Integration of Speech and Text Processing Modules into a Real-Time Dialogue System*

Jan Ptáček[1], Pavel Ircing[2], Miroslav Spousta[1], Jan Romportl[2], Zdeněk Loose[2], Silvie Cinková[1], José Relaño Gil[3], and Raúl Santos[3]

[1] Charles University in Prague, Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{ptacek,spousta,cinkova}@ufal.mff.cuni.cz
[2] University of West Bohemia, Faculty of Applied Sciences, Dept. of Cybernetics
Univerzitní 8, 306 14 Plzeň, Czech Republic
{ircing,rompi,zloose}@kky.zcu.cz
[3] Telefónica I+D
Emilio Vargas St., No. 6., 28043 Madrid, Spain
{joserg,e.rsai}@tid.es

Abstract. This paper presents a real-time implementation of an automatic dialogue system called 'Senior Companion', which is not strictly task-oriented, but instead it is designed to 'chat' with elderly users about their family photographs. To a large extent, this task has lost the usual restriction of dialogue systems to a particular (narrow) domain, and thus the speech and natural language processing components had to be designed to cover a broad range of possible user and system utterances.

Keywords: automatic dialog systems, human-computer interaction, speech technologies, natural language processing.

1 Introduction

The COMPANIONS project represents a slightly different approach to automatic dialogue system development. Instead of designing strictly task-oriented system that would robustly cover a limited domain, the goal was to build a system that is more like an artificial 'companion', i.e., able to chat with the user and allowing him/her to develop some 'relationship' with the system. In order to reach this goal, the system is conceived much more broadly, even with the ability to express emotions to a certain degree.

The original plan was to develop a system that would be able to conduct a natural dialogue with elderly users, mostly to keep them company and helping them to stay mentally active. Since this is too broad a scope to be handled, it was decided to narrow the task to reminiscing about family photographs. The system was named 'Senior Companion' and was originally planned to be developed for two languages — English

* This work was funded by the COMPANIONS project (http://www.companions-project.org — EC grant number IST-FP6-034434) and partially also by the projects LC536, ME838, GAUK 52408/2008, MSM0021620838, GA405/09/0278 and UWB project SGS-2010-054.

P. Sojka et al. (Eds.): TSD 2010, LNAI 6231, pp. 552–559, 2010.

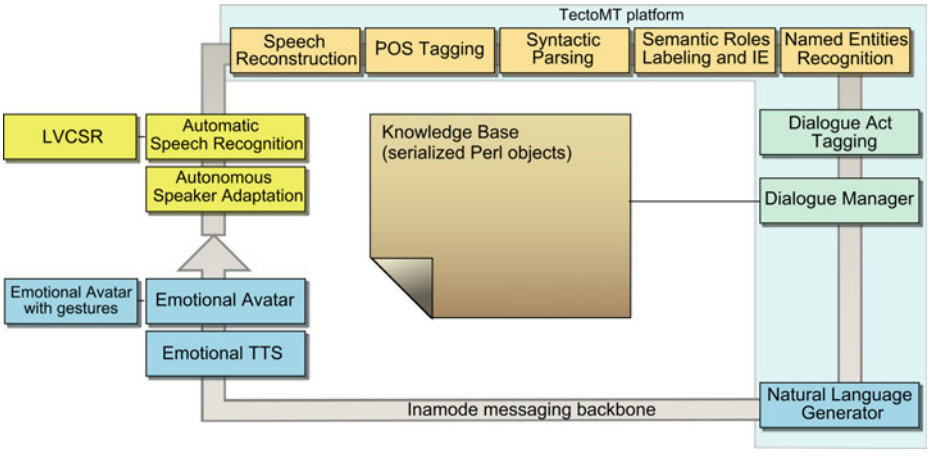

Fig. 1. Architecture of the system

and Czech. However, during the course of the project, the domain of the English system has slightly shifted and thus this paper describes the Czech system only.

In this paper, we will focus mostly on the problems related to the implementation and integration of the individual modules of the dialogue system prototype, whilst the scientific background of their development will be mentioned only briefly, with appropriate references to related work.

2 Architecture of the System

The architecture of the system is given in Figure 1. The Czech Companion consists of a number of independent modules (which in principle can reside on different machines) communicating via network messages encoded in XML. It is operating in an endless loop - each cycle of the processing loop starts with a user utterance and ends with the system's reply. There is a central messaging hub which controls the communication among the modules. Each module is geared towards a specific subtask, e.g. automatic speech recognition (ASR), natural language understanding (NLU), Embodied Conversational Agent (ECA) module and others, which are connected via a TCP/IP socket to the messaging hub.

