Generating a grammar for statistical training

R.A.Sharman, IBM(UK) Science Centre,Winchester SHARMAN at VENTA
F.Jelinek, T.J.Watson Research,Yorktown Heights JELINEK at YKTVMV
R.Mercer, T.J.Watson Research,Yorktown Heights MERCER at YKTVMV

The problem of parsing Natural Language

Parsing sentences of a Natural Language(NL) is an essential requirement for a variety of NL applications, and has been extensively studied. In particular, the sort of tasks which it would be desirable to do, include the ability to tag each word with its part-of-speech; to delineate with brackets, and label with a category name, each syntactic phrase; and to be able to adapt to different types of source material. Despite some 30 years of active research performing these tasks with a high degree of accuracy on unrestricted text is still an unsolved problem.

The conventional approach is a grammar, usually created manually by the encoding of some linguistic intuitions in some notation. Many grammars have a substantial *context-free grammar*(CFG) component, or are equivalent in computational power to CFG's. A standard parsing algorithm can then be used to obtain analyses of any given sentence. A discussion of the relevant concepts of parsing CFG'S is given in Hopcroft and Ullman, 1979, where the CKY algorithm, the first of the *chart parsing* techniques, is described. A recent example of a grammar, of CFG power, is the Generalised Phrase Structure Grammar (GPSG) given by Gazdar, Klein, Pullum and Sag, 1985. Simple context-free grammars, and systems derived from them, have the consequence that practical grammars of full Natural Languages tend to be very large; algorithms for parsing grammars are computationally expensive; and hand-written grammars are often incomplete, and are usually highly ambiguous. Consequently this method alone is unlikely to solve the problem of parsing NL's.

An extension to context-free grammars which considers each rule to be associated with a *probability* is called a *probabilistic context-free grammar*, or P-CFG (see Wetherill, 1980 for a full discussion). Conceptually, the probability of a rule for a given non-terminal symbol is the likelihood with which the rule is applied, as opposed to other rules for rewriting the same non-terminal label. With the addition of this extra piece of information it is possible to choose the parse which is the most likely from among all the ambiguous parses. The "most likely" parse is considered to correspond to the "correct" parse. A method exists for training such grammars with respect to some corpus, the *Inside-Outside Algorithm* (Baker, 1979). It is described in Jelinek, 1985(b), and has been used by, for example, Fujisaki 1987. Adapting a model with reference to a *training corpus* should enable the model to be used with greater success on an unseen *test corpus*. The problems of large grammars, expensive algorithms, incomplete and ambiguous grammars, are essentially the same as for simple context-free grammars. Additionally, estimates must be made of the probabilities of the rules in the grammar. Since the true values for a natural language, such as English, are not known, the quality of the estimates made is rather important. Any arbitrary initial guess may be a long way from the true value, and will need a great deal of training to achieve a good estimate. If the process of training is slow, then it may not be possible to do enough training to significantly improve the estimates. If the grammar being trained is not a good grammar, then it may not be possible to get a good solution to the parsing problem despite a very great deal of training.

It is assumed here that no restriction on vocabulary or sentence construction can be made. There are a number of significant applications, such as speech recognition and speech synthesis, for which this is the only reasonable assumption (see Jelinek, 1985(a)). The search for a method to perform fast, accurate parsing of unrestricted natural language sentences may require other models. One way forward is to attempt to use a formulation of grammatical knowledge which uses information in a more compact way. A popular method is the use of so-called *unification grammars* (derived in turn from GPSG). One attempt to design a computable form for such a grammar is described

by Sharman, 1989. An alternative step is to attempt to consider a development of the ID/LP notation introduced in GPSG.

Probabilistic ID/LP grammars

The idea of separating simple context-free rules into two, orthogonal rule sets, immediate dominance(ID) rules, and linear precedence(LP) rules, gives a notation for writing grammars called ID/LP. This technique is used in GPSG with a number of other techniques to represent linguistic facts in a compact and perspicuous formalism. Some of the other aspects of GPSG, such as the use of features to represent information in non-terminal categories, the use of feature co-occurrence restrictions, the use of feature specification defaults, and the definition of feature passing conventions, are not considered here.

