-speech in the lexicon is given. In section 3, several extended systems with different context-dependent lexicons are described. In section 4, an error-driven learning method is used to decrease memory requirement of the lexicon by keeping only positive lexical entries and make it possible to further improve the accuracy by merging different context-dependent lexicons into one after automatic analysis of the chunking errors. Finally, the conclusion is given.

The data used for all our experiments is extracted from the PENN WSJ Treebank (Marcus et al. 1993) by the program provided by Sabine Buchholz from Tilbug University. We use sections 00-19 as the training data and 20-24 as test data. Therefore, the performance is on large scale task instead of small scale task on CoNLL-2000 with the same evaluation program.

For evaluation of our results, we use the precision and recall measures. Precision is the percentage of predicted chunks that are actually correct while the recall is the percentage of correct chunks that are actually found. For convenient comparisons of only one value, we also list the $F_{\beta=1}$ value(Rijsbergen 1979): $\frac{(\beta^2+1)\cdot precision\cdot recall}{\beta^2\cdot precision+recall}$, with $\beta=1$.

1 HMM-based Chunk Tagger

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. This makes the process run in time linear to the length of the input string.

Our approach follows Skut and Brants' way by employing HMM-based tagging method to model the chunking process.

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

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

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 $t_i (1 \le i \le n)$* :

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

or

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

we have:

$$\begin{aligned}&\log P(T_1^n \mid G_1^n)\\ &= \log P(T_1^n) + \sum_{i=1}^{n} \log \frac{P(t_i, G_1^n)}{P(t_i)\cdot P(G_1^n)}\\ &= \log P(T_1^n) - \sum_{i=1}^{n} \log P(t_i) + \sum_{i=1}^{n} \log P(t_i \mid G_1^n)\end{aligned}$$

The first item of above equation can be solved by using 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 correspond to the language model component of the tagger while the third item corresponds to the lexicon component of the tagger. Ideally the third item can be estimated by using the forward-backward algorithm(Rabiner 1989) recursively for the first-order(Rabiner 1989) or second-order HMMs(Watson and Chunk 1992). However, several approximations on it will be attempted later in this paper instead. The stochastic optimal tag sequence can be found by maxmizing the above equation over all the possible tag sequences. This is implemented by the Viterbi algorithm.

The main difference between our tagger and other 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 sequence. Here, we use structural tags to representing chunking(bracketing and labelling) 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 part-of-speech 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

For example, in the following chunk tagged sentence(NULL represents the beginning and end of the sentence):

NULL [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 ./.] NULL

the corresponding structural relations between two adjacent input tokens are:

90(NULL He/PRP)
99(He/PRP reckons/VBZ)
99(reckons/VBZ the/DT)
00(the/DT current/JJ)
00(current/JJ account/NN)
00(account/NN deficit/NN)
99(deficit/NN will/MD)
00(will/MD narrow/VB)
99(narrow/VB to/TO)
99(to/TO only/RB)
00(only/RB #/#)
00(#/# 1.8/CD)
00(1.8/CD billion/CD)
99(billion/CD in/IN)
99(in/IN september/NNP)
99(september/NNP ./.)
09(./. NULL)

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 represnts 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 part-of-speech is added into the structural tag to represent more accurate models.

For the above chunk tagged sentence, the structural tags for all the corresponding input tokens are:

90_PRP_NP(He/PRP)
99_VBZ_VP(reckons/VBZ)
99_DT_NP(the/DT)
00_JJ_NP(current/JJ)
00_NN_NP(account/NN)
00_NN_NP(deficit/NN)
99_MD_VP(will/MD)
00_VB_VP(narrow/VB)
99_TO_PP(to/TO)
99_RB_NP(only/RB)
00_#_NP(#/#)
00_CD_NP(1.8/CD)
00_CD_NP(billion/CD)
99_IN_PP(in/IN)
99_NNP_NP(september/NNP)
99_._O(./.)