Such an approach allows to employ modules, no matter whether they are Windows-, Linux- or Mac-based and regardless of their programming language. Each module is wrapped into a connector that provides the messaging API. The connectors have been developed and tested for Java, Perl and Python. During the development phase, each partner has been running his/her modules at his/her site while being connected to a central hub over the network to eliminate the installation and remote maintenance efforts. Furthermore, the architecture allows to conveniently relocate any computationally demanding modules to a dedicated hardware. The hub and the connectors are developed by our Spanish project partner (Telefónica I+D) and known under the "Inamode" trademark.

To establish a natural dialogue, it is necessary for the system to be able to respond within a reasonable time (empirically within 3s). In order to achieve such a response time, we have identified modules that work with the same data, and grouped them together into one run-time "supermodule", running on a single machine. For an easy grouping of such modules, we have re-used the TectoMT platform [1], originally designed for the machine translation task. The TectoMT platform architecture provides access to a single in-memory data representation through a common interface, effectively eliminating the overhead of a repeated serialization and XML parsing.

3 Automatic Speech Recognition (ASR)

The speech recognition engine embedded in our application uses state-of-the-art ASR technologies enhanced with some innovative techniques developed in our research labs. Its acoustic model employs Hidden Markov Models (HMMs) trained using large speech corpora (over 220 hours of transcribed speech, more than 700 speakers), which grants a robust acoustic recognition independent of the speaker. This performance is further improved by our original speaker adaptation algorithm, which is completely hidden from the user and does not require the usual explicit offline training session, since it accumulates statistics from all the user's utterances that are recognized with high-enough confidence score. Once the amount of "confidently recognized" data is sufficient, the system estimates the feature MLLR adaptation matrices and uses them to transform the input speech vectors from that point onwards [2].

Another notable feature of our ASR system is the speech decoder itself. It is currently able to handle a lexicon with more than half a million words. The decoder works with standard n-gram language models and allows 2-pass recognition process where the bigram model is employed in the first pass and the resulting lattices are re-scored with higher-order n-grams in the second pass. Note that despite of this 2-pass technique the decoder still operates in real-time [3].

4 Speech Reconstruction

Instead of feeding the ASR output to the Natural Language Understanding modules for further processing directly, we have inserted an intermediate step called Speech Reconstruction. The reason is that NLU modules are typically trained on data coming from annotated text corpora; ours are no exception. Such corpora are usually based on newspaper text and differ significantly from the style, lexicon and register of a spontaneous speech dialogue. Moreover, in such dialogues it is very common to encounter ungrammatical sentences, speaker auto-corrections, repetitions and other irregularities, which are rarely present in written text, resulting in poor results of the subsequent NLU. The speech reconstruction module aims to transform the ASR output into standard grammatical sentences suitable for NLP tools by removing disluencies, changing word order, or even colloquial morphemes. Our system employs machine translation approach; we have trained the Moses statistical machine translation system to "translate" the recognized speech output into fluent written-text-like utterances. We have used 45.000 sentences from the manually edited and corrected ("reconstructed")

PDTSL corpus [4] to train Moses' translation model and a 10-million-word textual corpus for its language model.

5 Natural Language Understanding

The natural language understanding (NLU) pipeline starts with part-of-speech (POS) tagging. Its result is very important in the subsequent steps, such as dependency parsing or named entity recognition, as they rely on correct POS tags. The POS tagging software for the Czech language uses a large morphological dictionary [5] that assigns a set of possible POS tags to every word. Then, a machine learning algorithm is used to select the correct tag. The state-of-the-art tagger of Czech achieves 96% accuracy on PDT 2.0 test set. We have further enhanced the coverage of the dictionary on the Wizard-of-Oz training corpus [6].

Then, we use a robust dependency-oriented syntactic parser; its task is to assign every word its parent in the syntactic dependency tree of the input sentence. The state-of-the-art algorithm [7] is based on the Maximum Spanning Tree algorithm and trained on the PDT 2.0 [8] analytical layer. Its accuracy on the standard PDT 2.0 test set is 85%.