It is assumed in GPSG that dominance and precedence relations are independent. The independence of separate dominance rules does not seem problematical, but whether or not precedence rules are uniform across all phrase types does seem more contentious. As a result, ID/LP grammars tend to have a rather rich collection of ID rules, and a rather small collection of LP rules. This raises the interesting question of the possibility that ID and LP rules are not independent, but this possibility is not pursued here.

The notion of a grammatical relation, such as precedence or dominance, can be generalised to mean the propensity of a symbol to relate to another symbol. For example, a *noun phrase* has a propensity to contain a *noun*. Since it is clear that at least some noun-phrases do not contain nouns, this propensity will not be a certainty. However, the propensity of noun phrases to contain a noun will, presumably, be rather greater than, for example, the propensity of noun-phrases to (directly) contain a verb, or of other phrases to contain a noun. In other words, for any given phrase there is a probability distribution over the objects which can be its immediate constituents, which we can call the *dominance probability*. By a similar argument, there is also a probability distribution over the ordering of items in a phrase which we can call the *precedence probability*. Thus an ID/LP grammar which uses probabilities is a *probabilistic ID/LP grammar*, or P-ID/LP.

Computing the probability of a sentence

In order to use a P-ID/LP grammar analogously to a P-CFG we need to establish a way of computing the probability of a sentence, given the probability of ID and LP rules. The likelihood of the derivation of a sentence, W, from the topmost distinguished symbol, S, for an ID/LP grammar, can be determined from the independent likelihoods of each step in the derivation. Each step is the result of replacing some non-terminal symbol by some other terminal and non-terminal symbols, (the *dominance* relation), and the result of rearranging those derived symbols in the desired order (the *precedence* relation).

Thus, in the derivation of the sentence $S \stackrel{*}{\Rightarrow} W$ there is a sequence of individual steps, which may result in a derivation such as

$$S \Rightarrow \ldots \Rightarrow xAy \Rightarrow xBCy \Rightarrow \ldots \Rightarrow w_1 w_2 \ldots w_n = W$$

The immediate dominance rule $A \rightarrow B, C$ has been used to generate $xBCy$ from xAy at the intermediate step indicated. That is, the A was replaced by a B and a C without reference to their order. This rule is considered to have the conditional probability $P_D(B, C \mid A)$, or the dominance probability of B and C given A. Since this would have allowed both the rewriting BC, and CB, the precedence rule $B \prec C$ must also have been used. This rule is considered to have the probability $P_P(B \prec C)$, or the precedence probability of B before C. If the probability of the string xAy is $P(xAy)$, then the probability of the string $xBCy$ can be considered to be

$$P(xBCy) = P(xAy) \cdot P_D(B, C|A) \cdot P_P(B \prec C)$$

Thus, the derivation is treated as a stochastic process involving independent probability distributions governing the symbols which are produced in each step, and the order of the symbols. The result is the simple stochastic process conventionally modelled as production rule probabilities. The probability of a single derivation of W from S is denoted by $P(W|S, d_i)$, where d_i represents the i-th derivation. It is calculated from the product of all the individual steps in the derivation, and is clearly independent of the order of application of the individual steps in the derivation. The total probability of the word string W, or $P(W|S)$, is the sum of all the separate derivations from S to W, or $\sum P(W|S, d_i)$. The more useful quantity in parsing is the derivation for which the probability is the greatest, or the $\max_i P(W|S, d_i)$. This derivation is the most likely derivation, and thus defines the parse tree to be selected for the given sentence.

The general idea can be extended, in a straightforward way, not described here, for the case of unrestricted rules, where there are an unlimited number of symbols on the right hand side of a context-free rule. In this case the longer rules are modelled as successive applications of shorter rules. The advantage of this decomposition is that the size of the rule sets are to some extent governed by the size of the symbol set chosen, and do not involve the huge tail of low frequency rules typical of P-CFG grammars. For example, if there are n non-terminal symbols in the grammar, there can only be n^2 precedence relations between them, and an estimate of all of them can be made, avoiding the problem of unknown rules. Similarly the number of dominance rules is also restricted over the equivalent number of conventional phrase structure rules.

