<table>
<thead>
<tr>
<th>Title</th>
<th>Study on Optimal Spoken Dialogue System for Robust Information Search in the Real World</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>徐, 昕</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2016-09-26</td>
</tr>
<tr>
<td>DOI</td>
<td>10.14943/doctoral.k12405</td>
</tr>
<tr>
<td>Doc URL</td>
<td><a href="http://hdl.handle.net/2115/63374">http://hdl.handle.net/2115/63374</a></td>
</tr>
<tr>
<td>Type</td>
<td>theses (doctoral)</td>
</tr>
<tr>
<td>File Information</td>
<td>Xin_Xu.pdf</td>
</tr>
</tbody>
</table>

Hokkaido University Collection of Scholarly and Academic Papers: HUSCAP
Study on Optimal Spoken Dialogue System for Robust Information Search in the Real World

Author: Xin XU

Supervisor: Dr. Yoshikazu MIYANAGA

Examiners:
Dr. Kunimasa SAITOH
Dr. Takeo OHGANE
Dr. Hiroshi TSUTSUI

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy
in the Information and Communication Networks Laboratory
Graduate School of Information Science and Technology
August 2016
2016 Doctoral Thesis

Study on Optimal Spoken Dialogue System for Robust Information Search in the Real World

Division of Media and Network Technologies
Graduate School of Information Science and Technology
Hokkaido University
Xin Xu
August 22, 2016
Abstract

Recently, the spoken dialogue systems those enable users to intuitively and directly operate services and smartphones with voice commands and information search become popular. However, there is still a remaining challenge that there are not many users with the habitual and continual use of the spoken dialogue systems for information search in the real world, though most of them have devices in which the spoken dialogue system is implemented. To solve this challenge, three researches in different aspects have been done in this thesis, to realize an optimal spoken dialogue system for robust information search in the real world.

The first research practices human-centered design (HCD) to design a dialogue agent and a dialogue scenario promoting a daily use of the spoken dialogue interface, which is based on the cognitive science and the gamification theory. The author proposes a design concept of breeding a character, which is actually a dialogue agent, through taking care and having a dialogue in order to make users gradually feel that speaking to the dialogue agent is natural and fun. The real-world data prove the novelty of the proposed design, in which over 23% users keep speaking continually. More than 95% conversations from the dialogue agent are responded by the users.

The second research improves the efficiency and robustness of the dialogue management for information search based on the information theory. The author proposes two strategies to optimize question selection for information search and to decrease failures in information search mainly caused by mistaken queries. One strategy applies optimal question selection in a knowledge-based spontaneous dialogue system, which has been verified to be effective to assist the users’ operation for information search. The other strategy applies a robust and fast search method based on phoneme strings matching. It decreases the failures caused by the queries
containing incorrect parts. Experimental results show that the proposed search method increases search accuracy by 4.4% and reduces processing time by at least 86.2%.

The third research practices signal processing technologies to emphasize the usability of spoken dialogue systems. The author proposes a novel pitch detection method applying an adaptive filtering algorithm to restore the amplitude spectra of speech corrupted by additive noises. The periodic structures in the amplitude spectra are kept against noise distortion. Experimental results verify that the proposed pitch detection method achieved the highest robustness in a variety of noise type and noise level. With the high-accuracy pitch information, emotion recognition is going to be established in the next step of this research. Understanding speaker’s emotion helps to generate the appropriate dialogue actions to present superiority and differentiation to other modalities.

Furthermore, based on the above researches, this thesis proposes a dialogue structure to build a personalized dialogue system applying emotion recognition and multi-device interface for further real-world use in the future.
# Contents

1 Introduction 1
  1.1 The History of Spoken Dialogue System Research 1
  1.2 The Issues Focused in This Thesis 3
  1.3 Research Contributions 6
  1.4 Thesis’s Organization 7

2 Key Technologies of Spoken Dialogue Systems and Related Works 9
  2.1 The Structure of Spoken Dialogue Systems 9
  2.2 The Technology Development of Spoken Dialogue System 11
    2.2.1 Automatic Speech Recognition 11
    2.2.2 Natural Language Understanding 12
    2.2.3 Dialogue Management 12
    2.2.4 Natural Language Generation 13
    2.2.5 Text to Speech 14
    2.2.6 Emotion Recognition with Prosody Information 15
  2.3 Remaining Issue and Related Works 16
3 Spoken Dialogue System Design for a Habitual Use

3.1 In-House Trial of Spoken Dialogue System

3.1.1 Structure of the Trial System

3.1.2 Trial Evaluation

3.1.3 Discussion of the Spoken Dialogue Agent

3.2 Agent Characters’ Impressions against the User’s Tolerance

3.2.1 Preliminary Investigation for Selecting Characters

3.2.2 Experiment for Evaluating Users’ Tolerance

3.3 Design of User Interface for the Spoken Dialogue System Applying a Breeding Game

3.3.1 Target Users

3.3.2 Scenario Design

3.3.3 Evaluation of Scenario-based Acceptability

3.3.4 Game Implementation

3.3.5 Analysis of the Real-World Dialogue Log Data

3.4 Summary

4 Dialogue Management for Information Search

4.1 Dialogue Strategy of Optimal Question Selection

4.1.1 Spoken Dialogue System for Information Search in a Cooperative Manner

4.1.2 Question Selection for Quick Narrowing Down of Hit Data

4.1.3 Experimental Evaluations

4.2 Information Search by Robust and Fast Phonetic String Matching Method

4.2.1 Analysis of the Real World Queries
4.2.2 Investigation of the Relationship between the Query Length and DP Matching Pre-selection ............................................. 54
4.2.3 Acoustic Distance Derived from a Phoneme Confusion Matrix 56
4.2.4 Fast Two-pass Search Algorithm in Consideration of Acoustic Similarity .............................................................. 59
4.2.5 Experiments ........................................................................... 63
4.3 Summary ................................................................................. 74

5 Applying Prosody Information in Spoken Dialogue Systems 76
  5.1 Prosody Information .............................................................. 76
  5.2 Spectra Analysis and Pitch Detection ..................................... 78
    5.2.1 Fundamentals of Speech Production ................................ 78
    5.2.2 Introduction of Running Spectrum and Modulation Spectrum 80
    5.2.3 Pitch Detection ............................................................ 84
  5.3 Robust Speech Spectra Restoration for Pitch Detection .......... 87
    5.3.1 Adaptive Running Spectra Filtering designing .................. 87
    5.3.2 Pitch Detection with ARSF ........................................... 92
    5.3.3 Experiments ............................................................... 102
  5.4 Summary ................................................................................. 106

6 Conclusion and Future work ..................................................... 108
  6.1 Conclusion ............................................................................. 108
  6.2 Future Work ......................................................................... 109

Acknowledgement ...................................................................... 111

References .................................................................................. 112
List of Tables

3.1 Average score and standard deviation values .................................. 32
4.1 Results of subjective evaluation .................................................... 48
4.2 Average operating time for one search task ..................................... 48
4.3 The distribution of mistaken types within content-word-error cases ... 53
4.4 Number of hits by two web search engines ....................................... 54
4.5 Search accuracies and p-values of EDPED and EDPAD ................... 65
5.1 Comparison results at various noise conditions (GPE %) .............. 105
5.2 Comparison results at various noise conditions (SDFPE) ............. 106
List of Figures

2.1 Structure of a spoken dialogue system .................................. 10
3.1 Structure of the trial spoken dialogue system .......................... 22
3.2 Character agents used for the experiment ............................... 25
3.3 Demographics of Peratama users. ........................................ 27
3.4 Screenshots of Hey Peratama .............................................. 34
3.5 Utterance distribution ....................................................... 36
3.6 Distribution of major user-initiated dialogues topics .................. 37
4.1 Interface of the proposed application .................................... 42
4.2 The flowchart of the dialogue scenario ................................ 43
4.3 The distribution of mistaken queries in the different types and correct queries within the collected queries .......................... 51
4.4 The distribution of the length of a query in phonemes and the DP matching distances from the correct lyric for the real world incorrect queries ........................................ 56
4.5 Flowchart of the first pass search ....................................... 61
4.6 Flowchart of the second pass search ................................... 63
4.7 Relationship between hit rates and $I_c$ for various sizes of lyric database 66
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.8</td>
<td>Search accuracy and processing time with respect to $F_{th}$ in the case of 1-best</td>
<td>68</td>
</tr>
<tr>
<td>4.9</td>
<td>Search accuracy and processing time with respect to $F_{th}$ in the case of 20-best</td>
<td>69</td>
</tr>
<tr>
<td>4.10</td>
<td>Search accuracy of TDPAT and EDPAD</td>
<td>69</td>
</tr>
<tr>
<td>4.11</td>
<td>Average processing times and search accuracy of three search methods in the case of 1-best</td>
<td>72</td>
</tr>
<tr>
<td>4.12</td>
<td>Average processing times and search accuracy of three search methods in the case of 20-best</td>
<td>73</td>
</tr>
<tr>
<td>5.1</td>
<td>Schematic view of the human vocal mechanism (cited from [83])</td>
<td>79</td>
</tr>
<tr>
<td>5.2</td>
<td>Modulation spectra of clean speech</td>
<td>82</td>
</tr>
<tr>
<td>5.3</td>
<td>Modulation spectra of speech corrupted by 0dB white noise</td>
<td>83</td>
</tr>
<tr>
<td>5.4</td>
<td>Modulation spectra at 130 Hz in different conditions</td>
<td>84</td>
</tr>
<tr>
<td>5.5</td>
<td>AUTOCC method</td>
<td>87</td>
</tr>
<tr>
<td>5.6</td>
<td>Noise distortion in amplitude spectrum</td>
<td>88</td>
</tr>
<tr>
<td>5.7</td>
<td>Block diagram of the ARSF approach</td>
<td>89</td>
</tr>
<tr>
<td>5.8</td>
<td>The magnitude response of the adaptive high-pass filter with -20 dB attenuation</td>
<td>92</td>
</tr>
<tr>
<td>5.9</td>
<td>Flowchart of the proposed pitch detection method</td>
<td>94</td>
</tr>
<tr>
<td>5.10</td>
<td>Amplitude spectrum of clean speech</td>
<td>95</td>
</tr>
<tr>
<td>5.11</td>
<td>Amplitude spectrum of speech with 0 dB car noise</td>
<td>96</td>
</tr>
<tr>
<td>5.12</td>
<td>Amplitude spectrum of speech with 0 dB pink noise</td>
<td>97</td>
</tr>
<tr>
<td>5.13</td>
<td>Amplitude spectrum in speech with 0 dB talking babble noise</td>
<td>98</td>
</tr>
<tr>
<td>5.14</td>
<td>Autocorrelation contour in clean speech</td>
<td>99</td>
</tr>
<tr>
<td>5.15</td>
<td>Autocorrelation contour in speech with 0 dB car noise</td>
<td>100</td>
</tr>
</tbody>
</table>
5.16 Autocorrelation contour in speech with 0 dB pink noise . . . . . . . 101
5.17 Autocorrelation contour in speech with 0 dB talking babble noise . . 102
6.1 Structure of the future system . . . . . . . . . . . . . . . . . . . . . 110
Chapter 1

Introduction

1.1 The History of Spoken Dialogue System Research

Spoken dialogue system is described as “an interactive system which operates in a constrained domain” (Glass, 1999) [1]. It offers direct, simple, hands-free access to information with several technical factors contributing to.

The research of spoken dialogue systems can be traced back to “ELIZA” [2]. ELIZA is a system (precisely a computer program) of primitive natural language processing, which was created in 1960s at MIT Artificial Intelligence Laboratory. It only returns a response based on a superficial matching of words based on users’ input, without speech interface. So far, ELIZA has been regarded as a source of inspiration for programmers and developers of artificial intelligence (AI). Corresponding to the development of speech recognition and synthesis technologies around 1990, the research of spoken dialogue systems became popular. The forerunner is the MIT’s “VOYAGER” [3]. It functioned as navigation and guidance of the city, close
Chapter 1. Introduction

to the spoken dialogue service of the current smartphone in terms of the task domain. After it, “ATIS” project in the early 1990s supported by DARPA in the United States was carried out [4]. The researchers engaged in ATIS spun off to establish Nuance, Inc. and SpeechWorks, Inc., which brought a boom of spoken dialogue systems. The systems were employed in the interactive voice response (IVR) service with the telephone interface on the basis of a fixed grammar and dialogue flow described manually. As the spoken dialogue systems are introduced on a large scale in call centers, it was commercially successful.

For the recent trend, there are great developments in both the theory and practical use of spoken dialogue systems, especially in mobile devices [5]. Recent spoken dialog interfaces have moved beyond a mere keyboard replacement, to provide integrated voice search, speech understanding, and basic device operation. The main players are Apple’s Siri, Google’s voice search, Nuance’s Dragon Go!, Docomo Shabette Concier (Japanese) and systems in cars from several manufacturers. In addition, the spoken dialogue systems are not only on top of the smartphones, also for the robots. Japanese telecom company Softbank employs robot “Pepper” to do shop reception, and declares that the future robots that can recognize human emotion will change the way we live and communicate [6]. The family robot “Jibo” also gives an idea to bring spoken dialogue technologies to life with personal and emotional engagement [7].

Also new theories of spoken dialogue, deep learning is becoming a mainstream technology for speech recognition and has successfully replaced Gaussian mixtures for speech recognition and feature coding at an increasingly larger scale [8]. The classification model of support vector machines (SVM) and conditional random fields (CRF) [9] is introduced into the research of natural language understanding. In the field of dialogue management, though most systems are limited cross-turn persistence in the state of the dialog, some of them are using machine learning techniques such as reinforcement learning in partially observable Markov decision processes (POMDPs) [10], which has been in development in the research community for
about a decade.

1.2 The Issues Focused in This Thesis

Spoken dialogue systems can be presented by a wide range of domains, from simple weather forecast systems (systems ask the city name and give the weather information) to complex problem-solving, reasoning, applications (special dialogue systems for medical domain). Generally, spoken dialogue systems can be classified into two types: conversational (non-task) systems and goal-oriented (task) systems. It should be stressed that these are prototypical categories, and all dialogue systems do not fit neatly into one of them. Such as Apple’s Siri, it assists users to search information, to confirm or edit schedules by users’ commands, while takes the casual conversations at the same time.

The study in this thesis put particular emphasis on the goal-oriented type spoken dialogue systems, especially for information search task. The final goal of the study is to establish a spoken dialogue system as a daily and habitual access to information spaces through natural spoken language interaction and personal preferences.

Including Siri, many current commercial dialogue systems for information search are based on a dialogue strategy of One-Turn Q&A type that does not support complex dialogue scenario logics. On the other hand, some systems are based on a fixed scenario. Some conventional systems responds with a confirmation of the user’s request to avoid the misunderstanding by the speech recognition errors. However, these traditional dialogue strategies and designs degrade the usability of dialogue system. The focus has not been on the approaches for the habitual use of a spoken dialog interface in a long term. For example, as is investigated by some speech specialists [11], 85% of iPhone iOS 7 users have never used Siri. Not only Siri, other spoken interfaces also face the same issue. In [12], this phenomenon is discussed, and the necessary conditions for continual use of spoken dialogue systems, which
Chapter 1. Introduction

are also called “requirements for survive”, are summarized as follows:

- Requirement 1: should optimize agent presence and dialogue design to improve conversation frequency
- Requirement 2: should avoid failures of search and dialogue caused by the errors of speech recognition
- Requirement 3: should clearly show the superiority and differentiation to other modalities
- Requirement 4: should enhance the killer apps that are routinely used
- Requirement 5: should find more use cases of immediate real-time requirements

As Requirements 4 and 5 are more closely related to the specific applications and use cases, the study of this thesis concentrates on pursuing Requirements 1, 2 and 3 to realize an optimal spoken dialogue system for robust information search in the real world.

