

TD
2007
165

Doctoral Dissertation

Discourse Understanding in Task-oriented Spoken Dialogue Systems

Ryuichiro Higashinaka

Academic Year 2007

Graduate School of Media and Governance
Keio University

Acknowledgments

First and foremost, I would like to express my sincere appreciation to Professor Shun Ishizaki of Keio University for supervising this thesis and also for his continuous encouragement and support. I would also like to thank him for introducing me to the wonderful field of natural language processing when I was working on my master's.

I would also like to thank my co-supervisors, Professor Michiaki Yasumura and Professor Kuniaki Mukai of Keio University, Professor Katashi Nagao of Nagoya University, and Dr. Mikio Nakano of Honda Research Institute Japan Co., Ltd., for their reviewing this thesis and giving me valuable comments and suggestions.

Especially, I would like to thank Professor Katashi Nagao. He was my supervisor when I was working at the IBM Tokyo Research Laboratory as a student researcher. He passionately taught me basic skills in research as well as system development often through the night. He showed me how interesting it is to do research, which convinced me to take a similar path.

Similarly, I would also like to deeply thank Dr. Mikio Nakano. He was my supervisor when I joined NTT Communication Science Laboratories (NTT CS Labs.), NTT Corporation. He taught me everything I needed to be a professional researcher. A good part of my research skills, including paper writing, logical thinking, and system development, owes greatly to his strenuous support and patience. In fact, much of the work in this thesis was conducted under his supervision at NTT.

The research reported in this thesis was conducted at NTT CS Labs. I would like to thank the directors, executive managers, and group leaders for providing me the opportunity to conduct this research and also for their support. They are Professor Ken'ichiro Ishii of Nagoya University, Professor Shigeru Katagiri of

Doshisha University, Dr. Yoshinobu Tonomura of NTT CS Labs., Dr. Norihiro Hagita of ATR Intelligent Robotics and Communication Laboratories, Professor Hiroshi Murase of Nagoya University, Dr. Makino Shoji of NTT CS Labs., Dr. Naonori Ueda of NTT CS Labs., Professor Kiyoaki Aikawa of Tokyo University of Technology, Dr. Eisaku Maeda of NTT CS Labs, and Dr. Hideki Isozaki of NTT CS Labs.

I would also like to thank the members of the Dialogue Understanding Research and Knowledge Processing Research Groups at NTT CS Labs. for helpful discussions and suggestions. They include Masafumi Tamoto, Norihito Yasuda, Matthias Denecke, Kentaro Ishizuka, Yasuhiro Takahashi, Ryo Sato, Katsuhito Sudoh, Hajime Tsukada, Tsutomu Hirao, Hideto Kazawa, Akinori Fujino, and Taro Watanabe. Special thanks to Kohji Dohsaka, who, with his expertise in spoken dialogue systems, has been a great adviser throughout my research life at NTT, Noboru Miyazaki, who taught me the basics in speech processing and how to build experimental systems, and Jun Suzuki, who gave me helpful advice on support vector machines.

I would like to thank my colleagues Atsushi Nakamura, Yasuhiro Minami, and Masaaki Nagata for their helpful suggestions on the draft version of the papers related to this thesis. Thanks also go to Atsushi Fukayama, who taught me the basics of regression methods. I also thank my colleagues who joined NTT CS Labs. in the same year, Yoshinao Takemae, Sadao Hiroya, Naoki Saijo, Naomi Yamashita, and Shinji Watanabe, for their friendship and support.

Spoken dialogue systems research requires much effort in system development. I would like to thank Koichi Sugiura, Atsushi Miyazaki, Tetsuya Kubota, Yuichi Fujita, Ikumi Yamazaki, Dahchiun Shen, and Hiromi Sekine of NTT Advanced Technology Corporation, who helped me build the experimental systems described in this thesis. Thanks also go to Mamiko Kaneko, Yoko Shibata, Osamu Yamamoto, Hiroki Matsuura, Mariko Tomori, Kaori Kohno, Kaoru Yamada, Naoko Ohki, and Mika Koizumi of IR-Advanced Linguistic Technologies Inc. for their help in conducting dialogue experiments and dialogue data annotation. I also thank the members of the MIT Spoken Language Systems Group for making available their useful confidence-scoring software.

I would like to thank Professor Marilyn Walker, Joseph Polifroni, François

Mairesse, Swati Gupta, Professor Robert Gaizauskas, and Emma Barker, with whom I had the wonderful opportunity to work when I was a visiting researcher at the University of Sheffield. They gave me insightful comments and suggestions on various aspects of spoken dialogue systems, which greatly influenced the shape of this thesis. I would also like to thank Kazunori Komatani of Kyoto University, Ryoko Tokuhsa of Toyota Central R&D Labs., Inc., and Jason D. Williams of AT&T Labs, Inc. with whom I enjoyed having fruitful conversations and discussions on many issues in dialogue systems research. I am also grateful to Yukiko Katagiri of Keio University, whose fantastic illustrations helped me present this work on numerous occasions. Thanks also go to David Steenken of Kurdyla and Associates Co., Ltd. for helping me with English writing over the years.

I also feel indebted to the members of the Ishizaki Laboratory of Keio University, who made my student life an enjoyable and memorable one and also helped me fulfill the requirements of the doctoral program at Keio University. They include Associate Professor Kiyoko Uchiyama, Takuya Sakaguchi, Koichi Kamichi, Norifumi Watanabe, Jun Uchida, Toshikazu Samura, Tomoko Ohkuma, and Kai Seino.

Finally, I would like to thank my family for their wonderful support during this work. My sincere appreciation also goes to my grandmothers and grandfathers, who would have been so pleased to see me fulfill this work.

Abstract

The objective of the research presented in this thesis is to improve discourse understanding in spoken dialogue systems so that systems and users can interact more naturally using speech. Discourse understanding refers to a system's ability to understand user utterances based on the context of a dialogue. Until now, discourse understanding has been performed in a heuristic manner because the primary importance has been placed on developing workable systems. However, as systems become increasingly complex, systematic methods are needed if we are to achieve continuous progress in the field and also reduce the cost of system development. This thesis describes three approaches we developed to tackle three problems in discourse understanding in spoken dialogue systems.

The first problem is the lack of an evaluation measure for discourse understanding. In any field in engineering, it is important to have an evaluation measure. An appropriate evaluation measure would enhance activity in spoken dialogue systems research and make it possible to compare various approaches uniformly in an objective manner. Our approach to finding a suitable evaluation measure is to enumerate possible metric candidates and calculate their correlation with a system's performance through dialogue experiments.

The second problem is the lack of discourse information in confidence scoring of intention recognition results. One useful capability humans have during conversation is that they can detect understanding errors from the context of a dialogue. Computationally, this has been achieved by confidence scoring, which assigns reliability scores to understanding results on the basis of various features of a dialogue. However, little discourse information has been utilized to date. In our approach, we incorporate discourse-related features into the confidence scoring of intention recognition results, and show its effectiveness.

The third problem is the high cost of creating hand-crafted rules for ranking dialogue states for disambiguating user intentions. Spoken dialogue systems have to deal with the ambiguity arising from multiple speech recognition hypotheses as well as from natural language. Conventionally, this ambiguity is expressed as multiple dialogue states and hand-crafted rules have been utilized to rank the states with regard to the context. However, creating such rules is costly and, because of that, the rules have limited portability. In our approach, we automatically rank dialogue states using statistical information derived from dialogue corpora. We show the effectiveness of our approach in dialogue-state-ranking experiments.

Keywords: discourse understanding, speech understanding, spoken dialogue systems, evaluation measure, confidence scoring, statistical information

要旨

音声対話システムは、音声言語を用いて人間とやり取りを行うことで、さまざまなドメインにおいて、所定のタスクを遂行するシステムである。近年の音声認識・合成技術の進歩を背景に、商用サービスが始まるなど、多くの用途に使われ始めている。音声対話システムでは、ユーザとの複数回のやり取りを扱う必要から、談話理解（文脈を考慮したユーザ発話の理解）が重要となる。しかし、これまで音声対話システムの研究において、談話理解を本格的に扱った研究は少ない。人間のように文脈を上手に利用して相手の発話を理解することができれば、ユーザがシステムに短い発話で意図を伝えることが可能になったり、音声認識に誤りがあってもシステムが文脈からユーザ発話を正しく理解するといったことが可能になり、より使いやすい音声対話インターフェースが実現できると考えられる。本研究は、現状における音声対話システムの談話理解研究の3つの問題点をそれぞれ解決することで、音声対話システムの談話理解性能の向上を目指す。

第一の問題点は、談話理解の評価尺度が確立されていないことである。一般的に、研究分野が発展するためには評価尺度の確立が重要である。客観的に談話理解部が評価できれば、段階的に改良を行うことが可能となる。現状では談話理解部を評価するための評価尺度が存在しない。我々は、談話理解部の改善が、どれほどシステム全体のパフォーマンスの改善を導くかを、最も適切に予測し得る尺度が最も適切であると考え、さまざまな談話理解の尺度を列挙し、システム全体のパフォーマンスとの相関が最も高くなる尺度を選択することで、談話理解の評価尺度を決定する手法を提案し、実験的に得られた幾つかの有用な評価尺度を示す。

第二の問題点は、ユーザ意図の理解結果の信頼性評価に談話情報が使われていないことである。人間は、相手の声が聞きとりにくい環境において、自分の理解が合っているかを文脈により判断することができる。システムは自身の理解が正しいかを判断する場合、信頼性評価を用いる。これは、音声認識の分野で用い

られる技術で、様々な特徴量から総合的に値の成否を判定するものである。従来、ユーザ意図の理解結果の信頼性評価では主に、音声認識に関する特徴量しか使われてこなかった。理解結果がユーザとシステムのやり取りであることを考えるとこれは不十分であると考えられる。我々は、ユーザ意図の理解結果の信頼性評価に、談話において守られるべき基準であるグライスの公準に基づく特徴量を導入することで、高精度に信頼性を評価できることを実験により示す。

第三の問題点は、ユーザ意図の理解結果の曖昧性解消をルールで記述するとコストが高いことである。音声対話システムは、ユーザ発話に対して生じる複数の音声認識候補や自然言語そのものによって生じる曖昧性を扱う必要がある。従来、この曖昧性は複数の対話状態として表現され、手書きのルールによってこれらの対話状態を文脈的に適切な順に並び替えることで曖昧性解消を行ってきた。しかし、手書きのルールの作成には専門家による多大な労力が必要でありコストが高い。また、システムのドメイン毎に曖昧性解消のルールを作成する必要があるため、このコストは、音声対話システムの普及の妨げになりうる。我々は、対話コーパスから対話状態の並び替えに有効な統計情報を獲得し、その統計情報により自動的に対話状態を高精度で並び替える手法を提案する。また、実験によりその有効性を示す。

Keywords: 談話理解, 音声理解, 音声対話システム, 評価尺度, 信頼性評価, 統計情報

Contents

1	Introduction	1
1.1	Background and Motivation	1
1.2	Architecture of a Spoken Dialogue System	4
1.3	Discourse Understanding in Spoken Dialogue Systems	7
1.4	Problems Addressed in This Thesis	9
1.4.1	Lack of Evaluation Measure for Discourse Understanding .	10
1.4.2	Lack of Discourse Information in Confidence Scoring of In- tention Recognition Results	10
1.4.3	High Cost of Hand-crafted Rules for Ranking Dialogue States	11
1.5	Overview of Our Approaches	12
1.5.1	Creating an Evaluation Measure for Discourse Understanding	12
1.5.2	Incorporating Discourse Features into Confidence Scoring of Intention Recognition Results	12
1.5.3	Ranking Multiple Dialogue States by Corpus Statistics . .	13
1.6	Thesis Outline	13
2	Creating an Evaluation Measure for Discourse Understanding	15
2.1	Introduction	15
2.2	The Need for an Evaluation Measure in Discourse Understanding	17
2.3	Problem	17
2.4	Approach	20
2.4.1	Metric Candidates	20
2.4.2	Labeling	22
2.4.3	List of Metric Candidates	24
2.4.4	Performance Measure	26

2.5	Data Collection	27
2.5.1	Systems	27
2.5.2	Experiment	28
2.6	Data Analysis	29
2.6.1	Correlations of the Metric Candidates	29
2.6.2	Obtained Regression Models	29
2.6.3	Commonality in Regression Models	32
2.6.4	Important Factor Analysis of Regression Models	32
2.7	Summary	35
3	Confidence Scoring of Intention Recognition Results	39
3.1	Introduction	39
3.2	Intention Recognition in Spoken Dialogue Systems	41
3.3	Need for Confidence Scoring of Intention Recognition Results	42
3.4	Conventional Methods	44
3.5	Proposed Method	46
3.5.1	Discourse Features	48
3.5.1.1	Features Related to the Maxim of Quantity	50
3.5.1.2	Features Related to the Maxim of Quality	50
3.5.1.3	Features Related to the Maxim of Manner	52
3.6	Experiment	53
3.6.1	System	53
3.6.2	Data Collection	56
3.6.3	Data Screening	57
3.6.4	Feature Extraction and Labeling	59
3.6.5	Confidence Model Training	62
3.6.6	Evaluation	63
3.6.7	Impact of the Discourse Features	66
3.6.8	Analysis of Successful Cases	68
3.7	Summary and Future Work	70
4	Ranking Multiple Dialogue States by Corpus Statistics	73
4.1	Introduction	73
4.2	Discourse Understanding Using Multiple Dialogue States	74

