CHAT To Your Destination

Fuliang Weng[1] Baoshi Yan[1] Zhe Feng[1] Florin Ratiu[2] Madhuri Raya[1] Brian Lathrop[3]
Annie Lien[1] Sebastian Varges[2] Rohit Mishra[3] Feng Lin[1] Matthew Purver[2] Harry Bratt[4]
Yao Meng[1] Stanley Peters[2] Tobias Scheideck[1] Badri Raghunathan[1] Zhaoxia Zhang[3]
[1]Research and Technology Center, Robert Bosch LLC, Palo Alto, California, USA
[2]Center for the Study of Language and Information, Stanford University, California, USA
[3]Electronics Research Lab, Volkswagen of America, Palo Alto, California, USA
[4]Speech Technology and Research Lab, SRI International, Menlo Park, California, USA

{fuliang.weng,baoshi.yan,zhe.feng,madhuri.raya}@us.bosch.com
{varges,fratiu,mpurver,peters}@csli.Stanford.edu
{brian.lathrop,rohit.mishra}@vw.com
{harry}@speech.sri.com

Abstract

In the past few years, we have been developing a robust, wide-coverage, and cognitive load-sensitive spoken dialog interface, CHAT (Conversational Helper for Automotive Tasks). New progress has been made to address issues related to dynamic and attention-demanding environments, such as driving. Specifically, we try to address imperfect input and imperfect memory issues through robust understanding, knowledge-based interpretation, flexible dialog management, sensible information communication, and user-adaptive responses. In addition to the MP3 player and restaurant finder applications reported in previous publications, a third domain, navigation, has been developed, where one has to deal with dynamic information, domain switch, and error recovery. Evaluation in the new domain has shown a good degree of success: including high task completion rate, dialog efficiency, and improved user experience.

1 Introduction

In the past few years, we have been developing a robust, wide-coverage, and cognitive load-sensitive spoken dialog interface CHAT under a joint NIST ATP project with Bosch RTC, CSLI of Stanford University, ERL of VW of America, and STAR lab of SRI International. The CHAT system is specifically designed to address imperfect speech and imperfect memory of human users, when they use the system to interact with devices and receive services while performing other tasks—typically, these tasks are their primary, and sometimes even critical tasks, such as driving.

Examples of imperfect speech are speech disfluencies, incomplete references to proper names, and phrase fragments, while examples of imperfect memory include very limited number of names memorized or non-exact names memorized. Imperfect speech and memory happen quite often. In one reported Wizard-Of-Oz experiment for the restaurant finder domain [Weng et al 2006], 29% of the proper names used by people were partial names. The imperfect speech and memory issues accompanied with multi-tasking pose a big challenge to the development of a robust dialog system. Over the course of the project, we have developed a number of technologies in various modules of the dialog system to deal with these two issues [Weng et al 2004; Zhang and Weng 2005; Mirkovic and Cavedon 2005; Pon-Barry et al 2006; Varges 2005; Purver et al 2006]. Specifically, in this paper, we describe progress made over the past year when a navigation domain and related use cases are introduced. Evaluation conducted for the navigation domain shows high task completion rates and user satisfaction.

The paper is organized as follows: Section 2 describes the updated CHAT system architecture and its functionality; Section 3 is devoted to approaches used to address the imperfect speech and memory issues; Section 4 gives a description of data collection setup, evaluation scenarios, as well as evaluation results; finally, we conclude with a comparison with other work.

2 The CHAT System and Its Functionality

The CHAT system has adopted many state-of-art technologies and has grown beyond its heritages over the years. This progress is reflected in several core aspects, including the spoken language understanding (SLU) module, the dialog manager (DM), the content optimizer (CO), the knowledge management (KM), the response generation (RG), as well as the overall system architecture.

The SLU module integrates multiple understanding strategies with components such as edit region detection algorithm [Zhang and Weng, 2005; Zhang et al 2006][1], partial name identifier, shallow semantic parser, and deep structural parser. This approach enables understanding at finer levels when faced with imperfect input from the distracted multi-tasking user, and/or from speech recognition errors.

The DM, originated from the CSLI dialog manager [Lemon et al 2002], follows the information-state-update approach [Larsson and Traum 2000]. It uses a dialog move tree to keep track of multiple dialog threads and multiple applications [Mirkovic and Cavedon 2005; Purver et al 2006]. The latest version also supports mixed initiative dialogs for all the three domains.

