A Statistical Approach to Parsing Ill-Formed Input

1G－6

Pierre Hudry　Yasuharu Den
ATR Interpreting Telecommunications Research Laboratories
e-mail: {hudry,den}@itl.atr.co.jp

1 Introduction

This paper describes an algorithm used for parsing spontaneous speech and attempting to deal in an efficient way with three types of ill-formedness: false starts, filled pauses, and substitutions. The approach proposed to recover from these frequently occuring errors is based on a stochastic language model called Bayesian Language Inference (BLI). After a brief review of previous work on parsing ill-formed input, a general overview of the BLI algorithm will be given, emphasizing characteristics that make it an appropriate tool capable of partially dealing with disfluencies and substitutions. This paper will focus on how that partial information can be reorganized in order to accurately parse sentences displaying these kinds of ill-formedness, leading to the description of a statistical processing system capable of analysing ill-formed input with mathematically sound consideration of full syntactic context.

2 Parsing Ill-Formed Input

There have been a great number of semantics-free approaches to the problem of parsing ill-formed sentences. From simple pattern-matching techniques to chart-based methods, previous work led to a certain number of characteristics to be expected from a parsing system capable of dealing with simple kinds of ill-formedness:

- precise classification of ill-formedness handled
- consideration of full syntactic context
- similar computational costs in parsing well and ill-formed input

Chart-based methods suffer from a left-right bias making it difficult to take into account right context. In order to overcome this difficulty, island-driven chart-parsing [1] and combinations of bottom-up and top-down parsing [2] are two ideas which led to further exploration in the field. However those methods fail to choose the *best* possible parse of the input. Moreover this lack of quantitative context analysis also introduces higher computational costs.

The use of statistical models, while allowing automatic training of stochastic grammars, also provides the quantitative analysis needed in the disambiguation process. But simple local models like n-gram models and probabilistic context-free grammars, or even the more complex lexicalized grammar formalisms only give us general information about how likely a structure is to appear *anywhere* in a given sentence. However, within recent years, new models have been proposed based on the idea that rule expansion should also take into account broader linguistic context. Among those, a stochastic language model called Bayesian Language Inference (BLI) has the advantage of considering full syntactic context while performing strictly local calculations.

3 Bayesian Language Inference

BLI is a stochastic language model developped by H.Lucke [3] based on a context-free language formalism i.e. a set of terminal and non-terminal symbols and a set of rewriting rules. Assuming that the observation sequence has been divided into segments, the BLI algorithm first determines the parse tree topology for a given sentence, but without making assumptions on the nature of each node. This is done by conjecturing the existence of all possible nodes spanning any sub-sequence of the sentence. Recursive calculation of the probability that each node rewrites as the sub-sequence it spans is then performed, in a way similar to calculations of inside probabilities in the Inside-Outside training algorithm. Using prior parameters obtained through training, division points in the parse tree are determined by minimizing entropy.

Given this blank tree structure, the BLI method then assign non-terminal symbols to the nodes of the tree. For each node, it divides the input sentence in two parts (Figure 1):

- inner evidence e_u^-, part of the sequence actually produced by node u
- outer evidence e_u^+, remaining part of the evidence

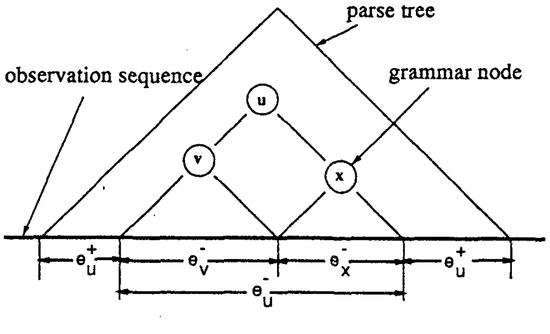

Figure 1: Definition of inner and outer evidence

e stands for the entire observation sequence spanned by the tree. In this probabilistic framework, the assignment task for each node u is simply that of finding the vector $\mathrm{BEL}(u) = P(u|e)$ and assigning to u the non-terminal symbol with highest probability. In order to calculate $\mathrm{BEL}(u)$, two auxiliary vectors are defined, $\lambda(u) = P(e_u^-|u)$ and $\pi(u) = P(u|e_u^+)$. $\lambda(u = nt)$ and $\pi(u = nt)$ are the probabilities that node u stands for non-terminal nt, the first probability being based on inner evidence, the second on outer evidence. All the λs and πs can be determined recursively using only local calculations, and the belief vector is then given by:

