A Probabilistic Parsing Method for Sentence Disambiguation

T. Fujisaki, F. Jelinek, J. Co‹ke, E. Black, T. Nishino+

IBM Thomas J. Watson Research Center
P.O. Box 704, Yorktown Heights, N.Y. 10598
+Tokyo Denki University

1. Introduction

Constructing a grammar which can parse sentences selected from a natural language corpus is a difficult task. One of the most serious problems is the unmanageably large number of ambiguities. Pure syntactic analysis based only on syntactic knowledge will sometimes result in hundreds of ambiguous parses. Martin [15] reported that his parser generated 455 ambiguous parses for the sentence:

List the sales of products produced in 1973 with the products produced in 1972.

Through the long history of work in natural language understanding, semantic and pragmatic constraints have been known to be indispensable for parsing. These should be represented in some formal way and be referred to during or after the syntactic analysis process. AI researchers have been exploring the use of semantic networks, frame theory, etc. to describe both factual and intuitive knowledge for the purpose of filtering out meaningless parses and to aid in choosing the most likely interpretation. The SHRDLU system [22] by Winograd successfully demonstrated the possibility of sophisticated language understanding and problem solving in this direction. However, to represent semantic and pragmatic constraints, which are usually domain sensitive, in a well-formed way is a very difficult and expensive task. To the best of our knowledge, no one has ever succeeded in doing so except in relatively small restricted domains.

Furthermore, there remains a basic question as to whether it is possible to formally encode all of the syntactic, semantic and pragmatic information needed for disambiguation in a definite and deterministic way. For example, the sentence

Print for me *the sales of stair carpets.*

seems to be unambiguous; however, in the ROBOT system pure syntactic analysis of this sentence resulted in two ambiguous parses, because the "ME" can be interpreted as an abbreviation of the state of Maine[9]. Thus, this simple example reveals the necessity of pragmatic constraints for the disambiguation task. Readers may claim that the system which would generate the second interpretation is too lax and that a human would never be perplexed by the case. However, a reader's view would change if he were told that the the sentence below had been issued previous to the sentence above.

Print for ca *the sales of stair carpets.*

Knowing that the speaker inquired about the business in California in the previous queries, it is quite natural to interpret "me" as the state of Maine in this context. A problem of this sort usually calls for the introduction of an appropriate discourse model to guide the parsing. Even with a sophisticated discourse model beyond anything available today, it would be impossible to take account all previous sentences: The critical previous sentence may always be just beyond the capacity of the discourse stack.

Thus it is quite reasonable to think of a parser which disambiguates sentences by referring to statistics which encode various characteristics of the past discourse, the task domain, and the speaker. For instance, the probability that the speaker is referring to states and the probability that the speaker is abbreviating a name, are useful in disambiguating the example. If the probabilities of the above are both statistically low, one could simply neglect the interpretation of the state of "Maine" for "me". Faced with such a situation, we propose, in this paper, to employ probability as a device to quantify language ambiguities. In other words, we will propose a hybrid model for natural language processing which comprises linguistic expertise, i.e. grammar knowledge, and its probabilistic augmentation for approximating natural language. With this framework, semantic and pragmatic constraints are expected to be captured implicitly in the probabilistic augmentation.

Section 2 introduces the basic idea of the probabilistic parsing modeling method and Section 3 presents the experimental results when this modeling method is applied to parsing problems of English sentences and of Japanese noun compound words. Detailed description of the method are given elsewhere.

