DISCRIMINATION ORIENTED PROBABILISTIC TAGGING

Yi-Chung Lin, Tung-Hui Chiang, and Keh-Yih Su

Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.

Abstract

Conventional tagging models, with parameters estimated by the widely used maximum likelihood estimator, usually fail to achieve satisfactory performance in real applications. Since they achieve lexical disambiguation indirectly and implicitly via estimation, these models are usually unable to cover the statistical variation in the real text. In this paper, a discrimination oriented learning algorithm is proposed to directly pursue the goal of lexical disambiguation, so that the modeling error and the estimation error due to insufficient training data can be compensated. A 42% reduction in error rate, has been observed in the task of tagging Brown Corpus by using this proposed method.

1. Introduction

Tagging part of speech (or lexical disambiguation) in a sentence is an important problem in natural language processing. Traditionally, this task is achieved by ruling out the lexical ambiguities with a parser. However, as pointed out by Church[1], a parser is usually not capable to rule out all of those undesired ambiguities. Thus, passing all the combinations of different grammatical categories to the parser still let the problem to be unsolved. However, if there is a mechanism to select only a few combinations to the parser, with high possibility to be correct, it not only reduces the total processing cost, as parsing is a very expensive process, but also enhances the power of disambiguation, as fewer parse tree will be generated.

Several algorithms have been proposed in the literature to select the corrective category from all the possible tags for a given word. Greene and Rubin[2] developed TAGGIT with 3300 *context frame rules*. Each rule deletes one or more candidates from a list of possible tags for each word when its context is satisfied. TAGGIT achieves accuracy rate about 77% in the task of tagging Brown Corpus. Leech, Garside and Atwell[3] tag *LOB Corpus* with CLAWS[4], which is a bigram model with an IDIOMTAG procedure applied after initial tag assignment and before disambiguation, and 96.7% corrective tagging has been reported. Church[1] used a trigram model to tag Brown Corpus and achieved 95%-99% (depend on the definition of *correct*) accuracy. DeRose[5] developed VOLSUNG, which is similar to CLAWS, and reached accuracy rates of 96% without idiom tagging and 99% with idiom tagging for the LOB Corpus. It also achieves 96% accuracy rate for tagging Brown Corpus. Recently, several probabilistic models based on trigram are investigated on different corpus[6][7], and have made some improvement.

All above models (except TAGGIT) use parameters estimated by maximum likelihood estimator. Correct disambiguation, however, depends only upon correct rank ordering of different category sequences. Therefore, maximizing likelihood does not imply minimizing the error rate of disambiguation[8][9]. Thus, a discriminative learning procedure is required to tune the model parameters to achieve high performance. Furthermore, due to insufficient amount of training data and incompleteness of model knowledge, the statistical variation between the testing set and the training set is usually not well characterized in those conventional approaches, therefore, minimizing the error rate in the training set does not necessarily imply maximizing the disambiguation accuracy in real applications. To achieve satisfactory result in real applications, this discriminative learning procedure must also be robust.

In this paper, a discrimination oriented learning procedure is proposed to fine tune the model parameters. Parameters are adjusted to shift the correct category sequence to the top rank among different combinations of categories during learning process. Great improvement, 42% reduction in error rate, have been observed in the task of tagging Brown Corpus.

2. Simulation Setup

2.1. Corpus Preparation

Brown Corpus is selected in this paper to compare different approaches, because it is the most well-known and widely-used corpus. Using *sentence closer* tag [10] as the delimiter between sentences, we extract 1,147,474 words (including sentence markers), of 54,597 sentences from Brown Corpus. No morphological analysis is done in preparing the training set. So words with different characters (such as *advantage* and *advantages*) are considered as different words. In the same way, the tags *PPS*, *MD* and *PPS+MD*, for the words *he*, *will* and *he'll*, are also treated as three different tags. Based on this, we construct a dictionary with 49,705 different words and a tag set with 187 different tags (not 87 tags stated in[10]).