Firstly, the study takes advantage of the gamification theory to design a dialogue agent and dialogue scenarios to foster the user’s affection on the character, which promotes the users to continually use the spoken dialogue interface. The affection is also supposed to enhance the user’s tolerance against the mistakes caused by speech recognition (ASR) or natural language understanding (NLU), which is the weak point of spoken dialogue systems.

Secondly, the study proposes dialogue management strategies to improve the user experience for searching information. To optimize the spoken dialogue system for information search, a novel strategy of question selection has been verified to be effective to assist users’ operation in a knowledge-based spontaneous dialogue system. Furthermore, to avoid the information search failures in dialogue, especially as the
traditional full-text search method is highly compromised when the queries contain incorrect parts due to mishearing or misrecognition, a robust and fast matching method based on phoneme strings decreases the failures, and the most appropriate answer candidates are presented.

Thirdly, to deeply explore the superiorities of spoken dialogue system to other modalities such as touch screen or keyboards for information search, the study in this thesis intends to apply emotion recognition. Compared with other modalities based on only text or command information, the responses of spoken dialogue system are supposed to be more appropriate with the speaker’s emotion information. However, it is revealed that the extraction of the prosody information that plays an important role in emotion recognition is a difficult issue in the real-world environment. Therefore, my study focuses on the detection of speech pitch period for practical applications. The proposed method intelligently restores speech modulation spectra according to the estimated noise conditions. The noise distortion that influences the amplitude spectral structure is much diminished. Therefore pitch detection gains the better accuracy than other conventional methods under different noise conditions. With this novel pitch extraction method, a method for recognizing speaker’s emotion is going to be established in the future study with a multimodal model mixing speech prosody and text positive/negative polarity recognition.

Finally, this thesis also gives discussion on future study of optimization of the dialogue along to the user’s preferences for information search. One direction is the further challenge of personalization. The dialogue system will be applied to many types of devices, such as a set top box (STB) and an in-vehicle machine as well as smartphones to cover the users’ life scenes to learn more hobbies and preferences of the user via analysis of individual dialogue log data.
1.3 Research Contributions

The main contributions of this work, which will be discussed in details in Chapters 3, 4 and 5, are summarized as:

First, human-centered design (HCD) and gamification theory are practiced to design a novel spoken dialogue interface. The research proposes a new concept of breeding the dialogue agent through taking care and having dialogues. The stepwise growth of the agent creates the users’ expectations and intimacy, which motivate users to use the spoken dialogue interface habitually. The designed scenario obtained high evaluation scores from the subjective evaluation experiments. Many positive comments are collected: “feel no stress in talking to the character as it is a game”. The game application was released on Google Play. The daily active rate is 23% or more based on the analysis of the real world users’ log data. More than 95% conversations from the dialogue agent are responded by the users. The activity is much higher than a conventional spoken dialog system with a character agent that is tested in an in-house trial. Furthermore, my study investigated when and what topics the real world users are talking to the agent. It is found that users are talking to the agent most around the time of getting up and going to bed to confirm the schedule and weather, and the time of arriving home to enjoy chit-chats. These understandings help the design of the future dialogue system to provide proper topics automatically.

Second, a dialogue management strategy is proposed to optimally select questions to ask the users to help their refine search. A decision algorithm is applied to find the best questions to minimize the number of search refinement steps. In addition, the questions responding to the candidates that users showed interests in are preferentially selected. Based on the evaluation results, the application with the proposed strategy performs better than the conventional applications in terms of the users’ satisfaction with search results and the effort spent for reconsidering search keywords. Furthermore, a robust and fast search method based on acoustic
distance and a two-pass search strategy is introduced. The two-pass search uses an index-based approximate preselection for the first pass and a dynamic programming (DP) based string matching in the second pass. It is verified in the task of lyric search with a test set of the incorrect queries those are misheard or mismemorized. The experiments proved that applying the originally proposed acoustic distance improved search accuracy by 4.4% compared with the conventional search with edit distance. Though the search accuracy is expected to be more improved if the acoustic distances are calculated from the singing voice data, the proposed method offered a realistic and efficient solution with an easily-obtainable database of more general ordinary speech. The proposed method achieved real-time operation by reducing processing time by more than 86.2% with a slight loss in search accuracy compared with a complete search by DP matching with all lyrics. It is proved to be the most practical solution for acoustic confusion queries, considering the trade-off between high search accuracy and low computation complexity.

Third, my research proposes a new algorithm named adaptive running spectrum filtering (ARSF) to restore the amplitude spectra of speech corrupted by additive noises. Based on the pre-hand noise estimation, adaptive filtering is used in speech modulation spectra according to the noise conditions. The periodic structures in the amplitude spectra are kept against noise distortion. Since the amplitude spectral structures contain the information of fundamental frequency, which is the inverse of pitch period, ARSF algorithm is added into robust pitch detection to increase the accuracy. Compared with the conventional methods, experimental results show that the proposed method significantly improves the robustness of pitch detection against noise conditions with several types and SNRs.

1.4 Thesis’s Organization

This thesis is organized as follows:
In Chapter 2, the structure of the dialogue system and the development of elemental technologies are organized, in which the analysis of the issues and problems of the current state of the dialogue system are listed. In addition, the focused issues of this thesis and the related works are described.

Chapter 3 presents my first research that practices human-centered design (HCD) to design a dialogue agent and a dialogue scenario promoting a daily use of the spoken dialogue interface, which is based on the cognitive science and the gamification theory. It mainly expands the discussion of how to design an interactive system for continual use. Secondly, it presents HCD process and gamification theory practiced in the design process of the dialogue system. Then, subjective evaluation and the results of the dialogue scenario are presented. The end of the chapter describes the analysis of the real world users’ log data.

Chapter 4 presents my second research that improves the efficiency and robustness of dialogue management for information search based on the information theory. Firstly, a dialogue strategy of question selection is introduced. A decision algorithm is proposed to find the best questions to be asked to narrow down the candidates as quickly as possible according to the knowledge database. Then, a DP-based two-pass search strategy using the acoustic distance based on the phonetic confusion matrix is explained in details.

Chapter 5 presents my third research that practices signal processing technologies to emphasize the usability of spoken dialogue systems. Firstly, it discusses the role played in the spoken dialogue system by the prosody information. Then, a robust pitch detection method using speech signal processing technology ARSF is introduced.

Chapter 6 gives the conclusion. It summarizes the results obtained in my researches. Furthermore, the future dialogue system will also be described with the proposed technologies in this thesis and the concept of personalization.
Chapter 2

Key Technologies of Spoken Dialogue Systems and Related Works

In this chapter, the key technologies of spoken dialogue systems are described. First, the classic structure of spoken dialogue systems is presented in Section 2.1. Second, the technology development of each component are discussed in Section 2.2. Then, the issue in current representative commercial spoken dialogue systems is presented in Section 2.3. The related works are introduced as well.

2.1 The Structure of Spoken Dialogue Systems

The spoken dialogue system is composed of automatic speech recognition (ASR), natural language understanding (NLU), dialogue manager (DM), natural language generation (NLG) and text-to-speech synthesis (TTS), as shown in Figure 2.1. The ASR component receives a user’s spoken utterance and transforms it into a textual
hypothesis. Then, NLU parses the hypothesis and generates a semantic representation of the utterance. This representation is then handled by the DM component, which looks at the discourse and dialogue context to, for example, resolves anaphora and interprets elliptical utterances, and generates a response on a semantic level. Then the NLG component generates a surface representation, often in some textual form, and passes it to TTS which generates the audio output to the user.
2.2 The Technology Development of Spoken Dialogue System

2.2.1 Automatic Speech Recognition

The task of ASR systems is to interpreter acoustic speech signals into a sequence of words. ASR systems are separated in different classes by describing what type of utterances they can recognize.

1. Isolated word: Isolated word recognizers usually require each utterance to have quiet on both side of sample windows. It accepts single words or single utterances at a time.

2. Continuous speech: Continuous speech recognizers allows user to speak almost naturally, while the computer interpreters the content. Recognizer with continuous speech should utilize novel methods to determine utterance boundaries.

3. Spontaneous speech: Spontaneous speech recognizers try to recognize speech that is natural sounding and not rehearsed. ASR systems with spontaneous speech ability should be able to handle a variety of natural speech feature such as words being run together.

ASR systems need a set of models to compute probabilities for parameterizing the audio signal into features, and an efficient search algorithm. Gaussian mixture model (GMM) or hidden Markov models (HMMs) are utilized to represent the sequential structure of speech signals [13] [14].

Deep learning, referred as representation learning or unsupervised feature learning is a new area of machine learning (ML) recently. Deep learning is becoming a mainstream technology for speech recognition [8] [15] and has successfully replaced
Gaussian mixtures for speech recognition and feature coding at an increasingly larger scale.

### 2.2.2 Natural Language Understanding

NLU is an important field of natural language processing that deals with machine reading comprehension, with artificial intelligence and computational linguistics technologies.

A classic parsing approach of NLU is to use a context-free grammar (CFG) based grammar which is enhanced with semantic information. The same CFG may then be used for both ASR and NLU, which is famously implemented in the W3C-standard VoiceXML [16].

There have also been a lot of efforts in data-driven approaches to NLU to gain robustness. The opportunity for transfer of ideas between ML and NLP can be traced by the following algorithms:

- Conditional random fields for part-of-speech tagging [17]
- Latent Dirichlet Allocation for modeling text documents topics [18]
- Sequence-to-sequence models for machine translation [19]

### 2.2.3 Dialogue Management

As described in [20], the most common tasks of DM can be divided into three groups:

1. Contextual interpretation: Resolve for example ellipses and anaphora.
2. Domain knowledge management: Reason about the domain and access information sources.
3. **Action selection:** Decide what to do next.

The different knowledge sources needed by DM can be separated into dialogue and discourse knowledge, task knowledge, user knowledge and domain knowledge. User knowledge can be used to adapt the system’s behavior to the user’s experience and preferences. Domain knowledge management includes models and mechanisms for reasoning about the domain and for accessing external information sources, such as an SQL database for music information or a graphical information system. On the other hand, for complex and analytical tasks interactive systems like HITIQA [21], it especially meant for answering explanatory questions like: why, how, list etc.

Furthermore, the introduction of the framework of the reinforcement learning based on dialogue examples and reward, energetic research has been performed on the model, such as POMDP [10]. It should be noted that assumes interaction flow conditions, or information state, such as a slot for performing task. Model based on this machine learning interesting, it has been reported that high performance can be obtained in the evaluation by simulations. On the other hand, in the constructed model is completely black box, or found later trouble, even as you try to add a vocabulary and concepts, there is a problem that cannot be maintained.

### 2.2.4 Natural Language Generation

NLG is the process of automatically generating natural language on the basis of a non-linguistic information representation. This may be, for instance, information from a database or an abstract message specification provided by the dialogue manager of a dialogue system. Most works of NLG are aimed at the production of written texts. In recent years, a number of new template-based systems have seen the light, including TG/2 [22], Theune et al.[23], YAG [24].

Based on the development of statistical natural language analysis, data-driven methods have led to improvements of the performance of capabilities of NLG sys-
tems. The work in [25] described content selection by means of data-driven methods. Duboue and McKeown [26] developed a two stage approach to content selection based on statistical techniques. Their method employs clustering to derive content selection rules for the purpose of automatic generation of biographies. Ballesteros et al. proposed two approaches based on SVM classifiers to map semantic and syntactic structures [27] for NLG.

### 2.2.5 Text to Speech

TTS, which also called Speech synthesis, is the artificial production of human-being speech. The TTS procedure mainly consists of two phases, usually called natural language processing (NLP) or high level, and digital signal processing (DSP) or low-level synthesis. If the output is to be generated by an embodied conversational agent (ECA), facial animation is also needed [28].

In recent years, researches have been done to provide high quality synthetic speech with no compromising in terms of naturalness and intelligibility in TTS systems.

To attain the natural speech, different models are available such as text as language models, grapheme to phoneme models, full linguistic analysis model and complete prosody generation model.

In complete prosody generation model, the quantities like phrasing, stress are determined to generate naturalness bearing synthetic voice. Towards generating such a speech, an explicit prosodic model is required. HMM is one of the best models currently in use for most of the TTS [29] [30]. Though many researches have been done in this stream, better solution is required.
2.2.6 Emotion Recognition with Prosody Information

Speech is usually analyzed considering both: the prosodic information and the spoken content. Therefore, to deeply explore the superiorities of speech to other modalities such as touch screen or keyboards, some spoken dialogue systems intend to apply the recognition of the emotion of the speaker.

By analyzing the emotion from speech, dialogue agent or robot can estimate the human feelings, and performs a response answer suitable to the situation. For instance, Japanese telecom company Softbank employs robot “Pepper” to do shop reception, and declares that the future robots that can recognize human emotion will change the way we live and communicate [6].

Commonly the speech emotion is classified in categories. The number of categories is mostly 4 until 7, such as anger, boredom, disgust, fear, joy, neutral and sadness [31].

In the design of a speech emotion recognition system, an important issue is the extraction of suitable features that efficiently classifies different emotions. Many studies have shown that prosody information provide a reliable indication of the emotion [32] [33] [34]. The most commonly used prosody features in speech emotion recognition are as follows [32]:

- Fundamental frequency (calculated as the inverse of pitch period): mean, median, standard deviation, maximum, minimum, range (max-min), jitter, and ratio of the sample number of the up-slope to that of the down-slope of the pitch contour.

- Energy: mean, median, standard deviation, maximum, minimum, range (max-min), linear regression coefficients, shimmer, and 4th order Legendre parameters.

- Duration: speech rate, ratio of duration of voiced and unvoiced regions, and
duration of the longest voiced speech.

However, for emotions those have nearly the same arousal level often it is not possible to classify them correct. Features are needed that make it possible to differentiate emotions like anger and joy better, comparing with only prosody features. These features depend on the human voice and are called quality features. Therefore, formants feature set including: first and second formants and their bandwidths are also used for emotion recognition.

With the extracted features, GMM and support vector machine (SVM) are mainly used to correctly classify the emotion. Recently, HMM and deep neural network (DNN) are also applied [34] [35].

Among the prosody features, pitch information is essential. Pitch detection methods usually use short-term analysis techniques, which means that a score $f(T|x_m)$ is calculated by a function of the candidate pitch periods $T$ for every frame $x_m$. In speech processing literatures, a wide variety of pitch detection methods has been proposed. However, accurate and robust pitch detection in noisy environments still remains as a difficult and important issue in real application.

2.3 Remaining Issue and Related Works

By the analysis of the use situations of the commercial spoken dialogue systems, it is still difficult to say that spoken dialogue interface has deeply penetrated users’ daily life yet. In order to realize the habitual use of spoken dialogue systems especially for information search, the following approaches are proposed to solve this remaining issue:

- Designing attractive characters and dialogue scenario based on gamification theory and human centered design (HCD), which makes the users feel like speaking to.
• Improving dialogue management strategies to optimize question selection for easy information search and to decrease failures in information search mainly caused by mismemory and speech recognition errors.

• Estimating the prosody information including pitch to improve the usability of spoken dialogue systems.

For the related works of designing spoken dialogue systems, Yankelovich et al. analyzed the issues of designing speech interface in [36]. Besides the speech recognition errors, the natures of speech, such as the lack of visual feedback and the organization of information, also generate negative impact on the user experience for smartphones. It leads to the lack of motivation of using spoken dialogue interface. Several researches tried to use an agent to realize a natural spoken dialogue interface [37] [38], expecting to improve the motivation. In [37], a human-like dialogue agent was applied for museum guide. The results indicated that the agent was useful to engage people in dialogue interactions. The agent’s emotion and personality are verified to be important for natural dialogues in [38]. Also, some works were carried out to incorporate spoken dialogue into games in a useful and entertaining way. Bell et al. introduced an interactive computer game in which spoken and multimodal dialogue for dialogue data collection [39]. Minami et al. developed a quiz system using speech, sounds, dialogue, and vision technologies [40]. Both of the researches proved that the game elements motivate users to speak to the systems, which help to complete the dialogue tasks. However, the focus has not been on the approaches for the habitual use of a spoken dialogue interface in a long term.