2 The Baseline System

As the baseline system, we assume $P(t_i / G_1^n) = P(t_i / p_i)$. That is to say, only the current part-of-speech is used as a lexical entry to determine the current structural chunk tag. Here, we define:

- Φ is the list of lexical entries in the chunking lexicon,

- $|\Phi|$ is the number of lexical entries(the size of the chunking lexicon)
- C is the training data.

For the baseline system, we have :

- $\Phi=\{p_i, p_i \exists C\}$, where p_i is a part-of-speech existing in the training data C
- $|\Phi|$=48 (the number of part-of-speech tags in the training data).

Table 1 gives an overview of the results of the chunking experiments. For convenience, precision, recall and $F_{\beta=1}$ values are given seperately for the chunk types NP, VP, ADJP, ADVP and PP.

Type	Precision	Recall	$F_{\beta=1}$
Overall	87.01	89.68	88.32
NP	90.02	90.50	90.26
VP	89.86	93.14	91.47
ADJP	70.94	63.84	67.20
ADVP	57.98	80.33	67.35
PP	85.95	96.62	90.97

Table 1 : Results of chunking experiments with the lexical entry list : $\Phi=\{p_i, p_i \exists C\}$

3 Context-dependent Lexicons

In the last section, we only use current part-of-speech as a lexical entry. In this section, we will attempt to add more contextual information to approximate $P(t_i / G_1^n)$. This can be done by adding lexical entries with more contextual information into the lexicon Φ. In the following, we will discuss five context-dependent lexicons which consider different contextual information.

3.1 Context of current part-of-speech and current word

Here, we assume:

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_i w_i) & p_i w_i \in \Phi \\ P(t_i / p_i) & p_i w_i \notin \Phi \end{cases}$$

where

$\Phi=\{p_i w_i, p_i w_i \exists C\}+\{p_i, p_i \exists C\}$ and $p_i w_i$ is a part-of-speech and word pair existing in the training data C.

In this case, the current part-of-speech and word pair is also used as a lexical entry to determine the current structural chunk tag and we have a total of about 49563 lexical entries($|\Phi|$=49563). Actually, the lexicon used here can be regarded as context-independent. The reason we discuss it in this section is to distinguish it from the context-independent lexicon used in the baseline system. Table 2 give an overview of the results of the chunking experiments on the test data.

Type	Precision	Recall	$F_{\beta=1}$
Overall	90.32	92.18	91.24
NP	90.75	92.14	91.44
VP	90.88	92.78	91.82
ADJP	76.01	70.00	72.88
ADVP	72.67	88.33	79.74
PP	94.96	96.48	95.71

Table 2 : Results of chunking experiments with the lexical entry list : $\Phi=\{p_i w_i, p_i w_i \exists C\}+\{p_i, p_i \exists C\}$

Table 2 shows that incorporation of current word information improves the overall $F_{\beta=1}$ value by 2.9%(especially for the ADJP, ADVP and PP chunks), compared with Table 1 of the baseline system which only uses current part-of-speech information. This result suggests that current word information plays a very important role in determining the current chunk tag.

3.2 Context of previous part-of-speech and current part-of-speech

Here, we assume :

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_{i-1} p_i) & p_{i-1} p_i \in \Phi \\ P(t_i / p_i) & p_{i-1} p_i \notin \Phi \end{cases}$$

where

$\Phi=\{p_{i-1} p_i, p_{i-1} p_i \exists C\}+\{p_i, p_i \exists C\}$ and $p_{i-1} p_i$ is a pair of previous part-of-speech and current part-of-speech existing in the training data C.

In this case, the previous part-of-speech and current part-of-speech pair is also used as a lexical entry to determine the current structural chunk tag and we have a total of about 1411 lexical entries($|\Phi|$=1411). Table 3 give an overview of the results of the chunking experiments.