Parsing is followed by a semantic analysis the output of which is a "tectogrammatical" transform of the dependency parse tree, simplified in its structure but enriched by various semantic attributes at its nodes. The semantic parsing involves (a) the assignment of one of the 69 semantic roles, (b) coordination detection, (c) argument structure assignment, (d) partial ellipsis resolution and (e) pronominal anaphora resolution. These attributes are filled using the *fnTBL* toolkit trained on the PDT 2.0 corpus. Post-parsing detection and correction of ungrammatical edges caused by long user utterances is rule-based.

The resulting semantic tree is matched against tree fragments using a tree querying engine PML-TQ [9]. The queries cover topics from the Dialogue Manager (DM; see below). The DM then stores the extracted information in a form of Perl objects, which are eventually saved on disk after each dialogue turn.

The DM also needs to know which named entities, such as personal names, appear in a given user utterance. The Czech SVM-based Named Entities Recognizer [10] we use achieves standard NER F-measure of 70%.

6 Dialogue Manager

The dialogue is driven by a Dialogue Manager (DM) component developed for the Czech Senior Companion. We build upon the idea and implementation of Field et al. [11], modeling the flow of the dialogue by a set of hand-crafted transition networks. We have re-implemented Field's DM and modified it to suit the type of dialogues as observed in our collected data.

The basic abstraction is a collection of interconnected transition networks, each representing a single topic. The nodes in a network represent various states of the dialogue, while the arcs provide possible continuations of the dialogue. Each arc bears a *test* and *action*. In our implementation, both are Perl code statements. The test is used to find out which arcs are suitable continuations given the current dialogue state. Once

a continuation arc is selected, its action code is executed. The action performs calls to DM API, updating DM's data structures to reflect the new dialogue state. Finally, the action assembles a "recipe" (a formal structure describing the content of the system response) for the NLG component to generate the system reaction in plain text. Several arcs may be traversed before an action orders the DM to send all the collected "recipes" to the NLG module.

We use an additional layer of Perl objects added between the DM and the NLU data-structures. These objects (persons, events, photos) model the knowledge acquired in the course of the dialogue and provide basic reasoning as well (e.g. it can tell age from a date of birth). Each object's property is able to store multiple values with varying level of confidence, and values restricted to a defined time span. For each topic, we have gathered a number of questions and remarks the system is supposed to say when it has the initiative, as well as replies to anticipated user turns. Here the large amount of collected dialogues has proved to be an irreplaceable resource, despite being used "only" for manual tuning of the DM networks. There is one initialization arc for each network, which is always traversed during each sub-dialogue in the active topic. The initialization arc leads to a central node as in Figure 2. The eccentric beams rooted in the central node take care of the various sub-dialogues. A sub-dialogue is picked at random, taking into consideration past traversals and of course making sure that the test on the entry arc of a sub-dialogue holds. Once the end of the sub-dialogue beam is reached, the DM is redirected back to the initialization arc.

The DM can also handle simultaneous dialogues with multiple users connected over the Inamode interface or over the XMPP/Jabber instant messaging protocol. The user-dependent data are grouped into objects, and the DM keeps track of the user by using an additional level of indirection. Prior to processing every incoming message, a pointer is updated to point to current user's data.

7 Natural Language Generation

Natural Language Generation (NLG) is implemented as a module of its own right, instead of a simple template approach typically used in previous dialogue systems. There are several reasons for that: Czech features a very rich inflection and morphosyntactic agreement (also related is its pro-drop feature), and its "free" word order used to convey emphasis and the topic-focus articulation. The NLG accepts a (grammatically grossly underspecified) tree-shaped template, which contains only the content to be formulated in the output natural language. These tree templates (called "recipes" in the previous section) are collected from the edges in the DM's transition network. They use linearized syntax that draws from the Graphviz dot notation [12]. The notation describes the tree by a set of statements specifying node properties and/or dependency relations between them. The integration with the DM is implemented using references to DM's Perl objects. Each reference is expanded to a set of additional statements, providing the subsequent NLG proper with the lexically and morphologically important features (e.g. gender) of those objects.

