Tagging an Unfamiliar Text With Minimal Human Supervision

Eric Brill and Mitch Marcus *
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, Pennsylvania 19104
{brill,mitch}@unagi.cis.upenn.edu

Abstract

In this paper, we will discuss a method for assigning part of speech tags to words in an unannotated text corpus whose structure is completely unknown, with a little bit of help from an informant. Starting from scratch, automated and semiautomated methods are employed to build a part of speech tagger for the text. There are three steps to building the tagger: uncovering a set of part of speech tags, building a lexicon which indicates for each word its most likely tag, and learning rules to both correct mistakes in the learned lexicon and discover where contextual information can repair tagging mistakes. The long term goal of this work is to create a system which would enable somebody to take a large text in a language he does not know, and with only a few hours of help from a speaker of the language, accurately annotate the text with part of speech information.

Introduction

Much of language learning involves the discovery and classification of linguistic entities, and learning the relationships that hold between objects of different classes. We are currently developing a set of automata that are able to learn some of the classificatory aspects of human language with little or no human guidance. In particular, work is being carried out on a set of computer programs that take a large corpus of text as input and from observing regularities in the corpus, these programs are able to learn information about morphology, word classes, and phrase structure.

The main tool used in this work to deduce linguistic information is distributional analysis. We adopt many ideas originally proposed by Zellig Harris [Harris 1951; Harris 1962]. Harris proposed a number of procedures that use the distributional behavior of relevant linguistic entities to uncover structural information about languages. The language learning procedures we use, although operating on different structural levels, all share one thing in common. They all learn structure using the tool of distributional analysis where the distributional behavior of an element is estimated from its behavior in a large corpus. Distributional analysis takes place over local and astructural environments. This means that environments such as *seven words to the right*, *subject of sentence* and *leftmost daughter of the phrasal head* are ruled out. Disallowing nonlocal environments constrains the set of possible environments that need to be considered when carrying out a distributional analysis. Disallowing structural environments also greatly constrains the set of possible environments, as well as ensuring that information about environments can readily and reliably be extracted from any large text corpus.

Figure 1 lays out the general framework under which this research is being carried out. The system begins in a language-naive start state. From the start state it is given a corpus of text as input and arrives at an end state, which is a level of language maturity allowing for morphological analysis, part of speech tagging and phrase structure analysis. For a description of the morphology and word class modules, see [Brill 1992b]. Phrase structure learning is described in [Brill and Marcus 1992].

In this paper, we will describe one module of the language learner in detail. This module annotates text with part of speech labels. Prior to receiving the unannotated text as input, no information about the lexicon or the syntax is known. Tagging information is then extracted from the corpus using distributional analysis. In its current incarnation the module allows us to take a text in a language we do not know, and with only a few hours of help from an informant familiar with the language of the text, we can build a tagger which is able to tag the text with fairly high accuracy. Note that we cannot even prespecify a tag set for the unfamiliar corpus. In a constrained corpus of computer manuals, a semantic tag such as *Computer Name* might be appropriate, whereas such a tag would not be appropriate in a less constrained corpus such as the Brown Corpus[Francis and Kučera 1982].

*This work was supported by DARPA and AFOSR jointly under grant No. AFOSR-90-0066, and by ARO grant No. DAAL 03-89-C0031 PRI.

