An Interactive Speech Interface for Summarizing Agile Project Planning Meetings

Shelly Park
University of Calgary
Department of Computer Science
2500 University Drive NW
Calgary, AB, T2N 1N4, Canada
parksh@cpsc.ucalgary.ca

Ehud Sharlin
University of Calgary
Department of Computer Science
2500 University Drive NW
Calgary, AB, T2N 1N4, Canada
ehud@cpsc.ucalgary.ca

Jörg Denzinger
University of Calgary
Department of Computer Science
2500 University Drive NW
Calgary, AB, T2N 1N4, Canada
denzinge@cpsc.ucalgary.ca

Frank Maurer
University of Calgary
Department of Computer Science
2500 University Drive NW
Calgary, AB, T2N 1N4, Canada
maurer@cpsc.ucalgary.ca

CHI 2006, April 22–27, 2006, Montreal, Canada.
ACM 1-xxxxxxxxxxxxxxxxxx.

Abstract

In this paper we present an autonomous meeting summarizer that transcribes an agile planning meeting and produces a textual summary of the discussion. We explore the issues involved in designing a speech-based interactive system that communicates with humans in a natural language. The inherent nature of ambiguity in conversational speech is overcome by suggesting a list of possible phrases to listen for. The system interacts with users in an interview-style dialogue for data collections. This is possible because we used the highly constrained structure and terminologies of agile planning meetings to make the approach successful.

Keywords

Human-to-computer Speech Communication, Speech-based User Interface, Spoken Dialogue Summarizer

ACM Classification Keywords

H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces – Natural language.

Introduction

The motivation for this project is to design a sociable robot that can participate in agile project planning meetings and produce a summary of the meeting. We want to assess a role that a robot can play in agile software development processes by understanding the impact a robot has on human behaviors. In order for a robot to participate in human-centered conversational meetings, we need to develop a user interface that can adapt to human speech communications. The goal of the current phase of the project is to develop a method that can be used to improve speech recognition accuracy and produce a transcript and a textual summary of the meeting using domain-specific information on agile project planning meetings.

As computers play a larger role in the automation of data acquisition and storage, creating a computer system that can understand and interact with humans in natural languages is becoming important in human-computer interactions [1]. Although Iwanska argues that natural language is a powerful knowledge representation system [2], computers do not understand the meaning behind sentences. Thus we define transcription of the meeting as all words that the system can successfully recognize and summarization as the ability to extract sentences based on the relevance of the context of the meeting.

The User Study

To understand our users better, we participated in the agile project planning meetings and observed the user behaviors. An agile project planning meeting called *Scrum* is for software developers to report their progress, future short term plans and any obstacles they encountered to the team [3]. Agile software engineers believe that the most efficient and effective method of conveying information to and within a development team is face-to-face conversations [4]. The meeting is comprised of the team and a scrum master and the meeting is generally guided by the scrum master. The meeting provides an ideal testing environment for an interactive meeting summarizer system because of its highly constrained conversational topics and a well structured order of speakers. The meeting only lasts about 15 minutes and each person speaks no more than 2 or 3 minutes. The summary should be categorized into progress, plans and problems for each person based on what they report during the meeting.

Structure of the Meeting

One of the biggest challenges for the meeting summarizer is that computer doesn’t understand the meaning of what people are saying. Thus our first manifestation of the summarizer is to create an interview-based interaction style. The users are asked to provide information for each category of the summary. The computer will ask:

- What did you do since the last meeting?
- What are your future plans?
- Do you have any problems that you would like to discuss with the team?

However, just asking three questions makes the interface seem monotonous, unsocial and unfriendly to the users. Kotelly argues that the success of a voice interface comes from a well designed user interface with a careful choice of words that can guide the users through the system. The usability study by Kotelly shows that anthropomorphized, first person approach can elicit better experience for users by emotionally engaging the users to try interacting with the system [5].

Therefore, the system tries to show some social behaviors. The system greets the user both at the beginning and the end of the meeting. It thanks the user for their help. The computer will also signal to the user that it is paying attention to what the user is saying by acknowledging with "Ok".

Speech Interface

The speech recognition technology has been substantially investigated for the last 40 years but we have yet to successfully produce a system that can interact with humans at the level of normal human-to-human speech communications. While many menu driven speech recognition applications have been developed, speech recognition of longer freeform conversational speech has been less successful.

