From: AAAI Technical Report WS-92-01. Compilation copyright © 1992, AAAI (www.aaai.org). All rights reserved.

Training Stochastic Grammars From Unlabelled Text Corpora

Julian Kupiec and John Maxwell
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Abstract

The paper describes various aspects and practicalities of applying the "Hidden Markov" approach to train parameters of regular and context-free stochastic grammars. The approach enables grammars to be trained from unlabelled text corpora, providing flexibility in the choice of syntactic categories and text domain. Part-of-speech tagging and parsing are discussed as applications. Linguistic considerations can be used to develop constrained grammars, providing appropriate higher-order context for disambiguation. Unconstrained grammars provide the opportunity to capture patterns that are not covered by a specific grammar. Experimental results are discussed for these alternatives.

Introduction

The analysis of large text corpora has become increasingly popular as a means of instantiating models of natural language. Methods of statistical pattern recognition have likewise received more attention as a means of parameter estimation and classification. Such approaches cater to the following:

1. Relative likelihoods of alternative interpretations, enabling the resolution of ambiguity on the basis of most likely interpretation.
2. The induction of models that cover a wide variety of constructions that appear in corpora of unrestricted text. One strategy is to start with a suitably unrestricted grammar and rely on parameter estimation to rule out the patterns that are not found in the training corpus. Only this strategy is considered here, though many alternative techniques (e.g. cluster analysis) also exist for automatically characterizing patterns in corpora.

The Baum-Welch algorithm (Baum 1972) (also called the Forward-Backward algorithm) is the prevalent method in speech recognition for training hidden Markov models (HMM's). It can also be used for training part-of-speech models from unlabelled text (Jelinek 1985). In the brief description that follows, notation is based on that of (Levinson et al. 1983). Grammars can be represented in terms of probabilistic transition networks. A hidden Markov model represents a transition network for a regular grammar, comprising states $\{c_1 ... c_n\}$ that correspond to part-of-speech categories. The probability of a transition between states c_i and c_j is labelled by the element $a[i,j]$ of the transition matrix A. Each state c_j has its own matrix of word probabilities. The probability that word k in the dictionary is generated by a transition to state c_j is given by the element $b[j,k]$ of the output matrix B. An element $I[j]$ of the initial matrix I gives the probability that a word sequence starts with category c_j. The model is considered to generate a sequence of words via a sequence of state transitions; a word being generated at each state. The part-of-speech category of each word must correspond to the state that generated it. Thus the word "of" could only be generated by the *preposition* state, whereas the word "do" could be generated by the *verb* state or the *noun* state (do – the musical note). We would expect the probability of the former to be much higher than the latter.

Given a training corpus, the training algorithm is responsible for assigning the parameters of the matrices (A, B, I). Once this has been done, the Viterbi algorithm (Viterbi 1967) can be used to find the most likely state sequence for any given sequence of words. The state sequence defines the corresponding part-of-speech categories for each word. Performance in excess of 95% correct category assignment has been reported using HMM-based taggers. (Jelinek 1985; Merialdo 1991; Kupiec 1992).

In the case of context-free grammars, (Baker 1979) describes the Inside/Outside algorithm for grammars in Chomsky normal form and (Kupiec 1992b) describes an algorithm that does not have this restriction. A context-free grammar can be represented by a recursive transition network. This involves the addition of states that model nonterminal categories. Initial probabilities in the matrix I serve to describe production

probabilities in this arrangement.

Unlabelled Training Corpora

This section provides an informal intuition regarding the training algorithms. The reader is referred to the references for formulaic details. Using a labelled training corpus the probabilities in (A, B, I) can be estimated directly from frequency counts. Such a corpus can be viewed as output from an *observable* markov source. The categories are known for each word, so the state sequences involved in generating the words the words of the text are explicit. The number of times a transition is made from state c_i to c_j can be determined by simply counting.

The state sequences in an unlabelled corpus are *hidden* from view. The following iterative strategy is then used to estimate the number of times a transition is taken from c_i to c_j. The strategy assumes some initial values have been assigned to (A, B, I). For convenience, the training corpus can be split up into sentences. Consider one such sentence:

$(w_0, w_1, w_2 ... w_Y)$

For any word w_x $(0 \leq x \leq Y)$ the probability of generating w_x at state c_i then making a transition to c_j while jointly generating the whole sentence can be determined from the sentence and the current values of (A, B, I). This probability is summed over all words in all sentences to find the expected number of transitions that were made from c_i to c_j. Expressing this as a fraction of the total expected number of transitions from c_i to all other states, a new estimate of $a[i, j]$ can be found. In fact the matrices (A, B, I) can all be re-estimated from their current values, and the training corpus. The process is repeated until the estimates converge, at which point they are guaranteed to provide a local maximum of the likelihood of generating the training corpus (Baum 1972).

