Journal of Theoretical and Applied Information Technology
15 May 2012. Vol. 39 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DIALOGUE CONTROL TECHNIQUE BASED ON CONFIDENCE CONFIRMATION ALGORITHM IN AUTOMATIC SPEECH RECOGNITION SYSTEM

[1]Dr.C.P. SUMATHI, [2]V. MEENAKSHI, [3]Dr.T. SANTHANAM
[1] Assoc. Prof., Dept. of Comp.Sci., S.D.N.B. Vaishnav College for Women, Chennai-33, TN, India
[2] Asst. Prof., Dept. of Comp.Sci., Govt. Arts College [Autonomous], Salem-7, TN, India
[2] Assoc. Prof., Dept. of Comp.Sci., D.G.Vaishnav College, Chennai-33, TN, India
Email : [1]santsum@hotmail.com, [2]vaimeena@yahoo.co.in

ABSTRACT

Over the recent years, speech recognition technology has been making steady and significant progress. Together with advances in robust parsing techniques, natural language generation algorithms, and the advent of high-quality speech synthesis systems, it has paved the way for the emergence of robust spoken dialog systems. This paper has focused on the process involved in determining what the user has intended at each dialogue turn in a mixed initiative dialogue, conditioned on a recognizer word graph with associated word confidence scores. The dialogue component directly influences the initial selection process at least whenever it has provided a specific context. The dialogue manager is the one component that has not only local information from each server, but also global knowledge about a particular user's constraints. In this paper, the Confidence Confirmation Algorithm in the selection of recognition hypotheses in the context of human-machine interactions is described. Enhancements also made to other human language technology servers for the purpose of providing useful information to the dialogue manager, as well as new capabilities in the dialogue manager itself aimed at detecting and repairing problematic spots in the dialogue.

Keywords: *Speech Recognition System, dialog control, confidence confirmation algorithm, Hidden Markov Model.*

1. INTRODUCTION

Speech recognition systems generally rank order hypotheses by computing scores for utterance hypotheses. These scores are useful for preference ordering the hypotheses, but do not give a good indication of the quality of the recognition or how confident the system is that the decoding is correct. For applications to act on speech input, they must be able to assess the confidence that the input has been decoded correctly. This work combines and extends the work described in [1], [2], and is related to extending one feature of [3] for providing confidence annotation of speech recognizer output. The idea is to normalize decoded word strings and phone acoustic scores by scores produced by a less constrained search. [1] used an all-phone recognition to normalize the scores of the hypotheses, followed by Bayesian updating. Among other things, [3] also used the best matching observation for each frame (senone) to normalize the acoustic score for the hypothesis. For acoustic measure, 10ms frame-level observation score is used as the basis for the normalization. Recognizer of Sphinx-4 system [4] is used as speech recognizer. It is a Semi-Continuous HMM recognizer using a trigram language model. Acoustic observations are modeled in this system by senones [5]. Senones are tied hmm-state specific mixture weights for the Gaussian distributions used by the semi-continuous HMM system. In a speech recognition system, one of the most difficult aspects is to assure that the system understood correctly each user query, or, if not, that the system is able to recover gracefully and efficiently from the errors.

A tedious though effective strategy is to prompt the user at each turn, soliciting only one piece of information, subsequently verifying through a confirmation sub-dialogue that it has been correctly understood. A more natural interface would allow the user much greater freedom, but at the price of significantly higher perplexity. In such a mixed-initiative system, it becomes important to draw on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

as many constraints as possible to aid in the hypothesis selection task. Explicit confirmation can yield greater confidence in the validity of hypothesized utterances, but, again, at the risk of increased tediousness. The problem of utterance-level confidence annotation[6] can naturally be cast as a machine learning classification task: given the current user utterance, select a set of relevant features, and use them to classify the utterance as correctly understood or not. This paper discusses how the ABE (Airline Back End) database deals with the issues of hypothesis selection and verification. It utilizes a mixed initiative dialogue strategy supported by confirmation sub-dialogues that are invoked only when the system actively suspects the miscommunication.

This system poses interesting and challenging problems for dialogue systems in that the interaction is complex and involves multiple variables. Once these variables are specified, users can become quite confused and the dialogue can be derailed if a serious misrecognition occurs. In the remainder of the paper, both the hypothesis selection process and the method that is used to control dialogue management is described. Next the confidence confirmation algorithm, which as a policy only confirms when it detects an unexpected response from the user is also described.