For the related works of DM for information search, most spoken dialogue systems, such Apple’s Siri and Google’s voice search, are simply combining the speech interface with the search systems. The conventional search systems are mostly based on a domain-specific web search or full-text search [41] with the keywords. However, they cannot provide proper results unless those results include the keywords those the user has input. Furthermore, if the user’s information literacy is not high and
the target database is huge, the search effort for narrowing down the candidates will still be a great issue. In addition, for the robust search strategies against the queries those contain incorrect parts due to mishearing or misrecognition, the related works are investigated. Several researches attempted to use phonetic string matching methods to solve the search problem caused by misheard queries. They were verified to be more robust than the text retrieval methods. Ring and Uittenbogerd [42] tried to find the correct lyric by minimizing the edit distances between phoneme strings of queries and the lyrics. However, edit distance does not present the degree of confusability between phonemes. To model the similarity of misheard lyrics to their correct versions statistically, Hussein [43] introduced a probabilistic model of mishearing that is trained using examples of actual misheard lyrics from a user-submitted misheard lyrics website “kissthisguy”[44], and developed a phoneme similarity scoring matrix based on the model. The performance of this method depends on the size of the training database. As described in [43], a total number of 20788 pairs of the misheard lyrics and the correct lyrics are used. However, such a big database like “kissthisguy” is not available in other languages. It is impractical to collect sufficient misheard lyrics to build a practical probabilistic model. On the other hand, another important requirement for the search strategies of DM is that it must satisfy a real-time response considering the latency of spoken dialogue systems. In order to reduce the processing time, conventional high-speed DP matching processors use index or tree-structured data to pre-select the hypothetical candidates [45], [46]. As an example, [45] used a suffix array as the data structure and applied phoneme-based DP matching to detect keywords quickly from a very large speech database. In order to avoid an exponential increase in the processing time caused by increasing keyword length, it divided the original keyword into short sub-keywords. Then, it searched the sub-keywords on the suffix array by DP matching. If the DP distance between a sub-keyword and a path of the suffix array is not more than a predetermined threshold value, these paths remained as the candidates of search results. By repeating the DP matching process between the original keyword and
the candidates, the final result is detected. As well as other high-speed DP methods, the predetermined threshold for sub-keywords is proportional to the length of the queries. However, based on the author’s investigation, for some domains, such as lyric search, the distance values between the queries and the correct lyrics show no statistical relationship with the length of queries.

For the related work of pitch extraction in noisy environments, the commonly used detection methods are considered as the autocorrelation method (AUTOC) and the cepstrum method (CEP). The autocorrelation function in AUTOC is calculated by inverse Fourier transform of a squared amplitude spectrum of speech, i.e., power spectrum [47], [48]. CEP uses the logarithm of an amplitude speech spectrum [48]. They are verified as excellent detecting methods in clean speech. As it is well known, white noise distributes the energy uniformly along the frequency axis. It does not form prominent energy peaks. After exponential calculation on amplitude spectrum, the high energy parts which represent speech components are enhanced. Accordingly AUTOC has robustness against white noise. The authors in [49] proposed a new method as an extension of AUTOC. It adjusts the exponential factor of the amplitude spectrum according to SNR at each speech frame. As SNR decreases, the exponential factor is increased to distinguish the speech components from noise components. In a wide range of white noise, i.e., from low SNR to high SNR, the accuracy on pitch extraction is improved by this extension. On the other hand, logarithm prefers to extract the envelope of the amplitude spectrum. CEP is relatively robust against the noise whose energy is distributed in a narrow band, such as car noise. However, if the noise situation changes, the performances of the conventional methods are rapidly degraded. Since these conventional methods, i.e., AUTOC, provide excellent detection in clean speech, a process which can keep amplitude spectral structure close to the shape of clean speech against unspecific noise conditions is requested.
Chapter 3

Spoken Dialogue System Design for a Habitual Use

This chapter expands the discussion focusing on how to design a spoken dialogue system to be habitually used. Firstly, an in-house trial of spoken dialogue system is introduced. The discussion of the trial results evocates a utilization of character agent which is well designed for building intimacy. Secondly, the hypothesis that the impressions of the agent character have influences to user’s tolerance of dialogue system mistakes, is verified by the experimental results. Finally, a breeding game application with a dialogue agent is designed to realize a habitual use of the spoken interface for information search.

3.1 In-House Trial of Spoken Dialogue System

In order to investigate the use situation and issues of spoken dialogue systems, an in-house trial of a personal assistant application has been carried out. The application is based on a spoken dialogue system in a smartphone. It uses an dialogue agent to
assist users in operating the regular smartphone functions, such as contact, calendar, alarm, and a short message service in daily life. It also supports web search and weather forecast.

### 3.1.1 Structure of the Trial System

The spoken dialogue system used in the trial is based on a multimodal dialogue platform that runs stand alone on Android devices [50]. The platform acts as a middleware that can easily communicate with other native functions of Android devices. Figure 3.1 illustrates the architecture of the applied dialogue system. It comprises of four components. The “Dialogue Management” component analyzes what users want and responds with appropriate agent actions to the user. The “Understanding” component includes a lightweight natural language understanding engine, based on a large database of vocabularies and sentences concerning mobile operations and information search. The “Character Rendering” component implements the agent with a fully articulated 3D graphical body, which support PMD motion format [51]. The “Speech Synthesis” component uses a stand-alone HMM-based Japanese TTS engine “N2” [52]. Another feature of this dialogue system is that the users’ profile information is collected through daily dialogues. Base on the information, the proper responses are selected. To realize the speech input with a wide range of tasks, Android standard speech input module is used [53].

### 3.1.2 Trial Evaluation

Eight participants including four females and four males, who had an experience of using a smartphone, participated in the trial. Though the trial resulted that five participants gave good evaluations of operability, all participants used the application for less than three days. None of the participants continued using the application after one week. Furthermore, the most voted reason for not using the application...
was feeling shy about speaking, which was more than the votes for speech recognition errors. Most users reported that the motivation of speaking will increase if they become very familiar with the dialogue agent.

### 3.1.3 Discussion of the Spoken Dialogue Agent

The results of the trial bring a great challenge of how to motivate the user to use the dialogue interface habitually. The author proposes to introduce gamification into the design of spoken dialogue interface. The dialogue application is designed in the form of a breeding game using an animated character as the dialogue agent. The breeding
process is expected to make users be very attached to the agent, which increases motivation of speaking to it. This idea was inspired by the hit phenomenon of a breeding toy Tamagotchi that has been keeping people absorbed for a long period. The reason was explained in [54], that people have a high tendency to create and transfer their affections to the artificial companions regardless of the companions are devices or animal. Therefore, the affection on the character is expected to help and enhancing user’s tolerance against dialogue system mistake.

3.2 Agent Characters’ Impressions against the User’s Tolerance

So far, the applications using agent characters for interactive communications, which are based on the speech recognition technology, have been increasing [37] [38]. The reason using character is to engage people in dialogue interactions. Based on the in-house trial described in Section 3.1, the author sets up a hypothesis that the use of character agent enhances user’s tolerance against the dialogue system mistakes caused by ASR, NLU or other reasons. This section introduces an investigation that the impressions of the characters give influences to the user’s permissiveness to the mistakes of the spoken dialogue systems.

3.2.1 Preliminary Investigation for Selecting Characters

In order to determine the agent characters to be used in the experiment, a preliminary investigation is carried out with 25 participants, whose ages ranged from 23 to 53 years-old. For the nationality, 20 participants are from Japan, 2 participants are from the Republic of Korea and 3 participants are from the People’s Republic of China. Three types of human-type characters and five types of non-human-type characters are nominated as the targets. The participants were subjected to rank
the cuteness of the eight characters. As it is verified in another previous work of the author that there is a trend that the cute character makes the user more tolerant of the dialogue agent mistakes, both of a high level “cute” character and a low level one are selected. The character that received the highest evaluation is character b in Figure 3.2, and the character with the lowest evaluation is character a. Since many spoken dialogue systems trend to use human-type characters, a human-type character is added into the experiment in addition, which is character c in Figure 3.2. It’s evaluation is in the middle of character a and character b and does not have age-and gender-related changes.

3.2.2 Experiment for Evaluating Users’ Tolerance

With the selected characters, an experiment is carried out to evaluate the influence to users’ permissiveness. Twelve Japanese participants including 9 males and 3 females, whose ages ranged from 22 to 31 years-old, attended the experiment. They all have experiences of using spoken interfaces in mobile devices. However, they did not know the experiment purpose until it started. An operational support task and a quiz task were prepared for the evaluation experiment. These tasks are executed over the applications on a tablet device. The operational support task is highly expected to provide the proper answers according to the user’s purpose. On the other hand, the expectation of the quiz task for the proper answers is relatively low. Therefore the evaluation can be studied from the different aspects, in which the users request the correct answer in different level. A survey is carried out to the participants about “whether it could be tolerated when each character made a mistake in the answer”. Based on the participants’ answers, the degree of tolerance depends on whether the character is the non-human type or the human type.

Specifically, for the human type character, 42% of the participants in the operational support task and 50% of the participants in the quiz task answered “It couldn’t be tolerated.” On the other hand, more than 80% participants answered
“It could be tolerated.” in all tasks for the non-human type character. The reasons are summarized as follows:

- For the human type character: Compared with the non-human character, it is potentially regarded as a human. The expectations for the precision of search and the speech recognition are greater.

- For the non-person type character: as the appearance is different from human, it is revealed that the participants trend to tolerate the character’s mistakes with the impressions of “cute” and “innocent”.

Figure 3.2 Character agents used for the experiment
Chapter 3. Spoken Dialogue System Design for a Habitual Use

3.3 Design of User Interface for the Spoken Dialogue System Applying a Breeding Game

Based on the discussions and experiment results in the last two sections, my work proposes a design concept of breeding a dialogue agent, which is an animated non-human type character, through taking care and having dialogues, to establish intelligent and useful conversations. The stepwise growth of the agent creates users’ expectation and intimacy, which motivate them to use the spoken dialogue interface habitually.

In order to design the game efficiently according to the user experience on a smartphone, human-centered design (HCD) process is applied [55]. The design process and analysis of the real-world dialogue log data are described in this section.

3.3.1 Target Users

Instead of all smartphone users, the author narrowed down the target users into group with two specific features at the first stage of the game design. The first feature is that the users like breeding characters and the second one is that they accept speech interface. As investigated in [56], more than twice as many female users as male ones are playing breeding games. Furthermore, the author analyzed the users of a smartphone game Peratama [57], in which the users enjoy a character speaking words in a distinctive voice during the play. The character was designed in the shape of a egg named peratama that means “a talkative egg” in Japanese. Over 100 thousands users of Peratama, who are regarded to accept speech interface, were grouped by age and sex, which are shown in Figure 3.3. The proportion of female users in their 20s to 40s was 57.4%, which was the most of any user group. Based on the analysis, the target users are defined as 20s-40s female smartphone holders with the experiences of playing mobile games.
Furthermore, the author interviewed five females in their 20s who like playing mobile games, about their behaviors in using smartphones and playing mobile games. By capturing the interactions and activities involved, the author summarized the understandings of what the users wanted in the daily life.

**Understanding 1**: They like cute characters.

**Understanding 2**: If there is an element for collecting items in the game, they want to complete the collection to feel a sense of achievement.

**Understanding 3**: They want to play games in a short break or in a train without spared much effort.

In addition, in order to observe their behaviors related to speech interface, a total of 186 comments of *Peratama* were collected from Google Play, App Store, and visitor books from exhibitions. The most representative voices were selected: “I don’t like the unique progress process that a player cannot choose. The play style is so
humdrum.”; “It is not significant to see the character’s growth except the appearance change”; “The voice and the words of peratama are funny and sometimes heal my heart”. New understandings were discovered from these comments.

**Understanding 4**: They desire various types of interactions (play types).

**Understanding 5**: They expect significant changes in the breeding with an image of taking care of babies or pets.

**Understanding 6**: They tend to pursue the feel of healing from the breeding process and the characters.

### 3.3.2 Scenario Design

Based on all of the understandings of the target users, the original design concept is expanded into the subconcepts as below:

- **Subconcept 1**: Users take care of the dialogue agent to make it grow up. According to Understanding 1, 2 and 3, the character changes into various appearances with unique voices. The action of taking care is easy and able to get game incentives. The incentives are used to collect various items.
- **Subconcept 2**: According to Understanding 4, more than one type of play (taking care) will be prepared besides spoken dialogue. Users can freely choose the play type due to the situations. Meanwhile, the spoken interaction gets the greatest incentives.
- **Subconcept 3**: According to Understanding 5, users can obviously see the agent growing from the dialogue contents. The agent starts speaking baby talk then gradually becomes able to give intelligent conversation. Finally, it is able to help users via the dialogues, such as checking schedules or waking users up.
- **Subconcept 4**: According to Understanding 6, the dialogue should be fun and sometimes emotional. After the breeding process, users can still interact with
the agent to make users feel themselves are involved into the agent’s world.

Then, the scenario of a new breeding game with a spoken dialogue interface is designed. The game is called *Hey Peratama*, using the character peratama as the dialogue agent. The game scenario is mainly composed of five play scenes those are developed by the above subconcepts. In each scene, there are three scenario points, which are described below:

- **Scene 1 developed from Subconcept 1**: Users earn coins from breeding operations and collect the game items.
  - 1-1 Users are asked to breed peratama in a room. Users can take care of peratama by touching.
  - 1-2 Each touch earns coins. Users can buy different costumes for peratama with the accumulated coins. Once users dress up peratama with a new costume, peratama is endowed with a distinctive voice and appearance.
  - 1-3 The costumes are displayed in the silhouettes. Users can enjoy the imagination before they disclose the costumes.

- **Scene 2 developed from Subconcept 2**: Users can take care of peratama by accompanying peratama in executing events following assigned schedule.
  - 2-1 If there is an event, such as taking a walk or cooking, registered in peratama’s calendar, users can accompany peratama in executing the events. It is operated by playing a video of an event story.
  - 2-2 Coins can be saved by this operation.
  - 2-3 By collecting several new costumes, the intimacy level with peratama increases, which means peratama grows up. Consequently, the peratama’s talk grows more complicated.

- **Scene 3 developed from Subconcept 2 and 3**: Users can enjoy a conversation with peratama, which helps it grow and mature.
3-1 Users can talk with peratama.
3-2 Each conversation with peratama earns a lot of coins.
3-3 As the intimacy level increases, the conversation ability of the peratama gets better.

- Scene 4 developed from Subconcept 3: Via a conversation, peratama can assist users to register and to confirm the schedules.
  - 4-1 Users can also register their schedules in peratama’s calendar.
  - 4-2 Through conversations with peratama, users can register and confirm their own schedules.
  - 4-3 Once users register the schedules, it is possible to earn coins.

- Scene 5 developed from Subconcept 3 and 4: Users can use peratama in the situations of the daily life.
  - 5-1 Users’ favorite peratama can be placed on top of the home screen as a widget. Doing so lets users check the game status easily.
  - 5-2 After setting the alarm via conversation with peratama, users can be awakened by the voice of peratama.
  - 5-3 The images of the favorite peratama can be sent to friends over SNS or chatting apps.

### 3.3.3 Evaluation of Scenario-based Acceptability

In order to take the importance of context into account, scenario-based acceptability is evaluated. A total of 11 females 20-30 years-old, who are regarded as the target users, participated in the evaluation. Four were college students and the others were office workers. All users had smartphones and had experience playing mobile games. The evaluation was a one-on-one interview. First, the interviewer explained the scenario by a present sheet. The present sheet contained a detailed scenario description with users’ actions, screen sketch, and the scenario points which are mentioned in
Section 3.3.2. Then the interviewer asked the questions with a rating scale of acceptability for the scenario points of the five play scenes. The interview lasted one hour for each interviewee in order to minimize the burden on the interviewee.