Determining P-ID/LP grammatical relations

The values of the probability of each dominance and precedence rule must be known so that parsing can take place according to the scheme described above. These values can be determined either by observation, or by training.

In order to determine the values by observation, a large corpus of pre-analysed sentences can be inspected, and the dominance and precedence frequencies can be determined by counting. From these frequencies it is simple to compute the dominance and precedence probabilities.

Alternatively, values of the dominance and precedence probabilities can be determined by assuming some arbitrary initial estimate, and training on a corpus, in a way similar to that used in the Inside-Outside algorithm. This involves computing the likelihood of the topmost label, S producing each observed sentence, W, and collecting counts of observed dominance and precedence relations, which are then used to re-estimate the probabilities of dominance and precedence. This re-estimation can be done many times, until satisfactory estimates of the true probabilities are obtained.

Since a suitable corpus of pre-parsed sentences was available the first method was used for simplicity. This is called *adapting* the grammar to the corpus, to distinguish it from *training* which is the technical term used for estimates derived from iterative re-estimation algorithms.

Creating a P-ID/LP parser

The probability of a sentence can be calculated in an entirely analogous way to that done for P-CFG's, by using a modified form of the CKY algorithm. The CKY table holds a place for the probability of every possible substring which the grammar produces for a given sentence. This is calculated from the bottom up, re-using already completed calculations where necessary. The computational complexity of this task is known to be related to the cube of the sentence length (see Hopcroft, 1979), which is at least a polynomial calculation, rather than an exponential one.

Techniques for thresholding the computation can be used to speed up the parser. Because there can be fewer relations to compute than there might be rules in a P-CFG the parser may be faster.

Using a P-ID/LP grammar to parse English

This section describes an experiment to determine if P-ID/LP parsing, in the manner described above, is a useful technique. The necessary grammatical relations were obtained from a collection of unrestricted sentences of English taken from the Associated Press (AP) newswire material, consisting of about one million words of text, with a vocabulary of about 50,000 words, The sentences were available in a pre-parsed form, called a *treebank*, with about 45,000 different phrase types marked by the manual parsers. A simple CFG could require as many rules as phrase types to represent the complexity shown in this data. The treebank was used to determine the values of dominance and precedence probabilities, and to test the output of the parser. For these purposes the original treebank was divided into two equal parts, so that a grammar derived from one part could be tested on the other. A modified CKY parser was used to parse the sentences with the P-ID/LP grammar, and to select the most likely parses. The resulting system has the capability to parse any sentence of English, although it is adapted specifically to the AP corpus.

The following steps were taken:

1. A restricted set of 16 non-terminal symbols were derived from the 64 actually used in the treebank.
2. A restricted set of 100 terminal symbols were derived from the 264 actually used in the treebank.
3. The treebank was divided into two parts, one for adapting the grammar, and one for testing.
4. The word list of the adapting data was extracted, with the unigram frequency of each word, and the tag attached to the word. This word list is used to create:
   a. A lexicon of word-to-tag correspondences, with the unigram probability of each entry. This is used to generate the probability of tag assignments to words when those words are found in a sentence to be parsed.
   b. A probability distribution over terminal tags. This is used to predict the tag of a word in a sentence, when that word does not appear in the lexicon.
5. Initial estimates of the dominance relations for non-terminal symbols were derived from the treebank.
6. Initial estimates of the precedence relations for non-terminal and terminal symbols were derived from the treebank.

A variant of the CKY parser for unrestricted rules was used to compute the probability of each sentence. The completed parse includes the most likely tag assigned to each word, and the most likely constituents hypothesised over the substrings of the sentence. The best parse can be displayed as a conventional parse tree.

The results

A test set of 42 sentences was chosen at random, subject only to the restriction that the sentence length was less than 30 words. The average sentence length was 20 (as opposed to 23 for the AP as a whole). These sentences were parsed, and the results compared with manual analyses of the same sentences. For the purposes of comparison, parses were divided into the following classes to show the accuracy of the parse, relative to the manual parse: exact match; close approximation; incorrect parse; failed parse. Some examples of these categories are given in the Appendix, to show the type of assessment which has been made.

