A SPEECH UNDERSTANDING SYSTEM BASED UPON A CO-ROUTINE PARSER

James F. Allen

Computer Science Department
University of Toronto
Toronto, Ontario
Canada

Abstract

This paper gives a brief description of the speech understanding effort under Development at the Univ. of Toronto. The main purpose so far has been to produce a base from which further research into the "higher levels" of speech understanding (semantics, pragmatics, user models, syntax) may build on.

Some features of interest in this system are the syllable based pattern recognition, the dynamic reclassification of the input signal according to expectations, interactive pattern formation, dictionary retrieval by sound characteristics and finally, the use of an Augmented Transition Network grammar with a co-routine parsing scheme which can be guided by prosodic and semantic information.

Most of the emphasis in the paper is placed on the co-routine parsing scheme which is illustrated with a detailed example.

Introduction

This paper gives a brief description of the speech understanding effort under development at the University of Toronto. The main purpose so far has been to produce a base from which further research into the "higher levels" of speech understanding (semantics, pragmatics, user models, syntax) may build on.

The system is designed to handle continuous but not conversational speech. The speaker is required to speak carefully which can be loosely defined as speaking as though to a person who was not completely fluent in English. It is felt that advances made at this "careful" level would certainly be pertinent to the more general problems of normal conversational speech.

Developed so far is a first iteration of a speech system using extensive syntactic support. Some features of interest in this sytem are the syllable based pattern recognition, the dynamic reclassification of the input signal according to expectations, interactive pattern formation, dictionary retrieval by sound characteristics and finally, the use of an Augmented Transition Network grammar with a co-routine parsing scheme which can be guided by prosodic and semantic information.

Following the description of the system is a detailed example of how the system attempts to decode a sample utterance. A complete description of most parts of the system will be found in Allen [1974].

Description of the System

The underlying philosophy in the system is that decisions should be made at the latest possible moment, and furthermore, one should be able to reconsider these decisions later when new information is revealed. Because of the unreliability of the input signal, we consider this capability essential. However, it entails retaining large amounts of data at each decision level when an input utterance is being analysed. The system is implemented mostly in SPITBOL [Dewar 1971], a version of SNOBOL, with the signal processing part written in FORTRAN.

a) Signal Processing:

The utterance is recorded in a normal quiet room on a standard quality tape recorder. This signal is digitized at 20kHz by an A/D converter. All non-silent portions of the tape are then processed by a 256 point Fast Walsh Transform [Clark 1972] to obtain a frequency/intensity spectrum every 1/80th of a second. The silence threshold varies according to its surrounding segments. Thus if one segment is classified as non-silent, the preceeding segment is more likely to be found to be non-silent. (This captures some onsets of non-voiced fricatives which would otherwise be missed).

b) Segmentation and Classification:

Segmentation must occur at two levels, the syllable boundaries must be detected and then the syllable must be internally segmented into different basic sound types.

The syllable boundary detection

algorithm depends mostly on the overall signal intensity. Since vowels are usually more intense than their surrounding consonants, we normally expect vowels to occur at local maxima and to be preceded by a section of significantly rising intensity. We detect these areas of rising intensity and then use various heuristic restrictions to decide the location of the boundaries. This algorithm locates most boundaries, with failures usually only occurring on syllables with extremely reduced stress-levels.

Reflecting the belief that the greater the number of basic sound types one has, the more likely misclassification will occur, we decided to use only four types: vowel, silence, sonorant-consonant and (non-sonorant) consonant. Each type has associated with it a set of measures that are pertinent when distinguishing between the different phones that occur in that type. Adiacent similar segments in the signal are grouped together into a class of one of the above types and the average value and slope of the measures are retained. With each type there is a small set of distinctive features [Jakobson 1963] that must be consistent throughout the segments forming the class. When these features are found to change, a new class Cpossibly same type) is begun.

Using combinations of the classes we can form Stops (consonant, silence, consonant; consonant, consonant; silence, consonant) and Dipthongs (vowel, vowel). Both types of consonants may be labelled transitory or non-transitory depending on the duration and variability of measures in the class.

c) Pattern Forming and Matching

As stated above the basic pattern unit in this system is the syllable. It is felt that this unit provides much greater reliability and freedom from variation than the pseudo-phoneme based systems mainly because there is a considerably larger section of data being compared at one time.

Patterns are formed from example utterances using the above segmentation and classificiation methods. However, to ensure maximum accuracy, all decision points may be monitored and the results modified interactively. The data, the segmenting positions and the classifications can be easily accessed by the user as they are considered, and may be modified with a set of provided procedures. The most common changes made interactively have been for transitory/ non:transitory and short voiced consonant/short sonorant-consonant confusions.