The KM controls access to knowledge base sources and their updates. Domain knowledge is structured according to domain-dependent ontologies. The current KM makes use of OWL, a W3C standard, to represent the ontological relationships between domain entities.

The CO module acts as an intermediary between the dialog management module and the knowledge management module, controls the amount of content, and provides recommendation to users. It receives queries from the DM, resolves possible ambiguities, and queries the KM. It performs an appropriate optimization strategy based on the returned results [Pon-Barry et al 2006].

The RG module uses a hybrid rule-based and statistical approach. It takes query results from the KM via CO and generates natural language sentences as system responses to user utterances. The query results are converted into natural language sentences using a rule-based bottom-up production system. Finally, a scoring and ranking algorithm is used to select the best generated sentence [Varges 2005].

The architecture of the CHAT system is similar to its previous versions [Weng et al 2004; Weng et al 2006]. However, a couple of enhancements have been made to deal with multiple applications and random events from external devices or services. One enhancement is the introduction of an Application Manager (AP). The AP module isolates the application dependent information and operations from the core dialog system.

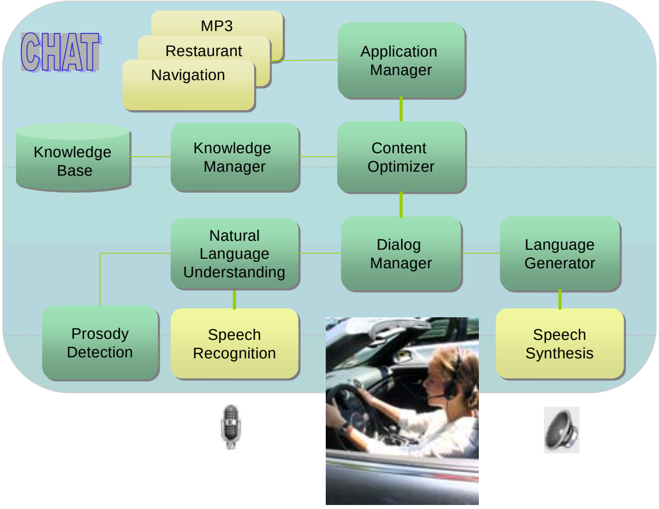

Figure 1 The CHAT System architecture.

Another major improvement is the modularity and configurability. The current version of the CHAT system is highly modularized and configurable. All the modules in Figure 1 are shared across the different domains. Domain specific models or parameters are supplied to the system in a configurable manner. Explicit on-the-fly domain switch becomes very simple – people can just say “switch to X” or other commonly used phrases to switch to the domain X. Implicit domain switch is also possible, where the users do not have to use explicit

[1] Edit region detection algorithms identify disfluent areas in an input utterance, such as hesitation, repeat, or correction. For example, “Get a, hmm, take me to Dave’s house”.

statements for switching to another domain. For example, having selected a desired restaurant in the restaurant domain, the user may then say “find me a fast route to restaurant XYZ”, without preceding this request with an explicit statement such as “switch to navigation”. However, due to extra burden on the system when all the applications are included, this feature is not set as a default. Additionally, because of the high modularity and configurability, it is much easier to add any new application.

In the added navigation domain, a few new functionalities are provided in the CHAT system:

1. Destination entry. One may either tell the system his or her address or use a selected Point Of Interest (POI) as the destination. In address input mode, different expressions for addresses, even ones with partial street or city names, are allowed.
2. Route selection. One may negotiate with the system with different criteria during the conversation for route selection. The criteria include distance, speed, mid-point, avoidance, or highway preference. For example, the driver can tell the system to find the shortest route to a destination that avoids a specific street. Or, he or she can also ask the system to find the fastest route that goes by a specific location and avoids highway.
3. Dynamic information or random events. While driving in a driving simulator, one can access dynamic information, such as the speed limit and the distance to destination. In addition, the dialog system can process random events, such as instructions from the navigation unit, and communicate the events to the user. In the meantime, the user can ask for a repeat of information if he or she did not hear the message clearly.

The three selected applications for the CHAT system, i.e., MP3 music player, restaurant finder, and navigation, represent three important application categories: entertainment, service, and navigation. When combining the three application domains together, the CHAT system can demonstrate a very useful scenario, which allows people to negotiate a POI (restaurant) or explicitly enter a destination; find the best route; and, on the way to the destination, access dynamic location-based information (“what is my next turn”) or entertain self (“play a song in playlist X”).

To give a concrete impression of what the CHAT system can do, an example dialog is given here:

User: Switch to restaurant.