$$\mathrm{BEL}(u) = \frac{\lambda(u)\pi(u)}{\lambda(u) \cdot \pi(u)}$$

where ab is componentwise vector product, and $a \cdot b$ is the dot product. This equation can be understood in the following way: for each non-terminal symbol nt, the probability that node u stands for nt (given the entire observation sequence) can be seen as the combination of two different sources of information: inner evidence ($\lambda(u)$) and outer evidence ($\pi(u)$). Full syntactic context is therefore considered, divided into inner and outer evidence.

4 Expanding BLI to Deal with Ill-Formed Input

The main idea behind this work is that whereas the BLI method uses λ and π vectors as auxiliary means to calculate BEL, the information these vectors contain is particularly useful as it is in parsing ill-formed input. Keeping in mind that the types of ill-formedness dealt with in this paper are false starts, filled pauses, and substitutions, let's now examine how a BLI parser trained on well-formed input might recover from each type of error, with a given parse tree structure already available.

In the case of false starts and filled pauses, also known as disfluencies, the problem is to identify and eliminate non-contributing portions of the input sequence. In the case of an isolated disfluency, spanned for instance by node v (Figure 1), ill-formedness will be detected at node u. However while the inner portion of the evidence including the disfluency will fail to bring the information necessary to determine $\lambda(u)$, the parser will give an accurate analysis of the outer portion of the evidence, $\pi(u)$ corresponding to well-formed input. Thus in the case of a disfluency occuring at node v, nodes x and u being in fact one identical node, the disfluency can be detected by directly comparing $\lambda(x)$ and $\pi(u)$. If the non-terminal symbols with highest probability for these two vectors are identical, the portion of the utterance responsible for the disfluency can then be eliminated and the parser proceed with calculations of λ and π vectors.

In the case no match is found between the non-terminal symbols yielded by the analysis of outer and inner evidence, we can then hypothesize that the ill-formedness involved here is due to a substitution. Assuming that we have a single substitution occuring, the π probabilities based on outer evidence can be calculated for each single word. We can then argue for the presence of a substitution whenever a contradiction appears between information brought by bottom-up analysis (the syntactic nature of the words, given by the inner evidence) and by top-down analysis (the expected non-terminal given the global sentence context, corresponding to outer evidence). Several ways of measuring this contradiction can be devised. But what appears to be the simplest one is directly comparing the values of BEL corresponding to the highest non-terminal probabilities for each word, and deciding that the lowest one indicates in which word the substitution occurs.

Yet two major problems remain to be addressed: how to deal with multiple errors and how to determine a possible parse tree structure from ill-formed input. A possible answer to the first question would be to use estimates for the π probabilities, which can't be calculated in the case of multiple errors. Such estimates are already provided by the original BLI algorithm in the form of prior probabilities. But more importantly, to complete the description of the proposed algorithm, the parse tree structure has to be determined in some way. This can be done by introducing noisy modifications in the probabilities of the rewriting rules. Thus if the input is well-formed, the chosen tree structure will not differ from the one found in the original BLI method. Moreover in the case of ill-formed input, the tree structure chosen will be the one using the greatest number of partially well-formed structures, which is what we would naturally tend to expect from such a system.

References

[1] Stock, O., R. Falcone, and P. Insinnamo, Bidirectional Charts: A Potential Technique for Parsing Spoken Natural Language Sentences, *Computer Speech and Language*, No. 3, pp. 219–237, 1989.

[2] Mellish, C., Some Chart-Based Techniques for Parsing Ill-Formed Input, *Proceedings of 27th Annual Meeting of the ACL*, pp. 102–109, 1989.

[3] Lucke, H., Inference of Stochastic Context-Free Grammar Rules from Example Data using the Theory of Bayesian Belief Propagation, *Proceedings of Eurospeech '93*, pp. 1195–1198, 1993.