PARSING SPOKEN UTTERANCES IN AN INFLECTIONAL LANGUAGE

M. Prakash*, G.V. Ramana Rao**, C. Chandra Sekhar** and B. Yegnanarayana**

* Dept. of Electronics and Computer Science, Regional Engineering College, Warangal, India
** Dept. of Computer Science and Engg., I.I.T., Madras, India

ABSTRACT

This paper proposes a new approach to parse spoken utterances suitable for inflectional languages such as Hindi. This approach exploits a lot of redundant syntactic and semantic information present in the inflections. In this approach, parsing is done at three levels: phrase level, clause level and sentence level. This is because each level requires a different parsing strategy. Phrase level parsing uses both the frame approach and the pattern matching approach. Therefore it has the directional freedom and the ability to focus upon the case markers as in the case frame approach and the robustness of the pattern matcher in the neighbourhood of a case marker. Clause level parsing allows the free occurrence of certain phrases within a clause. Sentence level parsing uses a simple pattern matching technique. The spurious and missing case markers are treated specially using metaknowledge.

1 INTRODUCTION

We are developing a dictation machine for continuous speech in Hindi. This machine is to be used for a restricted task like transcribing a carefully read text from short stories for children. It is to be independent of speaker, vocabulary and task. Like many speech recognition systems, our system consists of three expert modules: an acoustic-phonetic expert [1], a lexical expert [2] and a syntactic expert (includes surface level semantic knowledge also). Speech being a natural input, the output symbol string from the acoustic-phonetic expert may not correspond to the input utterance exactly. There will be several ambiguities and these will propagate to the word level through the lexical expert. Syntax and semantics can be used to disambiguate and recover the correct sentence from this erroneous string of words. Hindi is an inflectional language, and the inflections carry a lot of syntactic and semantic information. The objective of the work reported here is to address the issues of approaches to system development and parsing for the syntactic expert, taking the inflectional characteristics of the language into consideration.

For syntactic and semantic knowledge sources, the speech recognition systems built so far have used statistical models [3][4][5], linguistic models [6][7] and even a combination of these two [8]. The statistical models work locally. They concern either the semantic level, like trigram model [4], or the syntactic level, like tri-POS model [8]. All the systems working at the semantic level are task dependent. These models have an inherent limitation in that they are not capable of capturing the global syntax and the global semantics. To overcome this limitation, one has to resort to linguistic modeling. Although there is some work done on these lines for speech input itself [6][7], a lot of work has been done on a related topic, namely, processing ill-formed text [9][10]. However, these models cannot exploit the inflectional characteristics of Hindi, and hence a need for a new approach is felt.

We believe that the phonetic and inflectional characteristics of Indian languages lend themselves to the realization of a speaker, vocabulary and task independent system. Because of the phonetic nature of Indian languages, the characters of the language themselves can be used as symbols in the signal-to-symbol transformation stage and this greatly reduces the number of ambiguities when compared to English [1]. Hence, the burden on the syntactic expert will be less. Also, Indian languages are "free ward order" languages. Some of the information conveyed by word order in English is conveyed by the explicit use of case markers (a subset of inflections) in Hindi. The case markers and the clause boundaries can be hypothesized very reliably by the lexical expert in our system. These are the main reasons behind our new approach, which centers around case markers. In Section 2, we discuss our approaches to system development and parsing. The implementation details are presented in Section 3 and the conclusions are given in Section 4.

2 OUR APPROACH

The input to the syntactic expert module of our speech recognition system is a word lattice, i.e., a sequence of words with alternatives, which are identified by their begin and end times, and the quality of acoustic-phonetic match. The ambiguities normally present in a word lattice are of substitution, insertion and deletion types. In the context of ill-formedness in Natural Language Processing, substitution type of ambiguities corres-
pond to both unknown words and syntactic and semantic agreement constraint violations whereas the insertion and deletion types correspond to spurious and missing constituents respectively. In addition, the spoken utterances may have language specific ill-formedness like omission of words. Our aim is to recover the original utterance in the presence of these ambiguities.

Our approach to achieve this aim uses a three-phase parsing strategy in which parsing is carried out at phrase level, clause level and sentence level, in that order. This strategy treats case markers and clause boundaries as islands. Phrase level parsing relies heavily on local syntax and local semantics. This is where most of the robustness across languages like SP (Subject Phrase), DOP (Direct Object Phrase), LP (Locative Phrase) and ANP (Abstract Noun Phrase) are directly applicable. Hence, by relying less on such constraints, we can be found out locally. Clause level and sentence level parsing phases make use of global syntax and global semantics. The central idea of separating the parsing process into three phases is that they require different strategies. It aims at combining the positive features of the pattern matching and the case frame approaches.

2.1 Phrase Level Parsing

The input to this phase is the given word lattice, and the output contains, in addition to the given word lattice, phrases which can occur freely within a clause. The approach to any sub-system in this phase is as follows. We have two systems: System 1 consists of patterns which account for well-formed and language specific ill-formed utterances, and a metric to treat the speech specific ill-formedness due to the syntactic and semantic constraint violations. The rest of the speech specific ill-formedness is processed by a separate system, System 2.

System 1 makes use of a metric to rank the alternatives based on the syntactic and the semantic constraint violations. In many cases, the markers which are used to indicate these constraints are prone to errors. For example, the nasalisation marker which indicates the number could easily be missed. Also, the vowels used for gender agreement are highly confusable. Hence, by relying less on such constraints, we can expect a better performance. This can be achieved by using a metric to rank the alternatives, i.e., make more hypotheses. In any sub-system, each of the rules will be used idiomatically to stand for other cases/ phrases in the context of certain words. In our approach these can be treated more systematically.