System: Ok. Switched to the restaurant finder.

User: Can you find me a Chinese restaurant in Palo Alto with good service?

System: There are 11 Chinese restaurants in Palo Alto that have good service. Here are the first few: Little Garden Restaurant China Delight Mandarin Gourmet Restaurant Should I continue?

User: Hmm...tell me more about, Garden Restaurant?

System: Little Garden Restaurant is an expensive Chinese restaurant. Dress code is casual, service is good and rating is three stars. It’s on El Camino Real in Palo Alto.

User: Switch to navigation.

System: Ok, switched to Navigation system.

User: Take me to Little Garden Restaurant, but stop by Bosch Page Mill Office.

System: Ok, I will plan a route to Little Garden Restaurant that goes by Bosch Page Mill Office. Would you like anything else?

User: Get me the fastest route.

System: Ok, I will plan a fast route to Little Garden Restaurant that goes by Bosch Page Mill Office. Would you like anything else?

User: Um...nope.

System: Ok, I will plan a route now. Please wait … I calculated a route to Little Garden Restaurant that goes by Bosch Page Mill Office. It is 4.3 miles long and should take around 12 minutes. Now I am starting guidance...

User: Switch to music player.

System: Ok. Switched to mp3 music player.

User: What songs do you have by Alicia Keys?

System: There are 16 songs by Alicia Keys. Here are the first few: GoodBye, Jane Doe, and Caged Bird

User: Play “Caged Bird.”

System: Playing the song “Caged Bird.”

User: Add it to a playlist.

System: Which playlist do you mean, car or work?

User: What’s on the car playlist?

System: Interlude, A Womans Worth, and Whatever.

User: Add it to the car playlist then.

System: I added the song “Caged Bird” to car.

User: Switch to navigation.

System: Ok. Switched to navigation system.

User: Where am I right now?

System: You are on Page Mill Road in Palo Alto.

User: How far away am I from my destination?

System: You need to travel 2.5 miles to get to destination Little Garden.

During the conversation, click-to-talk is used to start the recognition. To reduce the effect of early speaking or early cutoff, we use a circular buffer to locate the start of an utterance, and use prosody information to identify precisely the ending of an utterance [Shriberg et al 2000]. This mechanism is integrated with the Nuance V8.5 recognizer.

In the next section, we will discuss the additional improvements made to address the issues of imperfect speech and memory.

3 Dealing with Imperfect Input and Memory

Two threads of research have been explored to deal with imperfect input: improve the robustness in the concerned modules; and provide error recovery strategies.

Improving robustness. To accommodate partial names in human utterances, separate ngram name models are trained on name databases of different classes for the SR module. A disfluency model is separately trained and integrated in the Statistical Language Model (SLM) for the recognizer. The partial or full proper names and disfluent regions are then identified by a proper name identifier and edit region detector, respectively. To understand the output from the recognizer, its SLU module adopts multi-component understanding strategies. A deep understanding component provides detailed information for each component in an utterance, which may be used for sophisticated dialogs. This module may also provide the boundary information for unknown proper names. On the other hand, a shallow semantic parser extracts domain-specific information, including flat or structured semantic classes. This provides a backoff strategy in the case the deep understanding module does not produce valid parses. These two components complement each other for better understanding and conversation.

Error recovery strategies. Individual understanding strategies do not always produce the correct interpretation in their 1st candidate. To correct errors, similarly, we experiment and integrate two different approaches: delay the final decision to a late stage; and design dialog strategies to clarify or confirm user’s intention. In the first approach, the SLU passes the top n-best alternatives as well as their likelihood scores to DM. The DM makes the final decision based on the n-best output from the SLU module, the possible dialog moves, and the dialog context (active dialog threads) [Purver et al 2006]. To deal with possible misunderstanding, we also developed dialog strategies such as clarification, confirmation, or even rejection when the system is not confident about its understanding. Another way to improve the communication is to convey back implicitly or explicitly the interpreted results and allow user to revise his or her constraint specification when any mismatch is noticed. Revision and addition of constraints onto previously stated ones are realized across all the three domains.

To handle imperfect memory issue, we continue our research in two directions: regulate the amount of information through presentation strategies; and allow the users to ask for the repeat of information already presented.