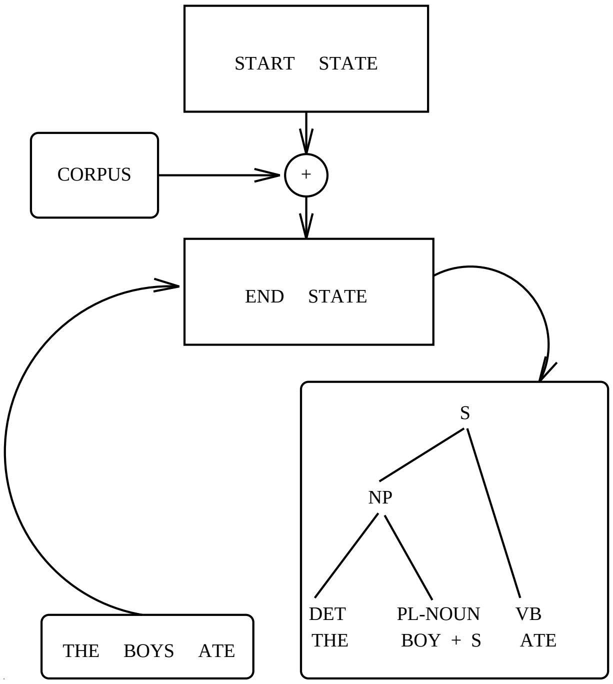

Figure 1: General Framework

Tagging Unfamiliar Text

This work was motivated by a real problem. We have access to a large number of text corpora in different languages, and we wished to apply our language learning techniques to these corpora. However, some of the learning modules need tagged text. None of the foreign language corpora are tagged. Not only were the corpora in foreign languages, but even the general content of many of the corpora was unknown to us. We wished to design a system to first find appropriate part of speech tags and then tag the text[1]. A completely automatic system would have been ideal, but this being out of reach we settled for a system that requires minimal supervision from a speaker of the language being processed.

There are a number of stages in tagging each unfamiliar corpus. First, a set of part of speech tags must be found by observing the distributional behavior of words in the text. Second, the most probable part of speech for all words which occur with sufficient frequency is determined. For instance, although *can* can have a number of different parts of speech in English, we need only determine at this point that *modal* is the most likely part of speech for *can*. A dictionary is built containing each word, and only the most likely part of speech for that word. Low frequency words are not entered into the dictionary at this point. This is because the method employed for discovering the most likely tag for a word relies upon distributional analysis, and there will not be sufficient information about the distributional behavior of low frequency words. After the dictionary is built, a small amount of text is tagged by assigning each word its most likely tag. It is known that if every word in a corpus is tagged with its most probable tag, accuracy of about 90% can be obtained (see [Brill 1992a]). Since the dictionary will not be completely accurate, our accuracy will be less than 90%. This small tagged text is then hand-corrected by our informant[2]. Rules are then learned automatically to correct errors in the dictionary and to correct cases where contextual cues indicate that a word is tagged incorrectly. The final step involves extracting suffix information on the words in the dictionary to allow us to assign a part of speech to words that occur infrequently. Each of these four steps will now be described in detail. As a pilot experiment, the steps were run on the Brown Corpus, a corpus of about 1.1 million words containing text from various different genres of written English.

Finding a Set of Part of Speech Tags

A number of proposals have been suggested for automatically finding word classes based upon the distributional similarity of words in a corpus [Harris 1951; Kiss 1973]. These methods work by defining a measure

[1] Many taggers have been built that achieve high accuracy (95-97% of words tagged with their appropriate part of speech)[Brill 1992a; Church 1988; Cutting et al. 1992; Derose 1988], but these taggers need to be trained on large amounts of tagged text and/or large dictionaries.

[2] The text that needed to be corrected had fewer than 8,000 words in our experiment.

of distributional similarity, and then clustering words into classes. While Kiss gave some evidence that this approach can be used effectively to class words, it has recently been shown that the approach is fairly robust and can be applied to large corpora with no human intervention [Brill et al. 1990; Brown et al. 1990; Brill 1991]. However, the word classes that result from such clustering are not error-free.

Instead of obtaining classes automatically by clustering all words in a particular corpus, we propose using these clustering methods to semi-automatically find word classes. In our current approach, only the 300 most frequently occurring words in the corpus are considered. The point of this clustering is not to find a class for every word, but only to try to elicit the salient word classes in the corpus. For each of these 300 words, we estimate from the corpus the probabilities of all other words in the corpus immediately preceding or following. If a word pair is seen fewer than three times, we consider its probability to be zero. Of all word bigrams in the Brown Corpus, 85.5% occur with frequency less than three. We have found that ignoring low frequency bigrams, while greatly reducing the computation time, does not affect the accuracy of word class formation. The vectors of adjacent word probabilities for each pair of the 300 words are then compared.

