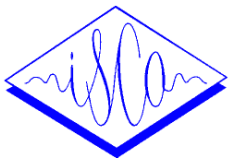


First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

Semantic Weights derived from Syntax-directed Understanding in DTW-based Spoken Language Processing

S. Bornerand, F. Néel and G. Sabah

LIMSI-CNRS B.P.133 91403 ORSAY CEDEX - FRANCE

Abstract

This paper deals with the integration of different levels of knowledge in a continous speech recognition framework. A system has been developed to carry out recognition and understanding processes in parallel. We present a method which allows the system to dynamically compute weights, taking semantic knowledge into account during the recognition process, which is based on a dynamic time warping (DTW) algorithm. The weights change during the course of an utterance recognition process, according to the interpretation of partial recognized sentences.

Nested into the DTW algorithm, a natural language processor (NLP) computes semantic weights. It builds an interpretation using the joining operation of the conceptual graph model. The syntax describes the joining rules of conceptual graphs. Then, we show how a set of joining rules can be represented by a conceptual grammar from which a parser can be generated.

The parsing process evaluates an interpretation of a partially recognized sentence according to a criterion using the overlapping rate of all the conceptual graphs which participate into the interpretation, in order to return a semantic weight.

1. Introduction

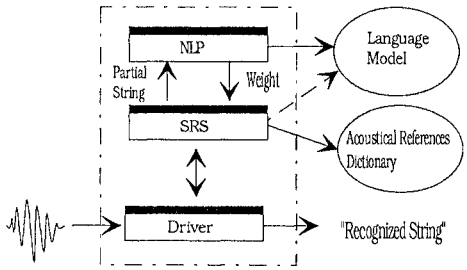
In automatic language processing, it is usual to distinguish spoken and written languages. Many researchers think that the same methods cannot be applied on both spoken and written language. In the papers [13] and [11], the authors present new methods intended to be better adapted to spoken language processing than a classical speech recognition system (SRS) combined with a classical natural language processor (NLP). However, these approaches may overcome the lack of the NLP methods as well as the lack of the SRS methods. The NLP methods are not simply badly-adapted but it is mainly the SRS methods which are not reliable enough for an NLP input. This problem can be addressed from another point of view. If the output of the SRS is not good enough, the NLP may be used as a producer instead of a consumer. So, the natural language processor produces forwards predictions to the speech recognition system. In this way, we can use the NLP methods with the SRS framework.

In natural language processing, although the lexical and syntactic levels seem to be standardized, we notice variations among certain approaches. The same problem arises in speech processing. What is the boundary between the recognition and the understanding processes ? How do we link the acoustical data with the conceptual structures ? This problem has to be addressed in a data exchange and process cooperation point of view.

In this paper, we propose an integrated system which assimilates the predictions of the NLP to weights. The syntax-directed understanding is a NLP method which uses syntactic rules to guide the semantic representation building. A grammar defines the semantics by a set of rules combining conceptual structures. A Conceptual Graph Model provides a compositional operation on conceptual structures which allows the semantic weight computation. For a prototype of the NLP-SRS cooperation, we thought that the DTW algorithm provides a flexible and simple framework which seems sufficient to take the weights into account.

In the next paragraph, we present the architecture of the integrated system. Section 3 shows the modification of the DTW algorithm, oriented toward weight management. Section 4 describes the implementation of a new NLP system based on a context-free grammar and on a conceptual graph model. Section 5 explains the technical aspects of the recognition and understanding cooperation.

2. System Architecture Overview



For each utterance, the driver calls the SRS. The DTW algorithm of the SRS starts with the comparison of words which occur in the first position in the sentences. At the beginning of each word, the SRS calls the NLP in order to build an internal representation of the new sequence from the old sequence and the new word. This process continues until the end of the signal. At each step, a pruning procedure is used to stop the progress of some sequences. At the end of the recognition, the driver gives the word sequence which has an end marker and a minimal score.