Regulated information presentation. During the conversation, user utterances are interpreted, and internal queries are constructed based on the constraints extracted from the utterances. These queries are sent to the Content Optimizer and Knowledge Manager for obtaining results that satisfy the constraints. Quite often, the results and their quantity would either overwhelm the user or leave them in a position where he or she does not know how to proceed. This can be a serious distraction or cognitive load problem in our investigation, as the user is occupied by other critical tasks, such as driving. One consequence is that people may not remember all the items enumerated, when the returned result list is long. In such case, the system proposes additional criteria so as to narrow down the results. In the event there is no result from the databases, the system proposes a relaxation of the constraints from the user. This has led to better user satisfaction [Pon-Barry et al 2006].

Information repetition. When the user focuses on other critical tasks, it is not always easy for him or her to remember the statements from the system.

One additional functionality allows the user to ask for the repeat of information just presented. This new functionality is very useful especially in the navigation domain where the navigation instructions occur at random and people may not always pay attention to the instructions at the time of speaking.

In addition, as mentioned earlier, the CHAT system allows the user to use partial names, anaphora, or ordinal references[2], which alleviates the imperfect memory issue and reduces the cognitive load of the user.

After the CHAT system is equipped with the above approaches and strategies, it shows a great improvement in terms of dealing with various phenomena caused by imperfect input and imperfect memory. Since most of these approaches and strategies are very collaborative in nature, they lead to a positive effect on user experience. This is partially reflected in the evaluation results reported in Section 4.

4 Experiments and Evaluation Results

For the navigation domain, the experimental setup is to drive and talk in a driving simulator. Three virtual cities are designed in the simulated environment with different streets, buildings, and businesses. Approximately 50 streets are setup in the tri city virtual environment – a limited number due to the cost of street design in the virtual world. Five different routes are designated to control the experiments and about 2500 restaurant names are included in the database for POI queries. Each restaurant is associated with a street name, a street number, and a city name. There is some duplication between city names and street names in the environment. Conducting experiments in a simulated environment addresses bias concern that arises when real cities are used for the task—some subjects may be more familiar than others in terms of streets and navigation. Using simulated environments also enables us to control the variation of different factors in the experiments, such as traffic.

As in the other two domains, WOZ data collection was used to bootstrap the development of the CHAT system for the navigation domain [Cheng et al 2004]. For the WOZ data collection, 20 subjects were recruited for performing navigation related tasks while driving in the three cities in the driving simulator. In addition, 14 subjects were recruited for dry runs, and 20 subjects were used for evaluation. The scenarios used in dryruns and evaluation are a subset of the scenarios used in the WOZ data collection.

The WOZ data collection gives us insight into how human subjects interact with an ideal dialog system, helps us in selecting research topics we need to address, and provides us data for improving the language coverage in both NLU and NLG modules.

Since the CHAT dialog system is designed as a task-oriented system and is not intended for any general conversation, careful attention was given to the development of the dialog tasks for the subjects to perform in the WOZ data collection, dry runs, and evaluation. Specifically, we developed the following two guidelines:

1. **Task-constrained**. We try to make goals of each task transparent and explicit (to form the intended mental context), so that the collected speech would not become irrelevant, unusable, or very sparse (see an example below).
2. **Language-neutral**. The language used in the instructions for communicating these task goals to the participant and in the scenario descriptions was created in such a way to avoid "copying behavior". One instruction explicitly asks the participants to "try to phrase your requests in your own words, rather than simply repeating the description of the scenarios".

We call this task design approach as *task-constrained and language-neutral*. This approach is used for both the restaurant finder and navigation domains. An example of a task description from the navigation is given here.

Task description: You have just picked up your business clients from the airport and would like to take them out to a reasonably priced lunch. You think that they would prefer Chinese food. Use the Navigation System to (1) find a Chinese restaurant, and (2) plan a route to the restaurant.

Eight task categories are used in the evaluation with examples such as "plan routes to destinations (e.g., restaurant POIs or address input)" and "query about road conditions". Each subject is given a practice trial and three test trials. The purpose of

[2] Examples of the ordinal references include "the second one", or "that last one".

the practice trial is to familiarize the subjects with the procedure and tasks, and to reinforce the language-neutral guideline. A total of 16 tasks from the eight task categories are designed, and they are designated to the three test trials. The evaluation procedure is very similar to the one used for the restaurant finder domain [Weng et al 2006].