Because it is the performance in the real applications (i.e., the testing set in our case) that we really care, the whole corpus are separated into two sets:

1. *Training set* — contains 919,247 words in 43,677 sentences, which is used to train the model parameters.
2. *Testing set* — contains 228,227 words in 10,920 sentences, which is used to estimate the accuracy rate of different tagging procedures.

2.2. Probabilistic Model

The purpose of lexical disambiguation is to find a correct part of speech sequence "$c_1, c_2, \cdots, c_N$" for a given sentence, "$w_1, w_2, \cdots, w_N$", where w_j is the j-th word of the given sentence and c_j is the part of speech assigned to the j-th word. This problem can be formulated as to find argmax$P(c_1^N|w_1^N)$, where c_1^N and w_1^N are the short-hand notations for "$c_1, c_2, \cdots, c_N$" and "$w_1, w_2, \cdots, w_N$" respectively. $P(c_1^N|w_1^N)$ can be further derived, using the multiplication rule in probability theory, as the following equation

$$P\left(c_1^N|w_1^N\right) = \prod_{j=1}^{N} P\left(c_j|c_1^{j-1}, w_1^N\right). \tag{1}$$

However, it is infeasible to directly estimate the parameter $P(\ c_j|c_1^{j-1}, w_1^N)$, for it demands a huge amount of data to train those a lot of parameters. To make it practical, assumptions must be made to simplify the evaluation process of $P(\ c_j|c_1^{j-1}, w_1^N)$. It is obvious that the correct category of a word in a sentence strongly depends on the word itself and the categories from the adjacent words. So, it is reasonable to make either of the following assumptions :

1. Assume $P(\ c_j|c_1^{j-1}, w_1^N) \approx P(c_j|w_j)P(c_j|c_{j-1})$. This is the bigram[1] model used in CLAWS[3].
2. Assume $P(\ c_j|c_1^{j-1}, w_1^N) \approx P(c_j|w_j)P(c_j|c_{j-2}, c_{j-1})$. This is the trigram model proposed by Church[1].

The probability $P(\ c_j|w_j)$ is called lexical probability, and $P(c_j|c_{j-1})$ or $P(c_j|c_{j-2}, c_{j-1})$ is called context (or transition) probability.

Using the above assumptions, the problem of lexical disambiguation can be formulated as to find argmax$(\prod_{j=1}^{N} P(c_j|w_j)P(c_j|c_{j-n}^{j-1}))$, where n=1 or 2. The *beginning of sentence* marker is assigned to c_0 and c_{-1} in the above formulation.

2.3. Baseline Performance

The context probabilities $P(c_j|c_{j-n}^{j-1})$ are first obtained from the training corpus by maximum likelihood estimator. For example, given a sentence "*I saw a beautiful girl*", one possible category sequence is "*pron v art adj n*", then the value of probability of *P(n|art,adj)* is estimated by *C(art adj n)/C(art adj)*, where *C(art adj n)* is the number of occurrences of the tri-POS[2] "*art adj n*" in the training corpus, and *C(art adj)*= Σ_X*C(art adj X)* where X is any possible tag. The lexical probability is estimated in a similar way.

Table 1 and 2 lists the performance of those bigram and trigram models. These results will be used as the baseline performance in the following tests. There are 1,147,474 words (including sentence markers) in the Brown Corpus, but only 40% of these words are *ambiguous* (i.e., words

[1] Based on the assumption that the next word which will be uttered depends only on the previous one or two words, bigram and trigram language models are widely used in speech recognition. Church[1] used the terms of bigram and trigram to indicate that the next category is strongly depends on the previous one or two categories, respectively. We will follow his notations in this paper.
[2] In this paper, a tri-POS is defined as a sequence three categories (i.e. sequence of "$c_{j-2}\ c_{j-1}\ c_j$". In the same way, bi-POS is defined as a sequence of two categories like "$c_{j-1}\ c_j$".