4.3	Previous Work	77
4.4	Approach	79
4.4.1	Statistical Information	80
4.4.1.1	N-gram Probability of a Dialogue Act Type Sequence	80
4.4.1.2	Occurrence Probability of a Dialogue State Update Pattern	81
4.4.2	Scoring of Dialogue States Using the Statistical Information	82
4.5	Experiment	82
4.5.1	Systems	83
4.5.1.1	Train Ticket Reservation Domain	83
4.5.1.2	Weather Information Service Domain	84
4.5.2	Data Collection	85
4.5.3	Annotating Reference Dialogue Acts and Dialogue States	86
4.5.4	Deriving Statistical Information	87
4.5.4.1	Trigram Probability of a Dialogue Act Type Sequence	87
4.5.4.2	Occurrence Probability of a Dialogue State Update Pattern	88
4.5.5	Evaluation	90
4.5.5.1	Offline Discourse Understanding	90
4.5.5.2	Ranking Experiment	92
4.5.5.3	Evaluation Criteria	92
4.5.5.4	Results	93
4.5.5.5	Impact of the Dialogue State Beam Width	95
4.5.5.6	Analysis of Successful Cases and Failures	97
4.6	Summary and Future Work	102
5	Conclusions and Future Directions	103
5.1	Summary of Contributions	103
5.1.1	Establishment of an Evaluation Measure for Discourse Understanding	103
5.1.2	Improvement in Confidence Scoring of Intention Recognition Results Using Discourse Information	104

5.1.3	Automatic Disambiguation of User Intentions Using Corpus Statistics	104
5.2	Future Directions	105
5.2.1	Incorporating Confidence Scoring into Ranking Dialogue States	105
5.2.2	Handling Other Discourse Phenomena	106
5.2.3	Towards More Conversational Spoken Dialogue Systems . .	108
	Bibliography	110
	List of Publications	123

List of Figures

1.1	A dialogue between a user and a system in the air traffic information service domain taken from the corpus of the DARPA Communicator project (Walker et al., 2002). Speech recognition hypotheses for user utterances are shown in parentheses after ‘recog:’.	3
1.2	Architecture of a spoken dialogue system.	5
1.3	An example of frame updates in a dialogue. (S means a system utterance and U a user utterance.)	8
1.4	Example of discourse understanding rules	9
2.1	An example of frame updates with corresponding reference frames. (R and H mean a reference frame and a hypothesis frame respectively.)	18
2.2	Labeling the slot values of a hypothesis frame.	22
2.3	Labeling the update of a hypothesis frame.	23
2.4	Distribution of actual and predicted task completion times by the support vector regression (SVR) model trained from WI + MR-1 + MR-2.	33
3.1	Updating an intention recognition result. (S, U and F indicate a system utterance, a user utterance, and a frame, respectively.)	42
3.2	Conventional methods. Word confidences $c_1 \dots c_9$ are associated with the slots.	44
3.3	Grice’s maxims of cooperativeness (Grice, 1975).	49
3.4	The number of turns required to complete the tasks in each grouping.	58

3.5	False acceptance rate (FAR)–false rejection rate (FRR) curves, and for the method that uses posterior probability, the conventional and proposed models.	65
3.6	False acceptance rate (FAR)–false rejection rate (FRR) curves for the proposed and conventional models and for the models that do not use D1, D3, and D12 as discourse features.	68
4.1	Architecture of a spoken dialogue system that utilizes multiple dialogue states for discourse understanding.	75
4.2	Example of discourse understanding using multiple dialogue states. (S, U, DA, and DS stand for a system utterance, a user utterance, a dialogue act, and a dialogue state, respectively.)	76
4.3	Example of discourse understanding using multiple dialogue states.	77
4.4	Example of a dialogue corpus.	80
4.5	Seven binary attributes to classify a dialogue state update.	81
4.6	Performance changes in the Mean Reciprocal Rank (MRR) with different dialogue state beam widths in the train ticket reservation (TRAIN) and weather information service (WEATHER) domains. The x-axis is on the log scale.	96
4.7	A typical case where the statistical information is successfully exploited. See also Fig. 4.8 for how dialogue states are updated in this piece of dialogue. Recognition confidence means the posterior probability of a speech recognition hypothesis estimated from acoustic and language model scores (Bouwman et al., 1999). . . .	98
4.8	The 2nd-ranked dialogue state after U2 and the top-ranked dialogue state after U3 in the piece of dialogue shown in Fig. 4.7. . .	99
4.9	A typical case where the statistical information has an adverse effect because of strong contextual constraints. See also Fig. 4.10 for how dialogue states are updated in this piece of dialogue. . . .	100

4.10	The top-ranked dialogue states with two different parameter sets; namely, $[\alpha = 1, \beta = 0, \gamma = 0]$ and $[\alpha = 1, \beta = 1, \gamma = 1]$ after the user utterance shown in Fig. 4.9. Aomori city was incorrectly preferred to Aomori prefecture to fill the place slot because the user's previous utterance was recognized to have a city name due to speech recognition errors.	101
------	--	-----

List of Tables

2.1	Labels given to each slot of a hypothesis frame.	22
2.2	Labels given to the update of each slot of a hypothesis frame. . .	24
2.3	Correlation coefficients (R) of the 26 metrics against task completion time and user satisfaction.	30
2.4	Squared correlation coefficients (R^2) and the root mean square error (RMSE) (in brackets) for multiple linear regression (MLR) and support vector regression (SVR).	32
2.5	Commonality between the trained support vector regression (SVR) models for task completion time. (Squared correlation coefficients (R^2) and the root mean square error (RMSE) in brackets.)	33
2.6	Commonality between the trained support vector regression (SVR) models for user satisfaction. (Squared correlation coefficients (R^2) and the root mean square error (RMSE) in brackets.)	34
2.7	Five dominating weighting factors for the prediction of task completion time.	35
2.8	Five dominating weighting factors for the prediction of user satisfaction.	36
3.1	Breakdown of the slot value samples for each grouping.	59
3.2	List of word-level features. Labels <i><not available></i> , <i><not used></i> , and <i><new></i> indicate the modifications we made to the features used in (Hazen et al., 2002).	60
3.3	List of utterance-level features. Labels <i><not available></i> , <i><not used></i> , and <i><new></i> indicate the modifications we made to the features used in (Hazen et al., 2002).	61

3.4	F-measure for the method that uses posterior probability, the conventional and proposed models.	64
3.5	Matrix of counts of correct and incorrect items for the conventional (conv.) and proposed (prop.) models.	66
3.6	F-measure for models each trained without D6 and one of the remaining discourse features.	66
3.7	Weights assigned to each of the discourse features in the six obtained confidence models. Averages and standard deviations of the weights are shown in the last column.	67
4.1	Examples of dialogue act type sequences and their per word tri-gram probabilities in the train ticket reservation domain.	87
4.2	Examples of dialogue act type sequences and their per word tri-gram probabilities in the weather information service domain. . .	88
4.3	The 18 dialogue state update patterns and their occurrence probabilities in the train ticket reservation domain. See Fig. 4.5 for the details of the binary attributes. Attributes 1-7 are ordered from left to right.	89
4.4	The 23 dialogue state update patterns and their occurrence probabilities in the weather information service domain.	90
4.5	Evaluation results for each combination of the weighting factors in the train ticket reservation (TRAIN) and weather information service (WEATHER) domains.	94

CHAPTER 1

Introduction

1.1 Background and Motivation

Due to advances in speech recognition and synthesis technologies, spoken dialogue systems, which communicate with humans using speech, have been increasingly studied and developed over the past decades. Currently, such systems can perform a wide variety of tasks, such as train ticket reservation (Core and Schubert, 1996), flight planning (Walker et al., 2002), delivery of weather information (Glass and Hazen, 1998; Zue et al., 2000; Nakano et al., 2001), tutoring (Litman and Silliman, 2004), and call routing (Gorin et al., 1997; Chu-Carroll and Carpenter, 1999). With their performance improving rapidly, they are already seeing some commercial success (Heisterkamp, 2001; Pieraccini and Lubensky, 2005; Feng et al., 2005). Such systems make it possible for us to communicate with machines by speech—our most familiar way of communication—making it possible for us to accomplish certain tasks intuitively and also with low cognitive load. They have the potential to greatly change our lifestyle.

Figure 1.1 shows an excerpt from a dialogue conducted between a user and a state-of-the-art spoken dialogue system in the flight reservation domain. As illustrated in this example, compared to speech command interfaces, which respond to user’s single utterances to perform simple tasks, such as turning on a light or switching television channels, spoken dialogue systems have to deal with rather complex tasks. With tasks becoming increasingly complex with exchanges requiring more than a few turns, instead of *speech understanding*, which understands independent user utterances, *discourse understanding*, which understands

user utterances from the context of a dialogue, is becoming important.

Good discourse understanding would offer many advantages to users in an interaction. For example, communication would become more efficient because the user would not need to tell the same thing to the system repeatedly. This feature can be achieved by keeping track a dialogue history (all previous exchanges of user and system utterances). By keeping track of dialogue histories, users would also be able to use referential expressions (e.g., pronouns such as *it*) or ellipses (zero pronouns), which occur frequently in human-human communication.

Sometimes, user utterances may be ambiguous. In such cases, the use of discourse information would be helpful in resolving ambiguities. Suppose that a user utters a station name in a train ticket reservation domain. If the system knows that the user has already uttered the destination, it would be possible for the system to infer that the station name means the place of departure. Similarly, a system could notice its misunderstanding by checking the conformity of its understanding to the discourse, preventing possible further miscommunication. If a system has a plan recognition capability, that is, a collection of predefined knowledge about how to execute a task, the system would be able to detect the user's intention quickly and could generate suitable suggestions (e.g., go to the bus stop and get on a bus) for task completion (e.g., go to the station).

Research on discourse understanding in spoken dialogue systems has been conducted mainly in two research fields: speech processing and natural language processing (NLP) in combination with artificial intelligence (AI). In speech processing, several approaches have attempted to incorporate contextual (discourse) information to improve speech recognition; e.g., to switch speech recognition grammars depending on dialogue phases. However, they mainly focus on improving speech recognition and do not deal with the main problem of discourse understanding, which is to understand user utterances with regards to the context in order to accurately estimate user intentions to complete certain tasks.

In NLP/AI research, much work has been done on discourse understanding based on plan recognition (Allen and Perrault, 1980; Carberry, 1990). However, hand-crafted rules are generally used to account for very specific phenomena in human-human dialogues. Therefore, it is acknowledged that plan-based approaches do not work sufficiently when there is an unexpected input from the

Sys: hello, what city are you leaving from?
User: hartford connecticut (recog: hartford connecticut)
Sys: a flight from hartford. where would you like to go?
User: orlando florida (recog: orlando florida)
Sys: traveling to orlando. on what date would you like to travel
User: october thirty first late morning (recog: october three first late morning)
Sys: traveling late morning. on what date would you like to travel
User: october thirty one (recog: october thirty one)
Sys: traveling on wednesday, october thirty first.
Sys: one moment please ... let's see ... flights from hartford to orlando ... if you'll just hold on, this shouldn't take too long ... gosh the network has been slow all day ... i'm glad you called though ... i'm not sure how long this will take ... this has been a long day, but i'm glad you called ... hummmmm. ... this is taking longer than usual ... i wonder if something is wrong ... i'll bet the network is down again!
Sys: okay i have a nonstop on delta airlines departing hartford at ten twenty a. m. arriving into orlando at one ten p. m. is that ok?
User: yes (recog: yes)
:
Sys: do you need a hotel in orlando?
User: yes please (recog: yes please)
Sys: would you like a hotel downtown or near the airport?
User: airport (recog: airport)
Sys: please specify your preferred hotel chain, for example, best western.
User: marriott (recog: marriott)
Sys: i've made a request for a hotel at the marriott. do you want to rent a car in orlando?
User: no car (recog: no car)
Sys: do you want a summary of your trip?
User: cancel the hotel (recog: cancel the hotel)
Sys: would you like me to summarize your trip?
User: yes (recog: yes)

Figure 1.1: A dialogue between a user and a system in the air traffic information service domain taken from the corpus of the DARPA Communicator project (Walker et al., 2002). Speech recognition hypotheses for user utterances are shown in parentheses after 'recog:'.

user, which cause speech recognition errors (Shinyama et al., 2000; Rich et al., 2001). Actually, one of the current problems in the NLP/AI approach is how to apply plan-based approaches to discourse understanding under the assumption that user utterances contain possible errors.

Although a number of systems are currently capable of discourse understanding in some way or other, discourse understanding has been largely handled in a heuristic manner because the emphasis in spoken dialogue systems research has been on the creation of workable systems. However, with current systems becoming increasingly complex, we are beginning to see the limitation of such ad hoc approaches. Therefore, the aim of our research is to introduce systematic methods that can improve discourse understanding in realistic spoken dialogue systems, making it possible for the systems to understand error-prone speech recognition results from the context of a dialogue to recognize user intentions more robustly. Although one may argue that future improvements in speech recognition will lead to successful application of NLP/AI approaches for robust discourse understanding, it has been suggested that speech recognition errors are unlikely to disappear in the near future (Lippmann, 1997). We therefore believe there is a strong need for this research.