To analyse an input utterance, the signal is first processed and classified as above (but without interaction). This gives an initial view of the utterance which can be used for predictive purposes. The similarity evaluation procedure matches a given word starting at a specified point in the utterance. It returns a similarity measure plus an indication of the ending position of the word. The similarity between a pattern and a section of input is evaluated by first aligning and matching the vowels and then working outwards. Transitory segments do not carry as much weight as the non-transitory ones. In fact, transitory patterns may be totally ignored if the surrounding segments in the pattern and the input are compatible. If, during the match, two different class types are aligned together, we attempt to reclassify the input segments in question. If the reclassification succeeds, we evaluate the similarity. This allows the pattern to evaluate the input in terms of what is expected rather than being forced to compare on some other terms dictated by the preprocessor. Note also that the syllable boundaries in the input do not have to be followed (although they usually are) as we can match a pattern starting at any point in the input.

d) Sound Characteristics: Syllable Features:

There is a great need for quick dictionary retrieval based upon a description of the desired sound, and also for ordering lists of candidate words in order of 'similarity to input' before matching has occurred. Both these requirements are satisfied in our system by using a syllable feature string. This is a string *of* letters which reflects various characteristics of the syllable. Similar sounding syllables should have similar feature strings. As an example, in our preliminary version the feature string consisted of four letters representing the characteristics:

Fricative present/ Not present/ Undeci ded

Sonorant-Consonant present/ Not present/ Undecided

The vowel has the distinctive features

Grave/ Acute/ Undecided

Compact/ Diffuse/ Undecided

Associated with the feature strings is a set of 6 transformations that successively generalize the string until all possible features strings will have 'matched[1] at least one of the transformed

strings. There is considerable room for improvement in the selectivity of the syllable feature string but the above example demonstrates its possibilities. (There are examples of its use in the parsing example later on in the paper).

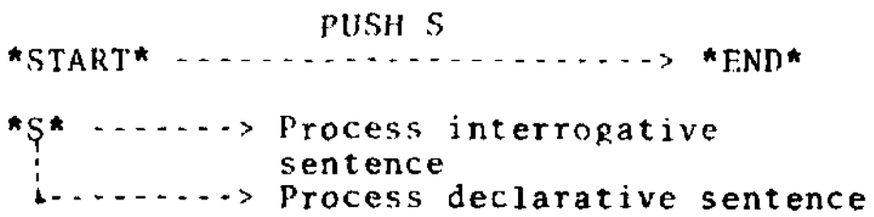
e) The Dictionary

The dictionary is composed of two separate tables. The first is the word table which is indexed by the actual word spelling and contains inflectional information, syntactic class, semantic features and pointers into the syllabic table. The syllable table is indexed by the syllable feature string with a unique suffix to make entries distinct. This table contains the actual patterns for the syllables plus back pointers into the word table. Given a fully or partially specified feature string, one can retrieve quickly all words that have a syllable with the specified features present.

f) The Grammar

The grammar is a fairly standard Augmented Transition Network Grammar as described by Woods [1973] with *one* addition. One can associate with an arc a Reordering function which is invoked if the arc succeeds. This function may reorder the arcs leaving the node that is being entered. This allows dynamic ordering of the arcs where one may use prosodic information from the utterance and also any acquired semantic knowledge. A priority factor may also be associated with an arc at this time which will automatically reduce the parse value if the arc is ever taken. This is best illustrated by an example.

e.g. let the A.T.N. be as follows



We associate a reordering function with the 'PUSH S' edge. As the node *S* is entered the fundamental frequency of the utterance is inspected. Assume it rises in pitch at the end of the utterance, we conclude that a question is the most likely sentence type. The interrogative arc is ordered first and if we decide that the utterance is very definitely a question we may also decrease the priority factor on the other arcs. This mechanism can save much time and wasted effort.

g) The Parser

The disadvantages of written text parsing schemes as applied to spoken input have been reviewed many times [Paxton 1973, Bates 1974] so 1 will not deal with them here. However, one way to eliminate many of the inadequacies is to allow separate paths in the grammar to be pursued independently. This is accomplished in our system by the controlled use of co-routine parsers, each having its own record of the state of the parse and an indication on how well it has matched the utterance so far. The co-routines share all the results from pattern matches on the input, and also share common portions of various data stacks.

F.ach co-routine parser operates in a depth first manner and may suspend at only one point, the place where the decision to accept or reject an arc is made. When they are re-invoked the decision will have been made for them by the controller. A co-routine only suspends when it finds an arc where a word match is acceptable, but the new overall parse value is below a given level. A co-routine returns when it finds a complete parse *or* when it tries to backtrack from the point in the A.T.N from which it was created.

The controller directs the parse strategy by deciding which co-routines *to* invoke and by setting the *acceptance* values for an individual word match and for the overall parse value.