2. Probabilistic Context-free Grammar

2.1 Extension to Context-free Grammar

A probabilistic context-free grammar is an augmentation of a context-free grammar [5]. Each of the grammar and lexical rules $\{r\}$, having a form of $\alpha \rightarrow \beta$, is associated with a conditional probability $Pr(r) = Pr(\beta \mid \alpha)$. This conditional probability denotes the probability that a non-terminal symbol α, having appeared in the sentential form during the sentence derivation process, will be replaced with a sequence of terminal and non-terminal symbols β. Obviously $\sum_{\beta} Pr(\beta \mid \alpha) = 1$ holds. Processes of sentence generation from a sentence symbol S by a probabilistic context-free grammar will be carried out in an identical manner to the usual non-probabilistic context-free grammar. But the advantage of the probabilistic grammar is that the probability can be computed for each of the derivation trees, which enables us to quantify sentence ambiguities as described below.

The probability of a derivation tree t can be computed as a product of conditional probabilities of the rules which are employed for deriving that tree t.

$$Pr(t) = \prod_{r \in D(t)} Pr(r)$$

Here r denotes a rule of the form $\alpha \rightarrow \beta$, and $D(t)$ denotes an ordered set of the rules which are employed for deriving the tree t. The next figure explains how the probability of a derivation tree t can be computed as a product of rule probabilities.

$$\begin{aligned} Pr(t) = {} & Pr(NP.VP.ENDM \mid S) \times \\ & Pr(DET.N \mid NP) \times \\ & Pr(\textbf{the} \mid \textbf{det}) \times \\ & Pr(\textbf{boy} \mid N) \times \\ & Pr(V.NP \mid VP) \times \\ & Pr(\textbf{likes} \mid V) \times \\ & Pr(DET.N \mid NP) \times \\ & Pr(\textbf{that} \mid \textbf{det}) \times \\ & Pr(\textbf{girl} \mid N) \times \\ & Pr(. \mid ENDM) \end{aligned}$$

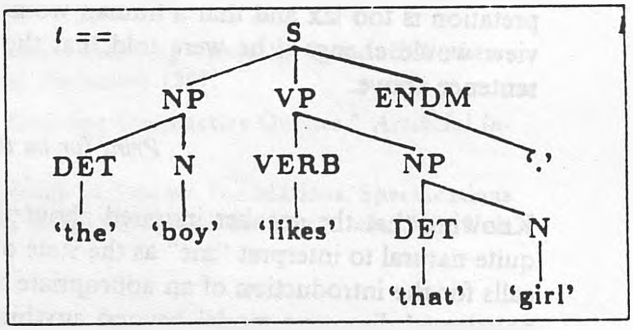

Fig. 1 Probability of a Derivation Tree

An ambiguous grammar allows many different derivation trees to coexist for sentences. From the viewpoint of sentence parsing, we say that a sentence is ambiguous when more than two parsed trees, say $t_1, t_2, \ldots$ are derived from the parsing process. Having a device to compute probability for a derivation tree as shown above, we can handle sentence ambiguity in a quantitative way. Namely, when a sentence s is parsed ambiguously into derivation trees $t_1, t_2, \ldots$ and a probability $Pr(t_i)$ is computed for each derivation tree t_i, the sum of the probabilities $\sum_i Pr(t_i)$ can be regarded as the probability that a particular sentence s will happen to be generated among other infinite possibilities. More interesting is the ratio denoting relative probabilities among ambiguous derivation trees:

$$\frac{Pr(t_j)}{\sum_k Pr(t_k)}$$

We can assume that it should denote the "likelihood" of each derivation tree. For example, consider the following English sentence *"Reply envelopes are enclosed for your convenience."* The sentence is ambiguous because it can be parsed in two different ways; the first being in the imperative mode, and the second in the declarative.

t_1: "Reply (that) envelopes are enclosed for your convenience." $\Rightarrow \dfrac{Pr(t_1)}{(Pr(t_1) + Pr(t_2))}$

t_2: "Reply envelopes (A kind of envelopes) are enclosed for your convenience." $\Rightarrow \dfrac{Pr(t_2)}{(Pr(t_1) + Pr(t_2))}$

These correspond to two different parsed trees, t_1 and t_2. By computing $Pr(t_1) + Pr(t_2)$, we can estimate the probability that the specific sentence *"Reply envelopes are ... "* is generated from among an infinite number of possible sentences. On the other hand, $Pr(t_1)/(Pr(t_1) + Pr(t_2))$ and $Pr(t_2)/(Pr(t_1) + Pr(t_2))$ give measures of likelihood for interpretations t_1 and t_2.