All of the points were rated on a four-level rating scale: 1-Strongly disagree; 2-Disagree; 3-Agree; 4-Strongly agree. The average scores and standard deviation for each scenario point are presented in Table 3.1. Since most of the scenario points obtain high scores, our design was verified as proper for the target users.

Furthermore, comments from the interviews were collected, such as “I feel no stress in talking to the peratama because it is a game,”; “I will talk to the peratama a lot to earn coins to buy things,”; “I feel a sense of achievement when peratama’s speech level is raised.”; “I will try use the calendar function when I forget taking my schedule book”. The positive results and comments encouraged us and gave us hints to improve the scenario. Four revisions of the game scenario were carried out on the basis of the evaluation:

1. Scene5-3 got the lowest score of acceptability, which is lower than 2.5. It is because most users feel troublesome to search the figures while chatting. It was deleted from the scenario to improve the usability.
2. As shown in Table 3.1, uses are positive to the incentives (coins) in Scene1-2, 2-2, 3-3 and 4-3. The author arranged the number of coins earned through the game plays to encourage users to use speech input. Conversations with the peratama and execution of the alarm and calendar assistants will earn 50 coins, while touch operation only results in 1 coin.
3. Many users claimed that they want more game items besides costumes, which will motivate them to collect the coins and play more. The items including the breeding room’s interiors were added.
4. Since some comments said that, they want peratama to call their names and to say warm words in the special days, the author added a function that can memorize users’ personal information and reflect it into the dialogues.
Table 3.1 Average score and standard deviation values

<table>
<thead>
<tr>
<th>Scene No.</th>
<th>Average</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene1-1</td>
<td>3.55</td>
<td>0.50</td>
</tr>
<tr>
<td>Scene1-2</td>
<td>3.91</td>
<td>0.29</td>
</tr>
<tr>
<td>Scene1-3</td>
<td>3.00</td>
<td>0.95</td>
</tr>
<tr>
<td>Scene2-1</td>
<td>3.27</td>
<td>0.75</td>
</tr>
<tr>
<td>Scene2-2</td>
<td>3.73</td>
<td>0.75</td>
</tr>
<tr>
<td>Scene2-3</td>
<td>4.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Scene3-1</td>
<td>3.82</td>
<td>0.39</td>
</tr>
<tr>
<td>Scene3-2</td>
<td>3.82</td>
<td>0.39</td>
</tr>
<tr>
<td>Scene3-3</td>
<td>3.91</td>
<td>0.29</td>
</tr>
<tr>
<td>Scene4-1</td>
<td>2.86</td>
<td>1.17</td>
</tr>
<tr>
<td>Scene4-2</td>
<td>2.82</td>
<td>1.11</td>
</tr>
<tr>
<td>Scene4-3</td>
<td>3.64</td>
<td>0.64</td>
</tr>
<tr>
<td>Scene5-1</td>
<td>3.55</td>
<td>0.66</td>
</tr>
<tr>
<td>Scene5-2</td>
<td>3.09</td>
<td>1.00</td>
</tr>
<tr>
<td>Scene5-3</td>
<td>2.45</td>
<td>1.08</td>
</tr>
</tbody>
</table>

3.3.4 Game Implementation

The game is implemented as an Android application. Screenshots of the game are presented in Figure 3.4. In order to maintain high motivation and prevent users from becoming bored with the game, the game interfaces and progress followed the gamification theory and usability standards in [58].

1. The game provides clear goals for users: the goals are divided into the short-term goal, midterm goal, and long-term goal. The short-term goal is to collect the coins by touching the peratama in the standby scene (Figure 3.4 (a)), accompanying the event (Figure 3.4 (b)), and talk training (Figure 3.4 (c)),
which are easy and cost little in terms of time. Once the goal is reached, the midterm goal is to collect the necessary costumes to increase the intimacy level with the peratama (Figure 3.4 (d)). Then, the long-term or final goal is to keep increasing the intimacy level to make the peratama grow and increase the dialogue maturity and the assistant abilities (Figures 3.4 (e) and (f)).

2. The first experience is encouraging and the play cycle makes users experience constant stress until the goal was achieved: the *Hey Peratama* game progressed rapidly in the early stages to make users get interested. However, the numbers of events and operations involving talking and touching per day were restricted, which give users moderate stress in daily play.

3. Consistently visualizing the user’s situation in the game and giving feedback are important: every time the intimacy level went up, a short tutorial explained what was new and what to do next. Sound effects were also provided after every operation.
Figure 3.4 Screenshots of *Hey Peratama*
3.3.5 Analysis of the Real-World Dialogue Log Data

The game application was released on Google Play [59]. There were 415 downloads in one week after the release. Over 23% of the total users played the game per day. All of them actively talked to peratama in the game. Comparing with the low motivation in the in-house trial mentioned in Section 3.1, 26% of users have played the game for more than three days. Many users commented that they hoped to talk more than the restriction. It indicated that our design was effective. Based on the analysis of log data, over 95% questions from peratama are answered. Furthermore user-initiated dialogues including chatting, weather checking, confirming schedule, also activated. It is proved that continual system-initiated dialogues built intimacy, which is activating user-initiated dialogues.

Furthermore, the author investigated the distribution of the number of utterance and the details of the utterance contents to understand how the user was using speech input. The analysis is based on the utterances from 4910 real world users. Figure 3.5 shows that there are 3 peaks of the number of utterances 1) from 7 o’clock to 8 o’clock, 2) from 20 o’clock to 21 o’clock, 3) from 0 o’clock to 1 o’clock. This distribution stays invariant in the period of one year. These time zones are around the time of getting up, the time of going home and the time of going to bed. Therefore these dialogues mainly took place at the users’ home. The tendency of the speech contents between the user and peratama is also analyzed. The classification of speech contents uses a bag-of-words based speech intention estimation for each utterance. Figure 3.6 shows the percentage of each utterance content in each time zone of 7 o’clock, 20 o’clock and 0 o’clock. Around 7 o’clock the topics were about weather and time, while the confirmation of the schedule from the users also occupied a relatively high proportion. From 20 o’clock to 21 o’clock, the chit-chats and the questions related to peratama took place in the dialogues most. Around 0 o’clock, a larger number of utterances contained the similar contents with those around 7 o’clock, by adding fortune-telling and asking the date.
Based on the above discoveries, it is supposed that a function that pushes a proper dialogue topic according to the time, will be effective for prompting users to use the dialogue system more habitually.
Chapter 3. Spoken Dialogue System Design for a Habitual Use

3.4 Summary

This chapter presents the studies on how to design a spoken dialogue system to be habitual use for information search. Based on the investigations of an in-house trial of spoken dialogue system and the analysis of agent characters’ impressions against the user’s tolerance, human-centered design and gamification theory are practiced to design a breeding game. A novel design concept of breeding a character through taking care and having a dialogue was proposed in order to make the users gradually feel that speaking to the dialogue agent is natural and fun. The designed scenario obtained high evaluation scores from the subjective evaluation experiments. After the release of the game on Google Play, the analysis of the real-world users also
positively supported our design, in which over 23% users are keeping doing spoken dialogue interactions in the game continually.
Chapter 4

Dialogue Management for Information Search

This chapter introduces the second point of my research, which is the work for optimizing the dialogue management for robust information search. As is known, an important use of spoken dialogue system is to obtain the information by natural utterances. Especially, it is regarded as a proper interface for the information illiterate. The general goal of this work is to realize effective interaction and robust search between people and the information resources such as database or internet.

First, a knowledge-based spontaneous dialogue system with the strategy of optimal question selection is proposed to assist the users’ operations. The task is targeted for searching recipes, as the cooking-related applications and researches have become more popular recently [60] [61] [62]. The proposed dialogue strategy actively asks the users questions to help them easily select the satisfactory recipes from among thousands of potential options. It aims to help the users, especially the novices who do not have abundant knowledge about recipes. Based on the results of evaluation experiments, it is concluded that the application with the proposed strat-
Another issue of the spoken dialogue systems for information search is the mistaken queries. A great amount of queries contain incorrect parts due to speech recognition errors as well as users’ mishearing or mis-memory. In order to improve the system performance for information search with these incorrect queries, a robust and fast phonetic string matching method is proposed in my work. Experimental results show that the proposed search method reduced processing time by more than 86.2%, compared with the conventional methods for the same search accuracy in the task of searching lyrics with the real-world misheard queries. Furthermore, some researches show that the recognition errors in unlimited-vocabulary speech recognition include a great amount the misrecognized words with similar morpheme strings, such as foreign names [63] [64]. It infers that the proposed search method is effective for recovering the search failures caused by speech recognition errors in spoken dialogue systems.

4.1 Dialogue Strategy of Optimal Question Selection

This section proposes a novel dialogue strategy to easily select the satisfactory and proper information from among thousands of potential options. It is applied in a recipe search application on mobile devices and helps the users especially the novices who do not have abundant knowledge about the recipes.

The application with the proposed dialogue strategy asks the users a series of questions related to various cooking categories including recipe genres and cooking needs in order to narrow down the potential recipes that meet the users’ wants. A decision algorithm is applied to find the best question to be asked to narrow down the
candidates as quickly as possible according to the recipe database. Furthermore, the questions responding to the recipes that the users viewed during the search process are preferentially selected.

4.1.1 Spoken Dialogue System for Information Search in a Cooperative Manner

The interface of the proposed recipe search application is shown in Figure 4.1. Area A is for user operation, and supports typing, speech input and button-press input. A brief operation guidance is also shown here. Area B is for showing the search keywords by which the results are refined. Area C is for displaying the search results: recipes with introduction in text and pictures. The application can “speaks” over the device’s speaker using a speech synthesis engine. The words are also displayed in a balloon.

To simplify the search process for the novice users, the proposed recipe search application employs a spoken dialogue system. The dialogue management is designed as a mixture of a user-initiative model and a system-initiative model. The flowchart of the dialogue scenario is shown in Figure 4.2. First, the application prompts an open question to ask the user, “What kind of recipes are you looking for?” The user can answer for example, “A recipe using chicken and cabbage”. The recipe results including those keywords are then refined. At the same time, the application prompts, “Please press here if confused by the number of recipe results” and displays a button. Once the user presses the button, the application starts to ask the user a series of questions related to the recipe categories, such as “Do you use an oven?” The user then simply presses a button or speaks into the microphone to choose yes or no, and the recipe results are consequently narrowed down until the user finds a satisfactory recipe. This design is intended to assist the user who has not prepared many keywords but expect an easy search. Even while being asked questions, the
user is allowed to view the details of recipes and freely input keywords and sentences to narrow down the number of recipe results.

Figure 4.1 Interface of the proposed application
4.1.2 Question Selection for Quick Narrowing Down of Hit Data

In order to minimize the number of search refinement steps, the question selection in the spoken dialogue system is based on the information gain according to the remaining recipes. The information gain is calculated by Equations 4.1 and 4.2, using the decision algorithm of iterative dichotomiser 3 [65].

\[
H(S) = - \sum_{x \in X} p(x) \log_2 p(x) \tag{4.1}
\]
Chapter 4. Dialogue Management for Information Search

\[ IG(A) = H(S) - \sum_{t \in T} p(t)H(t) \tag{4.2} \]

Entropy \( H(S) \) represents the average amount of uncertainty in the recipe set \( S \). \( p(x) \) is the proportion of the number of elements in attribute (or question) \( x \) to the number of elements in \( S \). \( T \) represents the subsets created from splitting set \( S \) by attribute (or question) \( A \). \( p(t) \) represents the proportion of the number of elements in \( t \) to the number of elements in set \( S \). Information gain \( IG(A) \) is the difference in entropy from before to after the set of recipes \( S \) is split on an attribute (or question) \( A \). So after each refinement, information gain is calculated for each remaining question. The question related to the category with the largest information gain is used on this iteration.

Furthermore, as users usually check a number of recipes that they are interested in before they make the final decision. Preferentially selecting the questions responding to the recipes that users are viewing is supposed to be helpful to reflect the user’s potential search purpose. Therefore, \( P(A) \) is induced as a weight of \( IG(A) \) in Equation 4.3, which is the proportion of the number of recipes categorized by \( A \) that have been viewed by the user during the search process to the number of all recipes that have been viewed. \( r \) is the smoothing parameter. The question related to the category with the highest \( PIG(A) \) value is regarded as the optimal candidate that keeps balance between the search efficiency and the users’ personalization. After each refinement, \( PIG(A) \) is calculated for each remaining question. The question with the largest \( PIG(A) \) is used on this iteration.

\[ PIG(A) = (P(A) + r) \times IG(A) \tag{4.3} \]

Based on a questionnaire of the users and an investigation of recipe websites, 81 recipe categories of cooking purposes are classified. They correspond to the cooking situations, cooking utensils, cooking methods, ingredients genres, cooking needs, cooking difficulty, dish origins and dish genres. Besides the normal categories, the
categories adapted to spoken queries, such as outdoor, diet and saving electricity are also included.

The learning algorithm named confidence-weighted linear classification (CW) [66] is used for classification and estimation. It allows for recipes to belong to multiple categories. The TF-IDF [67] value of nouns and adjectives in the texts of recipes are selected to build a bag-of-words model. By an evaluation involving 5-fold cross validation on 81 categories (on average each category uses 777 recipes for a test set), the average accuracy of classification is 94%.

4.1.3 Experimental Evaluations

To evaluate the usability of the proposed application, it was compared with a conventional keyword-based search application (Application1). Furthermore, a major commercial recipe search application (Application2) was also compared as a reference. The mobile device in the experiment was Google Nexus 7 tablet (OS: Android 4.1). The comparisons of subjective evaluations and operating time were both carried out.

Experiments were conducted with 7 participants (3 females, 4 males) aged from 20 to 40 years-old, including engineers and clerks. All of them were experienced smartphone users and the novice users of recipe search services. Application1 is almost the same with the proposed application only except the dialogue interface. It only allows keyword input with a keyword suggest function. Both the proposed application and Application1 stored 54,277 recipes in Japanese and they are implemented on the open-source search engine [68]. The Application2 is a widely-used recipes search application with a commercial database over 1 million recipes. All of the applications support speech input, in which the same Android standard speech input module is used [53]. The experiments were carried out in a quiet room. The participants were asked to imagine that they were in their own living room, and
trying to search for a satisfactory recipe with the mobile devices for cooking a dinner one hour later. No time limitation was set. For each application, a list of two designated tasks and one free task was given in which the participants were asked to finally decide on one recipe for each of these requirements. For examples, one designed task is searching a recipe of western food with spinach as an ingredient. They were allowed to add or change keywords that were related to the cases given. Before participants started the experiment, a brief guidance video was shown, and a few minutes of preliminary practice were allowed. The order in which each application was used was randomly determined for each participant.

Once the participants finished the operations of an application, they were asked to rate six aspects of their experience by using a 5-point Likert scale, on which 5 points were awarded for the most satisfactory experience, and 1 point for an unsatisfactory experience. The list of aspects is as described below:

1. Did you feel easy to operate the application?
2. Did you become clear about the interface after read the guide of the application?
3. Did you feel smooth to execute the next operation when you try to find the recipe?
4. Did you feel easy when you were considering the keywords?
5. Did you feel ok with the number of operations for inputting a keyword?
6. Are you satisfied with the recipe that you decided?