2. SYSTEM DESCRIPTION

The system performs a variety of domain-specific functions and in some sense the "application " that the dialog system interfaces to. The functions include access to information in the system database, retrieval of information on the web and domain-specific reasoning. The interfaces to web-based resources to obtain information about flights. Information includes schedules and prices for flights and locations, prices. This system incorporates domain-specific reasoning to deal with, for example, the resolution of ambiguous references and managing solution sets (for example, ranking flights on "desirability"). The system interacts with the database, which contains geographical information (about 500 destinations) and information about airlines. The database also contains information about how users might refer to various entities in the domain (for example airport names) and information about how the system should in turn refer entities when speaking to the user.

Base Architecture

The architecture behind this travel information system is Sphinx-4 architecture. The main components of Sphinx-4 work together during recognition process. When the recognizer starts up, it constructs the front end (which generates features from speech), the decoder, and the linguist (which generates the search graph) according to the configuration specified by the user. These components will in turn construct their own subcomponents. For example, the linguist will construct the acoustic model, the dictionary, and the language model. It will use the knowledge from these three components to construct a search graph that is appropriate for the task. The decoder will construct the search manager, which in turn constructs the scorer, the pruner, and the active list.

3. CONFIEDNCE CONFIRMATION ALGORITHM

Confidence Confirmation Algorithm is mainly concerned with the hypothesis selection process which is a complex process that involves several steps, including interactions among multiple servers. This process is represented by the algorithm given below.

"Confidence Confirmation Algorithm"

Step 1. Find the word graph representing multiple sentence hypotheses, with associated confidence scores for each word in the graph.

Step 2. Produce an N-best list of semantic frames, capturing alternative candidates for the meaning of the utterance.

Step 3. Get the most promising of these frames, taking into account possible discourse context, and present this candidate to the dialogue manager .

Step 4. The dialogue manager then decides whether this request is consistent with the prior dialogue. If some part of the query is problematic, it may do one of the several things:

i. Ask the user for explicit confirmation
ii. Seek and alternative hypothesis from the N-best list, that may be more appropriate pragmatically,
iii. Reject (delete) certain attributes that are both pragmatically inappropriate and poorly scoring,
iv. Initiate a sub-dialogue asking for confirmation.
v. Ask the user to keypad in the information.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Step 5. Pass on the information to the recognizer once again to get the confidence scores for each word.

Step 6. Stop.

The recognizer processes the recorded user waveform and produces a word graph with associated confidence scores for each word in the graph [7]. The confidence scores are based mainly on the log likelihood probabilities of the words, obtained from the acoustic models for their component phones. The confidence scores are obtained from a set of features that are combined into a single score using linear discriminant techniques[8]. In addition to the mean and minimum log likelihood score of the word in all of its possible local alignments, the combined score takes into account also the difference between the word's score and the best score obtainable over the same acoustic space, and also against the score of a "catch-all" model[9]. The number of competitors for the acoustic region is also taken into account. Fig.1 represents the block diagram for hypothesis selection and verification.

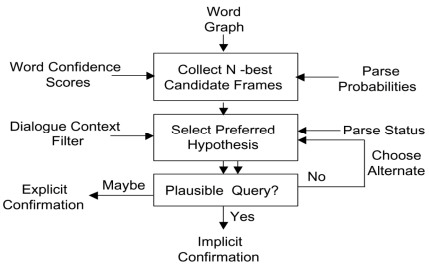

Figure.1 Block diagram for hypothesis selection and verification

The first step in hypothesis selection is to parse the recognizer's word graph into a set of candidate semantic frames. This is done with natural language system which parses from a context free grammar augmented with feature unification and a trace mechanism for movement. Acoustic and linguistic scores are combined to give an overall sentence score. In addition to the total combined score for each hypothesis, critical content words (e.g., cities and dates) retain their confidence score associated with the corresponding element in the semantic frame, for possible later consideration by the dialogue manager.

Each candidate semantic frame is also labeled according to its parse status, with one of four possible categories. "full parse", "robust parse", "phrase parse", and "no parse". "Full parse" means that a single coherent parse tree accounted for every word in the hypothesis. "Robust parse" means that every word was accounted for, but the parse structure consists of a sequence of parsed fragments with possibly interspersed licensed "skip words". "Phrase spot" means that large parts of the hypothesis may have been totally ignored, but certain critical, high scoring, content words were singled out for parsing. Even with all of these back-off mechanism, it is still the case that some user utterances are unparsable. The dialogue manager is responsible for providing a context-dependent response for the "no parse" category.

