COV Model and its Application in Chinese Part-of-Speech Tagging

Xing Fukun
Luoyang Foreign Languages University
471003 Henan
xingfukun@126.com

Song Rou
Beijing Language and Cultural University 100086 Beijing
songrou@126.com

Abstract

This article presents a new sequence labeling model named Context OVerlapping (COV) model, which expands observation from single word to n-gram unit and there is an overlapping part between the neighboring units. Due to the co-occurrence constraint and transition constraint, COV model reduces the search space and improves tagging accuracy. The 2-gram COV is applied to Chinese PoS tagging and the precision rate of the open test is as high as 96.83%, which is higher than the second order HMM, which is 95.73%. The result is also comparable to the discriminative models but COV takes much less training time than them. With symbol decoding COV prunes many nodes before statistics decoding and the search space of COV is about10-20% less than that of HMM.

1 Introduction

Part of Speech (PoS) can provide much useful information for most natural language processing tasks such as word sense disambiguation, chunk detection, sentence parsing, speech synthesis, machine translation and so on. Therefore lots of efforts have been made to build effective and robust models for automatic PoS tagging. According to Doug Cutting (1992), a practical PoS tagger should be "robust, efficient, accurate, tunable and reusable". With regard to efficiency the basic requirement for a PoS tagger is that training and test time should not be too long. And for a robust tagger the tagging accuracy should be as high as possible and can well deal with the sparseness data.

Most of the approaches to PoS tagging can be divided into two main classes, rule-based and statistics-based approach. In rule-based approaches, words are assigned tags based on a set of rules and a lexicon. These rules can either be manually crafted, or learned, as in the transformation-based error-driven approach of Brill (1995).

In the statistics-based approaches HMM is the representative of generative models and is widely used in PoS tagging (Church, 1988; Cutting et al. 1992; Thede & Harper 1999, Huang et al. 2007, etc.) .

Maximum Entropy model and Conditional Random Fields (CRFs) model are the representatives of discriminative models and are also applied in PoS tagging. Thanks to the flexibility of features selection these discriminative models achieve higher precision rates than the generative models in PoS tagging (Adwait, 1996; Lafferty, 2001 etc.). But the training of discriminative models is

Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 3–10,
Wuhan, China, 20-21 October 2014

time-consuming and requires high-quality computer processing power, which affects their applications in the real tasks.

Concerning all the characteristics of generative and discriminative models, we proposed a new model on the basis of HMM. The new model expand the observation from one single word to n-gram unit and between the neighboring units there is an n-1 gram part, which is shared by the neighboring units. So the new model is called Context OVerlapping (COV) model.

COV is a general sequence labeling model and has been applied to Chinese and English PoS tagging tasks. In these tasks COV achieves better performance than HMM and its performance is comparable to the discriminative models. Meanwhile its training time is much less than the discriminative models, which makes the model more efficient and robust in the real tasks.

The structure of the article is that: the first part will briefly introduce PoS tagging, in the second part we will introduce COV model. The third part will compare COV with HMM. The fourth part will address how to estimate parameters and handle sparseness data. The fifth part is about the algorithm of symbol decoding. The sixth part is about evaluation criteria and the seventh part presents the experiments and results. The final part is some discussions and future work to do.

2 COV Model

COV model is based on HMM. HMM is a form of generative model, that defines a joint probability distribution p(X,Y) where X and Y are random variables respectively ranging over observation sequences and their corresponding state sequences. In order to define a joint distribution, generative models must enumerate all possible observation sequences. For most domains, it is intractable unless observation elements are represented as isolated units, independent from the other elements in an observation sequence. More precisely, the observation element at any given time may only directly depend on the state at that time. This is an appropriate assumption for a few simple data sets, however most real-world observation sequences such as sentences are best represented in terms of multiple interacting features and even long-range dependencies between observation elements. Due to the observation independence assumption the performance of HMM is limited in PoS tagging. For example, here are 2 Chinese sentences:

(1) 市 长 /n 强调/v 深入/v a 细致/a 的/u 工作/vn 作风/n

(The mayor put emphasis on the careful working style.)