The divergence of the probability vectors is computed. Let P_1 and P_2 be two probability distributions over environments. The relative entropy between P_1 and P_2 is:

$$D(P_1||P_2) = \sum_{x \in Environments} P_1(x) * log \frac{P_1(x)}{P_2(x)}$$

Relative entropy $D(P_1||P_2)$ is a measure of the amount of extra information beyond P_2 needed to describe P_1. The *divergence* between P_1 and P_2 is defined as $D(P_1||P_2) + D(P_2||P_1)$, and is a measure of how difficult it is to distinguish between the two distributions. Below are the thirty word pairs from the 300 most frequently occurring words in the Brown Corpus deemed distributionally most similar according to the divergence of their distributional probability vectors.

HE SHE	COULD CAN
WE THEY	BUT ALTHOUGH
GIVE MAKE	WHILE ALTHOUGH
ME HIM	KIND NUMBER
IF WHEN	FIND TAKE
GET TAKE	ALTHOUGH SINCE
FIND MAKE	GET MAKE
THEM HIM	WHEN ALTHOUGH
IF THOUGH	MADE FOUND

MAKE TAKE	MEN CHILDREN
GIVE TAKE	MUST SHOULD
MEN PEOPLE	US THEM
FACE HEAD	CAME WENT
GET FIND	GIVE GET
SENSE KIND	TIME DAY
COULD WOULD	MIGHT MAY

We can use the distributional similarity scores to build a similarity tree for all 300 words. Begin with every word in its own class. Take the two classes that are most similar and merge them. Continue this until all words are merged into one class. Different areas of the tree will correspond to different word classes. This tree can be used to choose a tag set for the corpus. The informant is shown the tree and asked to find and name meaningful sections of the tree. Using the similarity tree can help uncover both syntactic and semantic word classes.

Finding the Most Likely Tag for Each Word

Let us now assume that we have settled upon a tag set[3]. The next step is to determine, for each word of sufficient frequency, the word's most likely tag. We carry this step out only on words that occur with sufficient frequency to allow for a reasonable approximation of the word's distributional behavior.

We first ask the informant to indicate which of the classes she can easily and quickly enumerate all members of. From the Penn Treebank, some such classes are TO (only applies to the word *to*) and punctuation tags. For the classes whose members are not easily enumerated, the informant is asked to choose a small number of words that she considers to be good exemplars for that class of words. These exemplars can be taken from the similarity tree discovered in the previous step. In the experiment we ran, the number of exemplars chosen ranged from 4 to 9. The *open* classes in our experiment, classes whose membership was not considered easily defined, were: adjective, adverb, determiner, modal, noun, preposition and verb. From these exemplar lists, a *distributional fingerprint* is created for each word class: the probability distribution for words preceding and following any of the words in the word class exemplar list.

Once these fingerprints are formed, each word is assigned to the class whose fingerprint most closely matches its own. The measure used to judge fingerprint similarity is divergence, which is described above.

The accuracy of this method was tested as follows. The entire Brown Corpus was used to estimate dis-

[3] For this pilot experiment, we based our tag set upon the Penn Treebank tag set to make it easier to quantify our results.

tributional behavior. A word list was made from a 15,607 word sample of random sentences from the corpus which consisted of all words not covered by the very small class tags and which did not occur on any exemplar list. Words that occurred fewer than three times in the random sample were removed from the word list. Such words accounted for about 20% of the total tokens. Each word on the word list was then compared to each word class fingerprint, and was assigned to the class it most closely resembled. To allow us to compare our results to a gold standard, we compared our answers to those deemed correct according to the Penn Treebank version of the Brown Corpus, with their tag set reduced to ours. The Penn Treebank has been tagged automatically and then manually corrected, to produce a very accurately tagged text. Fine-grained class distinctions were removed. For instance, the classes noun, proper noun, plural noun, plural proper noun and pronoun were all mapped down to the tag *noun*. The precise mapping is shown below[4].

