Using System Expectations to Manage User Interactions

Filipe M. Martins, Ana Mendes, Joana Paulo Pardal,
Nuno J. Mamede, and João P. Neto

Spoken Language Systems Laboratory, L^{2}F – INESC-ID
IST / Technical University of Lisbon
R. Alves Redol, 9 - 2° – 1000-029 Lisboa, Portugal
{fmfm,acbm,joana,njm,jpn}@l2f.inesc-id.pt
http://www.l2f.inesc-id.pt

Abstract. This paper presents a new approach to parse multiple data types in Dialogue Systems. In its initial version, our spoken dialogue systems platform had a single and generic parser. However, when developing two new systems, the parser's complexity increased and data types, like numbers, dates and free text messages, were not correctly interpreted. The solution we present to cope with these problems allows the system to rely on expectations about the flow of the dialogue based on the dialogue history and context. Because these expectations guide the parsing process, a positive impact is achieved in the recognition of objects in the user's utterance. However, if the user fails to match the system's expectations, for instance by changing the focus of the conversation, the system is still capable of understanding the input and recognizing the referred objects.

1 Introduction

DIGA (DIaloG Assistant) is a domain-independent framework for spoken dialogue systems [1] that was the basis of two distinct applications: a butler that controls an home intelligent environment; and an interface to remotely access information databases (like bus timetables). STAR, the Dialogue Manager of DIGA is frame-based: every domain is described by a frame, composed by domain slots that are filled with user requests until a service can be executed [2]. In the first working version of DIGA, the language understanding module of STAR grabbed every domain keywords in users utterances and matched them against the domain roles specified in the domain frame. Slots were filled with tokens collected solely by a generic parser, which is still being used. The unique functionality of the parser was to split the utterance into tokens. From the resulting set of tokens the relevant keywords were selected and used to fill the corresponding domain slots. Tokens not matching any slot were discarded. However, when creating two new telephone-based services (home banking and a personal assistant) we faced several challenges [3].

This paper addresses the challenges that arise at the parser level, to deal with ambiguity in user utterances during spoken interactions. Similar approaches can

A. Teixeira et al. (Eds.): PROPOR 2008, LNAI 5190, pp. 240–243, 2008.

be found on TRIPS architecture [4] where a parsing module with a linguistically motivated grammar is used [5]. Alternative parses are scored with hand-tuned factors coded into lexical descriptions and grammar rules. The VerbMobil project [6] uses three different parsers based on different approaches, which are allowed to run in parallel. The idea is to take the benefits each approach can deliver while overcoming their related problems. The RavenClaw framework [7] takes into account the dialogue flow to ease the interpretation of users utterances, by embedding this information into a statistical model. Grammar-rules are manually generated and domain-specific.

Next, we present our problem and solution (Sect. 2), then the evaluation (Sect. 3), and finally, conclusions and future directions (Sect. 4).

2 Using Expectations in Parser Selection

The problem with our parsing technique came to our attention when developing two new telephone-based dialogue systems. In the configuration of the parser for the home-banking domain, the main problem was to cope with account numbers as they usually are big and users prefer to spell them instead of reading them. When creating Lisa, a digital personal assistant, we faced serious difficulties when trying to write the domain objects' recognition rules.

To answer to these problems, the existing unique parser was replaced by a module that manages the execution of a set of parsers: the Parsing Manager (PM). This module allows the definition of parallel and independent sets of parsers through a divide-and-conquer approach. It is configured with an XML file that declares the data type of each parser and the sequence of parsers.

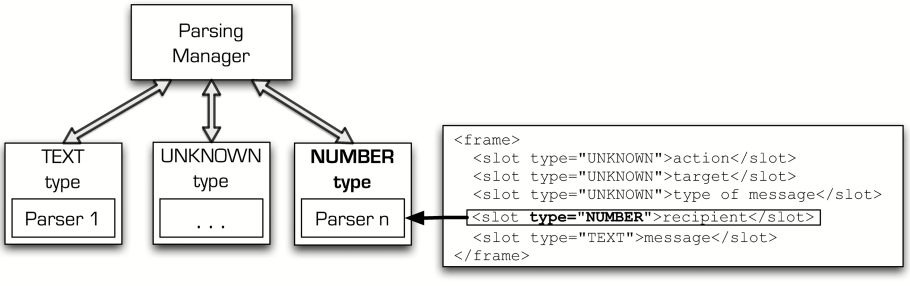

Fig. 1. Association between parsers and slot data types

Three parsers were built: `NUMBER`, which normalizes successive digits and numbers and corrects predictable recognition errors; `DATE`, which normalizes temporal expressions; and `TEXT`, which treats the input as a single chunk. With the definition of this set of parsers the dialogue manager only needs to select the adequate passer at each turn. In order to help the system with this decision, both the parser and the frame slot need to state its data types. Having the frame slots tagged with its data types, the system can inform the PM of the expected data

type for the next utterance. Having the parsers also tagged with their data type the PM knows which parser to use by selecting the matching data type. The association between parsers' and frame slots' type can be seen in Fig. 1. This information is used to select the parser expected to be most accurate. When the system takes the initiative and asks something to the user in order to fill an empty slot in the frame, the Interpretation Manager (IM) and the PM are informed about the expected data type to select the best expected parser.