The rule-based sentence generation process consists of a sequence of linguistically motivated steps: "formeme" selection, morphological agreement, addition of functional

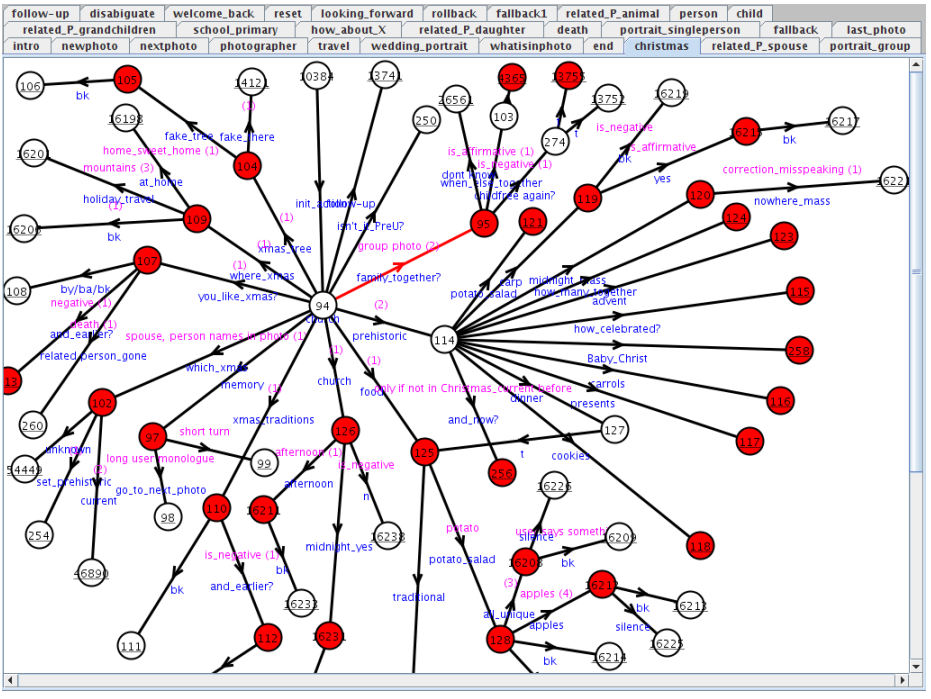

Fig. 2. Star patterned transition network for the topic of *Christmas* (red nodes mark a change in initiative)

words, inflection, word ordering, punctuation, and capitalization. The formeme selection phase is where the sentence takes on its surface-syntactic shape: the expanded input tree is traversed in a depth-first fashion, and a suitable morphosyntactic form is selected from the repertoire of forms available in Czech (i.e. prepositional phrase with the correct case, direct case, infinitive form, etc.). Once the formemes are established, the tree is processed by a cascade of operations that create the form selected and encoded by the formeme, so that the rules of Czech grammar are obeyed. This process includes, i.a., addition of functional words, proper morphological adjustments based on person, number and gender and realization of correct verb tenses. Finally, the word order is generated, as well as the correct punctuation and capitalization.

8 Embodied Conversational Agent

The system talks to the user through a graphically embodied conversational agent (ECA) which consists of a text-to-speech (TTS) system and a visual representation of a human character (the "avatar").

The TTS system is based on the Czech state-of-the-art speech synthesis system ARTIC [13] which employs the unit selection synthesis technique. Since one of the main goals of the COMPANIONS project is to enhance the naturalness of human-computer interaction as much as possible, the usage of a high quality emotional TTS

voice is essential. The ARTIC voice used for this purposes is based on an expressive speech corpus specially created for the COMPANIONS project using the Wizard-of-Oz corpus and a communicative function annotation scheme. Since it is difficult to classify human speech according to categorical or dimensional emotion models, we have settled for the assumption that a relevant affective state of ECA goes implicitly together with a *communicative function* of a speech act (or utterance), which is more controllable than the affective state itself. It means that we do not need to think of modeling an emotion such as 'guilt' per se, we expect it to be implicitly present in an utterance like 'I am so sorry about that' with a communicative function '(affective) apology'.

Apart from the TTS system, the ECA features a visual avatar. The final version of the Czech Senior Companion makes use of the Telefónica I+D avatar. It is a graphical visualization of a female head and torso with the capability of articulation, facial expressions and body gestures, which are triggered by special commands with both categorical and continuous parameters. A window with the avatar can be embedded e.g. in a web browser, which means that this ECA is able to run in various environments and modes of usage. The avatar as well as the TTS system is connected to the central messaging hub, as described in the Section 2. Both modules accept incoming messages and send outgoing messages to other modules. The activity of the avatar and TTS modules is coordinated by a proxy module. The proxy module accepts NLG output messages - every such a message contains text of the whole dialogue turn of ECA and the communicative functions of all the sentences in this text, as assigned by the Dialogue Manager. The proxy module parallelizes the work-flow by allowing the TTS system and the avatar to work simultaneously: TTS synthesizes sentences one by one and sends them to the input buffer of the avatar while the avatar is playing them stepwise. This measure was employed to minimize the latency of the system's response. The proxy module also communicates with a gesture module in order to transform the communicative functions assigned by DM into appropriate commands triggering gestures and facial expressions of the avatar. This process involves certain randomness causing the visual behavior of ECA be more natural. Synchronization of the avatar's articulation with synthesized speech is ensured by the TTS module, which generates an accompanying string of phones and their time-stamps.