The primary way to improve the speech recognizer is through grammar designs. The grammar constrains the possible words that it can recognize and thus improves the recognition accuracy by restricting the number of words that it should expect. There are two types of grammars: *rule-based* and *dictation*. The dictation grammars have fewer restrictions on what can be said but require higher quality audio to accurately predict the spoken words. The rule-based grammar allows the users to define the words and phrases that the system can recognize. The accuracy level of speech recognition is high due to the limited number of words and phrases that it recognizes. We follow a rule-based approach.

We used SAPI 5.1 for the speech recognition engine [6] and a simple desktop microphone. The main challenges of spoken dialogue summarization are detecting speech disfluencies, identifying the units of extractions, maintaining cross-speaker coherence and coping with speech recognition errors [7]. Zechner has studied speech disfluencies and sentence boundary detection by tagging parts of speech to find a pause in a speech [7]. However, the challenge is working with speech recognition error rate, especially when words sound similar.

Because the meeting is a progress report of their projects, people tend to use similar phrases to explain their progress. For example, the sentences will go "I have something", "I got something done", "I want to do something" or other similar phrases. Instead of using the default dictation grammar, we have formed a semi-rule-based system that suggests the computer what to listen for.

To form the rule-based grammar, we have divided the list of frequently occurring words based on the parts of speech. We have divided them into 9 possible sets: modal verb(*Mv*), verb(*V*), conjunction(*C*), preposition(*P*), noun(*N*), adjective(*A*), adverb(*Ad*), article(*At*) and pronoun(*R*). The verbs should contain all forms of verb tenses.

The list of words is compiled from previous meetings, computer science related literatures and MASE [8]. The log of task descriptions in MASE provides frequently occurring words that these particular software engineers will often use. Just putting massive list of generic words from a dictionary has proven to be ineffective as it doesn't narrow down the search domain.

Then we defined some of the phrases that the system should listen for based on the words list.

Example Grammar

Suppose *C*= {And, Or, But}, *Ad*={actually, lastly, simply, so, yet}, *Vp*={started, finished} and Vr={analyzing, studying}. The square bracket means the word is optional and *+ means any phrases are allowed.

Phrase = [*C*] [*Ad*] ([I] haven't *Vp* [*Vr*] [*Ad*] | [I] want to get *+ | *+)

The rule above can recognize a sentence like "And actually I haven't started analyzing yet"

If some sentences don't fall into the pre-defined phrases, we let the default dictation grammar recognize the sentence. Because we are looking forward to hearing certain phrases, we have a better chance of recognizing the phrase.

Small background noises can also trigger wrong recognitions. These white noises are usually transcribed as "So", "Ah", or "In". These words are ignored if they appear alone.

To test the effectiveness of the rule-based grammar recognition, we used archives of human transcribed meetings and created a new recording that is highly reflective of a typical agile project planning meeting. The sentences are intentionally spoken slowly and clearly. The real meeting had some background noises, people who spoke extremely fast, speech overlaps when two people spoke at the same time and people with accents. As the purpose of the current phase of the summarizer is obtaining the accuracy of the speech recognition and the summary, our preliminary testing is done with only one person. In the future, the system will handle multiple users interacting with the system at the same time.

The word error rate is calculated by counting the number of incorrect word recognition over the total number of words that it should recognize. The voice training has been done for about one hour. The same recording has been played 3 times. The error rate for the free dictation mode is 23% and the rule-based dictation is 12%. However, if the sample contains phrases that are not in the rules, the recognition error rate becomes the same as the free dictation mode. Table 1 contains an example of the sample dialogue and the resulting transcript.

Extracting a Summarization

The summary should contain only the important aspects of the meeting. Given that we are also working with an inaccurate transcript, we have to pick out sentences that seem more important. The first step is to re-word the sentences so they don't have a conversational tone. To do so, we have eliminated "I" in the beginning of the sentence and trailing adverbs.

The text summarization technique is categorized into abstraction and extraction. The extraction technique merely copies the most important sentences while abstraction technique involves paraphrasing the text. We used a simple extraction technique based on word relevance. The most popular way for summarizing a document is using statistical data based on the frequency of words appearance in a text to predict the most important sentences.