The capability to train on unlabelled text corpora affords flexibility in several respects. The laborious effort required to manually label a necessarily large amount of text is completely avoided. Grammars can be trained directly on corpora from a desired application domain. To use a different category set requires only a new dictionary describing what possible categories each word can assume.

In (Kupiec 1992) the dictionary in a tagger for English was replaced by a French dictionary then the tagger was trained on French text excerpted from the Canadian Hansards (the proceedings of the Canadian parliament). The resulting part-of-speech tagger for French required minimal extra effort to implement, and was found to have error rates commensurate with the tagger for English.

The approach described here permits context-free grammar rules to be changed at will during grammar development. This not only avoids the initial expense of manually parsing the training corpus, but also the possibility that it needs to be re-parsed due to changes in nonterminal categories.

Another situation in which flexibility of this approach may prove useful is when a category set includes domain specific semantic categories.

Sparse Data Considerations

During training, some transitions may never occur (e.g. two successive commas or determiners). In such cases the transitions would be re-estimated as having zero probability. Subsequently if one of these sequences did appear in text that was being tagged, the grammar would fail to accept the sentence. To preserve robustness in this situation transition probabilities that fall to zero during re-estimation are assigned appropriately small values so that all sentences can still be tagged. The situation also applies to context-free grammars, when the training corpus does not contain an instance of a particular rule, and subsequently an instance of it appears during parsing. In this case the production probabilities in the matrix I are also maintained as small positive values.

Word Equivalence Classes

When probabilities $b[j, k]$ are estimated for every word k in the dictionary, the B matrix contains a large number of parameters. Even if a very large training corpus is used, the fact that a large percentage of words in the dictionary typically only occur a few times in a corpus means the corresponding parameters would not be reliably estimated (more than 40% of the words in a dictionary for the Brown corpus only occur once). Consequently an alternative approach has been taken (Kupiec 1989) in which words are represented in terms of *equivalence classes*. Words are partitioned according to the set categories which they can assume. The members of the sparsely populated equivalence class {*adjective, adverb, verb*} are illustrated below:

clean	**direct**	**even**	**free**
further	**loose**	**pretty**	**sheer**
slow	**steady**	**upstage**	

Pooling words into equivalence classes greatly reduces the number of parameters involved, and enables reliable estimation (the 50,000 different word types in the Brown corpus are covered by just 410 equivalence classes). Furthermore, adding new words to the dictionary can generally be done without re-training, as they are likely to be covered by existing equivalence classes.

The approximation assumes that words in a given

class have similar distributions over their category set. This is generally not the case in for any given corpus, however across different corpora distributions generally are different, so the representation provides a degree of parameter smoothing for infrequently occurring words.

In contrast, many common words in a corpus can be estimated reliably. The most frequent 100 words in one training corpus accounted for approximately 50% of the total number of word tokens, providing adequate data for reliable estimation. If these words are assigned their own unique equivalence classes, they can assume different distributions over their associated categories.

Incomplete Dictionary Coverage

A dictionary containing over 200,000 word types was found to cover 95-97% of words in various actual applications. To perform robustly on unrestricted text, a tagging program must be able to predict the categories of the remaining "unknown" words that aren't in the dictionary. Such words are restricted to the set of open class categories (nouns, verbs, etc.) as opposed to closed classes, which can be exhaustively enumerated in the dictionary. A fixed prior probability could be used to associate an unknown word with a given open class category, however more accurate prediction can be achieved by making use of word suffix information. In languages such as English and French, inflectional and derivational suffixes provide useful clues to a word's part-of-speech category. To accommodate suffix-based prediction, conditional probabilities are pre-computed for various suffixes (130 were used for English). During the training phase a single equivalence class is used to represent unknown words. Suffix-based probabilities are then employed by the Viterbi algorithm when an unknown word is encountered. The effectiveness of the method is illustrated in the tagged nonsense passage shown in Figure 1. The passage is excerpted from (Weaver 1979) (Copyright 1979 by the National Council of Teachers of English. Reprinted with permission).

