Integrating Multiple Knowledge Sources For Improved Speech Understanding

Sherif Abdou, Michael Scordilis
Department of Electrical and Computer Engineering, University of Miami
Coral Gables, Florida 33124, U.S.A.

Abstract

In spoken dialog systems it is often the case that the sentence produced by the decoder with the highest recognition probability may not be the best choice for extracting the intended concepts. Lower ranking hypotheses may present better alternatives. In this paper, we show how to integrate multiple knowledge sources for the decision of selecting one of these hypotheses. A scoring schema combining information from the recognizer output, the parser, an utterance type classifier and dialog context is used. The scaling weights of the combined scores are determined automatically by an optimization procedure. Finally, we show the results of testing this approach and its performance compared to the approach of selecting the best recognition hypothesis.

1. Introduction and Review

A robust and user friendly dialog manager for spoken dialog systems aims at selecting the most probable interpretation for the user input. This process requires the integration of as many available knowledge sources in the process as possible. Such systems must be designed to work in an acceptable manner under the general constraints of imperfect speech recognition and spontaneous speech phenomena.

To improve the understanding of the user input, there exist many other knowledge sources besides the syntax and semantic knowledge that can be modeled and used. Typically, these are clustered into the categories of pragmatic and prosodic knowledge. Pragmatic knowledge includes inferring and tracking user plans, using context across clause and sentence boundaries, determining local and global constraints on utterances and dealing with definite and pronominal references. Prosodic knowledge includes accentuation, boundary tones and intonation modes.

Work in natural language processing has shown that these knowledge sources are important for the understanding process. Fink and Biermann [1] use historical information about a set of past dialogs to infer a general dialog structure and use it to predict the meaning of utterances in the current dialog. Mudler and Paulus [2] interleave the generation and verification of expectations. Once a partial recognition of a sentence has occurred, subsequent expectations based on this partial recognition are generated and used in recognizing the next portion of the utterance. In the MINDS system [3] reduced word error rates were achieved by utilizing a dialogue structure that could track the active goals, topics and user knowledge likely in a given dialog state. That knowledge is used to dynamically create a semantic case-frame network, whose transitions were utilized in generating the predictions used to constrain the word sequences allowed by the recognizer. These approaches are characterized as early disambiguation and generate restrictions for some partial hypotheses that prevent them from further investigation. This approach is computationally expensive as all the higher knowledge sources should propagate down to the word level.

In contrast, late disambiguation uses knowledge to either correct misrecognitions or to select among alternate hypothesis. Smith and Hipp [4] used dialog expectations and a Minimum Distance Parser. When faced with input that is extra-grammatical, the goal is to find a corresponding grammatical hypothesis that is closest in meaning to the given input and also have the minimum expectation cost. They used a small vocabulary and processed a single recognition hypothesis.

There have been some efforts to include dialog predictions in the n-gram language model, based on the notation of certain regularities in syntax and lexical distributions with different dialog states. Popović et al [5] defined the dialogue state by the system question presented to the user. Using this simple approach, the language model training corpus is split according to such dialog states and a separate language model for each dialog state is then trained. To overcome the problem of limited amount of training material, Wessel et al [6] proposed a linear interpolation of all dialog-state dependent language models and a global language model. They used the global context-independent model as a back-off. If user doesn't act as expected, this approach may lead to degradation rather than improvement in recognition rate.

Using prosody in speech understanding, Ostendrof [7] achieved a correct selection among pairs of ambiguous parses in 75% of the cases by using prosodic boundary and accent information. This approach requires a complete parse for the sentence which, for spontaneous speech, is not available most of the time. In the Verbmobile project [8] prosodic scoring was used for the word graph output of the speech recognition to structure the turns into prosodic phrases. This resulted in the improvement of both accuracy and speed of the parse result. The prosodic knowledge was used only at the utterance level.

In this paper, we integrate multiple knowledge sources and use them to select the best hypothesis for the user input. By permitting the dialog manager to guide the interaction it is possible to track the dialog state and thus estimate the expected semantic content of the user’s response. In addition, by using features of the intonation (F0) contour, it is possible to classify the utterance type as a statement or a question. The language understanding module, the parser, is allowed to process a large number of sentences provided by the decoder. By matching the semantic content of each hypothesis with dialog predictions and utterance type classification we aim to calculate an expectation score and a prosodic score. In combination with the recognition and parsing scores a weighted sum of the four scores can be computed for each hypothesis. The weighting factors are calculated by a learning algorithm to get the optimal values that are pertinent to a training data.
In the following part, section 2 includes the discourse model used for generating a prediction score. Section 3 includes the utterance type prosodic classifier. Section 4 describes the training algorithm used for weights computation and section 5 includes experimental results. The final conclusion and prospects for future work are included in section 6.