Its initial strategy is to invoke a parse with a very high overall parse value acceptance level. Every time this initial parse suspends, the state is stored and a new co-routine is created. Then, the original parse is re-invoked as though the edge failed. This continues until no more promising paths are found or a complete parse is discovered. If no complete parse is found, we then have a set of co-routine parsers at various stages along the initial parse path where previously rejected edges may be explored. The controller now invokes the co-routine with the highest overall parse value and lets it continue until its value drops below the second highest co-routine level. If there is no co-routine left that looks promising, the controller may create a new parser from a current one by rejecting the last accepted arc and continuing from there. The following example should illustrate the operation of the parsing system.

Example Decoding of a Sentence

This is an example of how the complete parser will decode a sentence. The data was obtained from a real analysis and an extended parse of an utterance by a single co-routine parser under a dummy controller. The complete implementation will present no further

conceptual difficulties and is considered to be imminent.

The example will be kept small so that a full discussion of what is happening may be possible. The sentence to be processed is 'Send me the student* and is processed in a system with a limited vocabulary of 29 words.

The utterance is preprocessed, initially segmented and classified, then this information is passed to the parser controller. The controller arbitrarily sets the minimum acceptable overall parse value to be the high value of 80 and sets the minimum acceptance value for an individual word match to be 50. Co-routine #1 is created and invoked at the starting point in the A.T.N.

Co-routine #1: Sentence parsed so far is ", parse value is?

We enter state S and invoke a reordering function. The fundamental frequency of the utterance is inspected and since the pitch does not rise at the end of the utterance it is decided that an interrogative sentence is very unlikely. The arc leaving state S that corresponds to processing a question is given a very low priority and is ordered last.

We try the arc which accepts a declarative sentence, meaning we must locate an initial noun phrase. A reordering function is invoked as we enter the node *NP* (noun phrase). The stress of the current input syllable is inspected and, since it is found to be stressed, the arc leaving *NP* which processes a determiner is given a low priority. In other words, an initial noun, pronoun or proper noun will be searched for before the determiner.

We try the match for a noun. There are 7 possible candidates. The syllable feature string of the input syllable is 'FYAC' (signifying frication present, sonorant-consonant present and vowel features Acute and Compact). Of the nouns, the first syllable of 'letter* is in closest agreement with respect to the syllable features, while 'course* and 'student' are the next best choices. These nouns are tried first when matching the candidates. No nouns succeed with a similarity score above the required score of 50. We match for a proper noun and find that 'John' succeeds with a value of 63. This is below the accepted level for the overall parse value, so we suspend.

Controller: We save the state of the parse as co-routine #2 and then re-invoke co-routine #1 to continue as though the word 'John' failed.

Co-routine #1: Parsed so far parse value is?

The nouns and proper nouns have failed, we try for a pronoun and finally for the low priority choice of the determiner. Both these possibilities fail and we find that a declarative sentence seems impossible since the initial noun phrase cannot be found.

We try for an imperative sentence. The first word needed is a verb. Of the verbs the word 'send' seems most likely to succeed on the basis of syllable feature similarity. In fact it does match with the value of 82. No other verbs are inspected at present sinee we found such a good correspondence. We proceed to try and find a noun phrase (object or indirect object) following the verb. The reordering function on node *NV* inspects the stress levels and decides that the determiner possibility has the lowest chance of succeeding. Both the noun and proper noun matching fail to produce an acceptable word, but the pronoun 'me' is accepted with a value of 73. This makes the overall parse value 76. (Derived from an average of slightly adjusted scores so that the lower score has more effect). This value is below our acceptable value of 80 so we suspend.

Controller: We save the state of the parse in co-routine #3 and make #1 reject the word 'me'.

A Review of the Co-routine States

#1 has parsed "Send" with value 82
#2 has parsed "John" with value 63
#3 has parsed "Send me" with value 76

Co-routine #1: Parsed is 'Send', value is 82

We continue as though 'me' failed. The last possibility to obtain a noun phrase following the verb is to try for a determiner. The word 'the' succeeds with a value of 67 making the overall parse value 70. We suspend.

Controller: We save the state of the parse as co-routine #4 and re-invoke #1 as though 'the' failed.

Co-routine #1: Parse so far is 'Send', value is 82

Failing to find a noun phrase, and then not finding a suitable alternative for the verb 'send' causes the imperative processing arc to fail. The last alternative is to parse for an interrogative sentence. However, an interrogative sentence was deemed

very unlikely by the initial re-ordering function. When we follow this arc, the parse value is set to the low value of 60. We suspend.

Controller: The co-routines are as follows:

#1 has parsed "" with value 60.
(expecting a question)

#2 has parsed "John" with value 63.

#3 has parsed "Send me" with value 76.

