Augmenting a Hidden Markov Model for Phrase-Dependent Word Tagging

Julian Kupiec
XEROX PALO ALTO RESEARCH CENTER
3333 Coyote Hill Road
Palo Alto, CA 94304

Abstract

The paper describes refinements that are currently being investigated in a model for part-of-speech assignment to words in unrestricted text. The model has the advantage that a pre-tagged training corpus is not required. Words are represented by equivalence classes to reduce the number of parameters required and provide an essentially vocabulary-independent model. State chains are used to model selective higher-order conditioning in the model, which obviates the proliferation of parameters attendant in uniformly higher-order models. The structure of the state chains is based on both an analysis of errors and linguistic knowledge. Examples show how word dependency across phrases can be modeled.

Introduction

The determination of part-of-speech categories for words is an important problem in language modeling, because both the syntactic and semantic roles of words depend on their part-of-speech category (henceforth simply termed "category"). Application areas include speech recognition/synthesis and information retrieval. Several workers have addressed the problem of tagging text. Methods have ranged from locally-operating rules (Greene and Rubin, 1971), to statistical methods (Church, 1989; DeRose, 1988; Garside, Leech and Sampson, 1987; Jelinek, 1985) and back-propagation (Benello, Mackie and Anderson, 1989; Nakamura and Shikano, 1989).

The statistical methods can be described in terms of Markov models. States in a model represent categories $\{c_1 ... c_n\}$ (n is the number of different categories used). In a first order model, C_i and C_{i-1} are random variables denoting the categories of the words at position i and $(i-1)$ in a text. The transition probability $P(C_i = c_x \mid C_{i-1} = c_y)$ linking two states c_x and c_y, represents the probability of category c_x following category c_y. A word at position i is represented by the random variable W_i, which ranges over the vocabulary $\{w_1 ... w_v\}$ (v is the number of words in the vocabulary). State-dependent probabilities of the form $P(W_i = w_a \mid C_i = c_x)$ represent the probability that word w_a is seen, given category c_x. For instance, the word "dog" can be seen in the states *noun* and *verb*, and only has a non-zero probability in those states. A word sequence is considered as being generated from an underlying sequence of categories. Of all the possible category sequences from which a given word sequence can be generated, the one which maximizes the probability of the words is used. The Viterbi algorithm (Viterbi, 1967) will find this category sequence. The systems previously mentioned require a pre-tagged training corpus in order to collect word counts or to perform back-propagation. The Brown Corpus (Francis and Kucera, 1982) is a notable example of such a corpus, and is used by many of the systems cited above.

An alternative approach taken by Jelinek, (Jelinek, 1985) is to view the training problem in terms of a "hidden" Markov model: that is, only the words of the training text are available, their corresponding categories are not known. In this situation, the Baum-Welch algorithm (Baum, 1972) can be used to estimate the model parameters. This has the great advantage of eliminating the pre-tagged corpus. It minimizes the resources required, facilitates experimentation with different word categories, and is easily adapted for use with other languages.

The work described here also makes use of a hidden Markov model. One aim of the work is to investigate the quality and performance of models with minimal parameter descriptions. In this regard, word equivalence

classes were used (Kupiec, 1989). There it is assumed that the distribution of the use of a word depends on the set of categories it can assume, and words are partitioned accordingly. Thus the words "play" and "touch" are considered to behave identically, as members of the class *noun-or-verb*, and "clay" and "zinc"are members of the class *noun*. This partitioning drastically reduces the number of parameters required in the model, and aids reliable estimation using moderate amounts of training data. Equivalence classes $\{Eqv_1...Eqv_m\}$ replace the words $\{w_1...w_v\}$ $(m \ll v)$ and $P(Eqv_i \mid C_i)$ replace the parameters $P(W_i \mid C_i)$. In the 21 category model reported in Kupiec (1989) only 129 equivalence classes were required to cover a 30,000 word dictionary. In fact, the number of equivalence classes is essentially independent of the size of the dictionary, enabling new words to be added without any modification to the model.

Obviously, a trade-off is involved. For example, "dog" is more likely to be a noun than a verb and "see" is more likely to be a verb than a noun. However they are both members of the equivalence class *noun-or-verb*, and so are considered to behave identically. It is then local word context (embodied in the transition probabilities) which must aid disambiguation of the word. In practice, word context provides significant constraint, so the trade-off appears to be a remarkably favorable one.