Tag	*Tags Mapped Down To This Tag*
Modal	MD
Determiner	DT PDT WDT
Noun	NN NP NNS NPS PP
Verb	VBN VB VBD VBZ VBP VBG
Adjective	JJ JJR JJS
Adverb	RB RBR RBS WRB
Preposition	IN RP

When we attempt to assign each word to its most probable class, the accuracy by type is 66%. In other words, 66% of the words on the word list were assigned to the class which they were most frequently assigned to in the Penn Treebank tagging of the Brown Corpus. Of the errors, 56% are from classifying words which should be nouns as adjectives. If instead we consider tokens and not types, things look better. Of the total number of tokens that occur in the 15,607 word sample and appear on the word list, 76% are tagged with their most probable tag. If words used to make the distributional fingerprints, as well as punctuation and other *closed* class tokens, are included in the word list, 84% of the words on the word list, by token, are tagged with their most probable tag.

[4] MD = modal, DT = determiner, PDT = predeterminer, WDT = wh-determiner, NN = sing. noun, NP = sing. proper noun, NNS = plur. noun, NPS = plur. proper noun, PP = personal pronoun, VBN = past part. verb, VB = base form verb, VBD = past tense verb, VBZ = 3rd pers. sing. present verb, VBP = non-3rd pers. sing. present verb, VBG = gerund or pres. part. verb, JJ = adjective, JJR = comparative adj., JJS = superlative adj., RB = adverb, RBR = comparative adv., RBS = superlative adv., WRB = wh-adverb, IN = preposition, RP = particle

Now let us consider actual tagging accuracy, how accurately this method tags a word with the correct tag for its context. While *modal* is the most probable tag for *can*, in the sentence *kick the can*, *can* should be tagged as a noun. Once again considering only words that occur with frequency greater or equal to three, if each of these words is tagged everywhere in the small test corpus with what the fingerprint comparison has found to be its most probable tag, an accuracy of 84% is obtained. We would now like to consider improving upon this performance.

Learning Rules to Improve Accuracy

In [Brill 1992a], we describe a rule-based part of speech tagger. This tagger works by first tagging every word with its most probable part of speech and then automatically learning a small set of contextual rules to improve tagging performance. The tagger has a small set of rule templates. Templates are of the form:

- If a word is tagged **a** and it is in context **C**, then change that tag to **b**, or
- If a word is tagged **a** and it has lexical property **P**, then change that tag to **b**, or
- If a word is tagged **a** and a word in region **R** has lexical property **P**, then change that tag to **b**.

At each stage of learning, a small sample text is tagged. This tagged text is then compared to the correct tagging, and the rule template instantiation whose application would result in the greatest error reduction in the sample text is added to the tagger's list of rules. The rule is applied to improve the tagging accuracy of the sample text, and rule learning continues on the corrected sample text. Rule discovery is completely automated. Some example rules learned by the tagger using the original Brown Corpus tag set were:

(1) Change a tag from infinitival-to to preposition if the next word is tagged as a determiner.
(2) Change a tag from verb to noun if one of the two previous words is tagged as a determiner.
(3) Change a tag from noun to verb if the previous word is tagged as infinitival-to.
(4) Change a tag from subject pronoun to object pronoun if the next word is tagged as a period.

With fewer than 100 such rules, performance comparable to stochastic taggers was obtained.

We used the rule-based tagger to improve upon the system's accuracy at tagging an unfamiliar text. In our experiment, we used about 10,000 words of text to train the tagger, and a separate 5,600 to test. We tagged the 10,000 words by mapping the proper tag indicated in the Penn Treebank to its reduced tag. In real use, the informant would have to tag the 10,000 word sample. However, we are only attempting to tag the higher frequency words, and so at this point about 20% of the 10,000 words are tagged as *unknown*, and need not be tagged by the informant. To tag the rest of the text, the informant need not start from scratch. This text can first be tagged by assigning every word its most probable tag from the dictionary built in the previous phase.