In the passage, unknown words can assume between five and ten open class categories depending on their suffix. The suffix probabilities can be calculated using untagged text; examples are shown below for the probability of the suffix *-ic*, for the equivalence classes $\{noun\}$ and $\{adjective\}$:

$$P(-ic \mid \{noun\}) = 0.0027$$
$$P(-ic \mid \{adjective\}) = 0.0988$$

The passage contains five definite tagging errors, which are indicated by tags marked each side with asterisks. Seventeen errors result if only a fixed prior is used. Three of the five errors are due to the assignment of "corandic" and "borigen" as adjectives instead of nouns. Their preferable assignment as nouns is indicated by their subsequent appearance after a determiner. This aspect of local word recurrence can be profitably modelled with a dynamic word cache and used to improve the prediction of unknown words (Kuhn & De Mori 1990).

Alternative Grammar Structures

Here, an *unconstrained* grammar refers to a grammar which has a uniform pattern of conditioning (and which can be constructed automatically). A *constrained* grammar refers to one which has been built to expressly reflect linguistic structure, and consequently has non-uniform conditioning. In addition, an *augmented* grammar is defined as an unconstrained grammar in which connectivity has either been deleted, or to which extra structure has been added to provide selective higher-order conditioning (perhaps based on linguistic considerations or as a result of analyzing errors made by the grammar).

Unconstrained regular grammars work well for training part-of-speech taggers. The results quoted later for the Brown corpus are for a first-order grammar. Second-order grammars are also commonly used (e.g. Meteer et al. 1991; Merialdo 1991).

We have experimented with unconstrained context-free grammars to investigate whether any linguistically reasonable dominance structure can be automatically inferred from an unlabelled corpus. A "headed" grammar was chosen for the experiment. Nonterminals C_i exist for every part-of-speech category i (a simple set containing $N = 9$ categories was used). A nonterminal rule C_i must immediately dominate a terminal c_i of the same category (thus the number of rule applications is limited to the size of the sentence). Each nonterminal C_i may also dominate a nonterminal to the left or right of the terminal. Thus the phrase " The big cat" may be inferred as either a determiner, adjective or noun phrase, and having a parse tree that indicates the dominance structure. The following schema illustrates the rules, in regular expression form:

$$C_i \Longrightarrow (C_x)\ c_i\ (C_y)$$
$$1 \leq i, x, y \leq N$$

In the above, brackets denote optional inclusion. Elements in the transition and initial matrices (A, I) were initialized to be equally likely. The elements of the output matrix B were assigned from a trained text tagger and thus reasonably accurate at the outset. A corpus of approximately 400 sentences was used for training. Initial results were somewhat encouraging. Nouns (as opposed to determiners or adjectives) dominated noun phrases, and verbs often dominated whole sentences. Prepositions and conjunctions however did not make satisfactory attachments. The context was then extended to two optional nonterminals to the left

```
Key to Tags:
------------
v3sg:      verb 3rd singular        npl:  plural noun
v:         uninflected verb         n:    noun
det:       determiner               prel: relative pronoun
pnom:      nominal pronoun          pobl: oblique pronoun
prespart:  present participle       npr:  proper noun
pastpart:  past participle          adj:  adjective

Corandic is  an  emurient grof with many  fribs ; it   granks from corite ,
n        is  det adj      n    prep adj   npl     pnom v3sg   prep n

an  olg which cargs like lange . Corite grinkles several other tarances ,
det n   prel  v3sg  prep n       n      v3sg     adj     adj   npl

which  garkers  excarp by   glarking the corite and  starping it   in
prel   *v3sg**  **n*** prep prespart det n      conj prespart pobl prep

tranker - clarped storbs . The tarances starp a   chark which is
n         adj     npl      det npl      v     det n     prel  is

exparged with worters , branking a   slorp . This slorp is  garped   through
pastpart prep npl       prespart det n       det  n     is  pastpart prep

several other corusces , finally frasting a   pragety , blickant crankle :
adj     adj   npl        adv     prespart det adj       adj      n

coranda . Coranda is  a   cargurt , grinkling corandic and  borigen .
npr       npr     is  det n         prespart  **adj*** conj **adj**

The corandic is  nacerated from the borigen by   means of   loracity .
det n        is  pastpart  prep det n       prep npl   prep n

Thus garkers finally thrap a   glick , bracht , glupous grapant , corandic ,
adv  npl     adv     v     det adj     adj      adj     n         **adj***

which  granks in   many  starps .
prel   v3sg   prep adj   npl
```

Figure 1: Predicting Unknown Words

or right of the terminal. No satisfactory result was obtained despite alternative initialization strategies for the A and I matrices, and different grammar structures of the same complexity. Many of the resulting parse trees contained systematic left or right branching. It appears that the grammar requires more constraint to obtain useful results. An alternative approach is described by (Pereira and Schabes 1992), in which information which constrains the estimation is included in the training corpus. They report successful results by training an unconstrained context-free grammar using a partially bracketed corpus in which constituent boundaries have been manually assigned.