Type	Precision	Recall	$F_{\beta=1}$
Overall	88.63	89.00	88.82
NP	90.77	91.18	90.97
VP	92.46	92.98	92.72
ADJP	74.93	60.13	66.72
ADVP	71.65	73.21	72.42
PP	87.28	91.80	89.49

Table 3: Results of chunking experiments with the lexical entry list : $\Phi = \{p_{i-1}p_i, p_{i-1}p_i \exists C\} + \{p_i, p_i \exists C\}$

Compared with Table 1 of the baseline system, Table 3 shows that additional contextual information of previous part-of-speech improves the overall $F_{\beta=1}$ value by 0.5%. Especially, $F_{\beta=1}$ value for VP improves by 1.25%, which indicates that previous part-of-speech information has a important role in determining the chunk type VP. Table 3 also shows that the recall rate for chunk type ADJP decrease by 3.7%. It indicates that additional previous part-of-speech information makes ADJP chunks easier to merge with neibghbouring chunks.

3.3 Context of previous part-of-speech, previous word and current part-of-speech

Here, we assume :

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_{i-1}w_{i-1}p_i) & p_{i-1}w_{i-1}p_i \in \Phi \\ P(t_i / p_i) & p_{i-1}w_{i-1}p_i \notin \Phi \end{cases}$$

where

$\Phi = \{p_{i-1}w_{i-1}p_i, p_{i-1}w_{i-1}p_i \exists C\} + \{p_i, p_i \exists C\}$,

where $p_{i-1}w_{i-1}p_i$ is a triple pattern existing in the training corpus.

In this case, the previous part-of-speech, previous word and current part-of-speech triple is also used as a lexical entry to determine the current structural chunk tag and $|\Phi| = 136164$.

Table 4 gives the results of the chunking experiments. Compared with Table 1 of the baseline system, Table 4 shows that additional 136116 new lexical entries of format $p_{i-1}w_{i-1}p_i$ improves the overall $F_{\beta=1}$ value by 3.3%. Compared with Table 3 of the extended system 2.2 which uses previous part-of-speech and current part-of-speech as a lexical entry, Table 4 shows that additional contextual information of previous word improves the overall $F_{\beta=1}$ value by 2.8%.

Type	Precision	Recall	$F_{\beta=1}$
Overall	91.23	92.03	91.63
NP	92.89	93.85	93.37
VP	94.10	94.23	94.16
ADJP	79.83	69.01	74.03
ADVP	76.91	80.53	78.68
PP	90.41	94.77	92.53

Table 4 : Results of chunking experiments with the lexical entry list : $\Phi = \{p_{i-1}w_{i-1}p_i, p_{i-1}w_{i-1}p_i \exists C\} + \{p_i, p_i \exists C\}$

3.4 Context of previous part-of-speech, current part-of-speech and current word

Here, we assume :

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_{i-1}p_iw_i) & p_{i-1}p_iw_i \in \Phi \\ P(t_i / p_i) & p_{i-1}p_iw_i \notin \Phi \end{cases}$$

where

$\Phi = \{p_{i-1}p_iw_i, p_{i-1}p_iw_i \exists C\} + \{p_i, p_i \exists C\}$,

where $p_{i-1}p_iw_i$ is a triple pattern existing in the training and $|\Phi| = 131416$.

Table 5 gives the results of the chunking experiments.

Type	Precision	Recall	$F_{\beta=1}$
Overall	92.67	93.43	93.05
NP	93.35	94.10	93.73
VP	93.05	94.30	93.67
ADJP	80.65	72.27	76.23
ADVP	78.92	84.48	81.60
PP	95.30	96.67	95.98

Table 5: Results of chunking experiments with the lexical entry list : $\Phi = \{p_{i-1}p_iw_i, p_{i-1}p_iw_i \exists C\} + \{p_i, p_i \exists C\}$

Compared with Table 2 of the extended system which uses current part-of-speech and current word as a lexical entry, Table 5 shows that additional contextual information of previous part-of-speech improves the overall $F_{\beta=1}$ value by 1.8%.