In this thesis, we deal with task-oriented dialogues. We do not deal with casual conversations, such as chats, because the topics that the system has to deal with may be too broad and difficult to process with current NLP techniques. In task-oriented spoken dialogue systems, it is crucial to obtain accurate user intentions for task success. For example, in the train ticket reservation domain, the system has to recognize from what station the user intends to depart and to what station the user wants to reach before the reservation of any train seats. The system has to elicit these pieces of information by dialogue. Therefore, this thesis mainly concerns how to accurately recognize user intentions from the context of a dialogue.

1.2 Architecture of a Spoken Dialogue System

Figure 1.2 shows the basic architecture of a spoken dialogue system. Although other architectures, such as those based on the information state model (Larsson

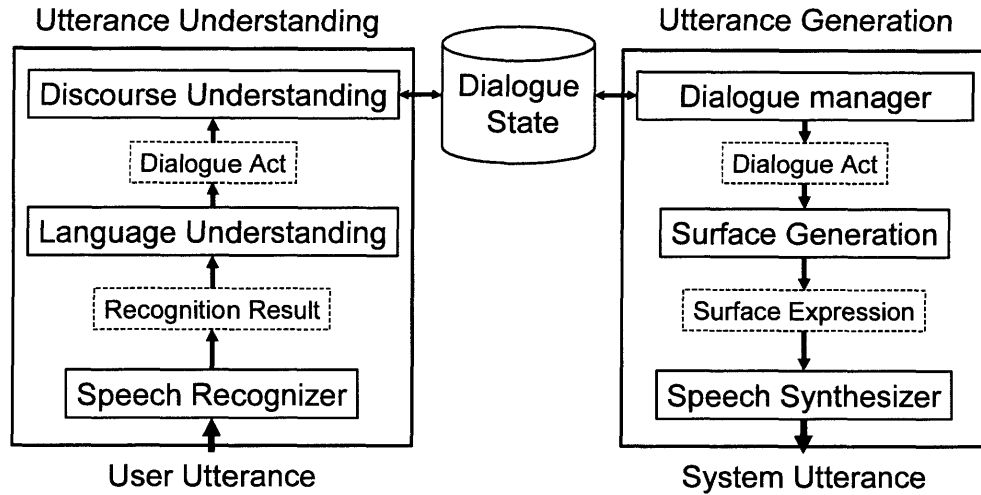


Figure 1.2: Architecture of a spoken dialogue system.

and Traum, 2000; Traum and Larsson, 2003), the hub-and-spoke model (Goddeau et al., 1994), or an Open Agent Architecture (OAA) framework (Bos et al., 2003) can be considered, any architecture will comprise components of similar functionalities. The components of a spoken dialogue system can be divided into those for utterance understanding and utterance generation. Utterance understanding components process incoming user utterances and those for utterance generation components create system responses.

There are three components for utterance understanding: speech recognition, language understanding, and discourse understanding. Speech recognition is performed by a speech recognizer that converts speech signals into a text string (speech recognition result). The speech recognition result is then processed by the language understanding component to derive a meaning representation, typically called a *dialogue act*. Obtaining a dialogue act from speech recognition and language understanding is often called speech understanding. The discourse understanding component then uses the dialogue act to update the current context represented by a *dialogue state*.

Based on a dialogue state, the dialogue manager decides *what to say* to the user in the form of a dialogue act. The surface generation component then creates a surface expression from the dialogue act. Compared to the dialogue manager, this component is concerned with *how to say* the system response. Finally, the speech

synthesizer generates speech signals from the surface expression and responds to the user by speech. The process of understanding a user utterance and generating a system response is repeated until the user completes his/her goal.

When receiving a user utterance, the system works as follows.

1. The speech recognizer receives a user utterance and outputs a speech recognition result, such as an N-best list (a list of N speech recognition hypotheses) or a word graph (a network representation of speech recognition hypotheses with each node in the network representing a word).
2. The language understanding component receives the speech recognition result. Syntactic and semantic analyses are performed to convert it into a meaning representation; that is, a dialogue act. A dialogue act typically comprises a *dialogue act type* that identifies the main intent of the user's utterance with its auxiliary information often encoded as attribute-value pairs.

For example, in the train ticket reservation domain, when the user says "I'd like to go to Tokyo", this utterance would create a dialogue act such as (*refer-destination place=Tokyo*) in which *refer-destination* is the dialogue act type, and *place=Tokyo* is the auxiliary information indicating that the destination place is "Tokyo".

3. The discourse understanding component receives the dialogue act, refers to the current dialogue state, and updates the dialogue state. Here, a dialogue state is a representation of the current context. We describe the action of the discourse understanding component in more detail in the next section.
4. The dialogue manager refers to the updated dialogue state, decides the next utterance, and outputs the next content to be spoken as a dialogue act. At the same time, the dialogue manager updates the dialogue state with its dialogue act so that the dialogue state can preserve the history of system utterances. For example, when the system wants to confirm whether the destination of the user is "Tokyo", it would create a dialogue act such as (*confirm-destination place=Tokyo*).

5. The surface generation component receives the dialogue act and produces the surface expression, namely, the next words to be spoken, possibly augmented with prosodic assignment. For example, “Do you want to go to Tokyo?” would be generated for (*confirm-destination place=Tokyo*). Prosodic assignment may be performed for the speech synthesizers to generate accented and sometimes emotional speech in the next step.
6. The speech synthesizer receives the surface expression and responds to the user by speech.

1.3 Discourse Understanding in Spoken Dialogue Systems

This thesis concerns the discourse understanding component of a spoken dialogue system. The function of the discourse understanding component is to appropriately update the dialogue state (i.e., the context) using the current dialogue state and the incoming dialogue act derived from a user utterance so that the system can make as appropriate a response as possible.

A dialogue state comprises a collection of bits of information concerning the dialogue. It typically includes the system’s recognition of a user intention (intention recognition result) as well as the history of intention recognition results, user utterances, and system utterances. Among these, we consider the intention recognition result to be the most important feature of the dialogue state because it is the system’s understanding of what the user intends to accomplish on the basis of all previous exchanges of user and system utterances.

We assume that an intention recognition result can be expressed simply by a frame expression (Bobrow et al., 1977), which is common in many practical dialogue systems and sometimes referred to as an electronic form or *E-form* (Goddeau et al., 1996). A frame/E-form is a bundle of slots that consist of attribute-value pairs concerning a certain domain. Therefore, an update of an intention recognition result means filling, changing, and deleting the slot values of a frame.

Figure 1.3 shows how the intention recognition results (frames) are updated in

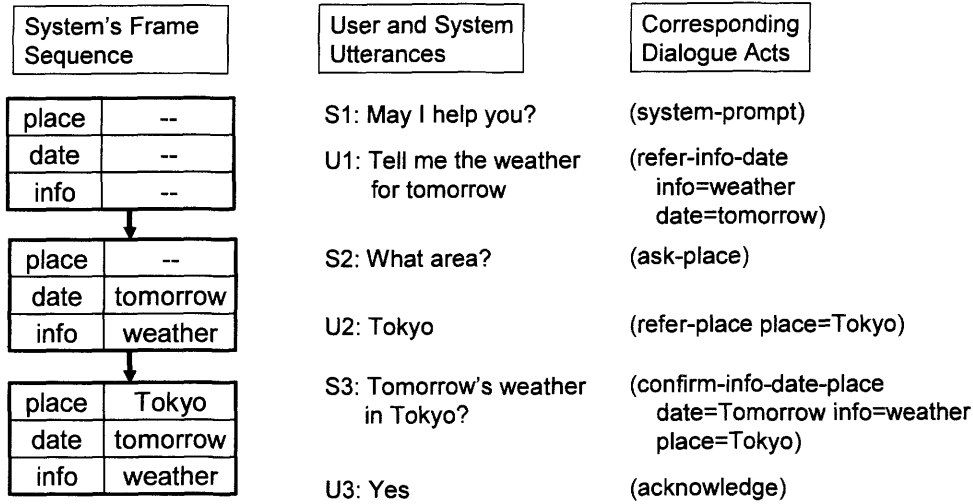


Figure 1.3: An example of frame updates in a dialogue. (S means a system utterance and U a user utterance.)

the course of a dialogue in a weather information system. It also shows dialogue acts corresponding to the user and system utterances. In this example, the intention recognition result consists of slots representing place, date, and information type (general weather, temperature, and probability of precipitation). At first, the slots contain no values. After several exchanges of utterances, the slots are updated and the system finally recognizes the correct user intention. Through this interactive process with the user, the intention recognition result gets closer to the true user intention.

The update of an intention recognition result is typically performed by a hard-coded process (Filisko, 2002) or by hand-crafted rules (Higashinaka et al., 2006). Figure 1.4 shows an example of a discourse understanding rule. This rule is interpreted as follows: “if the type of the user’s dialogue act is *refer-place* and the place slot in the frame has not been filled, then fill the place slot with the place name associated with the dialogue act”. The system generally possesses many such rules to appropriately update the dialogue state. Although the rule in Fig. 1.4 is a very simple one, there can be rules that may perform complex operations, such as filling certain slots with possible slot value candidates in previous user’s dialogue acts when some slot values are denied by the user, or choosing not to fill certain slots when there is enough evidence in the dialogue

```
if user-dialogue-act.type = 'refer-place' and
  not filled(place) then
  place <- user-dialogue-act.place
end
```

Figure 1.4: Example of discourse understanding rules

history that the dialogue act in question has an error.

Plans have often been used to represent dialogue states (Allen and Per-rault, 1980; Carberry, 1990; Litman and Allen, 1990; Chu-Carroll and Carberry, 1998). Traditionally, plan-based discourse understanding methods have been implemented mostly in keyboard-based dialogue systems. Although there have been some recent attempts to apply them to spoken dialogue systems (Allen et al., 2001; Rich et al., 2001), considering the current performance of speech recognizers and the limitations in task domains, we believe frame-based discourse understanding and dialogue management are sufficient (Chu-Carroll, 2000; Seneff, 2002; Bobrow et al., 1977).

There are also object-oriented approaches for the modeling of dialogue states (Sparks et al., 1994; Abella and Gorin, 1999). Such approaches model dialogue states as objects that encapsulate the necessary information and behavior for achieving sub-tasks in a dialogue. The dialogue progresses by making transitions among the dialogue states until the task as a whole is complete. As long as the discourse understanding results can be represented by frames, for example, by aggregating the objects' information, we believe our approach can be applied to these models as well.

1.4 Problems Addressed in This Thesis

In this thesis, we deal with three problems that we consider are important in improving discourse understanding in spoken dialogue systems.

1.4.1 Lack of Evaluation Measure for Discourse Understanding

In any field in engineering, it is important to have an evaluation measure. An appropriate evaluation measure enhances the research activity of a field because various approaches can be uniformly compared in an objective manner, making it possible to incrementally improve techniques. In the speech recognition research, the word error rate (WER) is a commonly acknowledged evaluation measure. It is not too much to say that the existence of the WER has led to the current good performance of speech recognizers. In addition, the concept error rate (CER) has been used to evaluate speech understanding.

Although there are such useful measures in speech recognition and speech understanding, no measure has been established for evaluating discourse understanding, which has made it necessary to evaluate it only on the basis of the system's total performance. Such evaluations, however, are greatly influenced by task domains and dialogue strategies. To stimulate research in discourse understanding and also to achieve incremental developments in the field, an appropriate evaluation measure is needed.

1.4.2 Lack of Discourse Information in Confidence Scoring of Intention Recognition Results

Not only do humans have better speech recognition accuracy than speech recognizers, but they also make better use of context to understand speech. One useful capability of humans is the ability to detect a misunderstanding from the context of a dialogue, which can prevent possible miscommunication at a later stage. A system that can accurately detect errors in its intention recognition result would provide improved discourse understanding because it could automatically correct erroneous slot values or have the discourse understanding component inform the dialogue manager that there may be errors in the slot values so that the dialogue manager can preferentially confirm them to make users correct errors promptly.

Recently, *confidence scoring*, a technique to assign reliability scores to speech recognition results, has been applied to detect errors in intention recognition results and has proved useful for dialogue management (Komatani and Kawahara,

2000; Singh et al., 2002; Dohsaka et al., 2003). In confidence scoring, confidence models are trained from data comprising various features. In current confidence scoring of intention recognition results, only the features related to speech recognition and speech understanding are used for the training of confidence models. Considering that the intention recognition result is a result of discourse understanding, the current features may be insufficient and the incorporation of discourse information may be necessary.

1.4.3 High Cost of Hand-crafted Rules for Ranking Dialogue States

Spoken dialogue systems have to deal with ambiguities arising from multiple speech recognition hypotheses as well as from natural language. Humans appropriately use context to resolve such ambiguities. Although ambiguity can be resolved by explicitly asking the user to disambiguate his/her utterances, there are also cases where ambiguity of a user utterance can be resolved using the user's succeeding utterances.