The next step is to use a simple heuristic to select the most promising candidate form the set of parsed frames. In the absence of any directives from the dialogue component, the system simply chooses the highest scoring full-parses theory, backing off to robust-parse, and finally phrase-spotting. However, it is often the case that the dialogue component has set up context conditions that will preferentially favor an otherwise sub-optimal theory. This can include a list of one or more semantic categories that are in focus and in some cases, individual words that are highlighted, or individual words that are to be selected against. For example, if the system has just asked the user for a return date, then all dates are given preferential treatment. Similarly, if it has just listed the cities it knows in a particular place, those cities will be highlighted.

Once the most promising hypothesis has been singled out, it is processed through context resolution and delivered to the dialogue manager for consideration. If all goes well, the new information is interpreted and a response is prepared that moves the dialogue plan closer to a conclusion. The alternate hypotheses are retained, but utilized only when there is reason to believe the selected hypothesis is erroneous.

4. DIALOGUE CONTROL

The dialogue manager is tasked with the difficult responsibility of determining how best to answer each user's query. With each turn, it processes the user's query, represented as a semantic frame, and prepares its meaning response, also represented as a semantic frame. The generation component converts the reply frame into a well formed reply string, to be spoken back to the user. The dialogue control is managed through the use of a dialogue control table. This table is a simple device for managing complexity. It enforces a linear organization of the complex planning tasks of dialogue management, and provides a high-level representation of dialogue activities in an outline

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

form. The table takes the form of a set of rules, specifying functions to be called when specified conditions are met. The conditions are tests (boolean, arithmetic, string match, etc.) on variables maintained in a dynamic dialogue state frame. The variables are initialized from the user's query (in context), and are augmented in the course of a dialogue turn by the various functions that are executed. It is up to the system developer to partition the dialogue tasks into a set of specific functions, and to choreograph the order in which, and conditions under which each function should be called. Ideally, each function has a very specific role, some having to do with verifying that the query is fully specified, others involved with retrieving the information from the database and still others involved with preparing the reply frame. A selected subset of the rules concerned with managing dates in shown in Table 1.

Table 1. Dialogue Control Table Connecting Dates

Week/Day/Rel.Date	ResolveRelativeDate
ReturenDate:Date	CheckInvalidDate
Hyplist&RejectedDate	SelectAlternateDate
RequestDateConfirmation	PromptDateConfirm
ConfirmDateDeny	RequestKeypadDate

Specific Knowledge Sources

In order to inform hypothesis selection at any point in the dialogue, several knowledge sources that are maintained and updated continually throughout the user's conversation with the system are used. The dialogue state is, of course one of the most useful of these knowledge sources. The dialogue state encodes parts of both sides of the conversation, in that it identifies any preceding system-initiated query as well as all user-specified constraints. The dialogue state also contains information on how far the user has come in the overall travel plan, which is helpful in determining if a particular dialogue move is likely. The system also retains in history a user model which is continually augmented as the dialogue progresses through the itinerary plan. It includes any as yet enforceable, such as an early specification of the return date or the mention of a desired fare class before the itinerary is completed. It also includes the particular details of the selected partial itinerary, which are useful for applying date and source constraints to later legs. In addition, a set of frame is maintained for alternative recognizer hypotheses of dates, for possible later reselection. If the original date is pragmatically implausible, missing data recognition [10] alternative dates are selected from this set. The alternatives are flushed each time a new leg is selected.

A detailed record-keeping mechanism for tracking source and destination throughout a dialogue was made and it was found these two keys became especially problematic, particularly in cases where the user may be attempting to travel to or from a city that is not in the recognizer's vocabulary. In these cases, the same misrecognition tends to occur repeatedly, as the recognizer continues to substitute the same incorrect hypothesis for the intended city, or the source/destination in question varies from query to query, as the recognizer hypothesizes different cities within its known vocabulary. By monitoring the patterns of source/destination keys from query to query, user will be able to decide when to prompt for verification or to solicit keypad input.

```
{c city history
      :source "MAA"  :source_status "inherited"
      :dest "BOM"  :dest_status "changed"
      :history {c city_history
         :source "MAA" :source_status "inherited"
         :dest "BOM"   :dest_status"repeated"
        :history {c city_history
                  :source "MAA"
                  :source_status  "first"
                  :dest "BOM"
                  :dest_status  "first" }}}
```

Figure.2. Example frame to represent city history.