The rule-based tagger learned 117 rules. The tagger was then tested on a separate 5,600 word sample. Initially, tagging accuracy was 84%, ignoring all words labeled as *unknown*. This level of accuracy was obtained by tagging every word with what was believed to be its most probable tag, regardless of context. After the rules were applied to the testing corpus, the accuracy was increased to 94%. Below are a few of the rules that were learned. Rules are of two types: correcting *most likely tag* mistakes for words, and using contextual information to tag more accurately.

(1) Change a tag from adjective to noun if one of the previous two words is tagged as a determiner.
(2) Change a tag from modal to verb if the word is *had*.
(3) Change a tag from adverb to preposition if the word is *as*.
(4) Change a tag from noanswer to verb if one of the previous two words is tagged as a noun.

Rule (1) helps correct the most prevalent mistake of the system: tagging nouns as adjectives. Rules (2-3) correct for mistakes made in automatically learning the most probable tag for a word. During the discovery of most probable tags by comparing each word to distributional fingerprints, some words are assigned the tag *noanswer*. Distributional fingerprints only make use of bigrams that occur at least three times in the corpus. It is possible for a word to occur three or more times in the corpus (our criterion for classifying a word), but no bigrams containing that word have sufficiently high frequency. If this is the case, words have no distributional fingerprint, and are incorrectly assigned to the class *noanswer*. Rule (4) is an attempt to properly tag some such words, by guessing that a word is likely a verb if it follows a noun.

Figure 2 shows a graph of the improvement in tagging performance as rules were applied to the test corpus.

Tagging Low Frequency Words

The final phase involves tagging low frequency words. Low frequency words present a difficulty, since there is not enough distributional information to get reliable distributional cues about their parts of speech. Part of speech taggers have been built that are fairly accurate at tagging words not seen in the training corpus. They do so by using morphological information. Since we do not want any language-specific information hard-coded in the learner, we cannot prespecify a set of affixes that indicate a particular part of speech. Instead, we use the 10,000 word accurately tagged text to compute for all three letter strings the most likely tag for a word ending in those three letters. Testing on the words marked *unknown* in the test corpus, tagging every word according to its suffix results in a 79.5% accuracy. Words for which no suffix information is available are tagged as nouns by default. The default tag can be determined by asking the informant the part of speech of a small number of low frequency words, and then determining the most frequent part of speech among these words.

The *unknown* category accounted for 22% of the words in the test corpus. Therefore, the accuracy in tagging the entire test corpus is (probability of low frequency word * accuracy in tagging low frequency word) + (probability of high frequency word * accuracy in tagging high frequency word):

$$.22 * 79.5\% + .78 * 93.9\% = 90.7\%$$

Given the level of accuracy suggested by this experiment, we see three primary uses for such a system. First, it provides a way to quickly tag an unfamiliar text when a very high accuracy rate is not needed. Second, it provides a way of bootstrapping the high accuracy tagging of a corpus by allowing a person to correct the inaccuracies in a corpus tagged using the above procedure, rather than tagging from scratch. Last, we hope that this work is a step toward fully automating the process of tagging unfamiliar text.

Improvements

Our pilot experiment shows that the approach outlined above holds promise as a way to accurately tag an unfamiliar corpus of text with only minimal help from an informant. We are currently pursuing a number of approaches toward improving the system. For some uses, the coarse tag set used in our experiment may not be sufficient. One possible approach is to first tag all words with a tag from the coarse tag set and then decide how to break the coarse tag into a number of more restrictive tags. Once this is done, a distributional fingerprint can be built for each of the restrictive tags, and words can be assigned to their proper class. Also, the rule-based tagger may be able to learn rules to improve the accuracy of tagging low frequency words that are tagged according to their last three letters.