The Basic Model

The development of the model was guided by evaluation against a simple basic model (much of the development of the model was prompted by an analysis of the errors in its behaviour). The basic model contained states representing the following categories:

Determiner	
Noun Singular	Including mass nouns
Noun Plural	
Proper Noun	
Pronoun	
Adverb	
Conjunction	Co-ordinating and subordinating
Preposition	
Adjective	Including comparative and superlative
Verb Uninflected	
Verb 3rd Pers. Sing.	
Auxiliary	Am, is, was, has, have, should, must, can, might, etc.
Present Participle	Including gerund
Past Participle	Including past tense
Question Word	When, what, why, etc.
Unknown	Words whose stems could not be found in dictionary.
Lisp	Used to tag common symbols in the the Lisp programming language (see below:)
To-inf.	"To" acting as an infinitive marker
Sentence Boundary	

The above states were arranged in a first-order, fully connected network, each state having a transition to every other state, allowing all possible sequences of categories. The training corpus was a collection of electronic mail messages concerning the design of the Common-Lisp programming language - a somewhat less than ideal representation of English. Many Lisp-specific words were not in the vocabulary, and thus tagged as *unknown*, however the *lisp* category was nevertheless created for frequently occurring Lisp symbols in an attempt to reduce bias in the estimation. It is interesting to note that the model performs very well, despite such "noisy" training data. The training was sentence-based, and the model was trained using 6,000 sentences from the corpus. Eight iterations of the Baum-Welch algorithm were used.

The implementation of the hidden Markov model is based on that of Rabiner, Levinson and Sondhi (1983). By exploiting the fact that the matrix of probabilities $P(Eqv_i \mid C_i)$ is sparse, a considerable improvement can be gained over the basic training algorithm in which iterations are made over all states. The initial values of the model parameters are calculated from word occurrence probabilities, such that words are initially

assumed to function equally probably as any of their possible categories. Superlative and comparative adjectives were collapsed into a single *adjective* category, to economize on the overall number of categories. (If desired, after tagging the finer category can be replaced). In the basic model all punctuation except sentence boundaries was ignored. An interesting observation is worth noting with regard to words that can act both as auxiliary and main verbs. Modal auxiliaries were consistently tagged as *auxiliary* whereas the tagging for other auxiliaries (e.g. "is" "have" etc.) was more variable. This indicates that modal auxiliaries can be recognized as a natural class via their pattern of usage.

Extending the Basic Model

The basic model was used as a benchmark for successive improvements. The first addition was the correct treatment of all non-words in a text. This includes hyphenation, punctuation, numbers and abbreviations. New categories were added for *number*, *abbreviation*, and *comma*. All other punctuation was collapsed into the single new *punctuation* category.

Refinement of Basic Categories

The verb states of the basic model were found to be too coarse. For example, many noun/verb ambiguities in front of past participles were incorrectly tagged as verbs. The replacement of the *auxiliary* category by the following categories greatly improved this:

Category Name	Words included in Category
Be	be
Been	been
Being	being
Have	have
Have*	has, have, had, having
be*	is, am, are, was, were
do*	do, does, did
modal	Modal auxiliaries

Unique Equivalence Classes for Common Words

Common words occur often enough to be estimated reliably. In a ranked list of words in the corpus the most frequent 100 words account for approximately 50% of the total tokens in the corpus, and thus data is available to estimate them reliably. The most frequent 100 words of the corpus were assigned individually in the model, thereby enabling them to have different distributions over their categories. This leaves 50% of the corpus for training all the other equivalence classes.

Editing the Transition Structure

A common error in the basic model was the assignment of the word "to" to the *to-inf* category ("to" acting as an infinitive marker) instead of *preposition* before noun phrases. This is not surprising, because "to" is the only member of the *to-inf* category, $P(W_i = \text{"to"} \mid C_i = \textit{to-inf}) = 1.0$. In contrast, $P(W_i = \text{"to"} \mid C_i = \textit{preposition}) = 0.086$, because many other words share the *preposition* state. Unless transition probabilities are highly constraining, the higher probability paths will tend to go through the *to-inf* state. This situation may be addressed in several ways, the simplest being to initially assign zero transition probabilities from the *to-inf* state to states other than verbs and the *adverb* state.