3.5 Context of previous part-of-speech, previous word, current part-of-speech and current word

Here, the context of previous part-of-speech, current part-of-speech and current word is used as a lexical entry to determine the current structural chunk tag and $\Phi = \{p_{i-1}w_{i-1}p_iw_i, p_{i-1}w_{i-1}p_iw_i \exists C\} + \{p_i, p_i \exists C\}$, where $p_{i-1}w_{i-1}p_iw_i$ is a pattern existing in the training corpus. Due to memory limitation, only lexical entries which occurs more than 1 times are kept. Out of 364365 possible lexical entries existing in the training data, 98489 are kept($|\Phi|$=98489).

$$P(t_i / G_1^n) = \begin{cases} P(t_i / p_{i-1}w_{i-1}p_iw_{ii}) & p_{i-1}w_{i-1}p_iw_i \in \Phi \\ P(t_i / p_i) & p_{i-1}w_{i-1}p_iw_i \notin \Phi \end{cases}$$

Table 6 gives the results of the chunking experiments.

Type	Precision	Recall	$F_{\beta=1}$
Overall	92.28	93.04	92.66
NP	93.50	93.53	93.52
VP	92.62	94.07	93.35
ADJP	81.39	72.17	76.50
ADVP	75.09	86.23	80.27
PP	94.12	97.12	95.59

Table 6: Results of chunking experiments with the lexical entry list: $\Phi = \{p_{i-1}w_{i-1}p_iw_i, p_{i-1}w_{i-1}p_iw_i \exists C\} + \{p_i, p_i \exists C\}$

Compared with Table 2 of the extended system which uses current part-of-speech and current word as a lexical entry, Table 6 shows that additional contextual information of previous part-of-speech improves the overall $F_{\beta=1}$ value by 1.8%.

3.6 Conclusion

Above experiments shows that adding more contextual information into lexicon significantly improves the chunking accuracy. However, this improvement is gained at the expense of a very large lexicon and we find it difficult to merge all the above context-dependent lexicons in a single lexicon to further improve the chunking accurracy because of memory limitation. In order to reduce the size of lexicon effectively, an error-driven learning approach is adopted to examine the effectiveness of lexical entries and make it possible to further improve the chunking accuracy by merging all the above context-dependent lexicons in a single lexicon. This will be discussed in the next section.

4 Error-driven Learning

In section 2, we implement a baseline system which only considers current part-of-speech as a lexical entry to deterrmine the current chunk tag while in section 3, we implement several extended systems which take more contextual information into consideration.

Here, we will examine the effectiveness of lexical entries to reduce the size of lexicon and make it possible to further improve the chunking accuracy by merging several context-dependent lexicons in a single lexicon.

For a new lexical entry e_i, the effectiveness $F_\Phi(e_i)$ is measured by the reduction in error which results from adding the lexical entry to the lexicon : $F_\Phi(e_i) = F_\Phi^{Error}(e_i) - F_{\Phi+\Delta\Phi}^{Error}(e_i)$. Here, $F_\Phi^{Error}(e_i)$ is the chunking error number of the lexical entry e_i for the old lexicon Φ and $F_{\Phi+\Delta\Phi}^{Error}(e_i)$ is the chunking error number of the lexical entry e_i for the new lexicon $\Phi + \Delta\Phi$ where $e_i \in \Delta\Phi$ ($\Delta\Phi$ is the list of new lexical entries added to the old lexicon Φ). If $F_\Phi(e_i) > 0$, we define the lexical entry e_i as positive for lexicon Φ. Otherwise, the lexical entry e_i is negative for lexicon Φ.

Tables 7 and 8 give an overview of the effectiveness distributions for different lexicons applied in the extended systems, compared with the lexicon applied in the baseline system, on the test data and the training data, respectively.

Tables 7 and 8 show that only a minority of lexical entries are positive. This indicates that discarding non-positive lexical entries will largely decrease the lexicon memory requirement while keeping the chunking accurracy.