2.2 Estimation of Rule Probabilities from Data

The Forward / Backward algorithm, described in [11], popularly used for estimating transition probabilities for a given hidden-Markov-model, can be extended so as to estimate rule probabilities of a probabilistic context free grammar in the following manner.

Assume a Markov model, whose states correspond to possible sentential forms which appear in a sentence parsing process of a context free grammar. Then each transition between two states of the Markov model corresponds to an application of a context-free rule that maps one sentential form into another. For example, the state $NP.VP$ can be reached from the state S by applying the rule $S \rightarrow NP.VP$ to a start symbol S, the state $ART.NOUN.VP$ can be reached from the state $NP.VP$ by applving the rule $NP \rightarrow ART.NOUN$ to the first NP of the sentential form $NP.VP$, and so on. Since each rule corresponds to a state transition between two states, parsing a set of sentences given as training data will enable us to count how many times each transition is traversed. In other words, it tells how many times each rule is fired when the given set of sentences is generated. For example, the transition from the state S to the state $NP.VP$ may happen most frequently because the rule $S \rightarrow NP.VP$ is commonly used in almost every declarative sentence; while the transition from the state $ART.NOUN.VP$ to the state every.$NOUN.VP$ may happen 103 times; etc. In a context-free-grammar, each replacement of a non-terminal symbol occurs independently of the context. Therefore, counts of all transitions between states $\alpha.A.\beta$ to $\alpha.B.C.\beta$, with arbitrary α and β, should be tied together.

Counting the transitions in such a way for thousands of sentences will enable us to estimate the rule probabilities $\{Pr(\beta \mid \alpha)\}$ which are the probabilities that left hand side non-terminal symbols α will be replaced with right hand side patterns β. The actual iteration procedure to estimate these probabilities from N sentences $\{B^i\}$ is shown below.

1. Make an initial guess of $\{Pr(\beta \mid \alpha)\}$ such that $\sum_\beta Pr(\beta \mid \alpha) = 1$ holds.
2. Parse each output sentence B^i . Assume that grammar is ambiguous and that more than one derivation path exists which generate the given sentence B^i. In such cases, we denote D^i_j as the j-th derivation path for the ith-sentence.
3. Compute the probability of each derivation path D^i_j in the following way:

$$Pr(D^i_j) = \prod_{r \in D^i_j} Pr(r)$$

This computes $Pr(D^i_j)$ as a product of the probabilities of the rules $\{r\}$ which are employed to generate that derivation path D^i_j .

4. Compute the Bayes *a posteriori* estimate of the count $C^i_\alpha(\beta)$ which represents how many times the rule $\alpha \rightarrow \beta$ was used for generating the sentence B^i .

$$C^i_\alpha(\beta) = \sum_j \left(\frac{Pr(D^i_j)}{\sum_k Pr(D^i_k)} \times n^i_j(\alpha, \beta) \right)$$

Here, $n^i_j(\alpha, \beta)$ denotes the number of times the rule $\alpha \rightarrow \beta$ is used on the derivation path D^i_j .

5. Normalize the count so that the total count for rules with same left hand side non-terminal symbol α becomes 1.

$$f_\alpha(\beta) = \sum_i \frac{C^i_\alpha(\beta)}{\sum_\gamma C^i_\alpha(\gamma)}$$

6. Replace $\{Pr(\beta \mid \alpha)\}$ with $\{f_\alpha(\beta)\}$ and repeat from step 2.

Through this process, the $\{Pr(\beta \mid \alpha)\}$ will approach the real transition probability[2,10]. This algorithm has been proven to converge [3].