(2) 市长/n 要/v 深入/v a 困难/a 的/u 群众/n 中间/f

(The mayor should care about those people in troubles.)

For the convenience of analysis we assume that in each sentence only “深入”(careful or care) has two parts of speech, adjective (a) or verb (v), and other words only have one PoS. If we use the first-order HMM model to predict the PoS of “深入” the prediction will be like:

$$\hat{Q}_1 = \arg\max_{X\{a,v\}} p(n)p(v|n)p(X|v)p(a|X)$$
$$p(u|a)p(vn|u)p(n|vn)p(\text{市长}|n)p(\text{强调}|v)$$
$$p(\text{深入}|X)p(\text{细致}|a)p(\text{的}|a)p(\text{工作}|vn)$$
$$p(\text{作风}|n)$$

$\hat{Q}_1$ denotes the state sequence of sentence (1) and X denotes the possible state of “深入”. For only “深入” is ambiguous and other words all have only one PoS, the formula can be simplified as:

$$\hat{Q}_1 = \arg\max_{X\{a,v\}} p(X|v)p(a|X)p(\text{深入}|X)$$

And as same as sentence (1) we can get the prediction formula of sentence (2) as:

$$\hat{Q}_2 = \arg\max_{X\{a,v\}} \text{p(X | v)p(a | X)p(深入 | X)}$$

Comparing the two formulae, we find that $\hat{Q}_1$ and $\hat{Q}_2$ are the same, which means that HMM tagger will not distinguish between the different PoSs of "深入" in the two sentences. In fact "深入" in sentence (1) is an adjective and in sentence (2) is a verb. So HMM must make one mistake either in sentence (1) or sentence (2). The mistake shows the limitation of HMM in PoS tagging.

In order to overcome the shortcomings of observation independence assumption of HMM and combine more context information into the model, COV model is proposed in this paper. The formalism of 2-gram COV is as follows and the formalisms of other n-gram COV (n>2) models can be gotten according to the 2-gram model.

In the 2-gram COV there is a basic state set $Q = \{q_1, q_2, ...q_s\}$. The observation sequence is S= $w_1 ... w_h$. The corresponding state of a 2-gram observation unit $w_{i-1}w_i$（2≤i≤h）is a state set $e_i = \{q_{i-1}^j q_i^j\}$, in which q_{i-1}^j is one of the basic states of w_{i-1} and q_i^j is one of the basic states of w_i. The state sequence $q_{i-1}^j q_i^j$ is called one state unit of the observation unit $w_{i-1}w_i$. It is notable that e_i is the state set when the word w_{i-1} and w_i co-occur, which is called Co-occurrence Constraint(CC). When w_{i-1} and w_i co-occur the amount of possible states of $w_{i-1}w_i$ will not be more than the amount of the combination of states of w_{i-1} and w_i.

The search for the state sequence with the highest joint probability can be computed like:

$$\hat{Q} = \arg\max \text{P(Q | S)} =$$

$$\arg\max \text{P(Q)P(S | Q)} \approx$$

$$\arg\max_{q_{i-1},q_i}(p(q_1)p(q_2 \mid q_1)\prod_{i=3}^{h} p(q_{i-1}q_i \mid q_{i-2}q_{i-1})$$

$$p(o_1 \mid q_1)\prod_{i=2}^{h} p(o_{i-1}o_i \mid q_{i-1}q_i))$$

Q denotes the state sequence and S denotes the observation sequence. $\hat{Q}$ denotes the final state sequence, whose joint probability is the highest.