3. Recognition Process

The connected word recognition process is based on a dynamic time warping (DTW) algorithm which uses a finite-state network with weighted transitions to encode the syntactic constraints. We present a method which makes it possible to dynamically compute the weights taking semantic knowledge into account during the recognition process. In [7] and [12], the weights are updated between two message processings and they are constant during the

recognition process. In our system, the weights change during the utterance recognition process, according to the syntax-directed understanding of partially recognized sentences. The DTW algorithm has been modified in order to make use of weights in this way [1].

3.1 Principle

The DTW algorithm is a deterministic optimum search method. The algorithm returns an optimal solution which is the recognized word sequence providing a minimal cumulated distance, called G. The optimum criterion used to compute G may be roughly expressed with the following formula:

(1) $G = MIN(\sum_i d_i)$ where d_i is a local distance between two primitive structures of acoustical pattern-matching at the time i.

The use of the semantic weight allows the computation of a score which reaches the best value when the cumulated distance is minimal and the weight is maximal. If a weight is converted into a penalty, a new optimum criterion is obtained and can be used to compute the weighted cumulated distance in the adapted DTW algorithm:

(2) $G' = MIN(\sum_i d_i$ - alpha log WEIGHT).

Therefore, the weights are not taken into account at the end of the recognition in order to find the best solution among a set of candidates. The weights are continuously integrated into the computation of the score.

3.2 Implementation

The DTW algorithm recursively computes the cumulated distance, named g, at each time i of the sampled speech signal. We introduced a cumulated weight, named w, which can be defined by the next local equations corresponding to the optimum criterion (2):

(3) Intra-word
$$\begin{cases} g_{i+1} = MIN_j (g_{i,j} + d_{i+1} + p^n_{i,j}) \\ p^n_{i+1} = p^n_{i,j} \\ w^n_{i+1} = w^n_{i,j} \end{cases}$$

Inter-word
$$\begin{cases} g_{i+1} = g_i + d_{i+1} + p^{n+1}_{i+1} \\ p^{n+1}_{i+1} = pen(w^{n+1}_{i+1}) \\ w^{n+1}_{i+1} = eval(w^n_i, m) \end{cases}$$

avec w^n_i : cumulated weight of an n-word sequence at time i;
d_i: local distance at time i;
g_i: cumulated distance at time i;
p^n_i : penalty for an n-word sequence at time i;
eval(): evaluation function to compute weight;
pen(): conversion function from weight to penalty;
m : new word added to the sequence.

The first equation system, Intra-word, defines the progress of the cumulated values through the elements of the comparison matrix for one word. The index , *j*, indicates the valid path.

The second equation system, Inter-word, defines the computation rules applied at word boundaries.

Our developments come directly from the AMADEUS recognition system designed at LIMSI by J.L Gauvain [3].

3.3 Semantic Weight

The weight is updated after each word sequence modification. The DTW algorithm calls the function, *eval()*, which activates the NLP module with the partially recognized sequence w^n_i and with the new input word *m* parameters. The NLP returns the semantic weight of the new partially recognized sequence, w^{n+1}_{i+1}. The result is converted into penalty with the following equation:

(4) $pen(w^{n+1}_{i+1})$ = - alpha LOG $eval(w^{n+1}_{i+1}, m)$

The factor , *alpha*, is used to adjust the penalty. It can continuously evolve in terms of the difference between the best candidate and the other candidates.

4. Understanding Process

The understanding process uses both syntactic and semantic information to construct a representation of the sentence. The semantic treatment is based on the compositional principle where compositional operations are defined to build the sentence meaning from each lexical term meaning. The syntactic treatment is based on a production rule system. The set of rules gives a guideline to perform compositional operations on conceptual structures associated to lexical terms of a sentence.

We used a conceptual graph model to represent semantic knowledge. A conceptual graph (CG) is a structure composed of concepts and relations. The join operation is the compositional operation on the conceptual structures. It is a partial pattern-matching operator on the conceptual structures. The join operation allows computation of the number of joined concepts.
A rigorous definition and detailed explanation of the conceptual graph model can be found in [10].