Stochastic grammars enable the probability of alternative parses to be used to rank the most likely ones. Thus they cater for "looser" grammars having higher coverage, and a correspondingly higher ambiguity. Efforts using simple constrained context-free grammars have been successful. Results have been encouraging even when using an equivalence class representation for most of the dictionary.

In (Kupiec 1989) a first-order regular grammar used for a tagger was augmented by including state chains that corrected some obvious errors that were being made because of insufficient context (e.g. dependencies between a past participle separated from a preceding auxiliary verb by one or more adverbs). The "headed" grammar mentioned earlier would likely benefit from being augmented. It may also be possible to train a constrained grammar, relax the constraints, then re-

train the grammar to obtain increased coverage while preserving linguistic well-formedness.

Training with Different Corpora

This section relates experiences with regard to training part-of-speech taggers on different corpora. A tagger was trained on approximately one million words of electronic mail messages (concerning the design of a programming language). It was subsequently used to tag a technical article concerning the drilling of deep wells. The most common error involved the mistagging of the word "well" as an adverb or adjective instead of a noun. Upon investigation, the informal nature of communication in electronic mail used the word "well" almost completely as an adverb. Using a tagger trained on text from *Grolier's American Academic Encyclopedia* resulted in fewer errors of this kind.

In turn, a tagger trained from Grolier's encyclopedia was used to tag excerpts from the electronic mail corpus. A commonly occurring error was the mistagging of "I" as a proper noun instead of a pronoun. Inspection showed that the training text from the encyclopedia was written almost exclusively in an impersonal style; the word "I" appearing often in phrases such as "King James I". Training a tagger using the encyclopedia was very appropriate for later use with it, but not for other text in which word usage was different.

The case conventions used for words also vary between corpora and affect the correctness of the tagging. For example, consider the words "do" and "van" in the names "Edson Arantes do Nascimento" and "Rembrandt van Rijn". In the Brown corpus "do" and "van" would be printed with an initial capital letter, facilitating their interpretation as proper nouns. In Grolier's (and other corpora) they often remain in lower case, leading to incorrect verb and noun assignments. Proper noun categories for these words may not be included in a dictionary at all; likewise other words that can be interpreted as proper nouns (e.g. surnames such as Baker, Mills, Rice etc.). It is thus advantageous to account for the dictionary coverage of proper nouns and case conventions when deciding upon a strategy to identify proper nouns for specific applications.

Another application involved the use of a tagger to tag a small corpus consisting entirely of questions (which occur rarely in the training corpora). The tagger performed poorly in this situation (an 11% error rate). Half of the errors involved questions containing the verb "do" (e.g. as in "When did World War II start ?"). The final word was tagged as a noun rather than a verb, as the appropriate context for disambiguation was lost by the time the end of the question was reached.

These examples illustrate how category usage can be corpus dependent. They also suggest that rather than attempting to train a tagger using text from the widest variety of corpora available, it may instead be worth trying to adapt a tagger locally to the text with which it is being used.

Evaluation

A manually labelled corpus enables the performance of the algorithms to be assessed. Currently, only the part-of-speech tagger has received a detailed evaluation (Kupiec 1992). A tagger was trained using unlabelled text from the Brown corpus with a dictionary built from the corpus (therefore no unknown words are present). The performance of the tagger can be ascertained by comparing its output with the manually assigned categories. Results are shown in Table 1.

The top two rows in Table 1 are results for training on untagged text and testing against the tagged text, for two different samples from the Brown corpus. The bottom two rows show results when the tagger was trained with smaller samples from a different untagged corpus containing material from a humor columnist. The latter is composed mainly of informal direct speech. A portion of an error matrix is shown in Table 2 (for the top row of Table 1), and the principal tagging errors are indicated in boldface. The conjunctions referred to in the Table are subordinate conjunctions. It is the authors' opinion that the percentage correct quoted for ambiguous words is a better indicator of performance than the percentage of total tokens which is typically reported in the literature. Unambiguous words in the dictionary are by definition correctly tagged. Counting punctuation marks as correct tokens is likewise undesirable from an evaluation standpoint.