2.3 Parsing Procedure which computes Probabilities

To find the most-likely parse, that is, the parse tree which has the highest probability from among all the candidate parses, requires a lot of time if we calculate probabilities separately for each ambiguous parse. The following is a parsing procedure based on the Cocke-Kasami-Young [1] bottom-up parsing algorithm which can accomplish this task very efficiently. By using it, the most-likely parse tree for a sentence will be obtained while the normal bottom-up parsing process is performed. It gives the maximum probability $\mathrm{Max}_j Pr(t_j)$ as well as the total probability of all parses $\sum_j Pr(t_j)$ at the same time.

The Cocke-Kasami-Young parsing algorithm maintains a two-dimensional table called the Well-Formed-Substring-Table (WFST). An entry in the table, $WFST(i, j)$, corresponds to a substring(i,j), j words in length, starting at the i-th word, of an input sentence [1]. The entry contains a list of triplets. An application of a rule $\alpha \rightarrow \beta\gamma$ will add an entry (α, β, γ) to the list. This triplet shows that a sequence of $\beta.\gamma$ which spans substring(i,j) is replaced with a non-terminal symbol α. (β: is the pointer to another *WFST* entry that corresponds to the left subordinate structure of α and γ : is the pointer to the right subordinate structure of α.)

In order to compute probabilities of parse trees in parallel to this bottom-up parsing process, the structure of this WFST entry is modified as follows. Instead of having an one-level flat list of triplets, each entry of WFST was changed to hold a two-level list. The top-level of the two-level list corresponds to a left hand side non-terminal symbol, called as LHS symbol hereinafter. All combinations of left and right subordinate structures are kept in the sub-list of the LHS symbol. For instance, an application of a rule $\alpha \rightarrow \beta\gamma$ will add (β, γ) to the sub-list of α.

In addition to the sub-list, a LHS symbol is associated with two variables - *MaxP* and *SumP*. These two variables keep the maximum and the total probabilities of the LHS symbol of all possible right hand side combinations. MaxP and SumP can be computed in the process of bottom-up chart parsing. When a rule $\alpha \rightarrow \beta\gamma$ is applied, $MaxP$ and $SumP$ are computed as:

$$MaxP(\alpha) = \max_{\beta,\gamma}(Prob(\alpha \rightarrow \beta\gamma) \times MaxP(\beta) \times MaxP(\gamma))$$

$$SumP(\alpha) = \sum_{\beta,\gamma}(Prob(\alpha \rightarrow \beta\gamma) \times SumP(\beta) \times SumP(\gamma))$$

This procedure is similar to that of Viterbi algorithm[4] and maintains the maximum probability and the total probability in $MaxP$ and $SumP$ respectively. $MaxP/SumP$ gives the maximum relative probability of the most-likely parse.

3. Experiments

To demonstrate the capability of the modeling method, a few trials were made to disambiguate corpora of highly ambiguous phrases. Two of these experiments will be briefly described below. Details can be found elsewhere.

3.1 Disambiguation of English Sentence Parsing

As the basis of this experiment, the grammar developed by Prof. S. Kuno in the 1960's for the machine translation project at Harvard University [13,14,18] was used with some modification. The set of grammar specifications in the Kuno grammar, which are in Greibach Normal form, were translated into a form which is more favorable to our method. The 2118 original rules were reformulated into 7550 rules in Chomsky normal form[1].

Training sentences were chosen from two corpora. One corpus is composed of articles from *Datamation* and *Reader's Digest* (average sentence length in words 10.85, average number of ambiguities per sentence 48.5) and the other from business correspondence (average sentence length in words 12.65, average number of ambiguities per sentence 13.5). A typical sentence from the latter corpus is shown below:

It was advised that there are limited opportunities at this time.

The 3582 sentences from the first corpus, and 624 sentences from the second corpus that were successfully parsed were used to train the 7550 grammar rules besides some lexical rules in each corpus.