For example, suppose a user says “from Tokyo” and then “to Osaka” in a train ticket reservation domain. Here, suppose also that “from” in the first utterance was inaudible because of noise and therefore not recognized by the system. In this case, the first utterance becomes “Tokyo”, making this utterance ambiguous because Tokyo could be the place of departure or the destination. This ambiguity can be resolved if the system can just wait for the next utterance “to Osaka”, which clearly indicates that the destination is Osaka. After this utterance, the system can understand that Tokyo is the place of departure. Compared to explicitly asking users for disambiguation, this approach could realize a more natural dialogue because of less system intervention.

Conventionally, such understanding has been achieved by allowing the system to keep multiple dialogue states as possible interpretations (e.g., Tokyo as the destination vs. Tokyo as the place of departure) and ranking them using hand-crafted rules in order to decide the best interpretation after each user utterance. However, such rules are costly to make and difficult to port across domains. In addition, this cost can lead to possible inaccuracy. An automatic method for

ranking dialogue states is therefore desired.

1.5 Overview of Our Approaches

1.5.1 Creating an Evaluation Measure for Discourse Understanding

Discourse understanding is a process of updating a dialogue state. Therefore, the discourse understanding performance can be evaluated by looking at how accurate the updated dialogue states are in comparison with their reference dialogue states that can be annotated later. However, by changing the ways they are compared, a variety of metrics can be considered as the evaluation measure candidates, which make it difficult to choose the most suitable one. Since our aim is to improve discourse understanding in order to improve the system's total performance, it is desirable that the improvement measured by any evaluation measure we might find also means the improvement of the system's total performance. For this reason, we believe that the metric that correlates the most with the system's total performance is the most suitable metric.

In our approach, to find such a metric, we first enumerate possible discourse-understanding-related metrics and obtain their correlation against the system's total performance through dialogue experiments using human subjects. We propose using the metric that has the highest correlation as the evaluation measure.

1.5.2 Incorporating Discourse Features into Confidence Scoring of Intention Recognition Results

We train confidence models to estimate the confidence of slot values in the intention recognition result. For a slot value to be correct, the speech recognition and speech understanding results that derived that slot value should be reliable. In addition, the slot value should be contextually adequate. For example, if a slot value has been previously denied or modified by the user, that value is likely to be wrong. On the contrary, if a slot value stays the same over multiple turns, that value is likely to be correct. Such information should be incorporated in the

confidence model training.

In our approach, we introduce a number of discourse-related features (called *discourse features*) that characterize the contextual adequacy of slot values in terms of *Grice's maxims of cooperativeness* (Grice, 1975). Grice's maxims are regarded as norms that should be followed in a collaborative dialogue.

By observing how a slot value violates or conforms to the maxims, it may be possible to represent the contextual adequacy of that slot value. We use the discourse features together with the features obtained from speech recognition results to train confidence models that can accurately classify slot values as correct or incorrect based on both the context and the speech recognition and speech understanding results.

1.5.3 Ranking Multiple Dialogue States by Corpus Statistics

In a dialogue, a dialogue state is updated by a sequence of dialogue acts by the user and system. Therefore, we can hypothesize that a dialogue state that has seen the most likely sequence of dialogue acts and updates is the most probable dialogue state. If we know how likely dialogue acts and dialogue states succeed one another, we can utilize such information to automatically rank dialogue states, disambiguating among possible interpretations to choose the best interpretation at each point in a dialogue.

To derive such information, we make use of a dialogue corpus annotated with reference dialogue acts and dialogue states. We derive two kinds of statistical information from the corpus: (1) the N-gram probability of a dialogue act type sequence and (2) the occurrence probability of a dialogue state update pattern. We use these two probabilities to score dialogue states for their automatic ranking, reducing the cost currently needed for hand-crafted rules.

1.6 Thesis Outline

This thesis comprises five chapters. Chapters 2, 3, and 4 respectively describe our approaches for creating an evaluation measure for discourse understanding

in spoken dialogue systems, for incorporating discourse features into confidence scoring of intention recognition results, and for ranking multiple dialogue states by corpus statistics to resolve the ambiguity in user intentions arising from multiple speech recognition hypotheses and speech understanding. Chapter 5 concludes this thesis and mentions future work.

CHAPTER 2

Creating an Evaluation Measure for Discourse Understanding

2.1 Introduction

Unlike simple speech understanding systems that understand a single user utterance and respond to it without taking context into account, spoken dialogue systems need to deal with multiple exchanges of utterances in the context of dialogues. This process is achieved by updating a dialogue state each time a user utterance is made. A dialogue state is a collection of bits of information that the system internally stores. Included in that information are the understanding result of the user utterances up to a certain point of time as well as grounding information, the user utterance history, and the system utterance history.

There is no well-established measure for evaluating discourse understanding. As a result, it has been evaluated only on the basis of the system's total performance, such as task completion rate, task completion time and user satisfaction estimated by questionnaires. However, such evaluations are greatly influenced by the task domains and dialogue strategies that the systems employ. This fact makes it difficult to compare various systems' discourse understanding. A measure that appropriately evaluates specifically discourse understanding capability would be useful for further improvement of discourse understanding components.

In evaluating single utterance understanding, which does not include discourse understanding, the *concept error rate* (CER) or the *keyword error rate* (KER) has been widely used as an evaluation measure (Glass et al., 2000). Using the CER

of discourse understanding results is one possibility. However, it may not be appropriate for the evaluation of discourse understanding, because it is unclear whether the CER correlates closely with the system's performance. A measure should have high correlation with what it is measuring. Since we seek to maximize the system's performance by improving the discourse understanding capability, the measure has to have high correlation with the system's total performance. Our aim is to find such a measure.

As our approach, we enumerate possible discourse-understanding-related metrics and obtain their correlation against the system performance through dialogue experiments using human subjects. The experiments have to be performed in several task domains utilizing various strategies to create a measure that can commonly be used across different systems. We can use the metric that has the highest correlation as the evaluation measure. It is also possible that the combination of some metrics will lead to a higher correlation. For such cases, we apply regression methods to create a single measure using the metric candidates. The methodology used here is similar to the one used in PARADISE (Walker et al., 1997) in that the impacts of various features of dialogues are assessed based on their correlation with the system's total performance. Instead of assorted features of dialogues, we focus specifically on discourse-related features and use their correlation to find appropriate evaluation measures for discourse understanding.

Section 2.2 explains the need for an evaluation measure for discourse understanding. In Section 2.3, using an example dialogue, we describe why conventional metrics, such as CER, cannot be used for the evaluation. In Section 2.4, our approach and various metrics concerning discourse understanding are described in detail. Then, in Section 2.5, we describe the dialogue experiments we performed to collect dialogue data using our dialogue systems. In Section 2.6, we show the correlation between each metric and the system performance and describe our attempt to create better measures using regression methods followed by detailed analysis of the obtained models. This chapter concludes with a short summary and some recommendations.

2.2 The Need for an Evaluation Measure in Discourse Understanding

A qualitative measure for evaluating each component in spoken dialogue systems would be useful for improving components. Speech recognition and language understanding modules have been evaluated using the word error rate (WER) and the CER or KER, respectively. Although they may not be the best evaluation measures for those components, they are intuitively reasonable and widely used among researchers and developers. Thanks to these measures, we can compare the speech recognition and language understanding components of various systems dealing with various tasks and strategies.

However, no well-established measure for discourse understanding exists, which makes it necessary to evaluate discourse understanding on the basis of the system's total performance, such as task completion rate, task completion time, and user satisfaction. Since the system's total performance varies depending on the system's task domains and dialogue strategies, when we compare several different discourse understanding components, task domains and dialogue strategies have to be fixed. Experiments have to be redone whenever the settings are changed, which makes the evaluation cost high.

A measure that can appropriately and specifically evaluate the discourse understanding capability would be useful for reducing the cost of dialogue experiments and making clear the performance of discourse understanding of various systems. Therefore, there is a strong need for an evaluation measure for discourse understanding.

2.3 Problem

What is an appropriate measure for discourse understanding? One candidate is the CER of a system's frames. However, this measure may not be suitable because its degree of correlation with system performance is uncertain. Since we seek to maximize the system's performance by improving discourse understanding, the measure should correlate highly with the system's total performance. There may be other measures that have higher correlation, and the straightforward use of

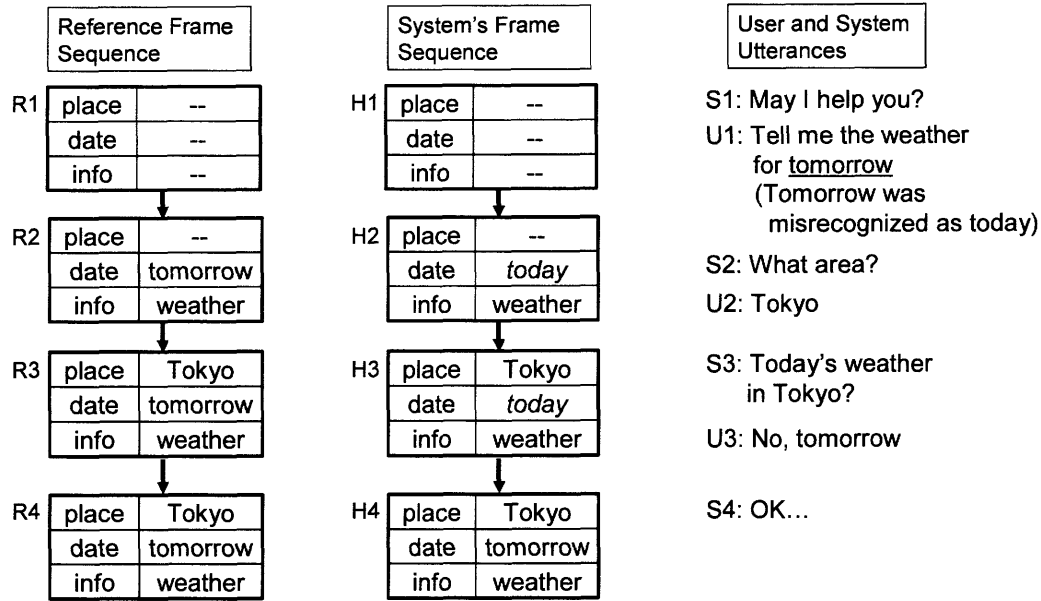


Figure 2.1: An example of frame updates with corresponding reference frames. (R and H mean a reference frame and a hypothesis frame respectively.)

the CER may lead to inappropriate evaluation.

Figure 2.1 shows an illustrative example of the problem. We call a system's frame a *hypothesis frame* and the correct frame that can be annotated later a *reference frame*. As a reference frame, we use the ideal discourse understanding result that takes all previous system and user utterances into account instead of using a frame that can be reached from the previous hypothesis frame and the succeeding user utterance.

In the example, part of the user's second utterance "tomorrow" is misrecognized as "today", and the system updated the initial frame (H1) to an incorrect frame (H2). Even after the user's next utterance "Tokyo", the wrong value "today" is still in the date slot (H3). This misunderstood item is later corrected by the user, who notices the error in the frame because of the incorrect system confirmation and corrects the value (H4). R1 to R4 show the references for the corresponding hypothesis frames.

We want to evaluate the system's frame sequence, which results from discourse

understanding. For systems that do not handle previous utterances, the CER is suitable for evaluating utterance understanding because the situation is similar to evaluating a single utterance understanding. However, when we take previous frames into account, the suitability of the CER becomes unclear. For example, there are cases where the resulting frame is wrong, but it may have been updated correctly in part.

Consider three metrics: the slot error rate, the update precision, and the CER. The slot error rate is the rate of wrong values in a frame. The update precision shows the ratio of incorrect slots within updated slots. The CER is the ratio of incorrect slots over the number of filled slots. Our definition of the CER may seem different from the one commonly used. However, since we focus on the discourse understanding result (a frame) instead of attribute-value pairs contained in previous user utterances, the number of substituted slots, deleted slots, and inserted slots over the number of filled slots can be considered to match the definition of the CER. They are derived as shown below.

1. Slot error rate
$$\frac{\# \text{ of incorrect slots}}{\# \text{ of slots}}$$
2. Update precision
$$\frac{\# \text{ of correctly updated slots}}{\# \text{ of updated slots}}$$
3. CER
$$\frac{\# \text{ of incorrect slots}}{\# \text{ of filled slots}}$$

The value of each metric is calculated for each pair of hypothesis and reference frames. In the example (Fig. 2.1), the average slot error rate for the hypothesis frames is $(1/3 + 1/3 + 0/3)/3 = 0.22$, the update precision is $(1/2 + 1/1 + 1/1)/3 = 0.83$, and the CER is $(1/2 + 1/3 + 0/3)/3 = 0.28$. These values encode the discourse understanding in some way. The slot error rate seems a reasonable measure, because a frame is the final result from the discourse understanding component and forms the basis for the next system utterance. However, notice that the slot that has an erroneous value “today” is inherited and counted as an error twice. The update precision, on the other hand, focuses only on the updated

slots, avoiding the shortcomings of the slot error rate, but the entire frame is not taken into consideration. The CER is only different from the slot error rate in that it focuses on the filled slots; it has the same double counting problem.

Currently, it is not clear whether the evaluation should focus on the frames themselves or the way they are updated in a dialogue, which makes it difficult to decide the most reasonable metric. Moreover, there may be other metrics that are more appropriate for the evaluation.