The most frequent error is the mistagging of nouns as adjectives. This is due to both the variability in their order in noun phrases, and the fact that semantic considerations are often required for disambiguation. In practice, the effect of the various errors depends largely on the tagger application. For example when using the tagger to delineate noun phrases in text, the mistagging of nouns as adjectives is not particularly serious. The choice of category set influences the error rate. For instance in the Penn Treebank category set (Santorini 1990) the *qualifier* category is subsumed by the *adverb* category and subordinate conjunctions are assigned the same tag as prepositions. The error matrix shows that at least 2,449 fewer errors would be made by the tagger if such distinctions were dropped.

Conclusions

The paper has described experiences using hidden Markov methods to train stochastic grammars for part-of-speech tagging and context-free parsing. Results in-

Nr. Words in Training Sample	Nr. Words in Test Sample		Nr. Ambiguous Words in Test Sample	
	Total	Correct	Total	Correct
442,151	443,246	424,361 (95.7%)	159,419	140,534 (88.1%)
443,246	442,151	423,097 (95.7%)	163,212	144,158 (88.3%)
118,906	443,246	421,016 (95.0%)	159,419	137,233 (86.1%)
66,122	443,246	418,223 (94.4%)	159,419	134,440 (84.3%)

Table 1: Performance of the Brown Corpus Tagger

Correct Tag	Incorrect Tags Assigned						
	Noun Sg.	Adj.	Adv.	Qual.	Part.	Prep.	Conj.
Noun Singular	-	**1,365**	65	5	0	4	0
Adjective	317	-	**658**	15	1	6	0
Adverb	29	175	-	**704**	32	**417**	182
Qualifier	9	22	**514**	-	0	1	**552**
Particle	0	2	9	0	-	392	0
Preposition	3	18	313	0	**507**	-	**624**
Conjunction (Sub.)	3	8	132	9	0	**607**	-

Table 2: Error Matrix

dicate that part-of-speech tagging can be done with high accuracy and flexibility. Attempts to train a unconstrained context-free grammar were not successful, however more conventional grammars can be trained from unlabelled text and used for parsing.

References

Baker, J.K. 1979. Trainable Grammars for Speech Recognition. Speech Communication Papers for the 97th Meeting of the Acoustical Society of America (D.H. Klatt & J.J. Wolf, eds): 547-550.

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

Grolier. *American Academic Encyclopedia.* Grolier Electronic Publishing, Danbury, Connecticut.

Jelinek, F. 1985. Markov Source Modeling of Text Generation. Impact of Processing Techniques on Communication (J.K. Skwirzinski, ed), Nijhoff, Dordrecht.

Kuhn, R. & De Mori, R. 1990. A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12: 570-583.

Kupiec, J.M. 1992. Robust Part-of-Speech Tagging Using a Hidden Markov Model. To appear in: *Computer Speech and Language.*

Kupiec, J.M. 1992b. Hidden Markov Estimation for Unrestricted Stochastic Context-Free Grammars. Proceedings of ICASSP-92, San Francisco, CA.

Kupiec, J.M. 1989. Augmenting a Hidden Markov Model for Phrase-Dependent Word Tagging. Proceedings of the DARPA Speech and Natural Language Workshop, 92-98, Cape Cod, MA. Morgan Kaufmann.

Merialdo, B. 1991. Tagging Text with a Probabilistic Model. Proceedings of ICASSP-91, 809-812, Toronto, Canada.

Meteer, M., Schwartz, R., & Weischedel, R. 1991. POST: Using Probabilities in Language Processing. Proceedings of IJCAI-91, 960-965, Sydney, Australia.

Pereira, F. & Schabes, Y. 1992. Inside-Outside Reestimation from Partially Bracketed Corpora. Proceedings of the DARPA Speech and Natural Language Workshop, Arden House. February.

Santorini, B. 1990. Part-of-Speech Tagging Guidelines For The Penn Treebank Project, Technical Report, MS-CIS-90-47. University of Pennsylvania, PA 19104.

Levinson, S.E., Rabiner, L.R. & Sondhi, M.M. 1983. An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition. *Bell System Technical Journal*, 62: 1035-1074.

Viterbi, A.J. 1967. Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm. *IEEE Transactions on Information Theory*, IT-13: 260-269.

Weaver, C. 1979. *Grammar for Teachers: Perspectives and Definitions.* (p.25). NCTE.