Once the probabilities for rules are thus obtained, they can be used to disambiguate sentences by the procedure described in section 2.3.

```
    SENTENCE
      PRONOUN      ( we )
      PREDICATE
        AUXILIARY    ( do )
        INFINITE VERB PHRASE
          ADVERB TYPE1 ( not )
  (A) 0.356 INFINITE VERB PHRASE
   :    VERB TYPE IT1( utilize )
   :    OBJECT
   :      NOUN        ( outside )
   :      ADJ CLAUSE
   :        NOUN        ( art )
   :        PRED. WITH NO OBJECT
   :          VERB TYPE VT1 ( services )
  (B) 0.003 INFINITE VERB PHRASE
   :    VERB TYPE IT1( utilize )
   :    OBJECT
   :      PREPOSITION  ( outside )
```

```
      :     NOUN OBJECT
      :       NOUN        ( art )
      :     OBJECT
      :       NOUN          ( services )
    (C) 0.641 INFINITE VERB PHRASE
      :   VERB TYPE IT1( utilize )
      :   OBJECT
      :     NOUN        ( outside )
      :     OBJECT MASTER
      :       NOUN        ( art )
      :       OBJECT MASTER
      :         NOUN        ( services )
      PERIOD
        ADVERB TYPE1 ( directly )
        PRD          ( . )
```

Fig. 2 Parse Tree for "We do not utilize . . ."

Figure 2 shows the parsing result for the sentence *"we do not utilize outside art services directly."*, which turned out to have three ambiguities.

As shown in the figure, ambiguities come from the three distinct substructures, (A), (B) and (C), for the phrase *"utilize outside art services."*. The derivation (C) corresponds to the most common interpretation while in (A) *"art"* and *"outside"* are regarded respectively as subject and object of the verb *"services"*. In (B), *"art service"* is regarded as an object of the verb *"utilize"* and *"outside"* is inserted as a preposition. The numbers 0.356, 0.003 and 0.641 signify the relative probabilities of the three interpretations. As shown in this case, the correct parse (the third one) gets the highest relative probability, as was expected.

Some of the resultant probabilities obtained through the iteration process for each of the grammar rules and the lexical rules are shown below.

Rules for "IT6"[1]

(0.11054)	IT6 → BELIEVE	--(a)
(0.10685)	IT6 → KNOW	--(b)
(0.08562)	IT6 → FIND	
(0.07628)	IT6 → THINK	
(0.03525)	IT6 → CALL	
(0.03280)	IT6 → REALIZE	
.	.	

Rules for "IT3"[2]

(0.16055)	IT3 → GET
(0.12447)	IT3 → MAKE
(0.11988)	IT3 → HAVE
(0.08132)	IT3 → SEE
(0.06477)	IT3 → KEEP
(0.06363)	IT3 → BELIEVE
.	.

Rules for "SE"[3]

(0.21602)	SE → AAA 4X VX PD	---(c)
(0.15296)	SE → PRN VX PD	---(d)
(0.15229)	SE → NNN VX PD	
(0.11965)	SE → AV1 SE	
(0.04730)	SE → PRE NQ SE	
(0.04457)	SE → NNN AC VX PD	
(0.02616)	SE → AV2 SE	
.	.	

Rules for "VX"[4]

(0.19809)	VX → VT1 N2
(0.10704)	VX → PRE NQ VX
(0.08790)	VX → VI1
(0.07500)	VX → AUX BV
(0.05455)	VX → AV1 VX
.	.

Fig. 3 Rule probabilities estimated by iteration

Numbers in the parentheses on the left of each rules denote probabilities estimated from the iteration process described in the section 3.3. For example, the probabilities that the words **believe**, and **know** have the part of speech **IT6** are shown as 11.1\% and 10.7\% on lines (a) and (b) respectively. Line (c) shows that a sequence AAA (article and other adjective etc.) 4X (subject noun phrase), VX(predicate) and PD (period or post sentential modifiers followed by period) forms a sentence (SE) with probability 21.6\%. Line (d), on the other hand, shows that a sequence PRN

[1] Infinite form of a mono-transitive verb which takes a noun-clause object

[2] Infinite form of a complex-transitive verb which takes an object and an objective compliment

[3] sentence

[4] predicate

(pronoun), VX and PD forms a sentence (SE) with probability 15.3 %. In such ways, the probability findings convey useful information for language analysis.