with two or more categories, and are called *ambiguous words*). Therefore, using word accuracy to measure performance is not a good way, because most words in the corpus can have only one category. In this paper, the word accuracy rate is reported on the *ambiguous word accuracy*, which is defined as N_A/N_W, where N_A is the number of ambiguous words which are correct tagged and N_W is the total number of ambiguous words in the corpus. The error rate of ambiguous words is defined as $1-N_A/N_W$. In the same way, the sentence accuracy rate is defined as N_C/N_S, where N_C is the number of sentences in which every word is correct tagged, and N_S is the number of sentences in corpus.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	55.65	91.66	8.34
trigram	64.96	93.95	6.05

Table 1 Baseline performance in the training set.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	53.34	91.04	8.96
trigram	55.34	91.44	8.56

Table 2 Baseline performance in the testing set.

Table 1 and 2 shows that the accuracy rate of trigram model in the training set is much better than that of bigram model, but, in the testing set, the performance of trigram model are just slightly better than that of bigram model. The high accuracy rate of trigram model in training set is due to the phenomena of *over-tuning[9]*. The large difference between the accuracy rate of the training set and that of the testing set for trigram model is mainly due to the insufficience of training data.

3. Discrimination Oriented Learning

In section 2.2, the disambiguation process is formulated as to find argmax$P(c_1^N|w_1^N)$, and the simplified form of $\prod_{j=1}^{N} P(c_j|w_j)P(c_j|c_{j-n}^{j-1})$ is used to calculate $P(c_1^N|w_1^N)$. For the convenience of real applications, a score function is defined in here as

$$\begin{aligned}
\text{Score} &= \log\left\{\prod_{j=1}^{N} P\left(c_j|w_j\right) P\left(c_j|c_{j-n}^{j-1}\right)\right\} \\
&= \sum_{j=1}^{N}\left\{\log\left(P\left(c_j|w_j\right)\right) + \log\left(P\left(c_j|c_{j-n}^{j-1}\right)\right)\right\} \qquad (2) \\
&= \sum_{j=1}^{N}\left\{S\left(c_j|w_j\right) + S\left(c_j|c_{j-n}^{j-1}\right)\right\},
\end{aligned}$$

where $S(c_j|w_j)=\log\left(P\left(c_j|w_j\right)\right)$, is called lexical score and $S(c_j|c_{j-n}^{j-1})=\log\left(P\left(c_j|c_{j-n}^{j-1}\right)\right)$, is called context score. Then, the lexical disambiguation process is to calculate the score function for all the possible category sequences of a input sentence, and to choose the category sequence which has the highest score. In baseline models, the parameters used to calculate the score function are estimated without considering the competing category sequences. So, they can not minimize the error rate in the training corpus.

In order to minimize the error rate of the training corpus, a discrimination oriented learning procedure[11][9] is adopted to tune the parameters (i.e., the lexical and context scores) in this paper. Without loss of generality, we use the bigram model and a sentence with three different possible category sequences to show how to tune the parameters. Assume that the sentence "*Press the left button*" has only one ambiguous word "*left*" with possible tags *v*, *n* and *adj*. The correct category sequence should be "*v art adj n*" in this case. The disambiguation process, before learning, is listed in Table 3. As the candidate 1, a wrong category sequence, has the highest score, an error is made.

		Press	the	left	button	sub total	total
candidate 1	@	v	art	v	n		
lexical score		0	0	-0.3*	0	-0.3	-2.38
contex score		-0.7	-0.52	-0.7*	-0.16*	-2.08	
candidate 2	@	v	art	n	n		
lexical score		0	0	-0.7	0	-0.7	-2.92
contex score		-0.7	-0.52	-0.3	-0.7	-2.22	
candidate 3	@	v	art	adj	n		
lexical score		0	0	-0.52*	0	-0.52	-2.42
contex score		-0.7	-0.52	-0.52*	-0.16*	-1.90	

Table 3 Disambiguation process before learning. The symbol @ is *beginning of sentence* marker. The marker * denotes those parameters which will be adjusted.