Initial comparison of expressions used in the navigation scenario/task descriptions and expressions used by the subjects shows that the copying behavior is largely avoided. We found that only 18.13% of the subject expressions mimic the scenario/task expressions. In quantifying the copy behavior, it is counted as a copy if an expression is used in a task description and a subject repeats this same expression. For example, in the task "get clarification of the most recent route instruction", if the subject says "clarify the most recent instruction", this is counted as a complete copy; if the subject says "clarify the last instruction", this is counted as half of a copy; and if the subject says "repeat the last instruction", this is counted as a non-copy. Certain expressions do not have a clear alternative, such as "the current location". In these cases, we do not count them as a copy, and there are only two of such expressions.

This initial result indicates that our guidelines are effective in the experiments.

Among other metrics, three major measurements are used in the evaluation of CHAT's performance for the navigation tasks: task completion rate, dialog efficiency, and user satisfaction. The task completion rate is defined as the percentage of tasks completed during the evaluation. The CHAT system reaches an overall 98% task completion rate for the navigation tasks. To measure the dialog efficiency, we use the number of turns required to complete a task. Here, one turn was defined as one user utterance to the system during a dialog exchange between the user and the system while attempting to perform a task. The CHAT system is able to complete the tasks with 2.3 turns on average. Although it is not directly comparable between the two different domains, this number is much smaller than the average number of turns needed for the restaurant finder tasks (4.1 turns) reported one year earlier. Using the user satisfaction rating system by CU-Communicator [Pellom et al 2000], we reached a score of 1.98 with 1 indicating "strong agreement" and 5 indicating "strong disagreement" to each of the following statements:

4. It was easy to get the information I wanted.
5. I found it easy to understand what the system said.
6. I knew what I could say or do at each point in the dialog.
7. The system worked the way I expected it to.
8. I would use this system regularly.

We computed a one-sample 2-tailed t-test to see if mean ratings for the navigation system was significantly different from the mean rating of 1.76 for the best of the CU Communicator Systems (i.e., goal user satisfaction rating). Results showed that this difference was not significant (t (19) = 1.17, p > .05). This suggests that participants were no less satisfied with our navigation system than those participants who evaluated the CU Communicator System.

To get a better understanding of the improvement, we examine the word recognition accuracy for the two domains: for the navigation tasks, the accuracies with and without Out-Of-Vocabularies (OOVs) included are 85.5% and 86.5%, respectively; for the restaurant finder tasks, the accuracies are 85% and 86%, accordingly. Thus, the improvements are more likely a result of the new or refined implemented approaches.

5 Conclusions

Previous dialog applications include travel planning, flight information, conference information, bus information, navigation, hotel reservation, and restaurant finder [Pellom et al 2000; Polifroni et al 2003; Bohus et al 2007]. However, these applications are independently developed using single or completely different frameworks. In our case, we have integrated three representative applications and allow explicit or implicit domain switch with shared dialog contexts. The most related work is the GALAXY-II [Seneff et al 1999]. However, in their work, different applications are managed by different turn managers.

In terms of content presentation, [Polifroni et al 2003] discussed ways of organizing the content based on fully automated bottom-up clustering, while our approach focuses on semi-automated but configurable strategies that make use of the system ontology, and on external domain configurations for content organization and presentation.

More sophisticated dialog management research has recently focused on collaborative aspects of human machine dialogs [Allen et al 2001; Lemon et al 2002; Rudnicky et al 1999]. However, such research on conversational dialog systems has typically focused on dealing with dialogs that users need to pay full attention to. In addition, most of this research only deals with simple expressions where the meanings are mainly embedded in the semantic slots. For research in which elaborated expressions are considered, the coverage is typically small. Another thread of research is targeted at broad coverage but simple dialogs, which is exemplified by the work at AT&T [Gorin et al 1997].

While extending the research on the collaborative aspects, our effort specifically focuses on dealing with the conversational phenomena in multi-tasking and distracting environments, specifically imperfect input and imperfect memory. While dealing with imperfect input can be traced back far in time [Carbonell and Hayes, 1983; Weng 1993; Lavie & Tomita 1993; He and Young 2003], the CHAT system integrates models ranging from disfluency, partial and full proper names, shallow semantic parsing, and deep structural parsing. The interpretation only occurs when all the contextual information and alternatives are gathered. For the imperfect memory issue, we explore information presentation and other strategies to enable the user to access the information comfortably. All these approaches and strategies lead to high task completion rate and dialog efficiency as well as user satisfaction across the three domains, especially for the navigation. Collectively, the CHAT system shows very interesting use scenarios and promising performance.

Acknowledgement

This work is sponsored by a NIST ATP funding, as well as Robert Bosch LLC and VW of America.