For the convenience of computation, we insert 2 "*B*", whose state is "B" at the beginning of the sequence and insert 2 "*E* ", whose state is "E" at the end of the sequence. And then the above formula will be:

$$\hat{Q} = \arg\max_{q_{i-1},q_i}(\prod_{i=1}^{h+2} p(q_{i-1}q_i \mid q_{i-2}q_{i-1})$$

$$\prod_{i=1}^{h+2} p(o_{i-1}o_i \mid q_{i-1}q_i))$$

In this model there is an overlapping part between the neighboring observation units $w_{i-2}w_{i-1}$ and $w_{i-1}w_i$. For w_{i-1} is shared by the neighboring units, the corresponding states units of $w_{i-2}w_{i-1}$ and $w_{i-1}w_i$ should also share the same overlapping state. If $q_{i-2}^k q_{i-1}^k$ is one state of $w_{i-2}w_{i-1}$ and $q_{i-1}^j q_i^j$ is one state of $w_{i-1}w_i$, then only if q_{i-1}^k is the same as q_{i-1}^j then it is possible to transmit from state $q_{i-2}^k q_{i-1}^k$ to $q_{i-1}^j q_i^j$, otherwise there is no transition path from $q_{i-2}^k q_{i-1}^k$ to $q_{i-1}^j q_i^j$. The constraint $q_{i-1}^k = q_{i-1}^j$ is called Transition Constraint (TC).

$\hat{Q}$ is a sequence consisting of h+1 2-gram state units like:

$$B\hat{q}_1, \hat{q}_1\hat{q}_2, \hat{q}_2\hat{q}_3, ..., \hat{q}_{h-1}\hat{q}_h, \hat{q}_h E$$

（$\hat{q}_i \in Q$）

It is obvious that the final state sequence can be gotten from the above sequence.

3 Comparisons between COV and HMM

There are 3 different points between COV and HMM.

First, in the nth HMM if each observation has k states and then the amount of the history states will be k^n. But in the n-gram COV the amount of the history states will usually be smaller than k^n because of the Constraint of Co-occurrence. And then the search space of COV will also be smaller than HMM.

Second, in the nth order HMM the emission probability of q_t to o_t is only $P(o_t|q_t)$. But in the n-gram COV, there are n emission probabilities relevant to q_t and o_t, which are $P(o_{t-n+1}...o_t \mid q_{t-n+1}...q_t)$, …, $P(o_t...o_{t+n-1}|q_t...q_{t+n-1})$. For all of these emission probabilities are related to q_t and o_t, these observation units will make constraints on the possible state units.

Third, in the nth order HMM the transition probability from the history state to the current state is $P(q_i|q_{i-n},...q_{i-1})$. But in the n-gram COV the transition path must obey TC, which requires the overlapping part of the neighboring state units must be the same. If the neighboring state units obey TC the transition probability is the same as that in nth order HMM. If the neighboring state units don't obey TC there will be no transition path between them. With TC a great amount of paths are pruned, which makes the search space reduced.

Here is an example to illustrate the lattice building and tagging process by 2-gram COV. In particular, this example needn't any probability computation and can get the final state sequence just with symbol comparing.

1	2	3	4	5
B-*B*	*B*-领导	领导-强调	强调-深入	深入-细致
B-B	B-n	n-v	v-a	a-a
	B-vn		v-ad	ad-ad
	B-v			

6	7	8	9	10
细致-的	的-工作	工作-作风	作风-*E*	*E*-*E*
a-u	u-v	vn-n	n-E	E-E
	u-n			
	u-vn			

Table 1: An example to illustrate COV tagging process (For the space limitation the table is split to two)

In the above table each column is a 2-gram observation unit and the neighboring units share an overlapping part. For example, unit 2 is "*B*-领导"（*B*-leader） and unit 3 is "领导-强调"（leader-emphasizes）, "领导"（leader） is the overlapping part between unit 2 and unit 3. Unit 2 has 3 possible state units, which are "B-n, B-vn, B-v", and unit 3 has only one possible state unit, which is "n-v". With Transition Constraint only if the overlapping part of state unit 2 and state unit 3 is the same there can be a transition path. So in the state units of unit 2 only "B-n" is remained and the state units "B-vn" and "B-v" are all eliminated for their overlapping parts (vn and v) are not the same as the overlapping part of state unit 3 (n). The shadowed grids in the table are all the impossible states and are eliminated. In this example after the symbol comparing and elimination there remains only one path for the sentence and the path is the final tagging result. So this sentence is tagged without any probability computation but only with the symbol comparing. The process of symbol comparing and elimination is called symbol decoding.