Context	Positive	Negative	Total
POS_iW_i	1800	314	49515
$POS_{i-1}POS_i$	209	136	1363
$POS_{i-1}W_{i-1}POS_i$	2876	229	136116
$POS_{i-1}POS_iW_i$	2895	193	131368
$POS_{i-1}W_{i-1}POS_iW_i$	4083	155	98441

Table 7 : The effectiveness of lexical entries on the test data

Context	Positive	Negative	Total
$POS_i W_i$	6724	719	49515
$POS_{i-1} POS_{ii}$	357	196	1363
$POS_{i-1} W_{i-1} POS_{ii}$	13205	582	136116
$POS_{i-1} POS_{ii} W_i$	14186	325	131368
$POS_{i-1} W_{i-1} POS_{ii} W_i$	15516	144	98441

Table 8 : The effectiveness of lexical entries on the training data

Tables 9-13 give the performances of the five error-driven systems which discard all the non-positive lexical enrties on the training data. Here, Φ' is the lexicon used in the baseline system. $\Phi' = \{p_i, p_i \exists C\}$ and $\Delta\Phi = \Phi - \Phi'$. It is found that $F_{\beta=1}$ values of error driven systems for context of current part-of-speech and word pair and for context of previous part-of-speech and current part-of-speech increase by 1.2% and 0.6%. Although $F_{\beta=1}$ values for other three cases slightly decrease by 0.02%, 0.02% and 0.19%, the sizes of lexicons have been greatly reduced by 85% to 97%.

Type	Precision	Recall	$F_{\beta=1}$
Overall	91.69	93.28	92.48
NP	92.64	93.48	93.06
VP	92.16	93.66	92.90
ADJP	78.39	71.69	74.89
ADVP	73.66	87.80	80.11
PP	95.18	97.38	96.27

Table 9 : Results of chunking experiments with error-driven lexicon : $\Phi = \{p_i w_i, p_i w_i \exists C \; \& \; F_{\Phi'}(p_i w_i) > 0\} + \{p_i, p_i \exists C\}$

Type	Precision	Recall	$F_{\beta=1}$
Overall	88.68	90.28	89.47
NP	90.61	91.57	91.08
VP	91.80	94.08	92.90
ADJP	72.20	62.72	67.13
ADVP	70.53	78.90	74.48
PP	86.55	96.34	91.19

Table 10: Results of chunking experiments with error-driven lexicon : $\Phi = \{p_{i-1} p_i, p_{i-1} p_i \exists C \; \& \; F_{\Phi'}(p_{i-1} p_i) > 0\} + \{p_i, p_i \exists C\}$

Type	Precision	Recall	$F_{\beta=1}$
Overall	91.02	92.21	91.61
NP	92.36	93.69	93.02
VP	93.68	94.94	94.30
ADJP	78.28	71.46	74.71
ADVP	76.77	81.79	79.20
PP	90.67	95.37	92.96

Table 11 : Results of chunking experiments with error-driven lexicon : $\Phi = \{p_{i-1} w_{i-1} p_i, p_{i-1} w_{i-1} p_i \exists C \; \& \; F_{\Phi'}(p_{i-1} w_{i-1} p_i) > 0\} + \{p_i, p_i \exists C\}$

Type	Precision	Recall	$F_{\beta=1}$
Overall	92.84	93.21	93.03
NP	93.35	93.65	93.50
VP	93.97	94.67	94.32
ADJP	79.49	72.94	76.07
ADVP	79.47	85.91	82.57
PP	95.19	96.29	95.74

Table 12: Results of chunking experiments with error-driven lexicon : $\Phi = \{p_{i-1} p_i w_i, p_{i-1} p_i w_i \exists C \; \& \; F_{\Phi'}(p_{i-1} p_i w_i) > 0\} + \{p_i, p_i \exists C\}$

Type	Precision	Recall	$F_{\beta=1}$
Overall	91.99	92.95	92.47
NP	93.35	93.39	93.37
VP	92.89	94.36	93.62
ADJP	80.01	71.70	75.63
ADVP	73.40	87.32	79.76
PP	93.42	97.33	95.33