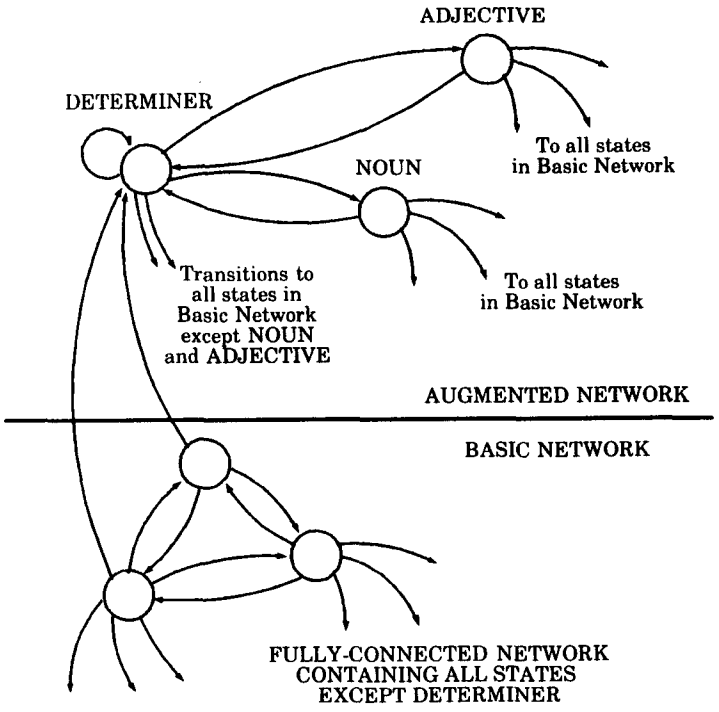

Figure 1: Extending the Basic Model

Augmenting the Model by Use of Networks

The basic model consists of a first-order fully connected network. The lexical context available for modeling a word's category is solely the category of the preceding word (expressed via the transition probabilities $P(C_i \mid C_{i-1})$. Such limited context does not adequately model the constraint present in local word context. A straightforward method of extending the context is to use second-order conditioning which takes account of the previous two word categories. Transition probabilities are then of the form $P(C_i \mid C_{i-1}, C_{i-2})$. For an n category model this requires n^3 transition probabilities. Increasing the order of the conditioning requires exponentially more parameters. In practice, models have been limited to second-order, and smoothing methods are normally required to deal with the problem of estimation with limited data. The conditioning just described is uniform - all possible two-category contexts are modeled. Many of these neither contribute to the performance of the model, nor occur frequently enough to be estimated properly: e.g. $P(C_i = \textit{determiner} \mid C_{i-1} = \textit{determiner}, C_{i-2} = \textit{determiner})$.

An alternative to uniformly increasing the order of the conditioning is to extend it selectively. Mixed higher-order context can be modeled by introducing explicit state sequences. In the arrangement the basic first-order network remains, permitting all possible category sequences, and modeling first-order dependency. The basic network is then augmented with the extra state sequences which model certain category sequences in more detail. The design of the augmented network has been based on linguistic considerations and also upon an analysis of tagging errors made by the basic network.

As an example, we may consider a systematic error made by the basic model. It concerns the disambiguation of the equivalence class *adjective-or-noun* following a determiner. The error is exemplified by the sentence fragment "The period of...", where "period" is tagged as an adjective. To model the context necessary to correct the error, two extra states are used, as shown in Figure 1. The "augmented network" uniquely models all second-order dependencies of the type determiner - noun - X, and determiner - adjective - X (X ranges over $\{c_1 ... c_n\}$). Training a hidden Markov model having this topology corrected all nine instances of the error in the test data. An important point to note is that improving the model detail in this manner does not forcibly correct the error. The actual patterns of category usage must be distinct in the language.

To complete the description of the augmented model it is necessary to mention *tying* of the model states (Jelinek and Mercer, 1980). Whenever a transition is made to a state, the state-dependent probability distribution $P(Eqv_i \mid C_i)$ is used to obtain the probability of the observed equivalence class. A state is generally used in several places (E.g. in Figure 1. there are two *noun* states, and two *adjective* states: one of each in the augmented network, and in the basic network). The distributions $P(Eqv_i \mid C_i)$ are considered to be the same for every instance of the same state. Their estimates are pooled and re-assigned identically after each iteration of the Baum-Welch algorithm.