2.4 Approach

As our approach, we enumerate possible metrics concerning frame sequences and choose those that have good correlation with the system’s performance as evaluation measures. We also combine the enumerated metrics to create a single measure by regression methods. It is likely that such a combined measure will have higher correlation by taking many aspects of frame values and updates into account.

Then, we perform dialogue experiments using human subjects and obtain both the value of each metric and the total performance of a dialogue. After collecting sufficient dialogue data, we calculate the correlation between the value of each metric and the total performance of dialogues. We treat the metric that has the highest correlation as the appropriate evaluation measure. In addition, by combining the metric values, we create a single measure by regression methods using all the values of metric candidates as explaining variables and the total performance as the explained variable.

To find a measure that is commonly applicable independent of task domains and dialogue strategies, dialogue experiments have to be performed using different task domains and different dialogue strategies.

2.4.1 Metric Candidates

Besides the slot error rate, update precision, and the CER, we came up with additional metrics that can be categorized into five groups depending on the viewpoints. They are all calculable by comparing hypothesis frames and reference frames. We consider it necessary that the evaluation can be achieved by simple

calculation, such as by comparing the hypothesis frame with the reference frame, so that it can be easily applied to various systems by developers and researchers in the field. The value of each metric in a dialogue is represented by the average value.

1. ***Metrics concerning slot values:*** Metrics comparing the values of every slot of a hypothesis frame with that of a reference frame. The slot error rate is one of them. And slot accuracy, insertion error rate, deletion error rate, and substitution error rate also are considered.
2. ***Metrics concerning updated slot values:*** Metrics comparing the values of only the updated slots. With these metrics, we can avoid the double counting of inherited errors in slots. Update precision is one of them. The calculation is performed in two ways. One concerns the correctness of updated slots in a hypothesis frame, the other the correctness of those in the reference frame. For example, update precision is the ratio of correctly updated slots in the updated slots, whereas update recall is the ratio of correctly updated slots over the slots that should be updated.
3. ***Metrics concerning filled slot values:*** Sometimes the aim of a task is not to fill every slot but to fill some of them. To reflect such cases, we propose metrics that focus only on the filled slots. These metrics are calculated for a hypothesis frame and a reference frame.
4. ***CER:*** The same as the conventional CER. It expresses the correctness of filled slots. The difference from the metrics concerning the filled slots is that this metric includes the insertion error. This metric is also calculated for a hypothesis frame and a reference frame.
5. ***A metric concerning a frame sequence:*** If the user intention is exactly recognized by the system, dialogue management is likely to work effectively, leading to improvement of total performance. Therefore, we propose a frame match rate, which is the rate that the hypothesis frame is exactly the same as the reference frame in all frames in a dialogue.

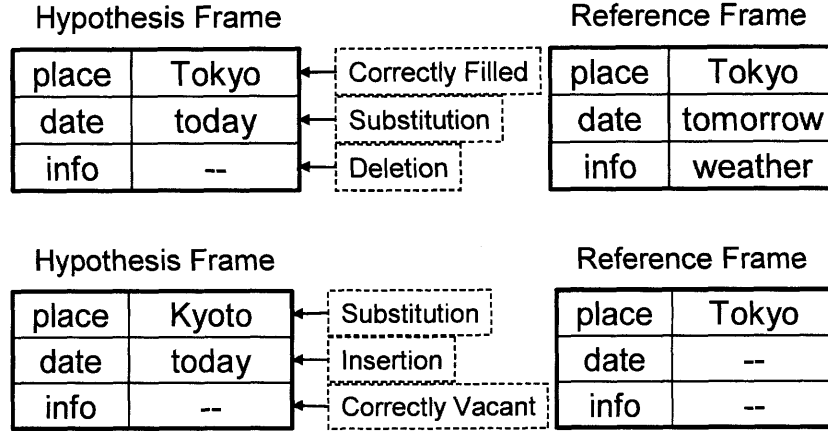


Figure 2.2: Labeling the slot values of a hypothesis frame.

Table 2.1: Labels given to each slot of a hypothesis frame.

Correctly Vacant	$Ref = Hyp$ and $Ref = Null$
Correctly Filled	$Ref = Hyp$ and $Ref \neq Null$
Insertion	$Ref = Null$ and $Hyp \neq Null$
Deletion	$Ref \neq Null$ and $Hyp = Null$
Substitution	$Ref \neq Hyp$ and $Ref \neq Null$ and $Hyp \neq Null$

2.4.2 Labeling

Here, we describe the procedure for deriving the values of metric candidates explained in the previous section. First, we label each slot of the hypothesis frame by comparing the corresponding slot in the reference frame. The reference frame has to be hand-crafted in advance.

The comparison is performed in two ways. One is a simple comparison of each value of the slots performed to see if the values are the same or different or if the slots have values at all. From this comparison, each slot of a hypothesis frame is given one of five labels (Fig. 2.2). Table 2.1 shows the labeling scheme. In the table, values of a certain slot of a hypothesis frame, a reference frame, and the previous frame are written as *Hyp*, *Ref*, and *Prev*, respectively. If a slot does not have a value, it is denoted as *Null*.

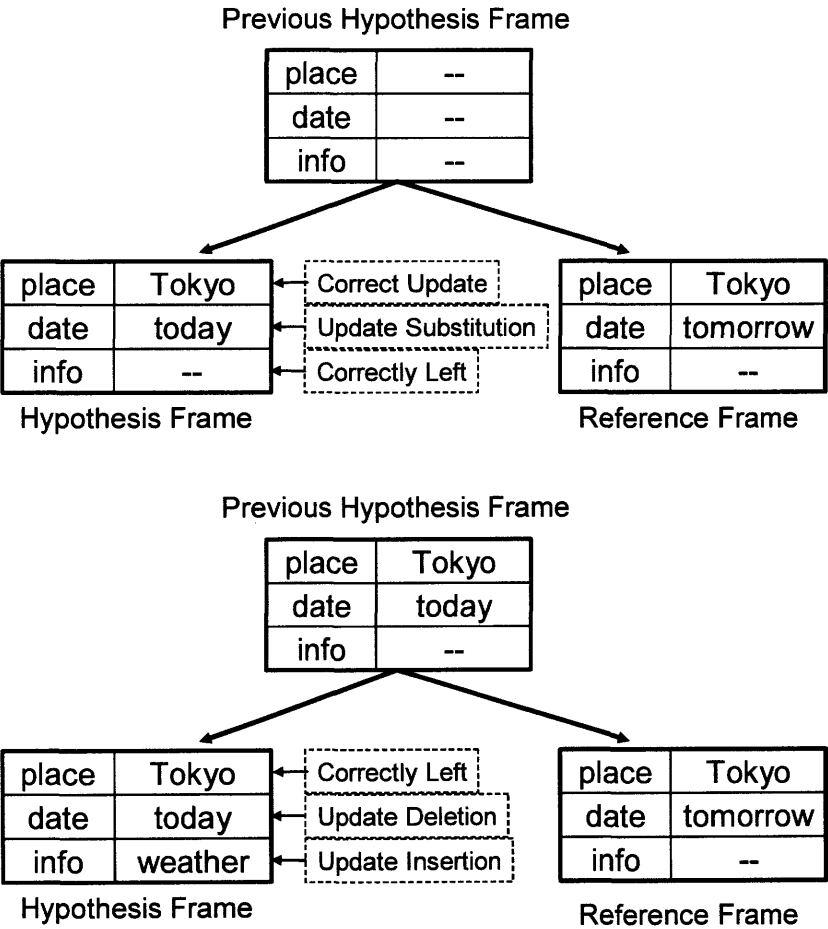


Figure 2.3: Labeling the update of a hypothesis frame.

Table 2.2: Labels given to the update of each slot of a hypothesis frame.

Correctly Left	$Prev = Ref$ and $Prev = Hyp$ and $Ref = Hyp$
Correct Update	$Prev \neq Ref$ and $Prev \neq Hyp$ and $Ref = Hyp$
Update Insertion	$Prev = Ref$ and $Prev \neq Hyp$
Update Deletion	$Prev \neq Ref$ and $Prev = Hyp$
Update Substitution	$Prev \neq Ref$ and $Prev \neq Hyp$ and $Ref \neq Hyp$

The other comparison is performed for changes from the previous hypothesis frame; “the difference between the previous hypothesis frame and the current hypothesis frame” is compared with “the difference between the previous hypothesis frame and the reference frame”. From this comparison, one of five labels is assigned to each slot of a hypothesis frame (Fig. 2.3). Table 2.2 shows the labeling scheme.

2.4.3 List of Metric Candidates

From the ten labels, we derive the values of metric candidates. The derivation formulae are presented below, where CV, CF, I, D, S, CU, CL, UI, UD, and US represent the number of slots labeled Correctly Vacant, Correctly Filled, Insertion, Deletion, Substitution, Correct Update, Correctly Left, Update Insertion, Update Deletion, and Update Substitution respectively. There are 26 metric candidates in all.

Metrics concerning slot values:

1. Slot accuracy

$$\frac{CV + CF}{CV + CF + I + D + S}$$

2. Insertion error rate

$$\frac{I}{CV + CF + I + D + S}$$

3. Deletion error rate

$$\frac{D}{CV + CF + I + D + S}$$

4. Substitution error rate

$$\frac{S}{CV + CF + I + D + S}$$

5. Slot error rate

$$\frac{I + D + S}{CV + CF + I + D + S}$$

Metrics concerning updated slot values in a hypothesis frame:

6. Update precision

$$\frac{CU}{CU + US + UI}$$

7. Correctly remaining rate in hypothesis

$$\frac{CL}{CL + UD}$$

8. Update insertion error rate in hypothesis

$$\frac{UI}{CU + US + UI}$$

9. Update deletion error rate in hypothesis

$$\frac{UD}{CL + UD}$$

10. Update substitution error rate in hypothesis

$$\frac{US}{CU + US + UI}$$

Metrics concerning updated slot values in a reference frame:

11. Update recall

$$\frac{CU}{CU + US + UD}$$

12. Correctly remaining rate in reference

$$\frac{CL}{CL + UI}$$

13. Update insertion error rate in reference

$$\frac{UI}{CL + UI}$$

14. Update deletion error rate in reference

$$\frac{UD}{CU + US + UD}$$

15. Update substitution error rate in reference

$$\frac{US}{CU + US + UD}$$

Metrics concerning filled slot values in a hypothesis frame:

16. Slot accuracy for filled slots in hypothesis

$$\frac{CF}{CF + I + S}$$

17. Insertion error rate for filled slots in hypothesis

$$\frac{I}{CF + I + S}$$

18. Substitution error rate for filled slots in hypothesis

$$\frac{S}{CF + I + S}$$

19. Slot error rate for filled slots in hypothesis

$$\frac{I + S}{CF + I + S}$$

Metrics concerning filled slot values in a reference frame:

20. Slot accuracy for filled slots in reference

$$\frac{CF}{CF + D + S}$$

21. Deletion error rate for filled slots in reference

$$\frac{D}{CF + D + S}$$

22. Substitution error rate for filled slots in reference

$$\frac{S}{CF + D + S}$$

23. Slot error rate for filled slots in reference

$$\frac{D + S}{CF + D + S}$$

24. CER for a hypothesis frame

$$\frac{I + D + S}{CF + I + S}$$

25. CER for a reference frame

$$\frac{I + D + S}{CF + D + S}$$

A metric concerning a frame sequence:

26. Frame match rate

$$\frac{\# \text{ of exactly correct frames}}{\# \text{ of frames}}$$

2.4.4 Performance Measure

System performance has been evaluated in many ways, but it is not certain what really is a valid system performance measure. We chose task completion time and user satisfaction, whose values are commonly used for system evaluations.

In this research, the aim of a dialogue is to complete a task. Efficiently completing a task is an important factor in improving system performance. Therefore, we employ task completion time to represent the performance of a dialogue. We also employ user satisfaction as determined by questionnaires, a method used by many researchers, such as (Walker et al., 2000). Although there is always controversy concerning the validity of questionnaires to estimate user satisfaction, no alternatives have been proposed.

2.5 Data Collection

2.5.1 Systems

We created three systems to perform the dialogue experiments for data collection. One is in a weather information service domain (**WI**), and the other two are in a meeting room reservation domain (**MR-1**, **MR-2**).

WI provides Japan-wide weather information. Users specify a prefecture name, a city name, a date, and an information type (general weather, temperature, and precipitation) to obtain the desired information. The system has four slots for understanding. It has a speech recognition vocabulary of 853. The language model is a trigram trained from the randomly generated texts of acceptable phrases.

MR-1 and MR-2 provide meeting room reservation service. Users specify a date, a room, and start and end times for the reservation. The systems has four slots for understanding. Both have a speech recognition vocabulary of 243. The language model is a trigram trained from the transcription obtained in advance using the same system. The difference between MR-1 and MR-2 lies in their discourse understanding components. Both systems create multiple dialogue state candidates ordered by priority after each user utterance and choose the highest ranked one as the best dialogue state. When deciding the best dialogue state, MR-1 preserves lower ranked dialogue states, whereas MR-2 discards them totally (See (Higashinaka et al., 2003b) for details).