The sentences were parsed in four ways: using a grammar which was unadapted(G1) or adapted(G2), and using a lexicon which was unadapted(L1) or adapted(L2), the adaptation being to the sample from which the test sentences were drawn. It would be expected that the system should do well on data which it was adapted to, and less well on data it was no adapted to. The resulting parses were manually inspected for errors.

The results for tagging are as follows:

TAGGING	G1,L1	G1,L2	G2,L1	G2,L2
correct tags	660	789	658	779
avg. correct tags/sent	15.7	18.8	15.6	18.5
% tags correct	80	96	80	95

The results for parsing are as follows:

PARSING	G1,L1	G1,L2	G2,L1	G2,L2
correct parses	8	10	12	18
similar parses	17	22	20	19
wrong parses	13	10	6	5
failed parses	4	0	4	0

The worst performance is the unadapted grammar and lexicon, and the best performance is the adapted grammar and lexicon. The adapted grammar with an unadapted lexicon, and the unadapted grammar with an adapted lexicon, are in between, and about as good as each other. On the basis of this data there is no reason to distinguish between these latter two cases.

Conclusions

Initial results indicate that tagging accuracy is quite good, but that parsing accuracy is less good. This should of course be compared to the performance of other parsers and grammars on material of similar complexity.

The performance of this system as a tagging tool is similar to a word tagging system using a Hidden Markov Model. Such a system would expect an error rate of no more than 5%, and perhaps as low as one or two percent. An exact comparison would have to take into account a consideration of the actual tags applied, the type of text used.

The performance of the system in parsing is not quite as successful as one would like, but there is some satisfaction to be obtained from the fact that correct, or nearly correct, parses account for 60% of the total even for the unadapted grammar, rising to 76% for the intermediate cases, and 88% for the fully adapted system. A large number of the causes of error can be seen to be related to semantic and pragmatic issues which the model does not, by definition, address. This holds out hope that the method may be capable of improvement by refining the precision of the relations modelled.

Some specific causes of imperfect parses were due to the choice of non-terminal symbols. The treebank symbols, while being perfectly adequate for manual parsing, are insufficiently precise for

an automatic system to distinguish between different categories of phrase. Either a better set of non-terminal symbols should be used, or features should be used to make existing categories capable of carrying finer distinctions.

Also, the precedence and dominance relations chosen are insufficiently precise to give good results. For example, the precedence relation is determined over all non-terminal symbols. It is possible that the order of symbols in some group of phrases is different than the order in another group, and that different precedence relations could express this.

It is planned to relax these restrictions in future work.

References

Baker, J.K. *Trainable Grammars for Speech Recognition*, in D.H.Klatt and J.J.Wolf (eds), *Speech communication papers for the 97th meeting of the Acoustical Society of America*, 1979

Fujisaki, T *A Probabilistic Modelling Methods for Language Parsing*, IBM research Report RC 13296, Yorktown 1987

Gazdar, G., E.Klein, G.K.Pullum and I.A.Sag *Generalised Phrase Structure Grammar*, Blackwell 1985

Hopcroft, J.E and J.D.Ullman *Introduction to Automata, Theory, Languages, and computation*, Addison-Wesley 1979

Jelinek, F. *The development of a Large Vocabulary Speech recognition system*, Proc. IEEE vol. 73, No. 11, 1985(a)

Jelinek, F. *Markov Source Modelling of Text Generation* Report of the Continuous Speech Recognition Group, IBM T.J.Watson Research Center, NY. 1985(b)

Sharman, R.A. *A Categorial Phrase Structure Grammar Development system*, IBM UKSC Technical Report 198, 1989(a)

Wetherill, C.S., *Probabilistic Languages: A Review and some Open Questions*, Computing Surveys, vol. 12, No. 4, 1980

Appendix - Examples of parse classification

The original sentence is shown (indicated by the prefix S:), with the manual parse(indicated by the prefix M:), and the automatic parse (indicated by the prefix A:). Part of speech tags have been removed for clarity. Each phrase is marked by a bracket pair which is labelled according to the type of phrase delimited. The labelling on the brackets should be self-explanatory, with N referring to a noun-phrase, V a verb-phrase, P a prepositional-phrase, and so on.