We have defined a conceptual grammar [2]. A production rule of this grammar defines a symbol as an operation of the process which builds the conceptual structure. An example of a rule follows:

```
1)groupe_nominal_2: DETERMINANT SUBSTANTIF
2)[GN:$0]-
   <-(CONST)<-[ENTITE:*x1]-
   ->(FORM)->[SUBSTANTIF:$2]-
    ->(ATTR)->[GENRE:*x2]-
     <-(ATTR)<-[GN:$0]
     <-(ATTR)<-[DETERMINANT:$1]-
      ->(ATTR)->[ESPECE:*x3]-
       <-(ATTR)<-[GN:$0]
      ->(ATTR)->[NOMBRE:*x4]-
       <-(ATTR)<-[GN:$0]
       <-(ATTR)<-[SUBSTANTIF:$2]
  ->(ATTR)->[PERSONNE]
3){ if (x4 == "singulier")
    { if (x3 == "définie") x1 = '#';
    }
    else
    { if (x4 == "définie") x1 = '#';
      else x1 = "{*}";
    }
  }
  ;
```

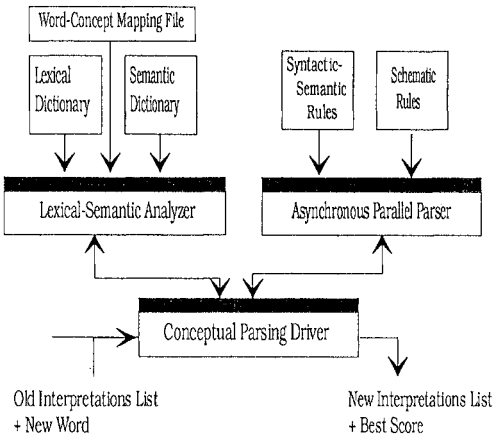
Each rule of the conceptual grammar has three parts. Part One is a standard rule derivation as it appears in a context-free grammar. Part Two is a conceptual graph which defines the schematic rule associated with the symbol *"groupe_nominal_2"* in the above example. The schematic rule provides a guideline to make a join between conceptual structures of each symbol, *"DETERMINANT"* and *"SUBSTANTIF"* hereinabove, in order to obtain the conceptual structure of *"groupe_nominal_2"*, with the following conventions:

- the sign *"$0"* is used to show the head concept of the CG and to attach it to *"groupe_nominal_2"* in order to define the new conceptual structure;
- the sign *"$1"* is used to specify a join between the concept *"[DETERMINANT:$1]"* and the head concept of the conceptual structure of the symbol *"DETERMINANT"*.

Part Three is used to describe the functional dependencies. A variable , *"xi"*, which appears in the reference field of a concept in part two can be used as a C language variable in the code fragment of part three.

The syntax-directed understanding process is the derivation of the conceptual grammar rules. A customized YACC tool [4] is used to create a non-deterministic parser from a conceptual grammar. An asynchronous running mode simultaneously parses several partially recognized sentences delivered by the SRS.

NLP Architecture



There are three types of components:
. the linguistic data bases (lexical and semantic dictionaries);
. the linguistic rule bases (syntactic-semantic and schematic rules);
. the automata.

The conceptual parsing driver receives a list of interpretations and a word. It calls the lexical-semantic analyzer. The analyzer returns a grammatical category and a conceptual graph coming from a join between a lexical graph and a semantic graph. For each interpretation, the driver calls the asynchronous parallel parser in order to pursue the syntax-directed understanding derivation in terms of both the grammatical category and the lexical-semantic graph. The return value is a new interpretation with a score. The conceptual parsing driver returns a new list of interpretations and the best score of them.

5. Processes Integration

The main actors of our spoken language processing system have been presented in the previous part. The aim is now to describe each data structure and specific procedure used by the processes in order to exchange and to share information.

5.1 Implementation Issues

5.1.a Recognizer Interface

The interface between the SRS recognizer and the NLP parser uses a CALL-based synchronisation. Each communication is supported by an eval() function call as shown in 3.3. NLP runs only when DTW requests a weight computation. The NLP processing must be asynchronous due to the difference of the word boundaries in each partially recognized sequence.