All three systems were developed using the spoken dialogue system toolkit WIT (Nakano et al., 2000). Their speech recognition engine is Julius (Lee et al., 2001) used with its attached acoustic model, and the speech synthesis engine is FinalFluet (Takano et al., 2001). Each system has two switchable dialogue strategies. One is to keep accepting user utterances until it has enough information to fulfill a task or the user explicitly requests a system response. The other is to confirm each user utterance.

2.5.2 Experiment

Using the three systems, we collected dialogue data for analysis. The dialogue data were collected using naive users in acoustically insulated booths.

Twelve subjects used WI. Each subject was given a task sheet listing what should be requested. They were instructed to complete the tasks one by one. We prepared eight task patterns. Together with the two dialogue strategies, each subject performed 16 dialogues, for a total 192 dialogues collected. Twenty-eight subjects used MR-1 and MR-2. Using four task patterns, two dialogue strategies, and two systems, each performed 16 dialogues, and 448 dialogues were collected.

After completing each dialogue, each subject was asked to fill out a questionnaire; the same one used in (Walker et al., 2000). The questionnaire is composed of nine questions concerning text to speech (TTS) performance, automatic speech recognition (ASR) performance, task ease, interaction pace, user expertise, system response, expected behavior, comparable interface, and future use and is on a 1-to-7 Likert scale.

We recorded system utterances, start and end times of user utterances, and dialogue states before and after the user utterance. The user’s voice and system’s voice were also recorded, and all user utterances were transcribed. Dialogues in which it took more than three minutes to complete the task were treated as failures. Task completion rates for WI, MR-1 and MR-2 were 95.8% (185/192), 91.1% (204/224), and 88.4% (198/224), respectively. The word error rates (WER) for WR and MR-1+MR-2 were 30.01% and 33.92%, respectively.

We hand-annotated reference frames. To avoid a large hand-labeling effort, we prepared an annotating tool that processes transcriptions to generate pre-reference frames, which were later corrected by human labellers. The correction took several hours for our dialogue data. Then, using the labeling scheme, we labeled each slot of the corresponding hypothesis frame and obtained all 26 values of the metrics for each dialogue. Task completion times were normalized using task patterns and dialogue strategies because task completion time can be greatly influenced by them. We used the total score of the questionnaire to represent user satisfaction.

2.6 Data Analysis

2.6.1 Correlations of the Metric Candidates

Table 2.3 shows the correlation coefficients of the 26 metrics against task completion time and user satisfaction. These are the results obtained when we used all the data: WI, MR-1, and MR-2. (Hereafter, we use WI + MR-1 + MR-2 to express the combined data of the systems.) For analysis, we used only successful dialogues for which task completion times and user satisfaction data were available. In WI + MR-1 + MR-2, there were 584 samples for task completion time after removing three 3σ outliers, and 587 samples for user satisfaction.

The update recall has relatively high correlation with a correlation coefficient -0.647 followed by -0.607 of frame match rate and -0.579 of update precision. The tendency is similar for user satisfaction, although the correlation coefficients are basically lower. By simple linear regression analysis with ten-fold cross validation, we found that update precision, update recall, and frame match rate explain 32.8%, 41.32%, and 36.31% of task completion time and 11.89%, 18.81% and 15.73% of user satisfaction respectively.

As a result, we can say that the update recall, frame match rate, and update precision are strong candidates for evaluation measures, especially the update recall.

2.6.2 Obtained Regression Models

We used two regression methods to create a single evaluation measure: *multiple linear regression (MLR)* and *support vector regression (SVR)*. For the MLR, the *m5'* method (Yong Wang and Ian H. Witten, 1997; Witten and Frank, 1999) was used for attribute selection instead of the greedy method. SVR is an optimization-based approach for solving machine learning regression problems based on support vector machines (Vapnik, 1995; Smola and Scholköpfung, 1998; Chang and Lin, 2001). We used a polynomial kernel expressed as

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d \quad \text{where } d = 2 \quad (2.1)$$

Table 2.3: Correlation coefficients (R) of the 26 metrics against task completion time and user satisfaction.

	Task completion time	User satisfaction
1. Slot accuracy	-0.554	0.336
2. Insertion error rate	0.117	-0.004
3. Deletion error rate	0.318	-0.210
4. Substitution error rate	0.450	-0.294
5. Slot error rate	0.554	-0.336
6. Update precision	-0.579	0.358
7. Correctly remaining rate in hypothesis	-0.437	0.309
8. Update insertion error rate in hypothesis	0.326	-0.179
9. Update deletion error rate in hypothesis	0.437	-0.309
10. Update substitution error rate in hypothesis	0.451	-0.296
11. Update recall	-0.647	0.441
12. Correctly remaining rate in reference	-0.182	0.099
13. Update insertion error rate in reference	0.572	-0.397
14. Update deletion error rate in reference	0.182	-0.099
15. Update substitution error rate in reference	0.386	-0.247
16. Slot accuracy for filled slots in hypothesis	-0.458	0.237
17. Insertion error rate for filled slots in hypothesis	0.127	-0.007
18. Substitution error rate for filled slots in hypothesis	0.441	-0.271
19. Slot error rate for filled slots in hypothesis	0.458	-0.237
20. Slot accuracy for filled slots in reference	-0.542	0.330
21. Deletion error rate for filled slots in reference	0.308	-0.199
22. Substitution error rate for filled slots in reference	0.452	-0.263
23. Slot error rate for filled slots in reference	0.542	-0.330
24. CER for a hypothesis frame	0.430	-0.247
25. CER for a reference frame	0.404	-0.214
26. Frame match rate	-0.607	0.406

We created regression models for each regression method using *task completion time normalized by the task pattern and the dialogue strategy*, and *user satisfaction* as the explained variables and the 26 metrics as explaining variables.

Table 2.4 shows squared correlation coefficients (R^2) and the root mean square error ($RMSE$) for the two regression methods. These are the results of ten-fold cross validation. When task completion time is the explained variable, most of the obtained regression models fit comparatively well and show validity as evaluation measures. For user satisfaction, the fit is not as good. The performance of SVR is similar to that of MLR.

One may notice that the regression models for MR-1+MR-2 perform better on both task completion time and user satisfaction than for WI. This is because, in WI, for certain city names, repeated misrecognition happened, which caused the system to have slots that have been substituted in the same dialogue a number of times. Since the metric values are represented by their average values in a dialogue, there are often cases where the ratio of substitution error is the same, but the number of times the error occurred is not. In such cases, it is difficult for regression models to achieve high correlation. As evidence, when we examined the individual correlation of each metric with system performance for WI and MR-1+MR-2, we found that the slot substitution error rate accounts for 11.3% of user satisfaction for MR-1+MR-2, whereas it accounts for only 0.19% of user satisfaction for WI.

Figure 2.4 shows the distribution of actual and predicted task completion times for the acquired model using WI + MR-1 + MR-2. The grouping of data, which appears as a horizontal line just above -1.0 in the vertical axis, means that dialogues with different actual task completion times were forcefully mapped to the same task completion times by the regression model since they have identical discourse understanding characteristics. This is attributable to possible differences in the duration of user pauses and speech intervals among the subjects and the limitations of using the average values of the metrics in a dialogue as the discourse features.

In the case of WI + MR-1 + MR-2, the obtained regression models explain 44 % of the task completion time, and 19.5 % of user satisfaction. In comparison with the case of a single metric, the regression methods provide a slightly better

Table 2.4: Squared correlation coefficients (R^2) and the root mean square error (RMSE) (in brackets) for multiple linear regression (MLR) and support vector regression (SVR).

	Task completion time		User satisfaction	
	MLR	SVR	MLR	SVR
WI	0.444 (0.718)	0.318 (0.809)	0.125 (1.514)	0.172 (1.470)
MR-1	0.409 (0.704)	0.445 (0.688)	0.158 (1.220)	0.212 (1.168)
MR-2	0.510 (0.690)	0.506 (0.708)	0.247 (1.114)	0.255 (1.105)
MR-1 + MR-2	0.474 (0.684)	0.483 (0.690)	0.198 (1.167)	0.245 (1.128)
WI + MR-1 + MR-2	0.429 (0.717)	0.440 (0.724)	0.180 (1.284)	0.195 (1.275)

prediction of system performance.

2.6.3 Commonality in Regression Models

To check whether a regression model trained from the data of one domain/system has commonality with that of another, we calculated R^2 and $RMSE$ for every combination of models. Table 2.5 shows the results for the SVR models with task completion time as the explained variable. Most of the R^2 values are around 0.4, suggesting that the model of one domain can be safely applied to that of the other. Since the performance of the model trained from WI + MR-1 + MR-2 shows sufficient performance against other models, this model can be used as a reasonable discourse evaluation measure. For this reason, hereafter, we only deal with models trained from WI + MR-1 + MR-2.

Table 2.6 shows the results for SVR models when user satisfaction is used as the explained variable. The tendency is similar for the MLR models.

2.6.4 Important Factor Analysis of Regression Models

Analyzing the obtained SVR models allows us to list up the possible major metrics for the prediction of the explained variables (Hirao et al., 2002). First, the objective function of SVR is defined as

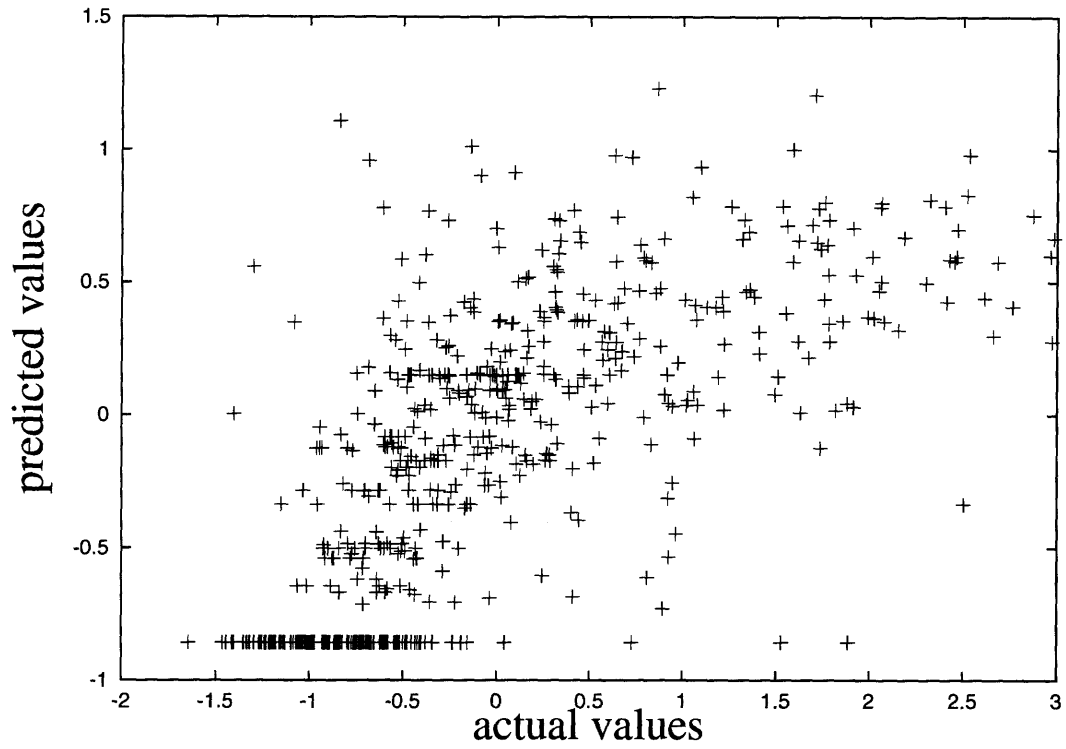


Figure 2.4: Distribution of actual and predicted task completion times by the support vector regression (SVR) model trained from WI + MR-1 + MR-2.

Table 2.5: Commonality between the trained support vector regression (SVR) models for task completion time. (Squared correlation coefficients (R^2) and the root mean square error (RMSE) in brackets.)

Training data \ Test data						WI + MR-1 + MR-2
	WI	MR-1	MR-2	MR-1 + MR-2		
WI	—	0.268 (0.794)	0.369 (0.792)	0.320 (0.793)	0.387 (0.749)	
MR-1	0.342 (1.050)	—	0.368 (0.812)	0.436 (0.718)	0.366 (0.837)	
MR-2	0.084 (1.349)	0.350 (0.789)	—	0.474 (0.702)	0.257 (0.956)	
MR-1 + MR-2	0.262 (0.999)	0.504 (0.636)	0.568 (0.647)	—	0.391 (0.773)	
WI + MR-1 + MR-2	0.495 (0.681)	0.487 (0.648)	0.523 (0.678)	0.503 (0.663)	—	

Table 2.6: Commonality between the trained support vector regression (SVR) models for user satisfaction. (Squared correlation coefficients (R^2) and the root mean square error (RMSE) in brackets.)