9 Conclusions

The presented implementation shows that even the most advanced speech and natural language processing technologies can be successfully transferred from a laboratory form to a software prototype that is working in real-time. However, it is clear that the various modules can still be enhanced to avoid errors that still occur, such as those caused by the Named Entity Recognizer or the co-reference module. Also, the speech reconstruction module is far from perfect it can handle simple disfluencies but fails on more complicated edits. Finally, the DM should be trained as well, using e.g. the technique of Partially Observed Markov Decision Processes (once we are able to collect more dialogue data by using our Wizard-of-Oz technique). In fact, the resulting dialogue corpus is and will be one the most valuable results of the project, at least for the Czech language; it can and certainly will serve to advance research on general

enhancement and domain-adaption all of the speech and natural language analysis and generation components (not just the DM).

Dialogue systems are difficult to evaluate, especially if not task-oriented. A preliminary evaluation done by both the COMPANIONS consortium partners and the project reviewers rated our system's overall performance to be very good. However, a proper evaluation suited for free-flowing conversation still remains to be done. Such evaluation is actually one of the primary goals of the final year of the COMPANIONS project.

References

1. Žabokrtský, Z., Ptáček, J., Pajas, P.: TectoMT: highly modular MT system with tectogrammatics used as transfer layer. In: StatMT 2008: The Third Workshop on Statistical Machine Translation, Morristown, NJ, USA, pp. 167–170. ACL (2008)
2. Zajíc, Z., Machlica, L., Müller, L.: Refinement approach for adaptation based on combination of MAP and fMLLR. In: Matoušek, V., Mautner, P. (eds.) TSD 2009. LNCS, vol. 5729, pp. 274–281. Springer, Heidelberg (2009)
3. Pražák, A., Müller, L., Psutka, J.V., Psutka, J.: Live TV subtitling - fast 2-pass LVCSR system for online subtitling. In: SIGMAP 2007, Barcelona, pp. 139–142 (2007)
4. Cinková, S.: Semantic representation of non-sentential utterances in dialog. In: SRSL 2009, Morristown, NJ, USA, pp. 26–33. ACL (2009)
5. Hajič, J.: Disambiguation of Rich Inflection (Computational Morphology of Czech). Nakladatelství Karolinum, Prague (2004)
6. Grůber, M., Legát, M., Ircing, P., Romportl, J., Psutka, J.: Czech senior COMPANION: Wizard of Oz data collection and expressive speech corpus recording. In: LTC 2009, Poznan, Polan, pp. 266–269 (2009)
7. McDonald, R., Pereira, F., Ribarov, K., Hajič, J.: Non-projective dependency parsing using spanning tree algorithms. In: HLT 2005, Morristown, NJ, USA, pp. 523–530. ACL (2005)
8. Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M.: Prague Dependency Treebank v2.0, CDROM, LDC Cat. No. LDC2006T01. Linguistic Data Consortium, Philadelphia, PA (2006)
9. Pajas, P., Štěpánek, J.: PML Tree Query 0.5 alpha (2008), `http://ufal.mff.cuni.cz/~pajas/pmltq/`
10. Kravalová, J., Žabokrtský, Z.: Czech named entity corpus and SVM-based recognizer. In: NEWS 2009: Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration, Morristown, NJ, USA, pp. 194–201. ACL (2009)
11. Field, D., Catizone, R., Cheng, W., Dingli, A., Worgan, S., Ye, L., Wilks, Y.: The senior companion: a semantic web dialogue system. In: AAMAS 2009, Richland, SC, pp. 1383–1384 (2009)
12. Ellson, J., Gansner, E.R., Koutsofios, E., North, S.C., Woodhull, G.: Graphviz - open source graph drawing tools. In: Graph Drawing, pp. 483–484 (2001)
13. Tihelka, D., Matoušek, J.: Unit selection and its relation to symbolic prosody: a new approach. In: INTERSPEECH 2006, Pittsburgh, PA, pp. 2042–2045 (2006)