Table 1 summarizes the experiments. Test 1 corresponds to the corpus of articles from *Datamation* and *Reader's Digest*, while Test 2 derived from the business correspondence. In both cases, the base Kuno grammars were successfully augmented by probabilities.

a.	Corpus	test1	test2
b.	Number of sentences used for training	3582	624
c.	Number of sentences checked manually	63	21
d.	Number of sentences with no correct parse	4	2
e.	Number of sentences where highest prob. was given to the most natural parse	54	18
f.	Number of sentences where highest prob. was not given to the right one	5	1

Table 1. Summary of English sentence parsing

3.2 Disambiguation of Japanese Noun Compound Word Parsing

Analyzing structures of noun compound words is difficult because noun compound words usually do not have enough structural clues for syntactic parsing[17]. Particularly in the Japanese language, noun compound words consist only of a few types of components, and pure syntactic analysis will result in many ambiguous parses. Some kind of mechanism which can handle inter-word analysis of constituent words is needed to disambiguate them.

We applied our probabilistic modeling method for disambiguating parsing of Japanese noun compound words. It was done by associating rule probabilities to basic construction rules of noun compound words. In order to make rule probabilities sensitive to inter-word relationship of component words, words were grouped into finer categories $\{N_1, N_2, N_3, \ldots N_m\}$. The base rules were replicated for each combination of right hand side word categories. Since we assumed that the right-most word of the right hand side inherits the category from the left hand side parent, a single $N \rightarrow NN$ rule was replicated to $m \times m$ rules. For these $m \times m$ rules, separate probabilities were prepared and estimated. The method described in the section 2.2 was used to estimate these probabilities from noun compound words actually observed in text.

Once probabilities for rules were estimated, the parsing procedure described in the section 2.3 was used to compute relative probability of each parse tree i.e. the likelihood of the parse tree among others.

In this experiment, we categorized words by a conventional clustering technique which groups words according to neighboring word patterns. For example, "oil" and "coal" belong to the same category in our method because they frequently appear in similar word patterns such as "~ burner", "~ consumption", "~ resources". 31,900 noun compound words picked from abstracts of technical papers [12] were used for this categorization process. Twenty eight categories were obtained through this process for 1000 high-frequency 2-character kanji primitive words, 8 categories for 200 prefix single-character words, and 10 categories for 400 suffix single-character words[16]. Base rules deriving from different combination of these 46 word categories resulted in 5582 separate rules. These base rules are displayed below.

<word> → <2 character kanji primitive word>

<word> → <word> <word>

< word > → < prefix single character word > < word >

< word > → < word > < suffix single character word >

5582 conditional probabilities of these rules were estimated from 28,568 noun compound words.

After training was successfully done, 153 noun compound words were randomly chosen, parsed by the procedure shown in the section 3.3 and the parse trees were examined by hand. The check was made whether the correct parse is given the highest probabilities. Among the 153 test words, 22 was uniquely parsed and 131 test words were parsed with more than two alternative parse trees. Among 131, in 92 cases, the right parses were given the highest probabilities.

Show below are parsing results for two noun compound words.

word 1: 中(medium) 規模(scale) 集積(integrated) 回路(circuit)

word 2: 小(small) 規模(scale) 電力(electricity) 会社(company)

(Word order is the same both in English and in Japanese).