Here the source is inherited in the last two utterances, whereas the destination was apparently first repeated and then changed. Note: MAA to represent Chennai International Airport and BOM to represent Mumbai International Airport.

Each source and destination city is entered into this history throughout the course of a single dialogue. This history is updated for each turn in which these values are present, either from the user utterance or from inheritance.

S	Welcome …Please login saying your login name those who already had an account Else say 'New User'
U	Kavitha
S	Please say the password…
U	August first[misrecognized]
S	Sorry. I didn't catch it. Please enter your password using softkeyboard
U	[Successful Enrollment]

Figure.3. Dialogue with the user involving logging on

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A status is stored along with the city, indicating whether the city was newly introduced in that turn, changed, repeated, or inherited from a previous turn. The record is stored in a nested frame structure, as illustrated in Fig.2. Dialogue where keypad entry is successful for enrolling the password during the logging on stage is depicted in Fig.3. For each query, containing source or destination keys, this record is consulted to determine if the values are consistent with what has appeared before in the dialogue. The city history is flushed wherever the flight is selected for the itinerary.

5. CONCLUSION

In a speech recognition system, although each separate server can be developed and trained on its own, it must function as part of an entire system, and do so in the context of a complex dialogue with a human user. The dialogue manager is the one component that has not only local information from each server, but also global knowledge about a particular user's constraints. In this paper, the confidence confirmation algorithm is described in the selection of recognition hypotheses in the context of human-machine interactions. enhancements is to be made to other human language technology servers for the purpose of providing useful information to the dialogue manager, as well as new capabilities in the dialogue manager itself aimed at detecting and repairing problematic spots in the dialogue. The dialogue component directly influences the initial selection process, at least whenever it has provided a specific context. While a set of N-best semantic frames is produced, most of the attention is directed towards the primary selected candidate. After perusal, several problematic situations trigger a response that involves confirmation requests and/or help messages. Sometimes components of the frame are ignored, either because the system can find no appropriate interpretation for them, they have low confidence scores, and/or they conflict with other information present in the same frame. The general strategy is to invoke confirmation sub-dialogues only when the user appears to make a surprise move. Similarly, alternative hypotheses are only considered when the top hypothesis leads to pragmatically implausible outcomes.

REFERENCES:

[1] Young, S. and Ward, W., Recognition Confidence Measures for Spontaneous Spoken Dialog, EUROSPEECH'93, September 1993.

[2] Chase, L., Rosenfeld, R., and Ward, W., Error-Responsive Modifications to Speech Recognizers:Negative N-grams, ICSLP 1994.

[3] Chase,L., Error-Responsive feed back Mechanisms for Speech Recognizers, Unpublished Ph.D. Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, April, 1997.

[4] Ravishankar, M.K., Efficient Algorithms for Speech Recognition, Unpublished Ph.D. Dissertation,School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 1996.

[5] Hwang, M. Y. and Huang, X. D., Subphonetic Modeling With Markov States - Senone, ICASSP'92, March 1992.

[6] Dan Bohus and Alex Rudnicky, Integrating Multiple Knowledge Sources for Utterance-Level Cofidence Annotation in the CMU Communicator Spoken Dialog System November 2002

[7] Hazen,T. Burianek.T, Polifroni.J. and Seneff.S., Integrating Recognition and Confidence Scoring with Language Understanding and Dialogue Modelling", Proc. ICSLP-2000.pp.1042-1045, Beijing, China, Oct., 2000.

[8] Marcel Katz, Hans - Günter Meier, Hans Dolfing, Dietrich Klakow, "Robustness of Linear Discriminant Analysis in Automatic Speech Recognition," Pattern Recognition, International Conference on, vol. 3, pp. 30371, 16th International Conference on Pattern Recognition (ICPR'02) - Volume 3, 2002.

[9] Akoyl.A., Erdogan. H., Filler model based confidence measures for spoken dialogue systems: a case study for Turkis, Acoustics,Speech and Signal Processing 2004 Proceedings(ICASSP'04) May 2004.

[10] Christophe Cerisara, Exploiting confidence measures for missing data speech recognition, Proceedings on Acoustics'08,Loria Umr 7503 – France, Dec2008.