Modeling Dependencies across Phrases

Linguistic considerations can be used to correct errors made by the model. In this section two illustrations are given, concerning simple subject/verb agreement across an intermediate prepositional phrase. These are exemplified by the following sentence fragments:

1. "Temperatures in the upper mantle range apparently from....".
2. "The velocity of the seismic waves rises to...".

The basic model tagged these sentences correctly, except for "range" and "rises" which were tagged as *noun* and *plural-noun* respectively[1]. The basic network cannot model the dependency of the number of the verb on its subject, which precedes it by a prepositional phrase. To model such dependency across the phrase, the networks shown in Figure 2 can be used. It can be seen that only simple forms of prepositional phrase are modeled in the networks; a single noun may be optionally preceded by a single adjective and/or determiner. The final transitions in the networks serve to discriminate between the correct and incorrect category assignment given the selected preceding context. As in the previous section, the corrections are not programmed into the model. Only context has been supplied to aid the training procedure, and the latter is responsible for deciding which alternative is more likely, based on the training data. (Approximately 19,000 sentences were used to train the networks used in this example).

Discussion and Results

In Figure 2, the two copies of the prepositional phrase are trained in separate contexts (preceding singular/plural nouns). This has the disadvantage that they cannot share training data. This problem could be resolved by tying corresponding transitions together. Alternatively, investigation of a trainable grammar (Baker, 1979; Fujisaki et al., 1989) may be a fruitful way to further develop the model in terms of grammatical components.

A model containing all of the refinements described, was tested using a magazine article containing 146 sentences (3,822 words). A 30,000 word dictionary was used, supplemented by inflectional analysis for words not found directly in the dictionary. In the document, 142 words were tagged as *unknown* (their possible categories were not known). A total of 1,526 words had ambiguous categories (i.e. 40% of the document). Critical examination of the tagging provided by the augmented model showed 168 word tagging errors, whereas the basic model gave 215 erroneous word tags. The former represents 95.6% correct word tagging on the text as a whole (ignoring unknown words), and 89% on the ambiguous words. The performance of a tagging program depends on the choice and number of categories used, and the correct tag assignment for words is not always obvious. In cases where the choice of tag was unclear (as often occurs in idioms), the tag was ruled as incorrect. For example, 9 errors are from 3 instances of "... as well as ..." that arise in the text. It would be appropriate to deal with idioms separately, as done by Garside, Leech and Sampson (1987). Typical errors beyond the scope of the model described here are exemplified by incorrect adverbial and prepositional assignment.

[1] It is easy to construct counter-examples to the sentences presented here, where the tagging would be correct. However, the training procedure affirms that counter-examples occur less frequently in the corpus than the cases shown here.

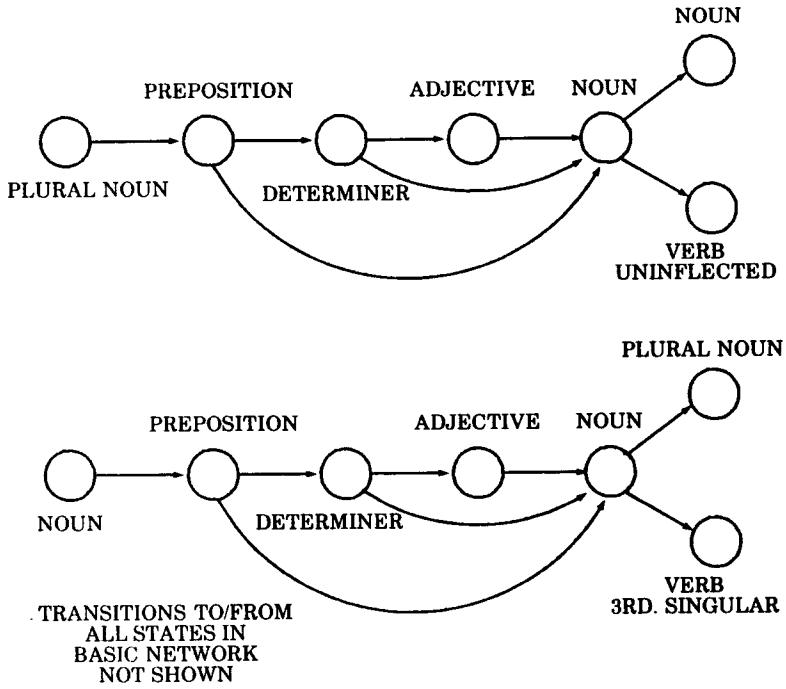

Figure 2: Augmented Networks for Example of Subject/Verb Agreement

For example, consider the word "up" in the following sentences:

"He ran up a big bill".
"He ran up a big hill".