Each interpretation makes use of an identifier to avoid redundancy during computation. The DTW algorithm can call NLP many times with the same parameters of the eval() function. The identifier allows verification before NLP runs. In [8], the NLP checks if the parsing configuration already exists in order to avoid tremendous search space expansion. In this system, the identifier indicates a current syntax-directed understanding configuration of a partially recognized sequence.

5.1.b Parser Interface

The driver can run many deterministic parsings in parallel. But, we have seen that the CALL-based synchronisation task is handled by the recognizer. Here, we focus our attention on the weight computation from the join operation on the conceptual structures. The join operation returns the number of joined concepts between two conceptual structures. We introduce a global overlapping rate of conceptual structures which have been joined together. The formula combines the number of concepts of an interpretation and the cumulated number of joined concepts across the parsing of this interpretation, in the following manner:

(5) rate = C / S where
C is the sum of all the values returned by the join operation;
S is the number of concepts of all the structures which participated into an interpretation. When any join operation has been made, the rate is 0 by definition.

The rate is returned to the recognition process as a weight. It is the returned value of the eval() function. The parser also returns a descriptor of the new parsing configuration to identify the interpretation.

5.1.c Driver Interface

In the recognizer interface, we have introduced the interpretation identifier to avoid duplicated parsing of the same partially recognized sequence. The driver manages the data structure which allows storage and identity-checking. A tree-structure is dynamically built up from the finite-state network explanation. It stores all the paths drawn by the DTW algorithm.

Each node structure contains at least both the parsing configuration descriptor, the score of an interpretation and a state of the network. Each node address is returned to the recognition process as an interpretation identifier.

Another important problem is the ambiguity of a word sequence. The driver takes care of the multiple parsing of the same sequence. It substitutes a list of interpretations for a single interpretation in the field of one node with the following rules:
. only the best interpretation yields a penalty that is returned to the recognizer;
. this penalty is subtracted from the others.

To sort the multiple interpretations, we introduce a double pruning. The main pruning is perform by the DTW algorithm adapted to a weighted cumulated distance. The second pruning is performed by the driver. An interpretation is deleted if the weighted cumulated distance plus the rest of its penalty exceeds the pruning threshold.

The second pruning takes place when the recognizer calls the parser before the computation of new interpretations. It is asynchronous.

5.2 Assessments

5.2.a Word-Pair Grammar

Our prototype of the SRS-NLP integration uses a word-pair grammar for several reasons. First, we assume that the perplexity of the word-pair grammar augmented with semantic weights is equivalent to the perplexity of a regular grammar. In this prototype, the speech module uses the word-pairs as basic syntactic constraints. Ungrammatical sequences may be recognized at this stage but the understanding module does not parse them. It provides semantic weights which makes it only possible to recognize a grammatical sequence and to take advantage of the well-formed conceptual structure of a partially recognized sequence.

Secondly, the use of a word-pair grammar is necessary when you want to recognize a large vocabulary with a complex language model. A regular grammar is obsolete to deal with this sort of task because the DTW algorithm has to build up a finite-state network which represents all the derivations of the grammar rules with a large vocabulary. That would require tremendous memory space. The word-pair grammar reduces this space to strictly the memory space needed to load the words.

Thirdly, you can use bi-grams when you put weights in the word-pair grammar. A bi-gram is a stochastic language model because it takes the transition probability of word into account. This extension of the word-pair grammar allows cooperation between a stochastic model and the conceptual grammar underlying a phrase structure grammar. The combined stochastic model and structural grammar improves recognition [9]. The stochastic model is able to predict local constraints at the word level. The conceptual grammar based on structural grammar is able to predict global constraints at the grammatical and conceptual level.

5.2.b Benefit

The system takes advantage of the strong integration of NLP and SRS. All the constraints coming from the syntax-directed understanding process can be applied as soon as possible. The word-pair grammar is powerful and simple. You can automatically learn the rules and the probability from large corpora. The conceptual grammar allows the conceptual structure building up.