Comparing candidate 1 and candidate 3 (the correct sequence), we find that the parameters *S*(v|*left*), *S*(v|*art*), *S*(*n*|v), *S*(*adj*|*left*), *S*(*adj*|*art*) and *S*(*n*|*adj*) are involved in the incorrect decision.

If we can increase the parameters *S(adj|left)*, *S(adj|art)* and *S(n|adj)*, and decrease the parameters *S(v|left)*, *S(v|art)* and *S(n|v)*, we can make the correct category sequence to have the highest score. To adjust these parameters, a score vector is first defined as

$$\begin{aligned}\vec{S} &= (s_1, s_2, s_3, s_4, s_5, s_6)\\ &= (S(adj|left), S(adj|art), S(n|adj),\\ &\quad S(v|left), S(v|art), S(n|v)),\end{aligned}$$

then the following equations are used to tune the score vector.

$$\vec{S}_{t+1} = \vec{S}_t + \triangle\vec{S}_t, \tag{4}$$

where

$$\begin{aligned}\triangle\vec{S} &= \varepsilon C H\left(\vec{X}, \vec{S}\right),\\ H\left(\vec{X}, \vec{S}\right) &= l'(d)\frac{1}{\left\|\vec{X}\right\|\sqrt{1-d^2}} T\left(\vec{S}\right)\vec{X},\\ d &= \frac{\vec{S}^t\vec{X}}{\left\|\vec{S}\right\|\left\|\vec{X}\right\|},\\ l'(d) &= \frac{d_0}{d_0^2 + d^2},\\ T\left(\vec{S}\right) &= \frac{\left\|\vec{S}\right\|\vec{E} - \vec{S}\vec{S}^t}{\left\|\vec{S}\right\|^3}.\end{aligned} \tag{5}$$

In equation (5), ϵ is a small constant to control the convergence speed of learning process, C is positive-definite matrix and d_0 is the window size[11]. The vector $\vec{X}$ in this example is $(1, 1, 1, -1, -1, -1)$, such that $\vec{S}^t\vec{X}$ is the difference between the score of candidate 3 and that of candidate 1. As the details of the learning process have already been investigated in the literature[11], we will not give the detail derivations here. Using the above equations, the disambiguation process after learning is listed in Table 4.

		Press	the	left	button	sub total	total
candidate 1	@	v	art	v	n		
lexical score		0	0	-0.35*	0	-0.35	-2.51
contex score		-0.7	-0.52	-0.74*	-0.20*	-2.16	
candidate 2	@	v	art	n	n		
lexical score		0	0	-0.7	0	-0.7	-2.92
contex score		-0.7	-0.52	-0.3	-0.7	-2.22	
candidate 3	@	v	art	adj	n		
lexical score		0	0	-0.48*	0	-0.48	-2.29
contex score		-0.7	-0.52	-0.48*	-0.11*	-1.81	

Table 4 Disambiguation process after learning. The marker * denotes those parameters which should be adjusted during learning.

After discrimination oriented learning, the accuracy of lexical disambiguation is improved greatly. Comparing Table 1, 2 , 5 and 6, the error rate of ambiguous words of bigram model is reduced from 8.96% to 5.66% in the testing set, i.e., about 37% error rate reduction. For trigram model, the error rate of ambiguous words in testing set is reduced from 8.56% to 6.4%, about 25% error rate reduction. In the training set, the decrease of error rate of bigram and trigram models are 48% and 58% respectively, but these improvements are not important because the error rate of real applications is approximated by the performance in the testing set not the training set.

One phenomena should be noticed in Table 5 and 6. Although the accuracy rate of trigram is much better than that of bigram in the testing set, the accuracy rate of trigram is worse than that of bigram in the testing set. This problem is due to the limited size of training corpus and will be discussed in next section.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	73.94	95.66	4.34
trigram	83.89	97.44	2.56

Table 5 Performance in the training set after learning.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
bigram	65.91	94.32	5.68
trigram	63.65	93.60	6.40

Table 6 Performance in the testing set after learning.