2. Generation of Dialog Expectations

Predictions are derived from what we know about the current state of the dialog. In some sense, predictions cover everything we expect to happen, and exclude events which are unlikely to happen. To be able to create predictions three data structures are used:

1) Knowledge base of domain concepts, which represent all objects and their attributes in the domain. We used the standard frame-based representation [9] which provides the capability to express inheritance and multiple relations between frames. Each utterance will be parsed into a combination of domain concepts, which constitute the meaning of that utterance.

2) Hierarchical plan trees, which represent a hierarchy of all possible abstract goals the user may have during the dialog. Plan trees are composed of individual sub-plan nodes, structured as AND-OR trees. Each plan node is characterized by possible sub-plans it can be decomposed into, and a set of domain concepts involved in trying to achieve this plan. The concepts associated with a plan tend to be restricted from previous dialog context. These restrictions on concept expansions are dynamically computed for each concept. The plan trees also define the traversal options available to the user.

3) The focus stack, which is used to keep track of currently active plans and subplans with their associated concepts. It is essential for processing the clarification subdialogs.

Our discourse model is based on Litman and Allen model [10] but customized by removing knowledge intensive details not needed for the coverage of information retrieval applications that represent the focus of our research.

In the analysis phase inference is made for each input user hypothesis to determine how it relates to the current dialog state. Two possibilities exist. In the first, the hypothesis represents a straightforward continuation of the domain plan, i.e. it represents a step to achieve the goal of the plan, and the system can proceed to the next sub-goal. In the second case, the user response initiates a new discourse plan which indicate the initiation of a clarification, correction, or meta-communication subdialogs. The scoring scheme for both plan types is as follows:

2.1. Domain Plans

First, we try to determine which plan steps are targeted by the present interaction. During one interaction cycle, several plan steps may be completed and many new plan steps may be initiated. Goals that have just been completed by this interaction and that are consistent with previous plan steps, located on the top of the focus stack, receive the highest score.

If the current goal doesn't match, we identify the next plan steps to which a user could possibly transit to. This may be the next “sibling” to the current step or another “child” of a “parent” or “grandparent” node, which represents a suspended plan so they receive lower scores. The overall score for a hypothesis is the sum of all the scores of the goals satisfied by that hypothesis.

2.2. Discourse plans

In real-world applications, speakers frequently diverge from the task at hand by initiating various types of subdialogs in order to digress and clarify. Discourse plans are domain independent and only a limited number of them can be executed at each point in the interaction. Litman and Allen [10] introduced the notation of discourse plans describing these digressions. In our model we included three types of discourse plans:

a) Clarifications: They represent user-initiated subdialogs, usually by questioning, to ask about some feature for a concept related to one of the current plans of the focus stack.

b) Corrections: They represent user-initiated subdialogs with the intention to correct part of an already constructed plan. They usually appear after system-explicit or implicit confirmations.

c) Meta_communications: They represent user-initiated subdialogs that refer to the dialogue itself, such as asking for repetitions or signaling nonunderstanding.

Discourse plans receive higher score than domain plans. This is done to reflect the higher priority of discourse plans than domain task plans. Discourse plans are used to control the flow of the dialog and missing them can lead to extra recovery turns or may be lead to a complete failure of the dialog. The absolute values for the scores is not as much important as their relative values, as they are scaled by weighting factors in the final score.

3. Utterance Type Classifier

Some researchers [14] claim that prosodic information is redundant for speech understanding and it just add intelligibility to the speech that may reduce the cognitive load of the listener. In automated dialog systems, with the unavoidable recognition and understanding errors, prosody may not be redundant but a required knowledge. Figure (1) includes an example of one of the cases from our data. Both of the two hypotheses received nearly equal recognition score and exactly the same parsing score. Output (2) was correctly selected by involving prosody.

The main target of this task is the classification of an utterance as statement or question using only acoustic features. It works in a parallel path with the speech recognizer. For the classifier we used CART_ style classification trees [11] as it can be inspected to gain an understanding of the contribution of the different used features. We used 10 features to construct the tree. All the features were extracted from the F0 contour, which was calculated based on algorithm by Veprek [12], with some preprocessing enhancements. Our decision to base all the used feature set only on the pitch contour was
inspired by literature results reported in [13] were listeners do the same classification task by just listening to F0 contours. The used features are:

- F0 range: the difference between the maximum and minimum F0 values within the utterance.
- F0 st_dev: standard deviation for the F0 values.
- End slope: slope of the regression line for the last 200 ms of the utterance.
- Penultimate slope: the slope of the regression line for the 200 ms before the end region.
- Top line slope: the slope of a regression line for the local maximum points.
- Bottom line slope: the slope of a regression line for the local minimum points.
- Register shape: a discrete value (1, -1) representing the shape of the top and bottom lines as converging or diverging.
- F0 difference: the absolute difference between the mean F0 of the end region and the mean F0 of the whole utterance.
- F0 ratio: the ratio between the mean F0 of the end region and the mean F0 of the whole utterance.
- F0 pen difference: the absolute difference between the mean F0 of the end region and the mean F0 of the penultimate region.
- F0 pen ratio: the ratio between mean F0 of the end region and mean F0 of the penultimate region.