Most times there may be more than one possible paths remained after symbol decoding and then the Viterbi algorithm will be applied to get the best tagging sequence. Although HMM also applies Viterbi for decoding, the search space of HMM is bigger than that of COV because COV has eliminated many impossible states in the step of symbol decoding.

4 Parameters estimation and strategy of handling sparseness data

There are 2 main parameters to be estimated in COV:

(1) P_t :State transition probability;

(2) P_e :State emission probability.

We apply the maximum likelihood to estimate these parameters from the tagged corpus. The details of the estimation will not be introduced here.

For the expansion of the observation the sparseness problem in n-gram COV is more serious than that in HMM. COV applies back-off strategy to deal with the sparseness data. The main idea is that if n-gram (n>2) $w_{i-n+1}...w_i$ is not in the n-gram vocabulary, which is gotten from the training corpus, it will be replaced by n-1 gram $w_{i-n+2}...w_i$. And if $w_{i-1}w_i$ is not in the 2-gram vocabulary then the state units of $w_{i-1}w_i$ will be replaced by the combination of states of w_{i-1} and w_i. If w_i is not in the unigram vocabulary it will be handled as same as in HMM.

5 Tagging Procedures and Decoding Algorithm

The main procedures of COV tagging is described in the following flow diagram.

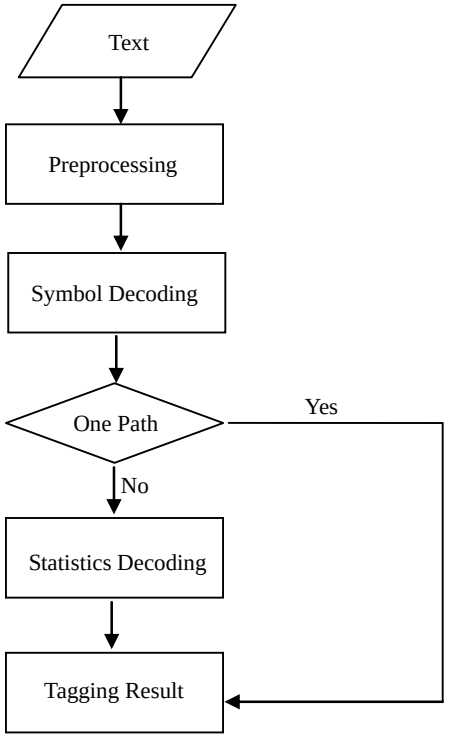

Figure 1: Flow diagram of PoS tagging by COV

There are two steps of decoding in PoS tagging by COV:

(1) Symbol decoding

(2) Statistics decoding

Statistics decoding applies Viterbi algorithm, which is explained in detail by Rabiner (1989) and will not be repeated here.

Here we will describe the symbol decoding algorithm in detail. First we define the suffix and prefix of a state sequence:

Suffix of $q_{i-n+1}...q_i$ is defined as $q_{i-n+2}...q_i$

Prefix of $q_{i-n+1}...q_i$ is defined as $q_{i-n+1}...q_{i-1}$

The symbol decoding algorithm is as follows:

Input: word sequence S= $w_0...w_h$ and all the possible state units of each n-gram unit

(1) Comparing the neighboring n-gram state units from left to right.

For any given neighboring observation units $s_{i-1} = w_{i-n}...w_{i-1}$ and $s_i = w_{i-n+1}...w_i$, they have the corresponding state unit sets e_{i-1} and e_i. And each state unit in the set is called state node.

For each node E_{i-1} in the state set of e_{i-1}, a comparison is made between the suffix of E_{i-1} and the prefix of the node E_i in e_i. If they are the same then a parent-child relation is built between the neighboring nodes E_{i-1} and E_i.

If node E_i in e_i has no parent node in e_{i-1} then E_i will be eliminated and if node E_{i-1} in e_{i-1} has no child node in e_i, E_{i-1} will also be eliminated.