The approach to parsing at the phrase level is based on the case frame approach and the pattern matching approach. It has the directional freedom and the ability to focus upon the case markers as in the case frame approach, and the robustness of the pattern matcher in the neighbourhood of a case marker. When a word lattice is given, to begin with, the NPrules and the VPrules are used to parse in bottom-up mode. Then, each of the sub-systems is invoked in turn in an order based on the dominance of the corresponding case marker in the parse tree, starting with the least dominating case marker. For example, the sub-system corresponding to the possessive case marker is invoked before that of genitive case marker because the genitive case marker always dominates the possessive case marker in the parse tree.

Each sub-system traverses the word lattice from left to right. Whenever it finds its case marker in the word lattice, it uses its System 1 to hypothesize various phrases. With the present case marker as an island, it takes each rule of System 1 in turn, and checks for the presence of the other categories on both the sides to make various hypotheses. If no hypothesis could be made, metarules of System 2 are used to diagnose the problem and remedy it. For example, a metarule belonging to the sub-system for the possessive case marker ‘ki:’(kid) is as follows: If the previous position contains an auxiliary verb form such as ‘ho:’(hold), then either delete ‘ki:’(kid) or change it to ‘ki:’(kki) (a clause connector) depending on whether ‘ki:’(kki) is already present or not. Finally, to account for the missing case markers, a set of metarules is added in those regions where no phrase is hypothesized.
2.2 Clause Level Parsing

During this phase, various clauses present in the given word lattice are hypothesized. Here, we use only one system which uses well-formed constraints to hypothesize the clauses. However, once a clause is hypothesized, we associate a score with it based on the scores of its components. The constraints are divided into two sets: a set of heuristic rules and a set of subject-verb agreement rules. An example for a heuristic rule is: if there is no subject phrase in a sentence hypothesis and if there is only one animate noun phrase, then treat that as the subject phrase by adding the appropriate case marker to it.

The clause level parsing proceeds as follows. The search for a clause in a word lattice starts from the beginning of the word lattice and from all the intermediate clause connectors present in the word lattice. The search starts from one position to the right of these also by skipping one word. Hypothetically, it could start from any position, but by looking at the sentence level rules, a good heuristic is that it can utmost start at one position after a clause connector. A clause connector is either included or excluded depending on its type. For example, clause connectors like ‘jhin’(जीता) are included while those like ‘ki’(किया) are excluded. As mentioned earlier, Hindi is a “free word order” language. The various phrases can occur in any order in a clause. Hence, each sequence of phrases constitutes a hypothetical clause. It ends either at the beginning of a clause connector or at the end of word lattice. A clause is actually hypothesized when the hypothetical clause satisfies all the well-formed constraints.

Spurious clause connectors are detected when no clause is hypothesized and are deleted. But it is harder to detect the missing clause connectors. Currently, we have only one heuristic to recover from such cases: if a hypothetical clause has two VPs, then hypothesize a clause connector immediately after the first VP. This works satisfactorily.

2.3 Sentence Level Parsing

During this phase, we use only patterns to account for the well-formed utterances and a metric to rank the alternatives based on how far they are from satisfying all the constraints of well-formedness. The metric is based on the syntactic and semantic constraint violations. Patterns are used to parse in the bottom-up mode.

3 IMPLEMENTATION AND DISCUSSION

Our syntactic expert module is being implemented on a Microvax-II using LISP. Various types of word lattices are used for the development of our system to account for the types of ambiguities as mentioned earlier. They are degraded word lattices, missing word lattices and spurious word lattices.

In degraded word lattices, corresponding to every word position in the input utterance, at least one word is always hypothesized. Since our lexical expert module is also under development, the input to the syntactic expert module is simulated. For each word in the input utterance, words which are close to it are found out based on a confusability matrix (this matrix reflects the confusion between any two characters by giving different numerical values) and by using a simple Approximate String Matching algorithm. Usually we get between 2 to 20 alternatives. These alternatives are rearranged in the word lattice randomly. From these word lattices we generate word lattices of varying degradations by changing the position of the correct words. So, in effect, these account for substitution type of ambiguities and unknown words. These word lattices are mainly useful in acquiring the metaknowledge required to parse in case of blocking in the normal process of parsing.

Charts are used as the central data structure. Currently, the system contains around 300 patterns, and the words in the lexicon have 60 lexical and semantic labels with 17 attributes in all. The approaches to parsing and system development adopted resulted in the smooth development of the system.

The number of alternatives and the length of input utterance seem to affect the recognition accuracy as one would expect. As the number of alternatives increases, the accuracy decreases and as the length increases, the accuracy increases. But, in both the cases, the change is not dramatic. However, the time required to parse increases non-linearly with both an increase of number of alternatives and length. Further, this increase is mainly in clause level parsing because of the free word order nature. Hence, a better strategy is required during this phase.

4 CONCLUSIONS

We have proposed a parsing strategy suitable for inflectional languages. This strategy treats case markers, which contain a lot of information and can be hypothesized reliably, as islands. The approach to parsing has three phases because each phase requires a different strategy. The approach to system development is such that different types of ambiguities are dealt with differently.
In our approach, the idiomatic usages are treated more systematically. These approaches resulted in the smooth development of the system. While these approaches worked well from the point of view of recognition, the strategy adopted in clause level parsing may have to be changed from the point of view of time complexity.

REFERENCES