For the tree training, 1850 utterances was selected from ATIS3 training set, with equal number of samples for each class. The training algorithm used a tenfold cross-validation procedure to avoid overfilling the training data. We selected another 260 utterances from ATIS3 testing set. For the test set the constructed tree yields an average accuracy of 81%. The most queried features were End_slope, Register_shape and F0_range, they constituted 70% of the tree questions.

Utterance transcription:
and what's the first flight in the morning
Recognizer output(1):
and with the first flight in the morning
Parser output(1):
Flight_Reservation:[Flight_Reference]{
WITH THE [Earliest](FIRST FLIGHT IN THE
[Time_Range]( [Time_spec](
[Period_Of_Day]( MORNING ) ) ) ) )
 recognizer output(2):
I'd what's the first flight in the morning
Parser output (2):
Flight_Reservation:[Request]{[Wh_form]{
WHAT'S [Flight Reference](THE [Earliest]
( FIRST FLIGHT IN THE [Time_Range](
[Time_Spec]( [Period_Of_Day]( MORNING )
) ) ) ) ) )

Figure 1. example for prosody usage in sentence selection

To translate the classification output of the utterance type classifier to a knowledge source, we use a simple heuristic for Syntax-Form matching. If the utterance is classified as a question and one of the hypotheses has a parse result that includes one of the questioning forms then we give a positive score to that hypothesis. Conversely, if the utterance is classified as a statement and one of the hypotheses has a parse result that include one of the questioning forms then we give a negative score to that hypothesis. All other cases receive a zero score.

4. Computing Weighting Factors

Besides the two knowledge sources described above, we also used a recognition score that is generated from the recognition engine for each hypothesis, and an understanding score generated from the parser that penalizes skipped words and fragmentation.

When we first implemented a disambiguation approach by integrating these four scores, initial set of weighting factors was chosen manually. This was done after analyzing some examples in order to tune the performance of each score to the desired behavior. To avoid the effort of manually tuning these weights we decided to switch to an automatic computation method from training data. This allows the retraining of the weights whenever more data are available and also when porting the system to a new application.

Weights training was achieved using "Least Squares Minimization/ Hill Climbing" algorithm that was introduced by Alshawi and Carter [15]. The main idea is that, an initial set of weights is calculated by minimizing the error function $E$, where

$$E = \sum_i (G_i - \sum_j W_j S_{ij})^2.$$ 

Here, $G_i$ is a training score selected manually for every sample of the training set to direct the search to the best hypothesis, $W_j$ is the scaling weight for knowledge source $j$, and $S_{ij}$ is the score of source $j$ when applied on sample $i$.

Setting the derivatives of $E$ with respect to $W$s to zero, we get a system of linear equations that are solved to get initial values for the weights. These initial values are adjusted by an iterative hill-climbing algorithm which at each iteration tries to find the weight change that achieves the maximum improvement in the scoring result (i.e., that maximizes the number of correctly selected hypotheses in the training data).

5. Experimental Results

To test the effect of using the integrated knowledge sources approach for selecting the best hypothesis, we made an experiment with two spoken dialog systems. One was a baseline system that just took the top candidate of the recognizer output and considered it for further processing. The other system used a selection strategy that depended on the proposed approach. We made ten scenarios for a travel-planning task. Nine were detailed scenarios and included the requirement to book a flight between two cities in specific dates with car and hotel arrangements. The last scenario was an open one. It just included the requirement for a business trip and the user was free to make his own plan. We collected 160 dialogues, from 8 subjects. Users were presented with a list of 20 dialog tasks, comprising the ten dialog scenarios repeated twice and shuffled in a random order. We made a hidden switch between the two systems after every successive trial. This experimental setting randomized the usage of the
scenarios between the two systems and the order in which they are presented to the subjects as much as possible. This avoids any bias that may result from the experience the user may get from using a scenario for one system before the other. The output from all the system modules is logged for subsequent analysis.

As a performance measure we used the percentage of correctly selected hypotheses which were manually counted by an analyst after judgment based on the transcription of user input and system log of hypotheses lists and calculated scores. The same measure was calculated after excluding cumulative errors, which build up when at a certain point an incorrect hypothesis is selected and incorporated into the dialog context, causing future dialog predictions scores to be inaccurate. Table 1 summarizes the results of the experiment.

Table 1. Comparison of Systems Performance

<table>
<thead>
<tr>
<th>Source</th>
<th>Testing with cumulative errors</th>
<th>Testing without cumulative errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline system</td>
<td>65%</td>
<td>69%</td>
</tr>
<tr>
<td>Proposed System</td>
<td>71%</td>
<td>78%</td>
</tr>
</tbody>
</table>

Future work will expand the prosodic score to include the cases of declarative questions, which have the same syntax form as statements. Also because of the relative reliability of our scores is not consistent across all the cases. A non-linear combination of the scores will be investigated for improved performance.

7. References