Original	Free Dictation	Rule-based Dictation
C: Hello. Let's start. What did you do since the last meeting? U: Um. I have transcribed the meeting manually. Um. C: Ok. U: I have analyzed the recordings done from the lab and they have a high background noise. C: Ok. C: Anything else? U: No, that's about it. C: Ok. What are your plans? U: I'm going to try the speech accuracy in a quieter room. C: Ok. C: Anything else? U: No. C: Do you have any problems that you would like to report to the team? U: No. C: Ok. Thank you.	Progress: - I have transcribed the meeting manually. - I haven't analyzed the recording stem from the lab and they have a high background noise. - know that's about it Plan: - I'm going to try the speech after seeing the choir room. Problem:	Progress: - I have transcribed the meeting manually. - I have analyzing recording done from the lab and they have the high background noise. - No That's about it. Plan: - I'm going to try the speech accuracy in acquiring. Problem:

Table 1: Here is an example conversation between a user and a computer. C is the computer and U is the user. The current phase of the project can only handle one person at a time, but the future work will involve multiple speakers. The table shows the speech recognition word error rate from both the default dictation mode and the rule-based mode. The underlined words denote mistaken words. If two words became one word, it is counted as two errors. If one word became two words, it is also counted as two errors.

However, the most frequently occurring words are unimportant words like prepositions, conjunctions and fillers such as "Um" and "Ok" in spoken-dialogue conversations. Thus simply using frequency of words is ineffective in determining the importance of a phrase.

To obtain a word relevance value, we analyzed the previous task descriptions and meetings to formulate a word list and assigned a low or high relevance value based on the relevance to the domain. If the combination of these relevance values for the words in a sentence is above a threshold value, the sentence will get included in the summary. For example, the word "and" has low relevance value, but "software" has high relevance value. Here is the summary produced from the above transcript.

Table 2: Summary Result	
Progress:	**▪ have transcribed meeting** **▪ have analyzing recording done from the lab and they have the high background noise**
Plan:	**▪ going to try the speech accuracy in acquiring**
Problems:	**N/A**

Table 2: The summary is returned in point forms for each of the categories: progress, plan and problems

Discussion

A carefully designed speech interface for a specific domain and a specific group of users can overcome the technological limitations in speech recognition or natural language understanding. To improve the speech recognition, we created a list of common phrases to listen for and ranked the sentences with relevance values to extract a textual summary. We used interview-style interaction to categorize people's report into progress, plans and problems. Our preliminary testing shows that a rule-based dictation can improve the word recognition rate, but more extensive testing is required to verify the effectiveness of the summarizer, especially in a multi-user setting. Humans behave differently with computers than with other humans [9]. Therefore we also need to evaluate the changes in social dynamics with the introduction of the summarizer during the meeting.

The limitation of the speech recognition system still poses problems as people must learn the speaking style recommended by the speech recognition software. The expectation of the interactive meeting summarizer should not be about how accurately the system can capture everything, but how accurately it extracted words that describe discussions during the meeting.

Acknowledgement

We would like to thank Ruth Ablett, Mike Ji and Brian David Fox. We would like to thank the members of EBE Software Engineering Lab for their participations.

Bibliography

[1] Weischedel, R., Carbonell, J., Grosz, B., Lehnert, W., Marcus, M., Perrault, R., Wilensky, R. White Paper: White paper on natural language processing. *Workshop on Speech and Natural Language HLT 89*, Association for Computational Linguistics (1989), 481-493

[2] Iwanska, L. *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, AAAI Press, Menlo Park, CA, USA, 2000

[3] Schwaber, K. *Agile Project Management with Scrum*, Microsoft Press, Redmond, WA, USA, 2004

[4] Principles Behind the Agile Manifesto http://www.agilemanifesto.org/principles.html

[5] Kotelly, B. *The Art and Business of Speech Recognition: Creating the Noble Voice*, Addison-Wesley Professional, Boston, MA, USA, 2001

[6] Speech SDK 5.1 for Windows Applications http://www.microsoft.com/speech/download/sdk51/

[7] Zechner, K., Automatic Generation of Concise Summaries of Spoken Dialogues in Unrestricted Domains, *SIGIR 2001,* ACM Press (2001), 199-207

[8] Maurer, F., Supporting Distributed Extreme Programming, *XP/Agile Universe 2002*, Springer-Verlag (2002), 13-22

[9] Jonsson, A., Dahlback, N., Talking to a Computer Is Not like Talking to Your Best Friend, *SCAI 1988, (1988)*, 53-68