The average scores are shown in Table 4.1. The greater the score is, the more positive the participant evaluates the application. Besides the score, the participant’s free comments of each aspect are also recorded. On aspect 1, the proposed
application achieved worse scores than the others. It is because the operation on spoken dialogue interface is quite new for the participants, getting used to when and how to answer was not easy. On aspect 2, Application1 achieved the best evaluation because of its simplest interface. However, the participant most positively evaluates the proposed application on aspect 3. The reason is supposed to be the speech guidance and the simple operation to choose yes or no to refine the search results. On the other hand, the proposed application received higher scores than other conventional applications on aspects 4 and 5. It is verified that the strategy using dialogue interface to simplify and to refine the search is effective, since it saved the participants' efforts to reconsider the keywords. Finally, the proposed application achieved the best score on aspect 6. It is ascribed to the proper question selection that provides useful keywords and information for the participants to search recipes. In addition, especially for the proposed application, an aspect “Did you feel the question asked by application helpful?” was asked. The average score is 4, which is relatively positive. The operating times including both real times (the real-world elapsed time) and sensory times (the elapsed time estimated by the participants) are also recorded. As shown in Table 4.2, the proposed application costs more time than others, as it speaks to the participants to ask the questions. However, different from Application2, the sensory time of the proposed application is shorter than the real time. It reflects that the participants felt less stress when using the proposed application.

Based on the results of the evaluation experiments, it is concluded that the proposed application performs better than the conventional applications in terms of the satisfaction with search results and the effort spent for reconsidering search keywords.
### Table 4.1 Results of subjective evaluation

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Score</th>
<th>Proposed Application</th>
<th>Application 1</th>
<th>Application 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspect 1</td>
<td>4.29</td>
<td>4.43</td>
<td>4.43</td>
<td></td>
</tr>
<tr>
<td>Aspect 2</td>
<td>4.29</td>
<td>4.71</td>
<td>3.57</td>
<td></td>
</tr>
<tr>
<td>Aspect 3</td>
<td>4.29</td>
<td>3.86</td>
<td>4.14</td>
<td></td>
</tr>
<tr>
<td>Aspect 4</td>
<td>3.29</td>
<td>3.00</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td>Aspect 5</td>
<td>3.71</td>
<td>3.57</td>
<td>3.29</td>
<td></td>
</tr>
<tr>
<td>Aspect 6</td>
<td>4.14</td>
<td>3.86</td>
<td>3.86</td>
<td></td>
</tr>
</tbody>
</table>

### Table 4.2 Average operating time for one search task

<table>
<thead>
<tr>
<th>Time (m)</th>
<th>Proposed Application</th>
<th>Application 1</th>
<th>Application 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Time</td>
<td>3.80</td>
<td>3.74</td>
<td>3.47</td>
</tr>
<tr>
<td>Sensory Time</td>
<td>3.14</td>
<td>2.95</td>
<td>3.55</td>
</tr>
</tbody>
</table>

### 4.2 Information Search by Robust and Fast Phonetic String Matching Method

As one of my work above has proved that applying spoken dialogue system makes information search easier, another research work focuses on improving the robustness of the spoken dialogue system for information search. Information search by voice is the technology underlying many spoken dialog systems those enable the users to access information by spoken queries. The information normally exists in a large database, and the query has to be compared with a field in the database to obtain the relevant information. Different from the spoken dialog technology for ATIS [4] style systems which emphasizes detailed semantic analysis, a robust search method is more essential for the dialogue management of information search. It needs to be robust to the unique conditions in the real world:
Incorrect queries due to users’ mishearing or mis-memory

Incorrect queries with ASR errors

My work proposes a robust and fast search method to improve the performance of spoken dialogue system with these incorrect queries. To achieve the robustness, “acoustic distance”, which is computed based on a confusion matrix of an automatic speech recognition experiment, is originally introduced. Then acoustic distance is applied into Dynamic-Programming (DP)-based phonetic string matching to identify the target contents that the incorrect queries refer to. Furthermore, as the latency of the spoken dialogue system is important, a two-pass search algorithm is proposed to realize real-time execution. The algorithm pre-selects the probable candidates using a rapid index-based search in the first pass and executes a DP-based search process with an adaptive termination strategy in the second pass.

The proposed search method is implemented for music information retrieval (MIR). Many commercial systems accept diverse queries by text, humming, singing, and acoustic music signals. Among these types of queries, text queries of lyric phrases are commonly used (lyric search) [69]. The effectiveness of lyric search systems based on full-text retrieval engines or web search engines is highly compromised when the queries of lyric phrases contain incorrect parts due to mishearing. Experimental proves the proposed search method to be the most practical solution for these incorrect queries, considering the trade-off between high search accuracy and low computation complexity.

4.2.1 Analysis of the Real World Queries

Several investigations were carried out on the collected real world queries. The statistical features of the queries and some issues peculiar to lyric search are presented in this section.
To analyze the queries of lyric phrases for MIR in the real world, a preliminary analysis investigated major Japanese question & answer community websites, “ok-wave” [70] and “oshiete goo” [71]. It was found that many questions used lyric phrases to request the names of songs and singers. As 1140 queries of lyric phrases asked by various questioners were collected, each query is compared with its corresponding lyric to categorize whether lyric phrases in the query are correct or not (correct query or incorrect query) and how they were mistaken. The lyrics and queries are written in Japanese or English, or a mixture of both.

Figure 4.3 shows the distribution of incorrect queries in the different types and correct queries within the collected data. The incorrect queries, which make up around 79%, are classified into the following types:

- Confusion of notations: Chinese characters in the queries are substituted for reading symbols (kana), and vice versa.
- Function-word-error: Only the function words (such as prepositions, pronouns, auxiliary verbs), which have little lexical meaning, are mistaken in the queries.
- Content-word-error: The content words (such as nouns, verbs, or adjectives), which have stable lexical meanings, are mistaken in the queries.
In the current full-text search methods, function-word-error and confusion of notations can be handled using a stop word list for filtering out the function words.
On the other hand, as the content words play more important roles in determining the search intention [72], content-word-error queries were further categorized into three subtypes, namely “acoustic confusion”, “meaning confusion” and “others”. The percentages and examples are listed in Table 4.3. The mistaken parts are marked in bold.

Acoustic confusion is defined as a replacement of a word with that of a similar pronunciation; or a replacement of the words of unknown spelling with reading symbol strings of a similar pronunciation. For the first example of acoustic confusion queries in Table 4.3, “/kotoganai/” and “/kotobawanani/” have similar pronunciations while the character strings have no common parts. In the second example, the Japanese syllable (kana) string is used as a query whose pronunciation is similar to the English phrase, “You’ve been out riding fences for so long now” in the target lyric. This was assumed to happen when users were not able to spell the foreign words that they heard in a song.

Meaning confusion is defined as a replacement of a word with its synonym or near-synonym. As shown in Table 4.3, in the first example of meaning confusion queries, “/anata/” is mistaken for “/kimi/”. Both of the terms refer to the same meaning “you” in Japanese. For the second example, “/tsuki/” and “/hoshi/”, which mean “moon” and “star”, are confused.

The “others” type contains word insertion, word deletion and other errors in the queries. From the analysis of collected examples, it is known that mistakes in the “others” type are caused by a variety of reasons, which include individual experiences or memories and other reasons. The analysis did not find a relationship between the mistakes and the lyrics.

As the acoustic confusion queries occupy about 19.3% of the collected queries (45.0% of content-word-error queries), it remains an important issue for lyric search.
### Table 4.3 The distribution of mistaken types within content-word-error cases

<table>
<thead>
<tr>
<th>Types of queries</th>
<th>Percentage</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Acoustic confusion</strong></td>
<td>19.3% (220 queries)</td>
<td></td>
</tr>
<tr>
<td>Ex. 1</td>
<td>Text (Japanese): 好きな事がない</td>
<td>好きな言葉は何</td>
</tr>
<tr>
<td></td>
<td>Pronunciation: /suki na kotogani/</td>
<td>/suki na kotobawanani/</td>
</tr>
<tr>
<td></td>
<td>Meaning: There is nothing I like.</td>
<td>What are your favorite words?</td>
</tr>
<tr>
<td>Ex. 2</td>
<td>Text: You’ve been out riding fences for so long now</td>
<td>ユーベーナウプラウドゥンシーンクスソーセングナウ</td>
</tr>
<tr>
<td></td>
<td>Pronunciation: /yumbihiiNNautora:idINgnu:eNNs hinzfoosoroNNguna:/</td>
<td>/yumbee naugurasu duNNchihNNkawusooseNNguna/</td>
</tr>
<tr>
<td></td>
<td>Meaning: You’ve been out riding fences for so long now</td>
<td>* no actual meaning</td>
</tr>
<tr>
<td><strong>Meaning confusion</strong></td>
<td>7.3% (83 queries)</td>
<td></td>
</tr>
<tr>
<td>Ex. 1</td>
<td>Text (Japanese): 君には何でも話せるよと</td>
<td>あなたには何も話せるよと</td>
</tr>
<tr>
<td></td>
<td>Pronunciation: /kimiiwanaNNdemohamase-yoo/to/</td>
<td>/aminanawenNNdemohamase-yo-to/</td>
</tr>
<tr>
<td></td>
<td>Meaning: I can say anything to you</td>
<td>I can say anything to you</td>
</tr>
<tr>
<td>Ex. 2</td>
<td>Text (Japanese): 月に願いを</td>
<td>星に願いを</td>
</tr>
<tr>
<td></td>
<td>Pronunciation: /hoshi ninegai /</td>
<td>/ho shin okochi NN</td>
</tr>
<tr>
<td></td>
<td>Meaning: pray to the moon</td>
<td>pray to the star</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td>16.3% (186 queries)</td>
<td></td>
</tr>
<tr>
<td>Ex. 1</td>
<td>Text (Japanese): 星から来た 子の見る夢は</td>
<td>星の子チョビンの見る夢は</td>
</tr>
<tr>
<td></td>
<td>Pronunciation: /ho shi karakita kono mijiru yume wa /</td>
<td>/ho shi kochohinNNkono mijiru yume wa /</td>
</tr>
<tr>
<td></td>
<td>Meaning: The dream that the child who came from the star has</td>
<td>The dream that child Chobin of the star has</td>
</tr>
</tbody>
</table>

My research focuses on the solution to the acoustic confusion issue for lyric search. The average length of 220 acoustic confusion queries collected is about 6 words. The word error rate of incorrect words is about 53.1% (the insertion errors were not included). In the text retrieval field, some fuzzy matching algorithms, such as Latent Semantic Indexing (LSI) and partial matching, were used by major commercial Web...
search engines [74] to improve the robustness against incorrect queries. Thereby, a search test was carried out to evaluate how robust web search engines are against 220 acoustic confusion queries collected. The test results are shown in Table 4.4. The number of “hits” is equal to the number of webpages mentioning the target lyric that are included in the top 20 results returned by a search engine. Correct queries mean the correct versions of the incorrect queries. Comparing the number of hits with the correct queries, the performance of both web search engines are severely degraded in case of incorrect queries.

According to this result, identifying a lyric containing the most similar part in the acoustic aspect of the query is expected to be a better solution for acoustic confusion than focusing on the textual or the semantic aspects.

Table 4.4 Number of hits by two web search engines

<table>
<thead>
<tr>
<th>web search engines</th>
<th>Web Search Engine 1</th>
<th>Web Search Engine 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>220 correct queries</td>
<td>175</td>
<td>157</td>
</tr>
<tr>
<td>220 incorrect queries</td>
<td>27</td>
<td>16</td>
</tr>
</tbody>
</table>

### 4.2.2 Investigation of the Relationship between the Query Length and DP Matching Pre-selection

Furthermore, the investigation of the relationship between the lyric query length and DP matching pre-selection is carried out. As introduced in Section 2.3, the conventional high-speed DP-based search method usually contains a pre-selection approach. It prunes out the improbable search paths by comparing the DP matching distance with the predetermined threshold proportional to the length of queries. To find out whether it is a practicable approach to the lyric search problem, the
DP matching distances between the queries and the correct lyric are also analyzed. Figure 4.4 shows the distribution of the analysis data. The horizontal axis is the phoneme number of each query, representing the length of the queries. The vertical axis is the DP matching distance between the queries and the correct lyrics. Figure 4.4 shows that the phoneme number of the queries is distributed in a broad range from 5 to 57. In addition, the distance values between the queries and the correct lyrics show no statistical relationship with the length of queries. Thereby, it is practically difficult for the conventional method, such as the one in [45], to find the appropriate threshold based solely on the length of queries.
Figure 4.4 The distribution of the length of a query in phonemes and the DP matching distances from the correct lyric for the real world incorrect queries

### 4.2.3 Acoustic Distance Derived from a Phoneme Confusion Matrix

In this section, the proposed acoustic distance is presented. Acoustic distance between two strings is calculated by DP matching with cost values derived from phonetic confusion probabilities instead of a constant cost value used for edit distance.
First, a phonetic confusion matrix is obtained by running a phoneme speech recognizer over a set of speech data and aligning the phoneme strings of recognition results with reference phoneme strings, which uses the same speech recognition experiment as in [75].

For the elements of the confusion matrix, \( g(p, q) \) means the number of instances of phoneme \( q \) obtained as recognition results by the actual utterances of phoneme \( p \). As “\( \phi \)” represents a null, \( g(\phi, p) \) means the number of instances of the wrongly inserted phoneme \( p \) (insertion) and \( g(p, \phi) \) means the number of instances of the deleted phoneme \( p \) (deletion). \( U \) represents the set of 37 phonemes including null.

For each phoneme \( p \), the phonetic confusion probabilities of an insertion \( P_{ins}(p) \), deletion \( P_{del}(p) \) and substitution for phoneme \( q \) \( P_{sub}(p, q) \) are calculated on the basis of the confusion matrix elements, by Equations 4.4 to 4.6.

\[
P_{ins}(p) = \frac{g(\phi, p)}{\sum_{k \in U} g(k, p)} \quad (4.4)
\]
\[
P_{del}(p) = \frac{g(p, \phi)}{\sum_{k \in U} g(p, k)} \quad (4.5)
\]
\[
P_{sub}(p, q) = \frac{g(p, q)}{\sum_{k \in U} g(p, k)} \quad (4.6)
\]

As a large value of \( P_{ins}(p) \) represents high confusability for an insertion of \( p \), it corresponds to the low cost of an insertion operation for \( p \) in string matching based on DP. Therefore the value of insertion cost \( C_{ins}(p) \) is calculated by Equation 4.7. In the same way, the value of deletion cost \( C_{del}(p) \) and substitution cost \( C_{sub}(p, q) \) are calculated from the corresponding phonetic confusion probabilities by Equations 4.8 and 4.9.

\[
C_{ins}(p) = 1 - P_{ins}(p) \quad (4.7)
\]

\[
C_{del}(p) = \frac{g(p, \phi)}{\sum_{k \in U} g(p, k)}
\]
\[
C_{sub}(p, q) = \frac{g(p, q)}{\sum_{k \in U} g(p, k)}
\]
Chapter 4. Dialogue Management for Information Search

\[ C_{\text{del}}(p) = 1 - P_{\text{del}}(p) \]  
(4.8)

\[ C_{\text{sub}}(p, q) = 1 - P_{\text{sub}}(p, q) \]  
(4.9)

Second, with the calculated cost values, edge-free DP matching between the phoneme strings \( S_1, S_2 \) is carried out by Equations 4.10 to 4.12. Here, \( S[x] \) is \( x \)th phoneme of phoneme string \( S \) and \( \text{len}(S) \) means the length of \( S \) \((S_1, S_2 \in S)\). \( D(i, j) \) designates the minimum distance from the starting point to the lattice point \((i, j)\). \( D_{S_1, S_2} \) is the accumulated cost of DP matching between \( S_1 \) and \( S_2 \), which is defined as the acoustic distance. It reflects acoustic confusion probability for each phoneme.