Rules For Correction

The system outlined above uses both rule-based and statistical techniques. We have combined the two methods by first using statistical techniques to extract information from a corpus, and then using a program to automatically learn rules that can patch up mistakes

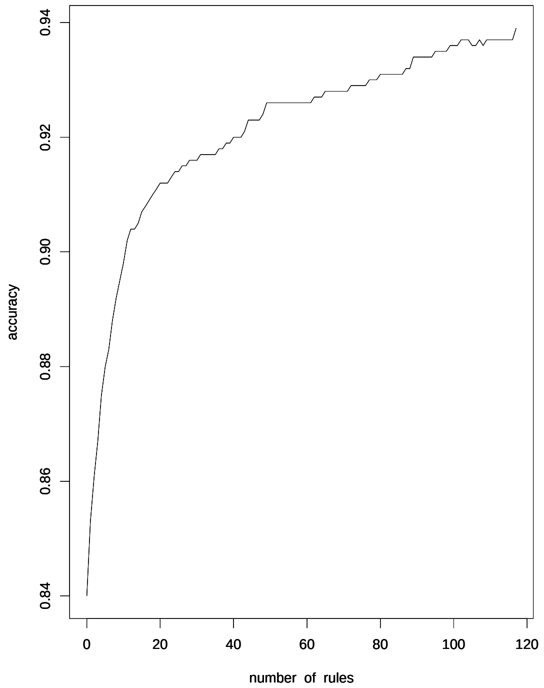

Figure 2: Improving Accuracy

made by the statistical techniques. Learning correcting rules can be an effective approach when the distribution of errors somewhat follows Zipf's Law [Zipf 1949]. If Zipf's Law is obeyed, then a small number of high probability error types will account for a large percentage of total error tokens. Such a distribution is amenable to the rule-based correction approach: the fact that there is a small number of high probability error types ensures that such errors can easily be uncovered in a small sample, and the fact that these errors account for a high percentage of total error tokens will ensure that remedying these errors will result in significant system performance improvement.

References

Brill, E.; Magerman, D.; Marcus, M.; and Santorini, B. 1990. Deducing Linguistic Structure from the Statistics of Large Corpora. *Proceedings of the DARPA Speech and Natural Language Workshop, June 1990*, pages 275–282.

Brill, E. 1991. Discovering the Lexical Features of a Language. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, CA. (Student Session).*

Brill, E. and Marcus, M. 1992. Automatically Acquiring Phrase Structure Using Distributional Analysis. *Darpa Workshop on Speech and Natural Language*, Harriman, N.Y.

Brill, E. 1992a. A Simple Rule-Based Part of Speech Tagger. *Proceedings of the Third Conference on Applied Natural Language Processing, ACL,* Trento, Italy. A longer version of this paper appears in the *Proceedings from the 1992 Darpa Workshop on Speech and Natural Language.*

Brill, E. 1992b. A Distributional Analysis Approach to Language Learning. Unpublished Dissertation Proposal, University of Pennsylvania. Philadelphia, Pa.

Brown, P.; Della Pietra, V.; Della Pietra, S.; and Mercer, R. 1990. Class-Based n-gram Models of Natural Language. In *Proceedings of the IBM Natural Language ITL, pp. 283-298,* Paris, France.

Church, K. 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. *Proceedings of the Second Conference on Applied Natural Language Processing, ACL*, 136-143.

Cutting, D.; Kupiec, J.; Pederson, J.; and Sibun, P. 1992. A Practical Part-of-Speech Tagger. *Proceedings of the Third Conference on Applied Natural Language Processing, ACL.*

DeRose, S.J. 1988. Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics* 14: 31-39.

Francis, W. and Kučera, H. 1982. *Frequency Analysis of English Usage. Lexicon and Grammar* (Boston: Houghton Mifflin).

Harris, Z. 1951. *Structural Linguistics.* Chicago: University of Chicago Press.

Harris, Z. 1962. *String Analysis of Language Structure.* The Hague: Mouton & Co.

Kiss, G. 1973. Grammatical Word Classes: A Learning Process and its Simulation. *Psychology of Learning and Motivation,* 7, 1-41.

Zipf, G. 1949. *Human Behavior and the Principle of Least Effort.* New York: Hafner Pub. Co.