#4 has parsed "Send the" with value 70.

We invoke #3 with the minimum acceptable overall parse value set to 70 (the second highest co-routine value) because we've exhausted all promising paths.

Co-routine #3: Sentence so far is 'Send me', value is 76.

We try for a noun phrase to be the object of 'send'. The stress inspection at this syllable indicates a determiner is most likely, so this arc is given top priority. The word 'the' succeeds with a value of 65 making the overall value now 69. We have dropped below 70 so we suspend.

Controller: We invoke co-routine #4 with a minimum acceptance value of 69.

Co-routine #4: Parsed so far is 'Send the', value is 70.

We try to find an acceptable noun to follow the determiner but none are successful. This causes the co-routine to return to the point from which we were created. We immediately return to the controller.

Controller: Co-routine #4 is destroyed and we invoke #3 with a minimum acceptance value of 63.

Co-routine #3: Parsed so far 'Send me the', value is 69

We try matching for a noun to follow the determiner. The word 'student' succeeds with a value of 67 which makes the overall value drop to 67 also. We have succeeded in finding a complete parse so we return to the controller.

Controller: The parse is accepted for the parse value is greater than any of the other possible paths.

Concluding Remarks and Discussion

It is convenient to break the concluding remarks down into two sections The first section on the lower levels, mainly the pattern matching, and the remaining section on the parsing scheme.

Pattern Matching

Interaction when forming patterns is considered critical to produce accurate patterns that truly reflect the data within a reasonable amount of time. The matching algorithm places considerable emphasis on the slopes of the measured values and their relative positions rather than exclusive interest in the actual physical values. Using the syllable as the matching unit allows this kind of comparison; a comparison which, at the phonemic level, would be too prone to variation. A departure from our basic sound types to a unified description of the syllable would probably provide further independence from the variations that are dominant in smaller unit schemes.

Higher Levels of Processing

As demonstrated to some extent in the example, the controller/co-routine mechanism allows for much of the freedom needed when dealing with spoken data. The separation of the actual individual parsing details from the direction of the overall parsing strategy allows for great versatility and ease of experimentation.

The use of prosodic information by the reordering functions gives much assistance in directing and disambiguating the parsing of the utterance. The advantages to be gained by using prosodies have been discussed in detail by Lea [1974]. This information is one of the few sources of knowledge that spoken text processors may access that is not often available to written text processors (except via punctuation) and we feel it should be exploited to the utmost. The reordering functions also provide an excellent communication interface to semantic and other high level modules which would be present in a full *system.*

Such a semantic module is sadly lacking in our system, all semantic processing is restricted to a few ad hoc functions, A more powerful independent module for the system is open for investigation at the present time. Also, user and dialogue models, which seem to promise great predictive and verification power, have not been developed yet to any significant extent.

One disadvantage of our parsing scheme at the present moment is the restriction of the processing to a left to right mode. Because of this one cannot take full advantage of the 'clearly recognized' words in the utterance. These reliable words could provide considerable predictive power and

guidance when trying to recognize surrounding words. However, it is felt that the general methodology of the controller/co-routine mechanism would be able to incorporate such localized parsing schemes without major structural changes.

As stated at the beginning of the paper, we have so far produced a basis upon which our further research is just beginning to develop.

Acknowledgements

I wish to thank John Mylopoulos, my supervisor, for his encouragement and support during the development of this work. Funding was gratefully received from the National Research Council of Canada.

References

1. Allen, J.F.; "A Prototype Speech Understanding System" M.Sc. Thesis, University of Toronto, 1974. Computer Science Tech. Report #77.

2. Clark, M., Swanson, J., Sanders, J.: "Word Recognition by Means of Walsh Transforms"; Conference on Speech Communication and Processing, 1972. IEEE CH0596-7, Cambridge, Mass.

3. Jakobson, R., Fant, G., Halle, M. ; Preliminaries to Speech Analysis: The Distinctive Features and their Correlates; M.I.T. Press, Cambridge Mass. , 1963.

4. Woods, W.A.; "An Experimental Parsing System for Transition Network Grammars"; in Natural Language Processing, R. RusinfEd.), Algorithmic Press, 1973.

5. Paxton, W., Robinson, A.; "A Parser for a Speech Understanding System"; Third Int. Joint Conference on A.I., Stanford University, August 20-23, 1973, pp.216-223.

6. Bates, M.; "The Use of Syntax in a Speech Understanding System:; IEEE Symposium on Speech Recognition, April 15-19, 1974.

7. Lea, W. ; Prosodic Aids to Speech Recognition IV; Technical Report PX10791, UNIVAC Corp., St. Paul, March 1974.

8. Dewar, R.B.K.; SPITBOL Version 2.0; Illinois Institute of Technology, 1971.