Table 13: Results of chunking experiments with error-driven lexicon : $\Phi = \{p_{i-1} w_{i-1} p_i w_i, p_{i-1} w_{i-1} p_i w_i \exists C \; \& \; F_{\Phi'}(p_{i-1} w_{i-1} p_i w_i) > 0\} + \{p_i, p_i \exists C\}$

After discussing the five context-dependent lexicons separately, now we explore the merging of context-dependent lexicons by assuming :

$$\begin{aligned}
\Phi = & \{p_{i-1} w_{i-1} p_i w_i, p_{i-1} w_{i-1} p_i w_i \exists C \; \& \; F_{\Phi'}(p_{i-1} w_{i-1} p_i w_i) > 0\} \\
& + \{p_{i-1} p_i w_i, p_{i-1} p_i w_i \exists C \; \& \; F_{\Phi'}(p_{i-1} p_i w_i) > 0\} \\
& + \{p_{i-1} w_{i-1} p_i, p_{i-1} w_{i-1} p_i \exists C \; \& \; F_{\Phi'}(p_{i-1} w_{i-1} p_i) > 0\} \\
& + \{p_{i-1} p_i, p_{i-1} p_{ii} \exists C \; \& \; F_{\Phi'}(p_{i-1} p_i) > 0\} \\
& + \{p_i w_i, p_i w_i \exists C \; \& \; F_{\Phi'}(p_i w_i) > 0\} + \{p_i, p_i \exists C\}
\end{aligned}$$

and $P(t_i / G_1^n)$ is approximated by the following order :

1. if $p_{i-1}w_{i-1}p_iw_i \in \Phi$,

   $P(t_i / G_1^n) = P(t_i / p_{i-1}w_{i-1}p_iw_i)$
2. if $p_{i-1}p_iw_i \in \Phi$,

   $P(t_i / G_1^n) = P(t_i / p_{i-1}w_{i-1}p_iw_i)$
3. if $p_{i-1}w_{i-1}p_i \in \Phi$,

   $P(t_i / G_1^n) = P(t_i / p_{i-1}w_{i-1}p_i)$
4. if $p_iw_i \in \Phi$, $P(t_i / G_1^n) = P(t_i / p_iw_i)$
5. if $p_{i-1}p_i \in \Phi$, $P(t_i / G_1^n) = P(t_i / p_{i-1}p_i)$
6. $P(t_i / G_1^n) = P(t_i / p_{i-1}p_i)$

Table 14 gives an overview of the chunking experiments using the above assumption. It shows that the $F_{\beta=1}$ value for the merged context-dependent lexicon inreases to 93.68%. For a comparison, the $F_{\beta=1}$ value is 93.30% when all the possible lexical entries are included in Φ (Due to memory limitation, only the top 150000 mostly occurred lexical entries are included).

Type	Precision	Recall	$F_{\beta=1}$
Overall	93.40	93.95	93.68
NP	93.60	94.64	94.12
VP	94.64	94.75	94.70
ADJP	77.12	74.55	75.81
ADVP	82.39	83.80	83.09
PP	96.61	96.63	96.62

Table 14: Results of chunking experiments with the merged context-dependent lexicon

For the relationship between the training corpus size and error driven learning performance, Table 15 shows that the performance of error-driven learning improves stably when the training corpus size increases.

Training Sections	$\|\Phi\|$	Accuracy	FB1
0-1	14384	94.78%	91.95
0-3	24507	95.19%	92.51
0-5	32316	95.28%	92.77
0-7	38286	95.41%	93.00
0-9	39876	95.53%	93.12
0-11	43372	95.65%	93.31
0-13	46029	95.62%	93.29
0-15	47901	95.66%	93.34
0-17	48813	95.74%	93.41
0-19	49988	95.92%	93.68

Table 15: The performance of error-driven learning with different training corpus size

For comparison with other chunk taggers, we also evaluate our chunk tagger with the merged context-dependent lexicon by cross-validation on all 25 partitions of the PENN WSJ TreeBank. Table 16 gives an overview of such chunking experiments.