Training data \ Test data						WI +
	WI	MR-1	MR-2	MR-1 + MR-2	MR-1+ MR-2	MR-2
WI	—	0.183 (2.407)	0.177 (2.438)	0.185 (2.414)	0.211 (2.122)	
MR-1	0.245 (2.502)	—	0.247 (1.320)	0.249 (1.309)	0.254 (1.395)	
MR-2	0.247 (2.347)	0.262 (1.202)	—	0.277 (1.176)	0.283 (1.236)	
MR-1+MR-2	0.246 (2.426)	0.253 (1.258)	0.262 (1.250)	—	0.267 (1.317)	
WI+MR-1+MR-2	0.193 (2.299)	0.187 (1.649)	0.195 (1.625)	0.199 (1.612)	—	

$$\begin{aligned}
 f(\mathbf{x}) &= \sum_{i; \mathbf{x}_i \in SVs} \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b \\
 &= \sum_{i; \mathbf{x}_i \in SVs} \alpha_i \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}) + b \\
 &= \mathbf{w} \cdot \phi(\mathbf{x}) + b
 \end{aligned} \tag{2.2}$$

where SVs is the set of support vectors, and $\phi(\mathbf{x})$ an explicit representation of new feature vectors \mathbf{x} mapped in the new feature space by the kernel. In the case of the 26 dimensions (features) in our original space and using a second-degree polynomial kernel, the dimensions of the new feature space become 378, and \mathbf{w} is written as

$$\begin{aligned}
 \mathbf{w} = & \left(\sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i1}^2, \dots, \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i26}^2, \right. \\
 & \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i1} x_{i2}, \dots, \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i25} x_{i26}, \\
 & \left. \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i1}, \dots, \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i26}, 1 \right)
 \end{aligned} \tag{2.3}$$

where $x_{i1} \dots x_{i26}$ are the values of the 26 metrics of the i th support vector. By gathering up the weighting factors by the metrics and by the combination of the metrics, we obtain the following weights:

2.7. Summary

Table 2.7: Five dominating weighting factors for the prediction of task completion time.

1.	$W(\text{Update precision})$	-0.154
2.	$W(\text{Update recall})$	-0.108
3.	$W(\text{Update precision}, \text{Update recall})$	-0.092
4.	$W(\text{Deletion error rate for updated slots in reference})$	0.084
5.	$W(\text{Update precision}, \text{Slot accuracy for filled slots in reference})$	-0.080

$$\begin{aligned}
W(x_1) &= \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i1}^2 + \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i1} \\
&\vdots \\
W(x_{26}) &= \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i26}^2 + \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i26} \\
W(x_1, x_2) &= \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i1} x_{i2} \\
&\vdots \\
W(x_{25}, x_{26}) &= \sqrt{2} \sum_{i; \mathbf{x}_i \in SVs} \alpha_i x_{i25} x_{i26}
\end{aligned}$$

We calculated all 351 weights (${}_{26}C_2 + 26$) from the obtained models. Tables 2.7 and 2.8 show the five dominant metrics or combinations of metrics for each model with their weights. The higher the weights are, the more significant the metrics or the combinations of metrics become. From the tables, one can see that the update precision plays a key role in the prediction of task completion time, and the update recall is the most important factor for improving user satisfaction. The frame match rate is also important for user satisfaction.

2.7 Summary

This chapter presented a method for creating an evaluation measure for discourse understanding in spoken dialogue systems. We enumerated metric candidates for the evaluation of discourse understanding and calculated their correlation

Table 2.8: Five dominating weighting factors for the prediction of user satisfaction.

1.	$W(\text{Update recall})$	0.144
2.	$W(\text{Update recall}, \text{Frame match rate})$	0.101
3.	$W(\text{Update precision}, \text{Update recall})$	0.101
4.	$W(\text{Frame match rate})$	0.100
5.	$W(\text{Correctly remaining rate in hypothesis}, \text{Update recall})$	0.086

with the system’s performance through dialogue experiments. We also created a single evaluation measure combining the metrics by regression methods to create a better measure. We found that update recall, frame match rate, and update precision had relatively good correlation with system performance, suggesting they are appropriate as evaluation measures. Above all, update recall can explain 41.3 % of the task completion time, and 18.8 % of user satisfaction. The use of the multiple linear regression (MLR) and support vector regression (SVR) methods revealed that the weighted sum of the metric values can create a measure that performs slightly better than a single metric. With the obtained regression model, 44 % of the task completion time and 19.5 % of user satisfaction can be explained. An analysis of the obtained SVR models also revealed that the update recall, update precision, and frame match rate play important roles in improving system performance.

Overall, we found that user satisfaction is more difficult to predict than task completion time. This can be attributable to the fact that we are dealing with task-oriented dialogues and that there exists a large variety of questionnaire results among subjects.

To conclude, we suggest using the update recall as an evaluation measure for discourse understanding in spoken dialogue systems. Update precision can also be used to support the evaluation. Considering that it is now common practice to combine the precision and recall metrics into an overall F-measure (harmonic mean) and that the two metrics are strong candidates for evaluation measures, the use of the F-measure can also be considered. In fact, we found that the F-measure has a higher correlation than the update precision and the update recall alone and explains 43.4% of the task completion time. Therefore, in

2.7. *Summary*

cases where the two metrics are available, we recommend the F-measure be used. We do not encourage the use of the obtained regression models as evaluation measures because they only offer a slight improvement and because we believe that the measure should be as simple as possible. With the measure, we can safely compare discourse understanding components of various spoken dialogue systems that deal with different task domains and dialogue strategies.

CHAPTER 3

Incorporating Discourse Features into Confidence Scoring of Intention Recognition Results

3.1 Introduction

For a spoken dialogue system to achieve certain tasks while conversing with users, the system has to correctly recognize user intentions. Here, we use the term *user intention* to express the information that the user has in mind and has to convey to the system in order to achieve his/her goal, such as extracting some particular information from the system.

Since users do not always convey their intentions in one utterance and speech recognition errors might occur, the system and the user normally have to exchange several utterances before the system finally recognizes the user's true intention. In such interactive intention recognition, after each user utterance, the system updates the intention recognition result, based on which the system performs dialogue management; namely, it decides what response it should make. Recently, *confidence scoring*, a technique for assigning reliability scores to speech recognition results, has been applied to detect errors in intention recognition results and has proved useful for dialogue management (Komatani and Kawahara, 2000; Singh et al., 2002; Dohsaka et al., 2003). If the detection is successful, the system can safely avoid unnecessary confirmations for reliable slots and ask questions about unreliable or unfilled ones preferentially.

In current confidence scoring for intention recognition results, since words/concepts in user utterances fill the slots, the confidence of words/concepts, which is typically calculated using various features obtained from speech recognition results and speech understanding results for single utterances, is used for the confidence of slot values.

However, this may be inappropriate because slot values are the results of discourse understanding, not the results of single utterance understanding. Consider a case where a slot is filled with a value that has once been denied or corrected by the user in a dialogue. The confidence of that value is likely to be lower than can be calculated for the word/concept in the utterance.

This chapter addresses this problem and proposes incorporating discourse features into the confidence scoring of intention recognition results. In our approach, we introduce a number of discourse-related features (called *discourse features*) that characterize the contextual adequacy of slot values in terms of Grice's maxims of cooperativeness, and use them together along with the features obtained from speech recognition results to train *confidence models* that classify slot values as correct or incorrect based on both the context and the speech recognition/understanding results. Since the features are only available for filled slots, we only deal with slots that have values.

Although this work does not aim at improving discourse understanding of spoken dialogue systems directly, we are hoping to obtain useful ideas for improving our speech understanding component through the process of confidence model training and the analysis of confidence models.

In the next section, we briefly outline the intention recognition process in spoken dialogue systems. In Section 3.3, we explain the need for the confidence scoring of intention recognition results. In Section 3.4, we introduce conventional methods and follow that with a detailed description of our proposed method and the discourse features in Section 3.5. In Section 3.6, we describe the experiments performed to verify the proposed method. In the last section, we summarize the chapter and mention future work.

3.2 Intention Recognition in Spoken Dialogue Systems

As described in Section 1.3, the intention recognition results are represented simply by frame expressions that consist of slot-value pairs (Bobrow et al., 1977; Goddeau et al., 1996). Intention recognition is performed by the discourse understanding component of a system by updating the slot values using speech recognition/understanding results.

In this work, we also assume that words in speech recognition hypotheses or concepts in language understanding results fill the slots, since filling slots with relevant words/concepts can be considered the most basic way of understanding user utterances and is the practice in many practical applications. Another assumption we have is that the system responses are generated using words/concepts in the slots.

To formulate, we model the interaction between the user and the system as follows: (1) The user sends words/concepts or sometimes commands to the system to change the slots. Here, commands represent user utterances that do not contain words/concepts, such as acknowledgments and denials. They have certain effects on the slots, such as the deletion of slot values. (2) The system responds to the user using the words/concepts stored in the slots. Then (1) and (2) are repeated until the user is satisfied with the system response. We believe that this model is general enough to encapsulate most of the task-oriented spoken dialogue systems in use today.

Figure 3.1 shows how the intention recognition result is updated in the course of a dialogue in a weather information system. In the example, “tomorrow” was misrecognized as “today” by the speech recognizer (U1), causing the system to have an incorrect value for the date slot (F2). The misunderstood item was later corrected by the user (U3), who noticed the error in the intention recognition result because of the system’s incorrect confirmation request (S3). Through the interactive process with the user, the intention recognition results get closer to the correct user intention (F1-F4).

Since the understanding of user utterances and the production of system utterances are both affected by the content of the intention recognition result, correctly

In the realm of discourse understanding, which works on top of speech understanding, there is also a growing body of research. Filisko (2002) proposed a context resolution server that specializes in reference resolution and ambiguity resolution in speech understanding results. Miyazaki et al. (2005) and Higashinaka et al. (2003b) both employ a multi-world model, in which multiple discourse understanding results are maintained as an ordered list to enable discourse-level ambiguity to be retained and resolved by succeeding utterances. The difference between the two models is that the former uses hand-crafted rules, and the latter uses statistical information derived from dialogue corpora for the ranking.

Although much work has been done in speech understanding and discourse understanding, it is still acknowledged that speech recognition errors are inevitable, and that speech recognition errors often cause a system to misunderstand the user's intention. In addition, ambiguities in natural language also make it difficult for a systems to correctly understand the user's true intention. Therefore, the dialogue manager has to confront the problem of handling unreliable and ambiguous intention recognition results.

Since the slot values are unreliable, one safe and simple approach for dialogue management is to confirm every item in the slots until all items in them are acknowledged by the user. However, too many confirmations are likely to make dialogues tedious, and when the system reduces the number of confirmations, the system is likely to deliver undesired information based on incorrectly recognized items. The system needs to find a balance between too many and too few confirmations. For this purpose, the system has to be able to detect exactly what item needs to be confirmed.

In speech recognition research, a technique called *confidence scoring* has been increasingly used to detect errors in speech recognition results. For example, it has been used for utterance verification (Rahim et al., 1997). It also helps transcribers find erroneous words/phrases in the recognized sentences, which speeds up the transcription process (Endo et al., 2002). Recently, this technique has also been applied to detect errors in intention recognition results and has proved useful for dialogue management.

Komatani and Kawahara (2000) and Dohsaka et al. (2003) used the confidence of the intention recognition results to adaptively change dialogue strategies, which

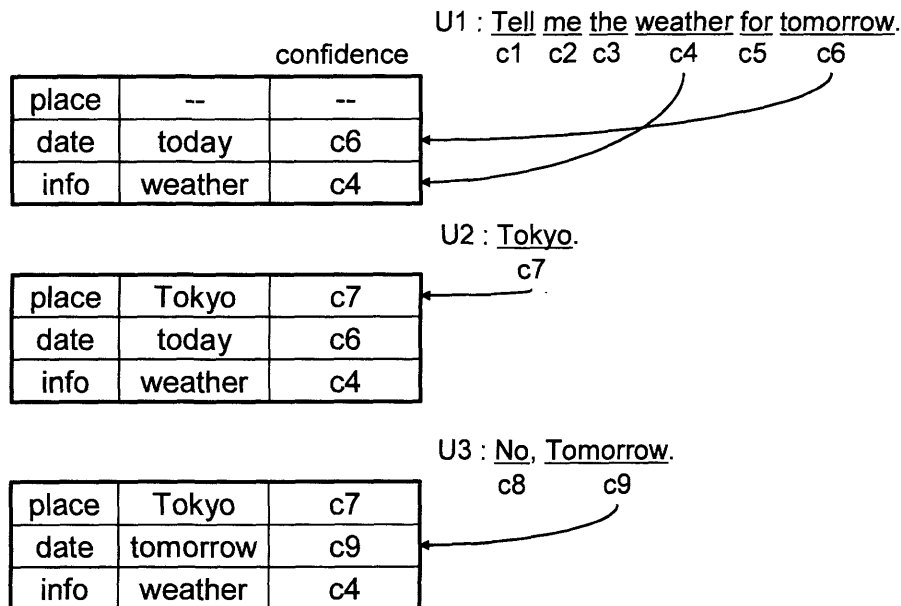


Figure 3.2: Conventional methods. Word confidences $c_1 \dots c_9$ are associated with the slots.

enables the system to confirm only the necessary items and avoid unnecessary confirmations. In this way, the task completion time was considerably reduced. The confidence of intention recognition results has also been used in order to better characterize the status of a dialogue state (called a state space) for the automatic learning of optimal dialogue management policies with reinforcement learning techniques (Singh et al., 2002).

Since estimating the reliability of the intention recognition results allows the dialogue manager to have a wider variety of choices as to how to respond to the user and enables the system to characterize the current state of a dialogue more accurately, there is a strong need for the confidence scoring of intention recognition results.