(2) Backward from right to left

A. If a node E_{i-1} is eliminated in step (1) for it doesn't have any child node in e_i, then the relation between E_{i-1} and its parent node E_{i-2} will also be eliminated.

B. If the parent-child relation between E_{i-2} and E_{i-1} is eliminated in step A and E_{i-2} doesn't have any child node then E_{i-2} will also be eliminated.

Backward to the left end of the sequence and the process of symbol decoding finishes.

Table 2 Symbol Decoding Algorithm

After symbol decoding the remaining nodes construct a node lattice. If there is only one path from left to right in the lattice then

decoding finishes and the state sequence is output. Otherwise Viterbi algorithm is used to calculate and select the most probable path.

6 Evaluation Criteria

We use the following criteria to evaluate the performances of COV.
(1) P_A: Overall precision rate
(2) P_M: Precision rate of the multi-class words,
(3)P_O: Precision rate of OOV (Out Of Vocabulary), not including the personal names, location names and organization names, etc.
(4) P_E: Error reduction rate, comparing with the baseline model.
All the above criteria have been introduced in Kupiec (1992) and Cutting (1992) etc and will not be repeated here.
(5) P_S : State certainty rate
In order to measure the statistics decoding complexity, we define State certainty rate P_S.

$$P_S = \frac{count(Total\_State\_Nodes)}{count(Observations)}$$

Count(Total_State_Nodes) denotes the total number of possible states for all the observations in statistics decoding. Due to the symbol decoding many states have been pruned in COV and the search space for statistics decoding is reduced accordingly. The level of search space reduction can be indicated by the criteria of P_s.

7 Experiments

7.1 Corpus and Preprocessing

The training and test data are all taken from the People's Daily of 2000 year, which has been segmented and manually assigned PoS tags by the Peking university. The division of corpus is as follows:

Group	Usage of corpus	Months	Amount of tokens
1	Training	Feb.	1050934
2		Feb.-June.	6142402
3	Open Test	Jan.	1235628
4	Close Test	Feb.	1050934

Table 3 Division of corpus

The baseline model is the 2nd order HMM, whose results will be compared with that of 2-gram COV.
Before training and tagging the corpus is preprocessed. All the named entities such as personal names, location names, organization names and all the digits are replaced by some particular symbols. For example, personal names are all replaced by "*PerN*".

7.2 Results

	P_A	P_M
2nd order HMM	96.54%	92.76%
2-gram COV	98.29%	96.44%
P_E	50.58%	50.83%

Table 4: Results of the close test.
Corpus of group 2 in table 3 is used as the training corpus.

	Group 1	Group 2
2nd order HMM	94.63%	95.73%
2-gram COV	95.53%	96.79%
3-gram COV	95.63%	96.83%

Table 5: P_A of HMM, 2-gram and 3-gram COV in open test.
The corpus of Group 1 and 2 are used as training corpus.
The above results show that 2-gram and 3-gram COV all outperform second order HMM. And 3-gram COV outperforms 2-gram COV, which indicates that with the expansion of observation the precision rate of COV will not decline but increase.

	Group 1	Group 2
2nd order HMM	90.75%	92.02%
2-gram COV	92.66%	94.24%
P_E	20.64%	27.85%

Table 6: P_M of HMM and COV in open test.
The result shows that COV has a better performance in tagging multi-class words than

HMM.

	Group 1	Group 2
HMM	53.21%	55.07%
COV(2-gram)	92.24%	93.99%
COV (unigram)	53.27%	55.35%

Table 7: P_O of HMM and COV

With regard to the 2-gram OOV, the OOV precision rate of COV is higher than 90%, which indicates that COV can well deal with the OOV problem when the observation unit is expanded.

We have done some experiments to compare the time cost and precision rate among HMM, COV and discriminative models such as MaxEnt and CRFs. For the limitation of computer processing power, we choose the People's Daily of January, 2000 as the training data and the first 5000 paragraphs of the People's Daily of February , 2000 as test data. The taggers are the MaxEnt tagger developed by Standford University and CRF++.