Extra information is required to assign the correct tagging. In these examples it is worth noting that even if a model was based on individual words, and trained on a pre-tagged corpus, the association of "up" (as adverb) with "bill" would not be captured by trigrams. (Work on phrasal verbs, using mutual information estimates (Church et al., 1989b) is directly relevant to this problem). The tagger could be extended by further category refinements (e.g. inclusion of a *gerund* category), and the single *pronoun* category currently causes erroneous tags for adjacent words. With respect to the problem of *unknown* words, alternative category assignments for them could be made by using the context embodied in transition probabilities.

Conclusions

A stochastic method for assigning part-of-speech categories to unrestricted English text has been described. It minimizes the resources required for high performance automatic tagging. A pre-tagged training corpus is not required, and the tagger can cope with words not found in the training text. It can be trained reliably on moderate amounts of training text, and through the use of selectively augmented networks it can model high-order dependencies without requiring an excessive number of parameters.

Acknowledgements

I would like to thank Meg Withgott and Lauri Karttunen of Xerox PARC, for their helpful contributions to this work. I am also indebted to Sheldon Nicholl of the Univ. of Illinois, for his comments and valuable insight. This work was sponsored in part by the Defense Advanced Research Projects Agency (DOD), under the Information Science and Technology Office, contract #N00140-86-C-8996.

References

J.K. Baker. Trainable Grammars for Speech Recognition. Speech Communications Paper. 97th. Meeting of Acoustical Soc. of America, Cambridge, MA, 1979.

L.E. Baum. An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process. Inequalities, 3, 1972. pp. 1-8.

J. Benello, A. Mackie, J. Anderson. Syntactic Category Disambiguation with Neural Networks. Computer Speech and Language, Vol. 3, No. 3, July 1989. pp. 203-217.

K. Church. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. Proc. Int. Conf. on Acoustics, Speech and Signal Processing, 1989. pp. 695-698.

(1989b) K. Church, W. Gale, P. Hanks, D. Hindle. Parsing, Word Associations and Typical Predicate-Argument Relations. Proc. Int. Workshop on Parsing Technologies, Pittsburgh PA, Aug. 28-31 1989. pp. 389-398.

S. DeRose. Grammatical Category Disambiguation by Statistical Optimization. Computational Linguistics, Vol. 14, No 1. 1988.

W.N. Francis, H. Kucera. Frequency Analysis of English Usage. Houghton Mifflin, 1982.

T. Fujisaki, F. Jelinek, J. Cocke, E. Black, T. Nishino. A Probabilistic Method for Sentence Disambiguation. Proc. Int. Workshop on Parsing Technologies, Pittsburgh PA, Aug. 28-31 1989. pp. 85-94.

R. Garside, G. Leech, G. Sampson. The Computational Analysis of English. Longman, 1987.

B.B. Greene, G.M. Rubin. Automatic Grammatical Tagging of English. Dept. of Linguistics, Brown Univ., Providence. 1971.

F. Jelinek. Self-Organized Language Modeling for Speech Recognition. Unpublished Technical Report, 1985. IBM T.J. Watson Research Center, Yorktown Heights, N.Y.

F. Jelinek, R.L. Mercer. Interpolated Estimation of Markov Source Parameters from Sparse Data. Proc. Workshop Pattern Recognition in Practice, May 21-23 1980. Amsterdam, The Netherlands. North-Holland.

J. Kupiec. Probabilistic Models of Short and Long Distance Word Dependencies in Running Text. Proc. DARPA Speech and Natural Language Workshop, Philadelphia, Feb. 21-23 1989. pp. 290-295.

M. Nakamura, K. Shikano. A Study of English Word Category Prediction Based on Neural Networks. Proc. Int. Conf. on Acoustics, Speech and Signal Processing, 1989. pp. 731-734.

L.R. Rabiner, S.E. Levinson, and M.M. Sondhi. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. Bell System Technical Journal, Vol. 62, No. 4, April 1983. pp 1035-1074.

A.J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. IEEE Trans. on Information Theory Vol. IT-13, April 1967. pp. 260-269.