1. Initialization:

\[ D(0, j) = 0 \ (0 \leq j \leq \text{len}(S_2)); \]  
(4.10)

2. Transition:

\[ D(i, j) = \min \begin{cases} 
D(i, j - 1) + C_{\text{ins}}(S_2[j]) \\
D(i - 1, j - 1) + C_{\text{sub}}(S_1[i], S_2[j]) \\
D(i - 1, j), \ (i f \ S_1[i] = S_2[j]) \\
D(i - 1, j) + C_{\text{del}}(S_1[i]) 
\end{cases} \]  
(4.11)

3. Determination

\[ D_{S_1, S_2} = \min \{D(\text{len}(S_1), j)\} \ (0 < j \leq \text{len}(S_2)); \]  
(4.12)
4.2.4 Fast Two-pass Search Algorithm in Consideration of Acoustic Similarity

Another important requirement for the spoken dialogue system and information search is to satisfy a real-time response. As the search algorithm of phonetic string matching methods is based on exhaustive DP matching, the computational complexity results are in the order of $m \times n \times I_t$ per query. Here $m$ is the length of the query, $n$ is the average length of a lyric and $I_t$ is the number of lyrics to search. Since commercial MIR systems usually provide hundreds of thousands of lyrics, the computational complexity is too high to realize a real-time search.

Therefore, a two-pass search strategy is proposed to be used in the DP-based phonetic string matching, which is based on acoustic distance, in order to realize a real-time search. It is realized through the following steps: off-line index construction, a rapid index-based search in the first pass and a DP-based search process with an adaptive termination strategy in the second pass.

Preliminary indexing is done as an off-line process. Theoretically, DP matching computation for the acoustic confusion distance between queries and lyric text should be done beforehand. However, this is impossible in reality because the number of query patterns is too large to be predicted.

An inverted index construction is preliminarily incorporated for the first pass search. The whole lyric set $L_t$ are converted into syllable strings using a morphological analysis tool such as Mecab [76]. Here $I_t$ represents the number of lyrics in the whole set. The syllable strings are converted into phoneme strings by referring to a syllable-to-phoneme translation table. Consequently, a phoneme string $S_{L(k)}$ represents a lyric $L(k)$ ($L(k) \in L_t$). Here $k$ is the lyric number. On the other hand, a list of linguistically existing units of $N$ successive syllables (syllable $N$-gram) $A_1 \cdots A_n$ are collected from the lyric corpus. The units are organized as index units for fast access, as shown in Figure 4.5. The acoustic distance $D_{S_{A_n}, S_{L(k)}}$ between the
phoneme strings of $A_n$ and $L(k)$ are pre-computed by Equations 4.10 to 4.12 and stored in the index matrix. It can be regarded as an index of acoustic confusion.

For the search process, firstly, by accessing the index described above, the first pass with a fast search is realized using the following steps. The flowchart is shown in Figure 4.5:

1. The input query $Q$ is converted into a syllable string $v$ by Macab.

2. By Equation 4.13 the syllable string is converted into syllable $N$-gram sets, $V_1, \ldots, V_m, \ldots, V_M$. Here, $v[m]$ is the $m$th syllable of $v$.

$$V_m = \{v[m], v[m+1], \ldots, v[m+N-1]\}; \quad (4.13)$$

3. $V_1, \ldots, V_m, \ldots, V_M$ are matched with the index units $A_1, \ldots, A_n, \ldots$. By accumulating the pre-computed and indexed distance values $D_{S_{A_n}, S_{L(k)}}$, the approximate acoustic distance $R(k)$ is calculated by Equation 4.14.

$$R(k) = \sum_{m=1}^{M} D_{S_{A_n}=V_m, S_{L(k)}} \quad (4.14)$$

4. To narrow the search space of lyrics, $L(k)$ with higher $R(k)$ is pruned off, and a lyric set $L_i$ containing $I_c$ ($I_c < I_l$) as the best lyric candidates is preserved for the second pass.

As seen in the four steps, the order of the syllable $N$-grams is not considered in the first pass.
Secondly, a DP-based search with an adaptive termination strategy in the second pass is done.

By means of the pre-selection in the first pass, the range of target lyrics is narrowed down to $L_{lc}$. DP matching with the lyrics in $L_{lc}$ is then carried out to calculate the precise distance. The candidates with the minimum acoustic distance $D(k)$ are indicated as the search results. Since the $R(k)$ is calculated as an approximate value of DP matching distance $D(k)$, after $L_{lc}$ is sorted by $R(k)$, the correct lyric with
the minimum $D(k)$ rises into the forward ranks in most cases. Thus, instead of the exhaustive DP matching over the entire set of pre-selected lyrics $L_{lc}$, a DP-based search with an adaptive termination criterion is proposed. The termination is adaptive to a cut off function $F$. The second pass search is designed as shown in the flowchart in Figure. 4.6.

Lyrics $L_{lc}$ are first sorted by $R(k)$ and then divided into $Z$ groups, thus each group has $I_c/Z$ lyrics. A DP matching calculation is executed in one group after another, while the cut-off function $F$ does not fulfill a terminating condition. Once the value of $F$ reaches a threshold $F_{th}$, the DP matching process is aborted at that group. Within the lyrics of the calculated groups, the lyrics are ranked in the order of the $D(k)$, and then the lyrics with lower distance values are provided as search results.
4.2.5 Experiments

Two segments of experiments were carried out in this research. First, the improvement of search accuracy by applying acoustic distance was evaluated. Second, the proposed method applying the acoustic distance and the two-pass search algorithm was compared with three conventional methods to evaluate its performance on both
search accuracy and processing time.

The results of the experiments were all obtained using a personal computer, with the specifications of Intel Core2Duo CPU 3.0GHz and 4G RAM.

4.2.5.1 Verification of Improvements in Search Accuracy by Applying the Acoustic Distance

Two exhaustive DP-based search methods using different distances were compared to evaluate the advantage of the acoustic distance. One method is Exhaustive DP applying Edit Distance (EDPED) of phoneme strings, and the other method is Exhaustive DP applying Acoustic Distance (EDPAD).

The test set consisted of 220 incorrect queries that were mistaken via acoustic confusion, the same as the queries used in Section 4.2.1. Also, a database of 10,000 lyric texts was collected. It contained both Japanese and English lyrics. The lyrics corresponding to the queries were included in the database.

As shown in Table 4.5, T-best (T =1, 20) represents the top T candidates of the ranked lyrics. The hit rate of T-best is defined as the rate of the total number of hits within top T candidates to the total number of search accesses (this is calculated as the search accuracy). EDPAD improves the hit rates by 2.8% and 4.4% respectively when the value of T in T-best is 1 and 20. A t-test was also conducted. When T in T-best is 20, the p-values are very low, which indicates that the proposed acoustic distance method achieved statistically significantly better performances than edit distance. Furthermore, an analysis of the queries that failed to identify the target lyric text using EDPAD method reveals that most of them are smaller than 6 syllables, indicating that the distance between the query and lyric texts was too close to make the target lyric distinguishable.
Table 4.5 Search accuracies and p-values of ED PED and EDPAD

<table>
<thead>
<tr>
<th>Search methods</th>
<th>ED PED</th>
<th>EDPAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-best</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hit rate(%)</td>
<td>49.5</td>
<td>50.9</td>
</tr>
<tr>
<td>p-value</td>
<td>0.203</td>
<td></td>
</tr>
<tr>
<td>20-best</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hit rate(%)</td>
<td>70.5</td>
<td>73.6</td>
</tr>
<tr>
<td>p-value</td>
<td>0.017</td>
<td></td>
</tr>
</tbody>
</table>

4.2.5.2 Evaluation of Search Accuracy and Search Time

The total performance of the proposed method applying the acoustic distance and the two-pass search algorithm, which are described in Section 4.2.4 is evaluated by lyric search experiments in this section.

Before the comparison experiment, in order to optimize the parameters of the proposed method, preliminary experiments were carried out.

First, a preliminary experiment has been done to determine the parameters for the first and the second passes. As described in Section 4.2.4, the following parameters in the proposed method need to be optimized:

- \( I_c \): the number of candidates in the first pass
- \( F \): the cut-off function in the second pass

To find corpus-independent parameters, 842 misheard lyric queries in English were collected from the website “kissthisguy”. Also, a database of 50,000 lyric texts was collected. The lyrics corresponding to the queries were all included in the database. Note that, the queries and lyric texts are entirely different from those used in Section 4.2.5.1.
First, an experiment was carried out to decide $I_c$. The first pass search using the index described in Section 4.2.4 was executed to investigate the relationship between search accuracy and $I_c$ to choose the best value for $I_c$. The results are shown in Figure 4.7. The horizontal axis shows the values of each tested $I_c$ from 100 to 2,000, and the vertical axis is the hit rate within $I_c$ candidates. Each line represents a different number of lyrics in the search space. The hit rates are almost saturated when $I_c$ is larger than 1500, in spite of the variation in the search space. Therefore, $I_c$ is set at 1500 in this research.

Figure 4.7 Relationship between hit rates and $I_c$ for various sizes of lyric database
Second, an investigation was undertaken to decide $F$. In most of the 842 queries, it was found that, by sorting the lyrics according to the approximate distance $R(k)$ and dividing them into groups, the target lyric has a significantly lower DP distance $D(k)$ than other lyrics in the same group. Based on the investigation above, $F$ is defined by Equation 4.15, where $D_{\text{min}}$ is the minimum value and $D_{\text{mean}}$ is the mean value of the group. The experimental results those reveal the relationship between processing time and search accuracy with respect to $F_{\text{th}}$ are shown in Figures 4.8 and 4.9, where the horizontal axis represents $F_{\text{th}}$, the right vertical axis represents the processing time, and the left vertical axis represents the hit rate. Figure 4.8 shows the results for the 1-best case, while Figure 4.9 shows the results for the 20-best case. Both figures show that the value of $F_{\text{th}}$ between 0.4 to 0.6 is the optimal threshold to reduce processing time without deteriorating search accuracy.

$$F = \frac{D_{\text{min}}}{D_{\text{mean}}}$$  \hspace{1cm} (4.15)
Figure 4.8 Search accuracy and processing time with respect to $F_{th}$ in the case of 1-best
Then, to evaluate the overall performance of the proposed method, both hit rate and processing time were compared with three conventional DP-based methods. All methods applied the proposed acoustic distance. The details are described below.

- “Two-pass DP search with Adaptive Termination (TDPAT)” is the proposed method described in Section 4.2.4. $F_{th}$ is tuned from 0 to 1. Considering the balance of index size and search accuracy, here $N$ of the syllable $N$-gram index is set to 3. A total of 50,000 entries of syllable 3-grams, which cover 92% of all syllable 3-grams in the collected lyric corpus, are prepared in the index. As all
the syllable 3-grams which exist in the queries are prepared, no search errors come from out-of-vocabulary syllable 3-grams in the experiment. The acoustic distance is normalized by the length of the corresponding DP path. The group size for the second pass is set at 100 lyrics, as $Z$ is 15. It is optimized by a preliminary experiment.

- “EDPAD” is an exhaustive DP-based search over the entire search space of lyrics, and this method is mentioned in Section 4.2.5.1.

- “High-speed DP search with Suffix Array (HDPSA)” is based on the method in [45]. In the experiment, since the input query and the database are both text, the texts were converted into syllable strings (instead of the phonemes originally used); and divided into syllable $N$-grams. Also, a suffix array recorded the boundary information of the lyrics in order to avoid matching queries across two lyrics. Here $N$ is set to 3 because this value resulted in better performance than when $N = 2$ or $N = 4$ in a preliminary experiment. The total threshold was tuned from 0 to 1.3 to find the optimal value balancing search accuracy and processing time.

- “Two-pass DP search with Distance-based Termination (TDPDT)” is a method that has almost the same processes as the proposed method, with the exception that the DP is terminated when the acoustic distance $D(k)$ exceeds a predetermined threshold value, that is tuned from 0 to 1.

The test set of 220 incorrect queries and 10000 lyric texts were the same with those used in Section 4.2.5.1.

First, to evaluate the robustness of TDPAT, a comparison with EDPAD and TDPAT is represented in Figure 4.10. Here, $F_{th}$ for TDPAT is set at 0.4, as optimized in Section 4.2.5.2. TDPAT maintains almost the same hit rate as the $T$ of $T$-best is varied from 1 to 40. As $T$ is over 40, there is also only less than 1.7% deterioration.
of search accuracy. As the processing time of TDPAT is 0.23 seconds per query, it is reduced by 89.3% compared with EDPAD. This improvement is due to the well-designed two-pass search algorithm that avoids losses occurring in the pre-selection and the adaptive termination processes.

The search accuracy and time complexity of three high-speed DP methods TDPAT, TDPDT and HDPSA are shown in Figures 4.11 and 4.12, where the horizontal axis represents processing time and the vertical axis represents hit rate. Each point in these figures indicates the processing time cost and the hit rate achieved when a particular threshold is set. Figures 4.11 and 4.12 show the results in the cases of 1-best and 20-best, respectively.

As shown in both figures, the performance of TDPAT is superior to that of HDPSA in terms of both processing time and search accuracy. In the case of 1-best, to achieve the same hit rate of 50.0%, TDPAT reduces processing time by a maximum of 96.5% compared with HDPSA. In the case of 20-best, to achieve the same hit rate of 70.0%, TDPAT reduces processing time by a maximum of 86.2%. These results indicate that the proposed search algorithm is more efficient than the conventional algorithms that determine the pruning threshold according to the length of the queries.

Also, TDPAT obtains higher search accuracy than TDPDT at the same processing times, especially for short processing times. It proves that the hypothesis of the definition for $F$ is correct and is effective in the search process.
Figure 4.10 Search accuracy of TDPAT and EDPAD
Figure 4.11 Average processing times and search accuracy of three search methods in the case of 1-best.
4.3 Summary

This chapter presents two strategies to optimize dialogue management for information search and to decrease failures in information search mainly caused by mistaken queries. First, the author proposed to select question by optimizing information gain and user preference for the spoken dialogue system. It is expected to assist users in
easily selecting satisfactory results by minimizing the number of search refinement steps. The evaluation experiments prove that the application with the proposed strategy performs better than the conventional applications in terms of satisfaction with search results and the effort spent for reconsidering search keywords.

Second, the author also proposed a robust and fast search strategy with a two-pass search algorithm to decrease information search failures caused by incorrect queries that are misheard or mismemorized. It uses an index-based approximate pre-selection for the first pass and DP-based search process with an adaptive termination strategy in the second pass. For the incorrect queries that are misheard or mismemorized, the experiments proved that applying acoustic distance improved search accuracy by 4.4% over edit distance. The proposed method achieved real-time operation by reducing processing time by more than 86.2% with a slight loss in search accuracy compared with a complete search by DP matching with all lyrics. It is proved to be the most practical solution for acoustic confusion queries, considering the trade-off between high search accuracy and low computation complexity. In addition, this proposed search strategy is expected to be effective for recovering the search failures caused by speech recognition errors in spoken dialogue systems as well, which can be inferred by other researches of morph-based spoken document retrieval and unlimited-vocabulary speech recognition [63] [64].
Chapter 5

Applying Prosody Information in Spoken Dialogue Systems

5.1 Prosody Information

To improve the usability of the spoken dialogue system, some researches [77] [78] were interested in using the speech prosody information, which represents the tune and rhythm of speech. Regarded as a part of the grammar of a language, prosody information is used to convey lexical meaning for stress (e.g. English), accentual (e.g. Japanese) and tone (e.g. Mandarin Chinese) languages. Taking Mandarin Chinese to give a further explanation, the lexical tone varieties define different words even though other acoustic characteristics are almost the same. For example, in Chinese, the disyllable sound “ji shu” with the lexical tones (1,4) means “cardinal number”, while the meaning changes to “technology” as the tones shift to (4,4). Therefore, in order to completely recognize Chinese language, not only the phonetic content, but also the prosody features are required [79]. Meanwhile, prosody also conveys non-lexical information such as intonation, which affects to differentiate
declarative sentences from questions. Furthermore, prosody is verified to convey more complicate non-lexical information: emotion. Many studies have shown that prosody information provide a reliable indication of the speech emotion [32] [33] [34] [80].