References

Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. Towards Conversational Human-Computer Interaction. *AI Magazine*, 2001.

Bohus, D., Raux, A., Harris, T., Eskenazi, M., and Rudnicky, A. Olympus: an open-source framework for conversational spoken language interface research, *HLT-NAACL 2007 workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology*, Rochester, NY, 2007.

Carbonell, J. and Hayes, P. Recovery Strategies in Parsing Extragrammatical Language. *American Journal of Computational Linguistics.* Vol 9, No. 3-4, 1983.

Cheng, H., Bratt, H., Mishra, R., Shriberg, E., Upson, S., Chen, J., Weng, F., Peters, S., Cavedon, L., and Niekrasz J. A Wizard-of-Oz framework for collecting spoken human computer dialogs. *Proc. of ICSLP-200*0, Jeju, Korea, 2004.

Gorin, A., Riccardi G., Wright. J., How may I help you? *Speech Communication*, Vol. 23, pp. 113-127, 1997.

He, Y. and S. Young. A data-driven spoken language understanding system. *IEEE Workshop on Automatic Speech Recognition and Understanding*. 2003.

Larsson, S. and Traum, D. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering,* 6(3-4), 2000.

Lavie, A., and Tomita, M. GLR* - An Efficient Noise-Skipping Parsing Algorithm for Context-Free Grammars. In Proceedings of IWPT-1993, Tilburg, The Netherlands, August 1993.

Lemon, O., Gruenstein, A., and S. Peters. Collaborative activities and multi-tasking in dialogue systems. *Traitement Automatique des Langues (TAL),* 43(2), 2002.

Mirkovic D. and Cavedon L., Practical multi-domain, multi-device dialogue management", *PACLING'05: 6th Meeting of the Pacific Association for Computational Linguistics*, Tokyo, 2005.

Pellom, B., Ward, W., and S., Pradhan. The CU Communicator: An architecture of dialog systems. *Proc. of ICSLP,* Beijing, 2000.

Polifroni, J., Chung, G., and Seneff, S. Towards automatic generation of mixed-initiative dialog systems from web content. *Proc. of Eurospeech*, 2003.

Pon-Barry, H. and F. Weng. Evaluation of presentation strategies in a conversational dialog system. *Proc. of ICSLP,* 2006.

Purver, M., Ratiu, F., and Cavedon, L. Robust Interpretation in Dialogue by Combining Confidence Scores with Contextual Features. *Proc. of Interspeech*, Pittsburgh, PA, 2006.

Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu, W., and Oh, A. Creating natural dialogs in the Carnegie Mellon Communicator system. *Proc. of Eurospeech*, 1999.

Seneff, S., Lau R., and Polifroni, J. Organization, communication, and control in the GALAXY-II conversational system . *Proc. of Eurospeech*, 1999.

Shriberg E., Stolcke A., Hakkani-Tur D. and Tur G. Prosody-Based Automatic Segmentation of Speech into Sentences and Topics, *Speech Communication* 32(1-2), 127-154, 2000.

Varges, S. Chart generation using production system. *Proc. of 10th European Workshop on Natural Language Generation*, 2005.

Weng, F. Handling Syntactic Extra-grammaticality. *Proceedings of the 3rd International Workshop on Parsing Technologies*, Tilburg, the Netherlands, August 10-13, 1993.

Weng, F., Cavedon, L., Raghunathan, B. Mirkovic, D., Cheng, H., Schmidt, H., Bratt, H., Mishra, R., Peters, S., Upson, S., Shriberg, E., Bergmann, C., Zhao, L. A conversational dialogue system for cognitively overloaded users. *Proc. of ICSLP*, Jeju, Korea, 2004.

Weng, F., Varges S., Raghunathan, B. Ratiu, F., Pon-Barry, H., Lathrop, B., Zhang, Q., Bratt, H., Scheideck, T., Mishra, R., Xu, K., Purvey, M., Lien, A., Raya, M., Peters, S., Meng Y., Russell, J., Cavedon, L., Shriberg, E., and Schmidt, H. CHAT: A Conversational Helper for Automotive Tasks. *Proc. of Interspeech*, Pittsburgh, PA, 2006.

Zhang, Q. and Weng, F. Exploring features for identifying edited regions in disfluent sentences. *Proc. of 9th IWPT*, Vancouver, Canada, 2005.

Zhang, Q., Weng, F and Feng, Z. A progressive feature selection algorithm for ultra large feature spaces. *Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL9th*, Sydney, Australia, 2006.