4. Merging Unreliable Parameters

Due to the limited size of training corpus, trigram model suffers the problem of *over-tuning*, which usually occurs when the number of available training data is not large enough compared to the number of parameters. In this situation, the learning process will be lead to a pseudo optimal point in the training corpus, which sometimes even degrades the performance in the testing set. This phenomena is shown in Table 5 and 6 that the performance of trigram in testing set is poorer than that of bigram, although the performance of trigram is much better than that of bigram in the training corpus. One way to overcome this problem is to replace the unreliable parameters of trigram, i.e., whose number of occurrences in the training corpus are below a threshold, with the more reliable parameters of bigram. For example, if the tri-POS (*art v n*) and (*prep v n*) occurred less than R times in the training corpus, then the parameter $S(n|v)$, instead of $S(n|art,v)$ and $S(n|prep,v)$, will be used in the learning process.

The merging procedure described above is similar to the *backing-off* procedure[12][7]. However, the proposed approach differs from the backing-off approach in that the parameters corresponding to bi-POS will be adjusted during learning process, instead of using them directly as backing-off procedure does. The threshold R is found to be insensitive in a wide range from 1 to 50 and is set to 20 in our simulation. Figure 1 displays the behavior of learning process for the case of merging the unreliable parameters and Table 7 shows the final performance.

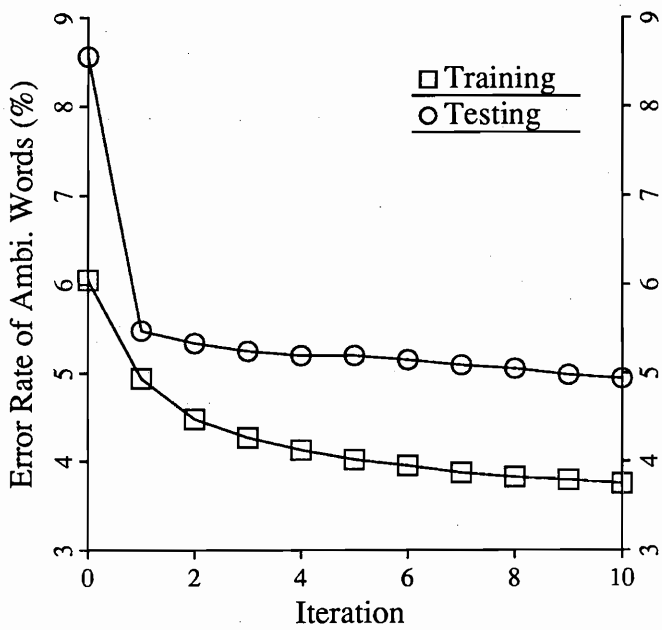

Figure 1 The error rates of merged trigram model during parameters-merged learning.

	Sentence Accuracy (%)	Ambi. Word Accuracy (%)	Ambi. Word Error Rate (%)
training	76.51	96.23	3.77
testing	69.76	95.05	4.95

Table 7 Final performance of merged trigram model in both training set and testing set.

The reason for the improvement of performance is : although trigram carries more discriminative informations, they are poorly estimated (or trained) for not having enough data, and thus is quite unreliable to be used in the testing set. To replace those unreliable parameters with more reliable parameters from bigram, although they carry less discriminative informations, we sacrifice a small amount of modeling error for reducing a large amount of estimation error in the testing set, thus to improve the performance in the testing set. Figure 2 shows the improvements made by discrimination oriented learning and parameters-merged learning.