For both of these cases, 5 alternative parse trees were given. Obtained parse trees were computed with relative probabilities, the likelihood, among other alternative parses. In the first sentence, the 5-th parse tree, which is the most natural, got the highest probability 0.43. In the second case, the 3rd parse tree, which is the most natural, got the highest probability 1.0.

	word 1 "medium scale integrated circuit"		
	structure of parsed tree shown in bracket notation	meaning implied from structure	prob.
1	medium [[scale integrated] circuit]	a medium-size "scale-integrated-circuit"	0.17
2	medium [scale [integrated circuit]]	a medium-sized integrated circuit which is scale (?)	0.04
3	[medium scale] [integrated circuit]	an integrated-circuit of medium-scale	0.19
4	[medium [scale integrated]] circuit	a medium-size circuit which is scale-integrated	0.17
5	[[medium scale] integrated] circuit	a circuit which is medium-scale integrated	0.43
case 2 "small scale electricity company"			
1	small [[scale electricity] company]	a small company which serves scale-electricity	0.0
2	small [scale [electricity company]]	a company which is small, serves electricity, and is something to do with scale	0.0
3	[small scale] [electricity company]	a company which serves electricity and which is small scale	1.0
4	[small [scale electricity]] company	a company which services small scale-electricity	0.0

5	[[small scale] electricity] company	a company which services small scale electricity (micro electronics?)	0.0

4. Concluding Remarks

N-gram modeling technique [20] has been proven to be a powerful and effective method for language modeling. It has successfully been used in several applications such as speech recognition, text segmentations, character recognition and others.[11,6,7,19,21] At the same time, however, it has proved to be difficult to approximate language phenomena precisely enough when context dependencies expand over a long distance. A direct means to remedy the situation is (a) to increase N of N-gram or (b) to increase the length of basic units from a character to a word or to a phrase. If the vocabulary size is M, however, the statistics needed for maintaining the equivalent precision in the N-gram model increase in proportion to M^N. The situation is similar in (b). Increasing the length of the basic unit causes an exponential increase in vocabulary size. Hence an exponential increase of the required statistics volume follows in (b) as well. This shows that the N-gram model faces a serious data gathering problem when a task has a long-context dependency. Obviously, the parsing of sentences creates this sort of problem.

On the other hand, the method introduced here aims to remedy this problem by combining a probabilistic modeling procedure with linguistic expertise. In this hybrid approach [7,8], linguistic expertise provides the framework of a grammar, and the probabilistic modeling method augments the grammar quantitatively.

Since the probabilistic augmentation process is completely automatic, it is not necessary to rely on human endeavor which tends to be expensive, inconsistent, and subjective. Also the probabilistic augmentation of a grammar is adaptable for any set of sentences.

These two important features make the method useful for various problems of natural language processing. Besides its use for sentence disambiguation demonstrated in the section 3.4, the method can be used to customize a given grammar to a particular sub-language corpus. Namely, when a grammar designed for a general-corpus is applied to this method, the rules and the lexical entries which are used less frequently in the corpus will automatically be given low or zero probabilities. Alternately, the rules and the lexical entries which require more refinement will be given high probabilities, thus the method helps us to tune a grammar in a top-down manner. The method is also useful for improving performance of top-down parsing when used for obtaining hints for reordering rules according to the rule probabilities.

In this way, although all possible uses have not been explored the method proposed in this paper has enormous potential application, and the author hopes that a new natural language processing paradigm may emerge from it.

Use of probability in natural language analysis may seem strange, but it is in effect a only simple generalization of common practice: Namely, the usual top-down parsing strategy forces a true or false (1 or 0) decision, i.e. to choose one alternatives from others on every non-deterministic choice point.

And most importantly, by use of the proposed method a grammar can be probabilistically augmented objectively and automatically from a set of sentences picked from an arbitrary corpus. On the other hand, the representation of semantic and pragmatic constraints in the form of usual semantic networks, frame theory, etc., requires a huge amount of subjective human effort.