For the further understanding of prosody, it is characterized by the following items at the phonetic level [81]:

- vocal pitch (fundamental frequency)
- loudness (acoustic intensity)
- rhythm (phoneme and syllable duration)

Many experiments studying emotional speech are based on the stylized emotion, which is presented by actors and actresses. The results indicate that a few categories of emotions can be reliably identified by listeners. Consistent acoustic correlates of these categories are analyzed.

For example:

- Excitement is expressed by high pitch and fast speed.
- Sadness is expressed by low pitch and slow speed.
- Hot anger is characterized by over-articulation, fast, downward pitch movement, and overall elevated pitch.
- Cold anger shares many attributes with hot anger, but the pitch range is set lower.

Especially the pitch features carry the information about intention, attitude or emotion expression from the user’s speech. However, conventional pitch detection methods are not robust enough in real noisy environments.
My work proposes a new algorithm named adaptive running spectrum filtering (ARSF) to restore the amplitude spectra of speech corrupted by additive noises. It realized the robust pitch detection in the real world environments against various noise situations.

5.2 Spectra Analysis and Pitch Detection

To give a better understanding of the proposed pitch detection method, some fundamentals of related speech signal processing are introduced in this section. Firstly, the mechanics of producing speech in human beings are described. Then the theories of running spectra and modulation spectra are presented.

5.2.1 Fundamentals of Speech Production

Reference [82] describes a schematic diagram of the human vocal mechanism. As shown in Figure 5.1 [83], air enters the lungs via the normal breathing mechanism. After air is expelled from the lungs through the trachea, the tensed vocal cords within the larynx are caused to vibrate by the air flow. The air flow is chopped into quasi-periodic pulses. Then the pulses are modulated in frequency in passing through the pharynx, the mouth cavity, and possibly the nasal cavity. Different sounds are produced depending on the positions of various articulators, such as jaw, tongue, velum, lips and mouth. The speech signals are slowly time varying signals. When analyzed within a sufficiently short period of time, such as from 5 to 100 msec, its characteristics are fairly stationary. However, over long periods of time, such over 1/5 seconds, the signal characteristics change to reflect the different speech sounds being spoken.
In order to represent the time-varying signal characteristics of speech, a parameterization of the spectral activity based on the model of speech production is used. The human vocal tract is essentially a tube, of varying cross-sectional area that is excited either at one end or at a point along the tube. Therefore, the transfer function of energy from the excitation source to the output can be described in terms of the natural frequencies or resonances of the tube, which is based on acoustic theory.
Such resonances are called formants for speech. They represent the frequencies that pass the most acoustic energy from the source to the output.

5.2.2 Introduction of Running Spectrum and Modulation Spectrum

Human phonetic judgments are remarkably robust in presence of variability from non-linguistic sources of information such as speaker variability, channel variability, the environment in which the speech was produced, recording equipment used for speech acquisition. For an example, people seem to be capable of focusing attention on the linguistic message during conversational speech. It works very well even in horrible noise situation. It can be considered that peripheral properties of human hearing must also have something to do with the ways speech evolved and is being used for human communication. Linear distortions and additive noise in speech signal show as a biases in the short-term spectral parameters. The rate of such extra linguistic changes is often outside the typical rate of change of linguistic components.

To investigate the further details of the noise distortion, speech modulation spectra are analyzed [84] [85]. To introduce the modulation spectra, speech signals $y(k)$ are segmented into frames by a window function $\omega(k, t)$ where $t$ is frame number. Short time Fourier transform of the windowed speech signals, $X(t, f)$, are calculated by Equations 5.1 to 5.4, where $\text{FT}[\cdot]$ denotes Fourier transform. Then, running spectra $|X(t, f)|$ are calculated by taking the absolute values of spectra $X(t, f)$, which is the same process to calculate speech amplitude spectra. Running spectrum is so called because the spectrum looks like it is running along the time axis. It represent the temporal properties of time varying amplitude spectra. As shown in Equation 5.5, the modulation spectra $X_m(f, g)$ are obtained by applying Fourier transform on the running spectrum $|X(t, f)|$ at each frequency [84]. $T$ is the total number of frames. $g$ is modulation frequency.
\[ y'(k) = \omega(k, t)y(k) = \begin{cases} y(k) & (N - 1)t \leq k < Nt \\ 0 & \text{others} \end{cases} \quad (5.1) \]

\[ Y(f) = FT[y(k)]_{k=0,\ldots,\infty} \quad (5.2) \]

\[ W(i, t) = FT[\omega(k, t)]_{k=0,\ldots,\infty} \quad (5.3) \]

\[ X(t, f) = \sum_{i=-\infty}^{\infty} Y(f - i)W(i, t) \quad (5.4) \]

\[ X_m(f, g) = FT[|X(t, f)|]_{t=1,\ldots,T} \quad (5.5) \]

Figures 5.2 and 5.3 show the modulation spectra of clean speech and speech corrupted by 0 dB white noise. Modulation spectra show some important characteristics related to noisy speech. First, the energy of the additive noise is distributed from 0 Hz to about 1 Hz modulation frequency. Most of the noise energy is concentrated near 0 Hz. On the other hand, the energy of speech signals is distributed in wide range. The crucial information of speech is concentrated in the area from 0 Hz to about 13 Hz. In addition, the part 2 to 4 Hz is quite important since it is related to
the variation of phonemes. The difference between noise and speech on the modulation spectra can be explained as the non linguistic spectra components contained in the noise change more slowly than the typical range of speech. The characteristics of modulation spectra are shown in Figure 5.4. Here the spectra in 130 Hz (normal frequency) are used as examples because 130 Hz is located in the region of human fundamental frequency.

Figure 5.2 Modulation spectra of clean speech
Figure 5.3 Modulation spectra of speech corrupted by 0dB white noise
5.2.3 Pitch Detection

Pitch detection methods also use short-term analysis techniques, which means that a score $f(T|x_m)$ is calculated by a function of the candidate pitch periods $T$ for every frame $x_m$.

In speech processing literatures, a wide variety of pitch detection methods has been proposed. However, accurate and robust pitch detection in noisy environments still remains as a difficult and important issue in the real world application.
Chapter 5. Applying Prosody Information in Spoken Dialogue Systems

As described in Section 2.3, AUTOC method is one of the most commonly-used algorithms. The autocorrelation function of the voiced frame is searched for the maximum value. And the location of the maximum contains information of pitch period. As shown in Figure 5.5, \( N_f \) is searched as pitch position. And pitch period equals to the value of \( N_f / F_s \). \( F_s \) is the sampling frequency of the signals. Autocorrelation is usually regarded as time-domain function. However, based on the theory of Wiener-Khintchine (Equation 5.6), the autocorrelation function \( R(n) \) in AUTOC is obtained by calculating IFFT on the 2-nd power amplitude spectra \( E(k) \) of speech signals. Similarly, in CEP method, the peak cepstral value is determined and the location indicates the pitch period. Cepstrum is the logarithm of an amplitude speech spectrum.

\[
R(n) = \frac{1}{N} |E(k)|^2 e^{\frac{2\pi i nk}{N}} \tag{5.6}
\]

Pitch detection methods typically fail in several cases, such as sub-harmonic errors, harmonic errors and noisy conditions. Especially the noises give great influence. When the signal to noise ratio (SNR) is low, most pitch detection methods are quite unreliable. It can be explained as that the various noise distortions impose many damages on speech amplitude spectra, which leads to pitch misdetection.

In case of white noise, the energy is uniformly distributed along the frequency axis. It does not form prominent energy peaks. After exponentiation calculation is taken on \( |E| \), the high energy parts which represent speech components are enhanced. So AUTOC performs robust against white noise [86]. Recently, considering the effects of noise and formants, Reference [49] proposed a new method. As shown in Equation 5.6, the part \( |E(k)|^2 \) is replaced by \( |E(k)|^p \). It adjusts the exponent \( p \) of the amplitude spectrum according to SNR of each speech frame. As the SNR decreases, which represents that noise gets serious, the value \( p \) is increased. Consequently, the high energy voiced part gets more enhanced. They are obviously outstanding.
compared with the low noise parts. The periodical structure of amplitude spectra is kepted. In the case of white noise and other wide-band noises, the accuracy of pitch extraction is improved by the new method in Reference [49]. However, the car noise distortion is originated from its periodicity. The energy is concentrated in a narrow band in the amplitude spectra of car noise. As presented in Figure 5.6, the prominent peak around 20 to 400Hz is generated by car noise. It changes the original periodical structure of the amplitude spectrum. Only increasing the value of $p$ does not ameliorate the detection accuracy well in the car noise background. On the other hand, logarithm calculation prefers to extract the envelope of amplitude power. Therefore, CEP method is relatively robust against the noise whose energy is distributed in a fairly narrow band, such as car noise. However, under most wide band noises, CEP gives unsatisfied performance. It is averagely weaker than other conventional detection methods in noise conditions. As discussed above, these conventional methods can not keep working well against the changing noise situation. This class of detectors can be considered as amplitude-spectra-based pitch detectors. They use the property that if the signal is periodic in the time domain, then the frequency spectrum of the signal will consist of a series of impulses at the fundamental frequency and its harmonics. Therefore, to keep high robustness against unspecific noise condition, a spectra restoration function is supposed to be applied into pitch detection. By the restoration, the energy distribution of the amplitude spectra is expected to get close to the spectra under clean condition. The periodicity in spectra is clearly retained, which helps to improve the detecting accuracy.
5.3 Robust Speech Spectra Restoration for Pitch Detection

5.3.1 Adaptive Running Spectra Filtering designing

According to the characteristics of speech modulation spectra, an adaptive running spectrum filtering (ARSF) process is proposed to restore the periodic structure of amplitude spectra against unspecific noises. It improves the accuracy of pitch detection. Figure 5.7 shows a block diagram of the ARSF approach. $|X(t, f)|$ means the running spectra of speech (also means the amplitude spectra). A low-pass filter with the fixed parameters and an adaptive high-pass filter are implemented separately.
Figure 5.6 Noise distortion in amplitude spectrum

on each running spectrum. The low-pass filter is a FIR-type low-pass filter with cutoff frequency of 13 Hz. It eliminates noise distortion in parts higher than 13 Hz modulation frequency. Some speech components also exist in this area. However, they are few and they do not contain important information about pitch, so that the low-pass filtering does not distort the accuracy of pitch detection.

The adaptive high-pass filter works by varying the filter parameters according to the noise level on the running spectrum. The noise level is pre-estimated.

As mentioned above, the highest noise energy is concentrated in $X_m(f, g \mid g=0)$, the component at 0 Hz in the modulation spectrum. Equation 5.7 shows that, as $X_m(f \mid f=f_n, g \mid g=0)$ is adapted close to the level of clean speech by high-pass filtering, the corresponding running spectrum $|X(t, f \mid f=f_n)|$ is restored. As $|X(t, f \mid f=f_n)|$ is restored at every normal frequency $f_n$, consequently, the whole amplitude spectra $|X(t, f)|$ are restored close to that of clean speech. It helps to keep the correct pitch
The distortions of the additive noises are variable and complicated in noisy speech. \(X_m(f,g)\) of each running spectrum varies by noise conditions. Therefore, the adaptive high-pass filter is designed for each running spectrum in order to adapt \(X_m(f,g)\). To design the filter well, the first thing is to estimate the increase on DC part (0 Hz) of the modulation spectrum.

\[
|X(t,f)|_{f=f_n} = \frac{1}{N} \sum_{g=0}^{N-1} X_m(f_{f=f_n}, g)e^{\frac{2\pi ig}{N}}
\]

\[
= \frac{1}{N}(X_m(f_{f=f_n}, g_{g=0}) + \sum_{g=1}^{N-1} X_m(f_{f=f_n}, g)e^{\frac{2\pi ig}{N}}) \quad (5.7)
\]

The distortions of the additive noises are variable and complicated in noisy speech. \(X_m(f,g)\) of each running spectrum varies by noise conditions. Therefore, the adaptive high-pass filter is designed for each running spectrum in order to adapt \(X_m(f,g)\). To design the filter well, the first thing is to estimate the increase on DC part (0 Hz) of the modulation spectrum.
Chapter 5. Applying Prosody Information in Spoken Dialogue Systems

\[ IIDCN(f) = 10\log_{10} \frac{|EM(f)_{clean}|^2}{|EM(f)_{nspeech}|^2} \] (5.8)

(IIDCN: Inverse of Increase in the DC part by Noise)

The adaptive high-pass filter is expected to be designed by assigning \( IIDCN \) to the attenuation ratio of filter corresponding to each frequency. The stopband edge frequency is set to 0 Hz. Theoretically, the value in 0 Hz modulation frequency of noisy speech is decreased to the level of clean speech after filtering. The calculation of \( IIDCN \) is important in this processing. As the absolute value of 0 Hz modulation spectrum, \( |EM(f|_{f=f_n})| \) is equal to the sum of the running spectrum amplitude that corresponds to \( f_n \) Hz by Equation 5.9:

\[
|EM(f|_{f=f_n})| = |X_m(f|_{f=f_n}, g|g=0)|
= |\sum_{t=1}^{T} X(t, f|_{f=f_n})|e^{2\pi t g|g=0}|
= |\sum_{t=1}^{T} X(t, f|_{f=f_n})||
\]
(5.9)

where \( T \) is the total number of frames. It depends on the length of the speech section and the frames size. The frame size is 46.3ms. \( t \) is frame number. Based on Equations 5.9 and 5.8, \( IIDCN \) is given as Equation 5.10:

\[
IIDCN(f) = 10\log_{10} \frac{|EM(f)_{clean}|^2}{|EM(f)_{nspeech}|^2}
= 10\log_{10} \frac{|\sum_{t=1}^{T} |X_{clean}(t, f)|^2}{|\sum_{t=1}^{T} |X_{nspeech}(t, f)|^2}
\]
(5.10)

where, \( |X_{clean}(t, f)| \) is \( |X(t, f)| \) of clean speech, and \( |X_{nspeech}(t, f)| \) is \( |X(t, f)| \) of noisy speech. Equation 5.10 shows that, with noise estimation, \( IIDCN \) is possible to be calculated in real noisy environment.
Chapter 5. Applying Prosody Information in Spoken Dialogue Systems

Assuming the first few frames in the speech section contain no speech but noise, the noise amplitude spectrum $|X_{\text{noise}}(f)|$ is estimated from these frames. The value of estimated $|X_{\text{clean}}(t, f)|$, $|X'_{\text{clean}}(t, f)|$, is calculated by Equation 5.11.

$$|X'_{\text{clean}}(t, f)| = \sqrt[2]{|X_{\text{nspeech}}(t, f)|^2 - |X_{\text{noise}}(f)|^2}$$

(5.11)

Here, if $|X_{\text{nspeech}}(t, f)|^2 - |X_{\text{noise}}(f)|^2 < 0$, $|X'_{\text{clean}}(t, f)|$ is set to 0.

$IIDCN'$ represents the estimated value of $IIDCN$, which is given by Equation 5.12.

$$IIDCN'(f) = 10\log_{10} \frac{\sum_{t=1}^{T} |X'_{\text{clean}}(t, f)|^2}{\sum_{t=1}^{T} |X_{\text{nspeech}}(t, f)|^2}$$

(5.12)

The estimated $IIDCN'$ is proposed to be used in ASRF in order to realize adaptive filtering.