5.2.c Drawback

The recognition system uses a speaker-dependent algorithm but the approach remains unchanged for a speaker-independent algorithm. The first difficulty concerns the acoustical unit. The word can not be the acoustical unit with vocabularies larger than 10.000 words. When the unit is smaller than the word, the evaluation function has to be only called at the word level.

The second problem concerns the composition operator in the understanding process. The response time is rather slow due to the complexity of the join operation working on conceptual structure. In the HMM systems [5] or in other systems based on statistical methods [1], the weight computation only uses arithmetic operators. They are more adapted to the numerical computer paradigm.

A last problem appears with the natural language requirements. In facts, the conceptual grammar is not presently able to deal with all natural language phenomena. For instance, our conceptual graph model does not yet support universal quantification.

6. Conclusion

In this paper, we propose a new integration of a recognition process and an understanding process to solve the difficulties of cooperation between the speech and natural language processing.

The recognition module is based on a continuous spoken word recognition algorithm. We adapted the DTW algorithm, using syntactical constraints and weights in order to take evolving semantic predictions into account during the utterance recognition.

The natural language processor is based on an YACC-based parser. We adapted a classic context-free grammar formalism to take into account the conceptual graph model and to guide the join operation on conceptual graphs. The conceptual structures are built up during the utterrance parsing. A conceptual structure is an interpretation of a recognized sequence. An evaluation function has been defined to compute semantic weights from the operations performed on conceptual structures.

A working prototype using a word-pair grammar has been developed to recognize the grammatical sequences and to take advantage of the well-formed conceptual structure of the partial recognized sequences.

Now, the definition of an application may be carried out to show the improvement of the speech recognition quality and to measure the response time in real life experiment with our SRS-NLP prototype.

Acknowledgement

This work was supported by the "GRECO-PRC Communication Homme/Machine", CNRS, France.

References

[1] S. Bornerand, F. Néel, G. Sabah, 1990, "Calcul dynamique de Pondération Sémantique dans un Algorithme DTW", JEP'90, Montréal, Canada, May 28-31, 1990.

[2] S. Bornerand, G. Sabah, 1990, "Syntax-directed Joining for Language Understanding Processing", Proc. of the 5th Annual Workshop on Conceptual Structures, Stockholm, SWEDEN, August 7, 1990.

[3] J.L. Gauvain, 1990, "AMADEUS: Principe et structure", notes et documents LIMSI.

[4] S.C. Johnson, 1975, "YACC: Yet Another Compiler Compiler", in UNIX Programmer's Manual, Bell Laboratories.

[5] K.F. Lee, 1989, "Automatic Speech Recognition: The development of the Sphinx System", Kluwer Academic Publishers, Boston, 1990.

[6] M.E. Lesk, 1975, "Lex - A Lexical Analyzer Generator", in UNIX Programmer's Manual 2B, Bell Laboratories.

[7] A.K. Matrouf, F. Néel, J.L. Gauvain and J. Mariani, 1990, "Adaptating Probability-Transitions in DP-matching Process for a Task-Oriented Dialogue", ICASSP'90.

[8] H. Murveit, R. Moore, 1990, "Integrating Natural Language Constraints into HMM-based Speech Recognition", ICASSP'90.

[9] H. Ney, A. Paeseler, 1988, "Phoneme-based continuous speech recognition results for different language models in the 1000-word SPICOS system", Speech Communication 7, 1988.

[10] J.F. Sowa, 1984, "Conceptual Structures - Information Processing in Mind and Machine", 500 pages, ed. Addison Wesley, 1984.

[11] S.J. Young, N.H. Russell, J.H.S. Thornton, 1988, "Speech Recognition in Vodis II", ICASSP'88.

[12] S.R Young, W.H. Ward, 1988, "Towards habitable systems: Use of world knowledge to dynamically constrain speech recognition", The 2nd Symposium on Advanced Man-Machine Interface Through Spoken Language, November 19-22, 1988, Hawai.

[13] P. Hayes, A. Hauptman, J. Carbonnell and M. Tomita, 1986, " Parsing spoken language, a semantic Caseframe Approach", COLING'86.