Type	Precision	Recall	$F_{\beta=1}$
Overall	96.40	96.47	96.44
NP	96.49	96.99	96.74
VP	97.13	97.36	97.25
ADJP	89.92	88.15	89.03
ADVP	91.52	87.57	89.50
PP	97.13	97.36	97.25

Table 16: Results of 25-fold cross-validation chunking experiments with the merged context-dependent lexicon

Tables 14 and 16 shows that our new chunk tagger greatly outperforms other reported chunk taggers on the same training data and test data by 2%~3%.(Buchholz S., Veenstra J. and Daelmans W.(1999), Ramshaw L.A. and Marcus M.P.(1995), Daelemans W., Buchholz S. and Veenstra J.(1999), and Veenstra J.(1999)).

Conclusion

This paper proposes a new error-driven HMM-based chunk tagger with context-dependent lexicon. Compared with standard HMM-based tagger, this new tagger uses a new Hidden Markov Modelling approach which incorporates more contextual information into a lexical entry by assuming $MI(T_1^n, G_1^n) = \sum_{i=1}^{n} MI(t_i, G_1^n)$. Moreover, an error-driven learning approach is adopted to drease the memeory requirement and further improve the accuracy by including more context-dependent information into lexicon.

It is found that our new chunk tagger singnificantly outperforms other reported chunk taggers on the same training data and test data.

For future work, we will explore the effectivessness of considering even more contextual information on approximation of $P(T_1^n | G_1^n)$ by using the forward-backward algorithm(Rabiner 1989) while currently we only consider the contextual information of current location and previous location.

Acknowledgement

We wish to thank Sabine Buchholz from Tilbug University for kindly providing us her program which is also used to extact data for Conll-2000 share task.

References

Abney S. "Parsing by chunks". *Principle-Based Parsing* edited by Berwick, Abney and Tenny. Kluwer Academic Publishers.

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

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

Bourigault D. "Surface grammatical analysis for the extraction of terminological noun phrases". *COLING-92*. Pp.977-981. 1992.

Brill Eric. "A corpus-based approach to language learning". *Ph.D thesis*. Univ. of Penn. 1993

Buchholz S., Veenstra J. and Daelmans W. "Cascaded grammatical relation assignment." *Proceeding of EMNLP/VLC-99, at ACL'99*. 1999

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

Church K.W. "A stochastic parts program and noun phrase parser for unrestricted Text." *Proceeding of Second Conference on Applied Natural Language Processing*. Pp.136-143. Austin, Texas, USA. 1988.

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

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

Grefenstette G. "Light parsing as finite-state filtering". *Workshop on Extended Finite State Models of Language at ECAI'96*. Budapest, Hungary. 1996.

Kupiec J. " An algorithm for finding noun phrase correspondences in bilingual corpora". ACL'93. Pp17-22. 1993.

Marcus M., Santorini B. and Marcinkiewicz M.A. "Buliding a large annotated corpus of English: The Penn Treebank". *Computational Linguistics*. 19(2):313-330. 1993.

Rabiner L. "A tutorial on Hidden Markov Models and selected applications in speech recognition". *IEEE 77(2)*, pp.257-285. 1989.

Ramshaw L.A. and Marcus M.P. "Transformation-based Learning". *Proceeding of 3th ACL Workshop on Very Large Corpora at ACL'*95. 1995.

Rijsbergen C.J.van. *Information Retrieval*. Buttersworth, London. 1979.

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

Veenstra J. "Memory-based text chunking". *Workshop on machine learning in human language technology at ACAI'99*. 1999.

Voutilainen A. "Nptool: a detector of English phrases". *Proceeding of the Workshop on Very Large Corpora*. Pp48-57. ACL' 93. 1993

Watson B. and Chunk Tsoi A. "Second order Hidden Markov Models for speech recognition". *Proceeding of 4th Australian International Conference on Speech Science and Technology*. Pp.146-151. 1992.