As an example, let us consider the service that helps to send a short text message. Firstly, the system needs to request the recipient's phone number: After receiving the user's response to the question, the interpretation manager uses the data type of the slot being asked to select the parser to be used. In the example, it is being asked a `NUMBER` and the adequate parser returns the intended result: '918765131'. This approach allows the system to focus on the expected data type which improves object recognition scores and performance, provided that the user keeps up with system's initiatives. If the user decides not to answer the system's question, the selected parser may fail to interpret the utterance. In this case, the IM requests a generic interpretation to the PM.

Moreover, and since this is a frame-based system, the user can state a set of parameters of the request in the same utterance: *I want to send a short text message to nine eighteen seven six five thirteen one*[1]. When this occurs, it is necessary to execute the parsers sequentially to maximize the object chunking process and the extracted information. The sequential execution of parsers is possible by the definition of parsers composed by a sequence of other parsers.

3 Evaluation and Results

To evaluate our solution, we built a test corpus of interactions between Lisa and a novice user. While the evaluation was being performed, system's expectations about the user next utterance were automatically added to the corpus. Afterwards, the corpus was manually annotated with the correct expected data type for each interaction. Comparing both annotations we evaluated the system for two data types: `NUMBER` and `TEXT`. The results[2] are shown on Table 1.

Table 1. Evaluation results

System's Expectation	Interactions	Hits	Non Hits
`NUMBER`	382	258	124
`TEXT`	49	44	5

Data type expectations were met 90% for `TEXT`, and 67.5% for `NUMBER`. The overall treatment of the user input improved by 28%, meaning that from the

[1] Although our system only processes the Portuguese language, we used English in order to allow a broader understanding of this paper.

[2] A "hit" happens when the system's expectations match the user utterance; when the user utters something unexpected by the system, a "non-hit" occurs.

total number of interactions with the user (1085), in 302 interactions the most adequate parser was used, because the system had the correct expectation about what would be the next user utterance.

4 Conclusions and Future Work

The technique of using the system's expectations about the user's next utterance improved the domain objects recognition accuracy. Nevertheless, only 40% of the interactions with the user benefited as only those provided the system with expectations. In the other 60%, the system did not create an expectation, and the generic parser was used. More parsers and grammars for new data types will be needed as new dialogue systems are built with this framework. A future enhancement is the inclusion of morphological, syntactic and even semantic linguistic-based interpretation. A more sophisticated parser is needed to identify the objects in the utterances and to explore the relations and dependencies between them. We plan to include another generic parser for the Portuguese language that we currently use for text analysis. The used grammar will need to be tailored to allow common spoken language phenomena and ungrammaticalities that usually do not occur in written language.

Acknowledgments. This work was funded by PRIME National Project TECNOVOZ number 03/165. Joana Paulo Pardal is supported by a PhD fellowship from Fundação para a Ciência e Tecnologia (SFRH/BD/30791/2006).

References

1. Neto, J.P., Mamede, N., Cassaca, R., de Oliveira, L.C.: The development of a multipurpose spoken dialogue system. In: EUROSPEECH (2003)
2. Mourão, M., Cassaca, R., Mamede, N.: An independent domain dialogue system through a service manager. In: Vicedo, J.L., Martínez-Barco, P., Muńoz, R., Saiz Noeda, M. (eds.) EsTAL 2004. LNCS (LNAI), vol. 3230. Springer, Heidelberg (2004)
3. Martins, F., Mendes, A., Viveiros, M., Paulo Pardal, J., Arez, P., Mamede, N., Neto, J.P.: Reengineering a domain-independent framework for spoken dialogue systems. In: Proc. Software engineering, testing, and quality assurance for natural language processing, Workshop of ACL (to appear, 2008)
4. Allen, J., Ferguson, G., Swift, M., Stent, A., Stoness, S., Galescu, L., Chambers, N., Campana, E., Aist, G.: Two diverse systems built using generic components for spoken dialogue (recent progress on TRIPS). In: ACL Demo Sessions (2005)
5. Swift, M., Allen, J., Gildea, D.: Skeletons in the parser: using a shallow parser to improve deep parsing. In: COLING, ACL (2004)
6. Rupp, C., Spilker, J., Klarner, M., Worm, K.: Verbmobil: Foundations of Speech-to-Speech Translation. In: Verbmobil: Foundations of Speech-to-Speech Translation (2000)
7. Bohus, D., Raux, A., Harris, T., Eskenazi, M., Rudnicky, A.: Olympus: an open-source framework for conversational spoken language interface research. In: Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technology. HLT-NAACL (2007)