Exact match

The parse produced is an exact match of the hand parsed sentence, except for certain totally predictable typographical conventions.

```
S: But casino profits plummeted .
M: But  [N casino profits N]  [V plummeted V]  ._.
A: [S But [N casino profits N] [V plummeted V] ._. S]
```

Equivalent parse

The parse produced is a reasonable parse of the sentence, and could have equally well been produced by the manual analysis.

```
S: The city has about 900 firefighters .
M: [N The city N] [V has [N about 900 firefighters N] V]  ._.
A: [S [N The city N] [V has [P about [N 900 firefighters N] P] V] ._. S]]
```

Prepositional phrase attachment problem

The parse is essentially correct, but a prepositional phrase has been attached in the wrong place, or has the wrong scope.

```
S: The pancreas , in addition to making digestive enzymes , is
   the body organ that produces insulin , a hormone that
   controls the level of sugar in the blood .

M: [N The pancreas N] , [P in [N addition [P to [Tg making [N
   digestive enzymes N] Tg] P] N] P] , [V is [N the body
   organ [Fr that [V produces [N insulin , [N a hormone [Fr
   that [V controls [N the level [P of [N sugar N] P] [P in [N
   the blood N] P] N] V] Fr] N] N] V] Fr] N] V] .

A: [S [N The pancreas N] , [P in [N addition [P to [V making
   [N digestive enzymes N] V] P] N] P] , [V is [N the body
   organ [Fr that [V produces [N insulin N] V] Fr] N] V] , [N a
   hormone [Fr that [V controls [N the level [P of [N sugar [P
   in [N the blood N] P] N] P] N] V] Fr] N] .  S]
```

Coordination problems

The parse is essentially correct, but a coordinated phrase has coordinated item of the wrong scope.

```
S: The bacteria can be transmitted in water and from bodies of
   dead birds .

M: [N The bacteria N] [V can be transmitted [P [P& in [N water
   N] P&] and [P+ from [N bodies [P of [N dead birds N] P] N]
   P+] P] V] .

A: [S [N The bacteria N] [V can be transmitted [P in [N water
   [ and [P from [N bodies [P of [N dead birds N] P] N] P] ]
   N] P] V] .  S]
```

Problems with commas

The parse is essentially correct, but a wrong decision has been made over the interpretation of one or more commas.

```
S: At Karen 's Collectables , owner Karen Walker had a present
   for Kennedy , a sterling silver tie-tack in the image of
   John Kennedy .

M: [P At [N [G Karen 's G] Collectables N] P] , [N owner [N
   Karen Walker N] N] [V had [N a present [P for [N Kennedy N]
   P] , [N a sterling silver tie-tack [P in [N the image [P of
   [N John Kennedy N] P] N] P] N] N] V] .

A: [S [P At [N [G Karen 's G] Collectables N] P] , [N owner
   Karen Walker [ had [N a present [P for [N Kennedy N] P] N]
   ] N] ,_, [N a sterling silver tie-tack [P in [N the image
   [P of [N John Kennedy N] P] N] P] N] .  S]
```

Wrong Parses

The best parse found is not the right parse. The manual parse is considered to be the right parse.

```
S: Oil-state senators are trying to block an amendment to the "
   windfall-profits " tax that would cost the oil industry
   $22.5 billion over the next decade .

M: [N Oil-state senators N] [V are trying [Ti to block [N an
   amendment [P to [N the " windfall-profits " tax [Fr that [V
   would cost [N the oil industry N] [N $22.5 billion [P over
   [N the next decade N] P] N] V] Fr] N] P] N] Ti] V] .

A: [S [N Oil-state senators N] [V are trying [Ti to block [N an
   amendment N] to [N the N] Ti] V] " [Si windfall-profits " [N
   tax [Fr that [V would [ cost [N the oil industry [P $22.5
   [N billion [P over [N the next decade N] P] N] P] N] ] V]
   Fr] N] Si] . S]
```

It is worth noting that even in the case of the so-called wrong parse there is much to commend the attempt. However, it is the judgement here that the parse is not sufficiently accurate to place further judgements on, and so it has been classified as wrong.

Failed Parses

No parse is found. This usually occurs because the thresholding technique used to speed up the calculation by eliminating low probability parses has eliminated all remaining parses. It is often a result of incomplete lexical and grammatical knowledge.