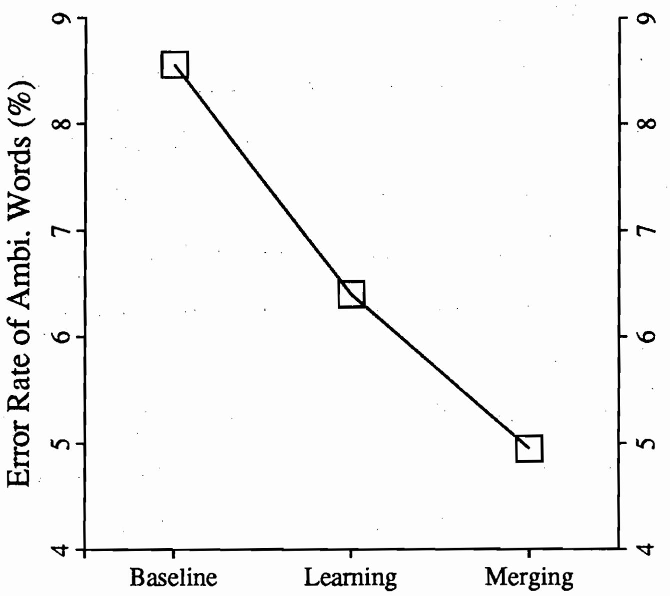

Figure 2 The performance improvement of trigram model. *Baseline* means parameters are estimated by maximum likelihood estimator. *Learning* means the parameters are tuned by discrimination oriented learning. *Merging* means the parameters are merged and then tuned.

5. Conclusion

Recently, probabilistic models are widely used for lexical disambiguation. In conventional probabilistic approaches, model parameters are estimated by maximum likehood estimator without considering the competing candidates, therefore, the can not minimize the error rate of lexical disambiguation. In this paper, a discrimination oriented learning method is proposed to tune the parameters. The method results in 37% and 25% error rate reductions of ambiguous words for bigram and trigram models in the testing set. To further improve the performance, a merging procedure is used to conquer the problem of over-tuning and make the model more robust. Using those merged parameters for learning, great improvement, 42% reduction in error rate, have been observed in the task of tagging Brown Corpus.

Reference

[1] K. W. Church, “A stochastic patrts program and noun phrase parser for unrestricted text,” in *ACL Proceedings of the 2nd Conference on Applied Natural Language Processing*, Austin, TX, USA, pp. 299–307, Feb 9-12 1988.

[2] B. B. Greene and G. M. Rubin, *Automatical grammatical tagging of English.* Rhode Island: Department of Linguistics, Brown University, 1971.

[3] G. Leech, R. Garside, and E. Atwell, "The automatic grammatical tagging of the LOB corpus," *ICAME News 7*, pp. 13–33, 1983.

[4] B. M. Booth, "Revising CLAWS," *ICAME News 9*, pp. 29–35, 1985.

[5] S. J. DeRose, "Grammatical category disambiguation by statistical optimization," *Computational Linguistics*, vol. 14, pp. 31–39, Winter 1985.

[6] B. Merialdo, "Tagging text with a probabilistic model," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, Canada, pp. 809–812, May 14-17 1991.

[7] B. Maltese and F. Mancini, "A technique to automatically assign parts-of-speech to words raking into account word-ending information through a probabilistic model," in *Proceedings of the 2nd European Conference on Speech Communication and Technology*, Genova, Italy, pp. 753–756, Sep 24-26 1991.

[8] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "A new algorithm for the estimation fo hidden markov model parameters," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, New York, USA, pp. 493–496, Apr 1988.

[9] K.-Y. Su and C.-H. Lee, "Robustness and discrimination oriented speech recognition using weighted HMM and subspace projection approaches," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toronto, Ontario, Canada, pp. 541–544, May 14-17 1991.

[10]W. N. Francis and H. Kucera, *Frequency analysis of English usage*. Houghton Mifflin Company, 1982.

[11]S. Amari, "A theory of adaptive pattern classfiers," *IEEE Trans. on Electronic Computers*, vol. EC-16, pp. 299–307, June 1967.

[12]S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 400–401, Mar. 1987.