	HMM	COV	MaxEnt	CRF 1	CRF 2
Training time	1mins	2mins	4.6hrs	63hrs	60 hrs
Test time	4mins	8mins	11mins	17mins	11mins
P_A	94.23 %	95.43 %	95.69%	95.67 %	95.80 %

Table 8: Training, test time and PA of different models

The template of MaxEnt is: w-1, w0, w+1, prefix of w0, suffix of w0, length of w0

The template of CRF1 is: w-1, w0, w+1, prefix of w0

The template of CRF2 is: w-1, w0, w+1, prefix of w0, suffix of w0, length of w0

The above data show that the precision rate of COV is higher than HMM, and comparable to the discriminative models. Moreover, training time of COV is much less than the discriminative models and almost at the same level as HMM. High precision rate and low time cost makes COV more competitive and practical than other models.

Training Group	HMM	COV	Reduction of Ps	Reduction rate of Ps
1	1.79	1.66	0.14	7.82%
2	2.03	1.57	0.46	22.66%

Table 9: Ps of 2nd order HMM and 2-gram COV

The above result shows that the search space in statistics decoding of COV is smaller than HMM.

We also count the tokens which can be tagged with symbol decoding.

Training Group	Tokens of Symbol Decoding	Percentage of Symbol Decoding	P_A
1	86187	6.98%	99.24%
2	92174	7.46%	99.42%

Table 10: Results of symbol decoding

The total tokens of test corpus is 1235631.

The above data shows that there are about 7% tokens which can be tagged with symbol decoding and without any probability computation. Moreover, the precision rate of symbol decoding is above 99%, which is much higher than the average precision rate.

The smaller search space and higher precision rate proves the efficiency and robustness of COV in PoS tagging.

We also conducted some experiments of English PoS tagging. The training and test data are from the Wall Street Journal (WSJ) in Penn Tree Bank. We use the texts of group 00 to 19 in WSJ as training data and group 00 to 04 as close test data and group 23 to 24 as open test data. The baseline model is also the 2nd order HMM. Results are as follows.

	P_A of	P_A of	P_M of	P_M of

	HMM	COV	HMM	COV
Close Test	97.85%	98.29%	94.85%	96.44%
Open Test	96.48%	96.79%	93.92%	95.18%

Table 11 Results of English PoS tagging Experiments

The above results show that COV also outperforms HMM in English PoS tagging.

8 Discussion

COV is not only suitable to PoS tagging task. We have applied it to the Chinese word segmentation, sentence boundary detection and chunk detection, in which COV also achieves satisfactory results. COV is not limited to the certain language but can be applied in the tagging tasks of different languages. Comparing with HMM, COV has the advantages of smaller search space and higher tagging precision rate. Comparing with the discriminative models, COV has the advantages of less training time and comparable precision rate. All of these prove that COV is a general, efficient and robust model for sequence labeling.

Meanwhile we also find that it is difficult for COV to combine more context and lexical features as discriminative models can do. For example, COV has not taken the suffix or prefix of a word into the model. In fact such information is important for guessing the PoS of unknown words. In the future we will make efforts to take more context and lexical information into the model and improve its performance.

References

L. Rabiner. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. of the IEEE, 77(2).

Church, K. 1988. A stochastic parts program and noun phrase parser for unrestricted text. Proceedings of Second ACL Applied NLP, 136-143.

Scott M. Thede, Mary P. Harper. 1999. Second-order hidden Markov model for part-of-speech tagging. In ACL 37, 175–182.

Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. Computational Linguistics, 21(4):543-565.

Julian Kupiec. 1992. Robust part-of-speech tagging using a hidden Markov model. Computer Speech and Language, 6(3):225-242.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 133-142.

Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In Proceedings of the 3rd Conference on Applied Natural Language Processing (ACL), pages 133-140.

Scott M. Thede and Mary P. Harper. 1999. A second-order hidden Markov model for part-of-speech tagging. In ACL, pages 175–182.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of ICML-01, pages 282-289.

Zhongqiang Huang , Mary P. Harper , Wen Wang. 2007. Mandarin Part-of-Speech Tagging and Discriminative Reranking. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 1093–1102.