The second thing is to set the cutoff frequency of the high-pass filter. As described in [84] and [85], the noise is concentrated in 0 to 1 Hz of modulation spectra. However, some speech information which contributes to pitch is contained in 0 to 1 Hz of modulation spectra. To search for the cutoff frequency that can provide best performance of pitch detection, a high-pass filter is tested in a pitch detection experiment. 15 Chinese words and 1 Chinese sentence are used. They are added by car interior noise and white noise with level of 0 dB and 10 dB. The stopband edge frequency of the high-pass filter is set to 0 Hz, and the attenuation is fixed to a certain value (the value at -5 dB, -10 dB and -20 dB are tried separately). Only the cutoff frequency is changed in the experiment. The results of pitch detection of each cutoff frequency are compared. Though 0.2 Hz and 0.3 Hz give better results for some words or in some specific noise situations, 0.1 Hz is verified to provide
the totally best performance among 0 to 1 Hz. Also 0.1 Hz provides best detection result in the clean speech.

Finally, the characteristics of the adaptive high-pass filter are decided as: FIR type high-pass filter with 0.1 Hz cutoff frequency; the stopband edge frequency of the high-pass filter is set to 0 Hz; the attenuation ratio is equal to the estimated $\text{IIDCN}'$ of each corresponding running spectrum. The magnitude response of the adaptive high-pass filter with -20 dB attenuation is described in Figure 5.8. It corresponds to $-20 \text{ dB } \text{IIDCN}'$. In addition, if $\text{IIDCN}'(f) = 0$ or $\text{IIDCN}'(f) > 0$, which means that there is no noise distortion, the filtering is not applied.

### 5.3.2 Pitch Detection with ARSF

As an algorithm of spectra restoration, ARSF is added in the pitch detecting process to keep the original (clean condition) periodic spectra structure against noises. The process is presented as follows: First the speech section is segmented into frames by
the Hanning window. In the experiment the frame length is 46 ms (512 sampling points by 11025 Hz sampling frequency). The shift of frames is 23 ms (256 sampling points by 11025 Hz sampling frequency). Then fast Fourier transform (FFT) is applied on each speech frame and its absolute value shows the amplitude spectrum. Since the formant characteristics of vocal tract adversely affect pitch detection, a band-limitation operation is implemented to diminish the frequency-multiplication contents of speech signals. It is actually done by keeping the values of 50 to 500 Hz part while setting the values of other parts to 0 in amplitude spectra. It corresponds to the region of human fundamental frequency. Then, ARSF is applied to the running spectrum of each frequency (band-limited from 50 to 500 Hz). In the first step, a low-pass filter with cutoff frequency of 13 Hz is applied. Secondly the adaptive high-pass filter is designed by the estimated $\hat{H}(\omega)$. After the ARSF process, the autocorrelation contour is obtained by applying inverse fast Fourier transform (IFFT) on 2-nd power restored amplitude spectrum. Finally, the positions of pitch peaks are detected in each speech frame. The flowchart of pitch detection with ARSF is described in Figure 5.9.

To prove the improvement of noise robustness by ARSF process, an example of one voice frame is presented. The amplitude spectrum and the autocorrelation contour of the frame in clean condition are shown in Figures 5.10 and 5.14. By various noise influences, the periodicity of amplitude spectrum is changed. It is shown by the dashed lines in Figures 5.11 5.12 and 5.13. As a result, the pitch periods are misdetected in the autocorrelation contours. It is shown by the dashed lines in Figures 5.15 5.16 and 5.17. After ARSF is used, the noise components are mainly filtered out while the speech components are kept. The information of periodic structure remains clear in the amplitude spectra. It is shown by the solid lines in Figures 5.11 5.12 and 5.13. As the pitch peaks stay significant by ARSF, pitch periods are correctly extracted in the autocorrelation contours. It is shown by the solid lines in Figures 5.15 5.16 and 5.17.
Figure 5.9 Flowchart of the proposed pitch detection method
Figure 5.10 Amplitude spectrum of clean speech
Figure 5.11 Amplitude spectrum of speech with 0 dB car noise
Figure 5.12 Amplitude spectrum of speech with 0 dB pink noise
Figure 5.13 Amplitude spectrum in speech with 0 dB talking babble noise
Figure 5.14 Autocorrelation contour in clean speech
Figure 5.15 Autocorrelation contour in speech with 0 dB car noise
Figure 5.16 Autocorrelation contour in speech with 0 dB pink noise
5.3.3 Experiments

Before the experimental results are presented, some elements of judgment criteria are introduced. The voiced error $e(n)$ is defined by Equation 5.13:

$$e(n) = P_d(n) - P_r(n)$$  \hspace{1cm} (5.13)
where $n$ is the frame number in speech sections. $P_d$ is the pitch period detected by each method. The reference values of pitch period $P_r$ is calculated by SAPD method [87]. If $|e(n)| > 10$ samples (it is $10/F_s=0.9$ms), it is called gross pitch error. The percentage of the frames in which gross pitch errors occur in the whole frames of the speech section is defined as GPE. The second type of pitch error is the fine pitch period error in which case $|e(n)| < 10$ samples. As a measure of the accuracy of the pitch detector, the Standard Deviation of the Fine Pitch Errors (SDFPE) is defined by Equation 5.14:

$$
\sigma_e = \sqrt{\frac{1}{N_i} \sum_{j=1}^{N_i} e^2(m_j) - \bar{e}^2}
$$

(5.14)

where $m_j$ is the j-th frame in the utterance for which $|e(n)| < 10$ samples. $N_i$ is the number of such frames in the utterance. $\bar{e}$ is mean of the fine pitch errors which is defined by Equation 5.15:

$$
\bar{e} = \frac{1}{N_i} \sum_{j=1}^{N_i} e(m_j)
$$

(5.15)

Both GPE and SDFPE are used as the criteria to estimate the robustness and accuracy of the detection methods.

The experiment uses 10 isolated Mandarin Chinese words. They are spoken by 6 females and 2 males. Four methods are compared in speech data interfered by noises. There are five types of noises included in the NOISEX-92 noise database. The noise levels are 0 dB, 5 dB and 10 dB. The experimental results of GPE are in Table 5.1. ARSF (ideal) means that ideal $|X_{\text{clean}}(t, f)|$ is used in ARSF. The ideal $|X_{\text{clean}}(t, f)|$ is the originally recorded clean speech before noise is added. ARSF (estimated) is the proposed method that is based on noise estimation. $|X'_{\text{clean}}(t, f)|$ is used. Table
1 shows that, only except for the case of 10 dB white noise in which Reference [49] method shows a little better performance, the proposed method with ARSF realizes the best robustness among various noise conditions. Also, ARSF (ideal) and ARSF (estimated) are compared. Their results are almost at the same level. It means that there are no big mistakes in noise estimation and that the idea of ARSF is correct and practical. However, the fact that 4 to 5% difference occurs in the cases of 0 dB car noise and 0 dB engine noise indicates an more accurate and smarter noise estimator remains as an important issue in the future research. Table 5.2 shows the SDFPE of four methods. The proposed method with ARSF achieves lower or the same SDFPEs in low SNR car, babb and engine noises. However AUTOC and Reference [49] show better performance in the cases of white noise, pink noise and high SNR of other noises. It is because that the noise level of each running spectrum in white noise, pink noise and high SNR of other noises is not serious. By-effects of filtering in ARSF give small influences, which degrade pitch information a little. For the next step, smoothing after ARSF is to be considered. By the way, the proposed method keeps SDFPE within a small range from 1.55 to 2.10 in all of the noise conditions. It is more stable than AUTOC and Reference [49] method.
Table 5.1 Comparison results at various noise conditions (GPE %)

<table>
<thead>
<tr>
<th>Noise</th>
<th>SNR</th>
<th>ARSF (ideal)</th>
<th>ARSF (estimated)</th>
<th>Ref.[6] Method</th>
<th>AUTOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>car</td>
<td>0</td>
<td>11.8</td>
<td>16.0</td>
<td>33.3</td>
<td>33.1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.7</td>
<td>7.2</td>
<td>20.8</td>
<td>21.6</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2.7</td>
<td>3.5</td>
<td>10.0</td>
<td>10.1</td>
</tr>
<tr>
<td>white</td>
<td>0</td>
<td>1.9</td>
<td>2.1</td>
<td>2.4</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.3</td>
<td>1.4</td>
<td>1.4</td>
<td>1.8</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.9</td>
<td>1.1</td>
<td>1.0</td>
<td>1.1</td>
</tr>
<tr>
<td>pink</td>
<td>0</td>
<td>4.9</td>
<td>7.0</td>
<td>12.0</td>
<td>15.1</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2.3</td>
<td>3.3</td>
<td>4.8</td>
<td>5.7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.6</td>
<td>1.8</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>engine</td>
<td>0</td>
<td>10.2</td>
<td>14.4</td>
<td>18.6</td>
<td>21.5</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.7</td>
<td>4.7</td>
<td>7.3</td>
<td>8.7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>2.0</td>
<td>2.3</td>
<td>3.9</td>
<td>4.1</td>
</tr>
<tr>
<td>babb</td>
<td>0</td>
<td>4.1</td>
<td>5.1</td>
<td>9.8</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>2.0</td>
<td>2.1</td>
<td>4.3</td>
<td>4.3</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.5</td>
<td>1.7</td>
<td>2.2</td>
<td>2.2</td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td>3.8</td>
<td>4.9</td>
<td>8.9</td>
<td>9.7</td>
</tr>
</tbody>
</table>
Table 5.2 Comparison results at various noise conditions (SDFPE)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>car</td>
<td>0</td>
<td>1.72</td>
<td>1.54</td>
<td>1.92</td>
<td>2.36</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.73</td>
<td>1.56</td>
<td>1.47</td>
<td>1.74</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.70</td>
<td>1.66</td>
<td>1.23</td>
<td>1.43</td>
</tr>
<tr>
<td>white</td>
<td>0</td>
<td>1.65</td>
<td>1.66</td>
<td>1.41</td>
<td>1.43</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.61</td>
<td>1.71</td>
<td>1.23</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.69</td>
<td>1.66</td>
<td>1.04</td>
<td>1.05</td>
</tr>
<tr>
<td>pink</td>
<td>0</td>
<td>1.84</td>
<td>1.66</td>
<td>1.56</td>
<td>1.54</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.77</td>
<td>1.72</td>
<td>1.28</td>
<td>1.34</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.55</td>
<td>1.59</td>
<td>1.19</td>
<td>1.09</td>
</tr>
<tr>
<td>engine</td>
<td>0</td>
<td>2.10</td>
<td>2.07</td>
<td>2.14</td>
<td>2.07</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.83</td>
<td>1.79</td>
<td>1.73</td>
<td>1.67</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.84</td>
<td>1.80</td>
<td>1.40</td>
<td>1.39</td>
</tr>
<tr>
<td>babb</td>
<td>0</td>
<td>1.91</td>
<td>1.81</td>
<td>2.12</td>
<td>1.91</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>1.67</td>
<td>1.72</td>
<td>1.72</td>
<td>1.70</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1.71</td>
<td>1.68</td>
<td>1.50</td>
<td>1.34</td>
</tr>
</tbody>
</table>

5.4 Summary

This chapter presents a new algorithm ARSF to restore the amplitude spectra. It is applied in robust noise detection of speech pitch period. The ARSF process intelligently rehabs speech modulation spectra according to the noise conditions. The noise distortion that influences the amplitude spectral structure is much diminished. Therefore the proposed pitch detection becomes more accurate than other conventional methods under different noise conditions. The future study will try to obtain
better models to estimate the noise amplitude spectrum in order to further improve the accuracy of pitch detection. Furthermore, the author will consider applying the extracted pitch information to recognize speaker’s emotion. Understanding speaker’s emotion helps to generate more appropriate dialogue actions to present superiority and differentiation to other modalities.
Chapter 6

Conclusion and Future work

6.1 Conclusion

This thesis proposes three researches to build an optimal spoken dialogue system for robust information search in the real world, which is expected to realize user’s the habitual and continual use of dialogue system in the daily life.

First, the research of designing dialogue takes advantage of the gamification theory to design a dialogue agent and a dialogue scenario to promote the habitual use. The real-world data also prove the novelty of this research, in which over 23% users were keeping speaking to the dialogue agent continually.

Second, the research of dialogue management proposes two strategies to improve the user experience for information search: firstly, a strategy of optimal question selection has been verified to be effective to assist users’ operation in a knowledge-based spontaneous dialogue system. Furthermore a robust and fast matching strategy based on phoneme strings decreases the failures caused by the queries containing incorrect parts. Experimental results prove that this proposed search strategy increases search accuracy by 4.4% and reduces processing time by at least 86.2%.
Chapter 6. Conclusion and Future work

Third, the author proposed a research of high-accuracy voice pitch detection against noise. The proposed method intelligently restores speech modulation spectra according to the noise conditions. The noise distortion that influences the amplitude spectral structure is much diminished. Even in a variety of noise types and levels, the proposed pitch detection method is able to verify the high robustness compared with the existing methods. Furthermore, the extracted pitch information is planned to be applied for recognizing speaker’s emotion in the future research, which help generate the appropriate dialogue action.

6.2 Future Work

The future direction for my research work is to utilize the spoken dialogue system to be a cross-device type interactive agent in the user’s daily life and be able to provide personalized information based on the user’s preferences.

The expected future system mechanism is shown in Figure 6.1. The dialogue agent will be applied to other devices, such as set top box (STB) and in-vehicle machines, such as GPS Navigator, as well as smartphone to cover more life scenes to support the user’s information search and device operation.

To make the spoken dialogue system more optimized according to user’s personal information, the technologies that understand the user’s emotion and build user model which reflects the user’s hobbies and preferences are required.

The following researches are going to be focused in my future works:

1. Robust emotion recognition in the real world: besides the pitch extraction as proposed in Chapter 5, other prosody features such as energy, duration information and quality features including formants information are going to be used. And the statistical models such as, hidden Markov model or deep neural network are going to be applied to correctly classify the emotion. Furthermore,
not only speech information, text and visual information are also considered to be combined to generate better recognition performance.

2. Automatic and active learning to build user’s profile: by means of analyzing the user’s emotion and extracting the keywords in dialogue log data, a user model (or profile) is expected to be established, with which the dialogue management executes the dialogue task more efficiently. Furthermore, the technologies of automatic question generation and situation estimation for active learning are required.
Acknowledgement

The author is deeply grateful to his supervisor, Prof. Yoshikazu Miyanaga of the Division of Media and Network Technologies, Graduate School of Information Science and Technology, Hokkaido University. He has taught the author innumerable lessons and insights on the workings of academic research in general. Thanks for his technical and editorial advice led the accomplishment of the present thesis. The author also thanks all of the members of the Information and Communication Networks Laboratory for the kindness and so much support for his research. The author also wishes to thank Prof. Kunimasa Saitoh, Prof. Takeo Ohgane and Prof. Hiroshi Tsutsui for taking time to examine the contents of this thesis.

The author is also grateful to his colleagues in KDDI CORPORATION and KDDI R&D Laboratories, Inc., for their kind understanding and supporting.

Last, the author thanks his family for all the love and support. Especially, the author would like to thank his wife Naoko for her understanding during the past few years. Her constant support and encouragement was in the end what made this dissertation possible.
References


[44] KISS THIS GUY, “Search by Song, Lyric or Artist”.

http://www.kissthisguy.com/


http://www.kddilabs.jp/newsrelease/2013/101001.html


References

[56] “User Behavior Analysis Research on Smart-phone Games of Mobaga and GREE”, Seed Planning, Inc. 2012 (Japanese)

[57] Peratama (Android Application in Google Play).


[60] COOKPAD.
   http://cookpad.com/

[61] Recipe Search.
   http://www.medc.jp/recipe-search/


References


[70] OKWAVE (Japanese).
    http://okwave.jp/

[71] 教えて!goo (Japanese).
    http://oshiete.goo.ne.jp/

References


References

[81] “What is Prosody?”.
http://kochanski.org/gpk/prosodies/section1/


List of Publications

[1] Peer-reviewed Journal


[2] International Conference


3. Xin Xu, Masaki Naito, Tsuneo Kato, Hisashi Kawai. “Robust and Fast Lyric Search Based on Phonetic Confusion Matrix”, in Proceedings of


[3] Domestic Conference