Acknowledgement

The authors are indebted to Dr. B. Mercer, and Dr. L. Bahl of the IBM Thomas J. Watson Research Center, for their valuable technical suggestions and comments.

Prof. S. Kuno of Harvard University has not only given the senior author permission to use the Kuno grammar, but also much valuable linguistic advice.

References

[1] A.V. Aho, J.D. Ullman, *The Theory of Parsing, Translation and Compiling*, Vol. 1, Prentice-Hall, 1972

[2] J.K. Baker, *Trainable Grammars for Speech Recognition*, internal memo, 1982

[3] L.E. Baum, *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions in Markov Chains*, Vol. 41, No. 1, The Annals of Mathematical Statistics, 1970

[4] G.D. Forney, Jr., *The Viterbi Algorithm*, Proc. of IEEE, Vol. 61, No. 3, 1973

[5] K.S. Fu, *Syntactic Methods in Pattern Recognition* , Vol. 112, Mathematics in Science and Engineering, Academic Press, 1974

[6] T. Fujisaki, *A Scheme of Separating and Giving Phonetic Transcriptions to Kanji-kana Mixed Japanese Documents by Dynamic Programming*.(in Japanese), Natural Language Workshop Report NL28-5, Inf. Proc. Soc. of Japan, 1981

[7] T.Fujisaki, *Handling of Ambiguities in Natural Language Processing* (in Japanese), Doctoral dissertation, Dept. of Information Science, Univ. of Tokyo, 1985

[8] T. Fujisaki, *An Approach to Stochastic Parsing*, Proc. of COLLING84, 1984

[9] L. Harris *Experience with ROBOT in 12 Commercial Natural Language Database Query Applications*, Proc. of IJCAI, 1979.

[10] F. Jelinek, *Notes on the Outside Inside Algorithm*, internal memo, 1983

[11] F. Jelinek, et. al, *Continuous Speech Recognition by Statistical Methods*, Proc. of the IEEE, Vol. 64, No. 4, 1976

[12] MT for Electronics, *Science and Technology Database*(in tape form), Japanese Information Center for Science and Technology, Vol. 26

[13] S. Kuno, *The Augmented Predictive Analyzer for Context-free Languages and Its Relative Efficiency*, CACM, Vol. 9, No. 11, 1966

[14] S. Kuno, A.G. Oettinger, *Syntactic Structure and Ambiguity of English*, Proc. of FJCC, 1963

[15] W.A. Martin, et al., *Preliminary Analysis of a Breadth-First Parsing Algorithm: Theoretical and Experimental Results*, MIT LCS Report TR-261, 1979

[16] T. Nishno, T. Fujisaki, *Probabilistic Parsing of Kanji Compound Words* (in Japanese), J. of Inf. Proc. Soc. of Japan, Vol 29, No.11, 1988

[17] T. W. Finin., *Constraining the interpretation of nominal compounds in a limited cpmtext*, In R. Grishman and R. Kittredge, editors, Analysing Language in a Restricted Domian, Lawrence Erlbaum Assoc., Hillsdale, 1986

[18] A.G. Oettinger, *Report No. NSF-8: Mathematical Linguistics and Automatic Translation*, The Computation Laboratory, Harvard Univ., 1963

[19] J. Raviv, *Decision Making in Markov Chains Applied to the Problem of Pattern Recognition*, IEEE, Trans. Information Theory, Vol. IT-3, No. 4, 1967

[20] C.E. Shannon, *Prediction and Entoropy of Printed English*, Bell Sys. Tech. J., Vol. 30, 1951

[21] K. Takeda, T. Fujisaki, *Segmentation of Kanji Primitive Word by a Stochastical Method*(in Japanese), J. of Inf. Proc. Soc. of Japan, Vol 28, No.9, 1987

[22] T. Winograd, *Understanding Natural Language* New York, Academic Press, 1972

[23] T. Winograd, *Language as a Cognitive Precess* Vol. 1 Syntax, Addison Wesley, 1983