3.4 Conventional Methods

Slots are typically filled with words in speech recognition hypotheses or with concepts in speech understanding results, and the acoustic, linguistic, or sometimes the semantic reliability of the words or concepts has been used for the confidence

of the slots. Figure 3.2 shows an example of the confidence scoring of the slots in Fig. 3.1, illustrating how the confidence of words $c_1 \dots c_9$ can be associated with the slots.

The simplest way to calculate a confidence score is to use the score that the speech recognizer outputs for words, e.g., the total acoustic and language model score or the word posterior probability (Wessel et al., 2001). When a slot is filled by a concept, the total or mean confidence of the words that form that concept is normally utilized. For example, an utterance “to Tokyo” might form a concept “arrival-city=Tokyo.” In this case, the confidence for this concept is calculated taking the summation or the mean of the confidence of words “to” and “Tokyo.”

To obtain more accurate scores especially for concepts, probabilistic modeling of a sequence of concepts has been proposed. Hacıoglu and Ward (2002), using concept N-grams, proposed constructing a concept graph on top of a word graph and calculating confidence of concepts in a fashion similar to calculating the word posterior probability. Lin and Wang (2001) propose a concept-based probabilistic verification model, which also exploits concept N-grams.

There are also approaches that use *confidence models* for confidence scoring. A confidence model is a kind of a classifier that scores or classifies words/concepts based on training data. Although data collection, feature extraction, and labeling procedures have to be performed before the training, the confidence model approach has proved particularly useful when various types of features, such as numeric values and symbolic values, have to be integrated for the scoring.

Hazen et al. (2002) integrate two levels of features in speech recognition hypotheses to train confidence models for words: word-level features that focus only on the reliability of the acoustic samples corresponding to the word, and utterance-level features that concern the appropriateness of the whole utterance in which the word is found. This integration is based on the assumption that if the whole utterance is unreliable, the word contained in that utterance is also likely to be incorrect.

In order to achieve more accurate scoring depending on the context, Pradhan and Ward (2002) proposed creating confidence models for concepts (or semantic frames) using previous system prompts in addition to the features obtained from the speech recognition results. They adopted this particular approach because

they focus on large vocabulary-based system-guided dialogues within the DARPA Communicator project (Pellom et al., 2000), in which user utterances are heavily influenced by previous system prompts.

There is also an approach that utilizes *pragmatic analyses* to score concepts uttered by the user (Ammicht et al., 2001). This makes several basic assumptions about what concepts the user should utter after a system response and uses the assumptions as rules to score the incoming concepts and rescore already recognized concepts. For example, when an already recognized concept seems to have been implicitly confirmed, the confidence of that concept is augmented.

Hirschberg et al. (2004) introduce a number of prosodic features, such as F0, the length of a pause preceding the turn, and the speaking rate, to detect misrecognized user turns in spoken dialogue corpora. Since users tend to change the way they speak when faced with inappropriate system utterances that originate from previous misrecognized utterances, they use the prosodic features of subsequent utterances to detect possible errors in previous user utterances. The problem they are dealing with is different from ours in that they do not evaluate user utterances immediately after speech recognition.

3.5 Proposed Method

Previous methods have been quite successful in providing reasonably good estimates of correct/incorrect for intention recognition results. However, we would like to pose a question: *Is it really appropriate to use the confidence of words/concepts for the confidence of intention recognition results?*

We argue that it may not be appropriate because the confidence of words/concepts is calculated separately from the context; that is, the intention recognition result is the system's understanding result of a discourse, not the result of understanding an independent utterance. There may be some cases where hypothesized words/concepts are not likely to fill the slots, as when the slot values contradict what has been said in a prior part of a dialogue. Ignoring the fact that intention recognition results represent the discourse may lead to inaccurate confidence scoring. Therefore, we propose incorporating discourse information into the training of confidence models.

To enable discourse information to be used in confidence model training, we have to find features to represent a slot value from the discourse point of view. We hypothesize that there is a principle that a valid discourse should satisfy and that any indication of violation of or conformity to this principle can be used to score a slot value in a discourse. We employ, as such a principle, Grice’s maxims of cooperativeness (Grice, 1975). Grice’s maxims are described as norms that should be followed in a collaborative conversation. Grice proposed four maxims, namely, maxims of Quantity, Quality, Relation, and Manner. Figure 3.3 shows the description of the maxims from (Grice, 1975). We created twelve discourse features, each one of them indicating possible violation of or conformity to the maxims. The derivation of the features are described in Section 3.5.1 in detail. Although there may be other principles or models for discourse, such as discourse plans (Allen et al., 2001; Rich et al., 2001), such high-level discourse principles may not be necessary when considering the speech recognition errors. Therefore, we only consider Grice’s maxims in this work.

Along with the discourse features, we also use acoustic and language model features of the words/concepts filling the slots because they have been proven useful in the literature. Having defined the features, we take the following steps in confidence model training: We (1) collect slot value samples through dialogue experiments with human users, (2) extract the discourse features and the acoustic and language model features for slot values and annotate them as correct/incorrect, and (3) train confidence models for slot values using the collected data.

As a confidence model training technique, we adopt one of the existing techniques (Hazen et al., 2002). For evaluation, we compare the performance of the obtained confidence models with that of the baseline models. The baseline here means models that only use acoustic and language models for the confidence model training. We also compare our models, for reference, with a method that only uses the posterior probability of words that the speech recognizer outputs, since posterior probability is widely used in the community for its simplicity.

Although the use of previous system prompts can be seen as incorporating discourse information into confidence scoring (Pradhan and Ward, 2002), our approach is different in that we deal with the discourse understanding result, not the

result of single utterance understanding, and in that our discourse features are represented by numeric values, not symbolic conditions for classifying user utterances. In addition, compared to the tasks in the Communicator project (Pellom et al., 2000), we focus on relatively smaller tasks with less system initiative and handle restricted utterances mainly consisting of user requests. Therefore, the use of previous system prompts is not expected to greatly improve confidence scoring in our case. However, if we had to handle a wider variety of utterances, our approach could be used together with the work of Pradhan and Ward (2002).

We also see (Ammicht et al., 2001) as an attempt to incorporate discourse information into the confidence scoring. However, they are also not particularly focusing on discourse understanding results but concepts in single utterances and their approach uses heuristic rules for the scoring, directly relating certain discourse phenomena with fixed effects, whereas our approach aims at finding useful features to express discourse information so that the features can be related to confidence scores by confidence model training based on training data.

3.5.1 Discourse Features

Here, we describe how we derive our discourse features. In all, we came up with 12 discourse features: one, seven, and four features in relation to the maxim of quantity, quality, and manner, respectively. Since we consider that the maxim of relation is automatically abided by in task-oriented dialogues—for example, in the weather information domain, the user and the system would not talk about booking flights or train tickets—we only focused on the remaining three maxims.

The discourse features are conceived following our assumption about the interaction between the user and the system; namely, the user sends words/concepts or sometimes commands to the system in order to change the slots, and the system responds to the user using the words/concepts stored in the slots. We argue that as long as the system follows this assumption, our features can be safely extracted. We also assume that the user’s true intention does not change during the dialogue. In what follows, we describe in detail each feature related to the maxims.

1. Maxim of Quantity:

- (a) Make your contribution as informative as is required (for the current purposes of the exchange).
- (b) Do not make your contribution more informative than is required.

2. Maxim of Quality:

- (a) Do not say what you believe to be false.
- (b) Do not say that for which you lack adequate evidence.

3. Maxim of Relation:

- (a) Be relevant.

4. Maxim of Manner:

- (a) Avoid obscurity of expression.
- (b) Avoid ambiguity.
- (c) Be brief (avoid unnecessary prolixity).
- (d) Be orderly.

Figure 3.3: Grice's maxims of cooperativeness (Grice, 1975).

3.5.1.1 Features Related to the Maxim of Quantity

The maxim of quantity suggests that one has to make one's contribution to the conversation as informative as necessary. The mention of a slot value that is the same as the one appearing in the previous system's confirmation request may not, therefore, be desirable. For example, the exchange

System : "Are you interested in the weather in Tokyo?"

User : "The weather in Tokyo"

corresponds to a case violating the maxim of quantity. Although the sequence may be a re-confirmation of the system's confirmation request, in terms of the maxim of quantity it is better for the user to provide more information about his/her intentions. Taking this into account, we conceived the following discourse feature D1:

- (D1) **Same keyword pair count:** Throughout the dialogue, count the number of times the system confirms the current slot value and the user mentions the same value in the next utterance. We use this count as the feature. A large value of this feature would mean that there have been a lot of un-informative interactions about a particular slot value, suggesting that the value may be wrong.

3.5.1.2 Features Related to the Maxim of Quality

The maxim of quality states that one should not say what one believes to be false. This can be interpreted as: the content of all the user utterances should be consistent. Therefore, any contradiction or inappropriateness among the system's recognized user intentions can be used as an indicator of a violation of the maxim of quality.

To describe how the intention recognition results (slot values) are recognized in the course of a single dialogue, we first introduce the idea of the *slot value sequence*, which represents the transition of values of a particular slot. For example, $\{null \rightarrow null \rightarrow Tokyo \rightarrow Tokyo\}$ is a slot value sequence for the place slot in F4 in Fig. 3.1. Here, the last value Tokyo is the current value whose confidence we aim to estimate, and *null* means that the slot does not have a value. Ideally,

if the user is following the maxim, the slot value sequence should consist of just one single value. By characterizing the slot value sequence from different points of views, we conceived the following seven discourse features (D2 through D8):

- (D2) **Slot purity:** In the slot value sequence, count the number of times the current value is found and divide that count by the number of non-null values in the sequence. We use this ratio as the feature. For example, when the value of the place slot changes $\{Tokyo \rightarrow Osaka \rightarrow Kyoto \rightarrow Osaka\}$, then the current value Osaka is found in two of the four values, making the slot purity $1/2$. This feature encodes the user’s consistency about a certain value. Therefore, a large value of this feature may suggest that the slot value is correct.

- (D3) **Top slot purity:** In the slot value sequence, for all the values that appear, count the number of times each value appears, find the highest count, and divide that count by the number of non-null values in the sequence. We use this ratio as the feature. When the value for the place slot changes $\{Tokyo \rightarrow Osaka \rightarrow Kyoto \rightarrow Osaka\}$, Tokyo, Osaka, and Kyoto are assigned the values of $1/4$, $1/2$ ($2/4$) and $1/4$, respectively. The maximum value is Osaka’s $1/2$; therefore, the top slot purity is $1/2$. This feature represents the slot purity of the dominating slot value in the sequence if there is any. If the top slot purity of a slot value is greater than its slot purity, it may be likely that the slot value is wrong.

- (D4) **Slot variety:** The number of different values that appear in the slot value sequence. For $\{Tokyo \rightarrow Osaka \rightarrow Kyoto \rightarrow Osaka\}$, there are three values Tokyo, Osaka, and Kyoto; therefore, the slot variety is 3. This feature encodes the user’s inconsistency, and a large value of this feature may suggest that the slot value is wrong.

- (D5) **Deny count:** The number of times the current value has been deleted. For example, consider the sequence $\{Tokyo \rightarrow null \rightarrow Kyoto \rightarrow Tokyo\}$. The current value Tokyo is once denied (set to null) by the user (later set to Kyoto). Therefore, the value is 1. If a certain value is correct, a cooperative user would not delete that value. A large value of this feature may suggest that the slot value is wrong.

- (D6) **Overwrite count:** The number of times the current value has been overwritten by other values. For example, consider the sequence $\{Tokyo \rightarrow Osaka \rightarrow Kyoto \rightarrow Tokyo\}$. The current value Tokyo is overwritten once by Osaka. Therefore, the value is 1. If a certain value is correct, a cooperative user would not overwrite/replace that value. Therefore, a large value of this feature may suggest that the slot value is wrong.
- (D7) **Continue count:** Starting backwards from the current value, count the number of times the current value is found in the slot value sequence *successively*. We use this count as the feature. For example, consider the sequence $\{null \rightarrow Tokyo \rightarrow Tokyo \rightarrow Tokyo\}$. Before the current value Tokyo, there are two Tokyo values. Therefore, the value is 2. Since the slot values have to be successively the same to yield a large value, this feature encodes the user's possible strong consistency about a certain value. Therefore, a large value of this feature may strongly suggest that the slot value is correct.
- (D8) **Different value count:** Starting backwards from the current value, count the number of times the current value is *not* found in the slot value sequence *successively*. We use this count as the feature. For example, consider the sequence $\{Tokyo \rightarrow Osaka \rightarrow Kyoto \rightarrow Tokyo\}$. There are two non-Tokyo values before the current value Tokyo. Therefore, the value is 2. This feature functions exactly opposite to the continue count (D7), as it encodes the user's possible strong inconsistency. A large value of this feature may suggest that the slot value is wrong.

3.5.1.3 Features Related to the Maxim of Manner

The maxim of manner states that one should avoid unnecessary prolixity as well as ambiguity. Therefore, if there are a large number of same/different words/concepts corresponding to a slot value appearing in user or system utterances, it may be an indication that the slot value is wrong. Note that these features focus on the user's and system's raw utterances or dialogue acts with concepts, not the slot value sequence and that these features encode what the system has observed within a dialogue rather than what the system has understood. Taking this into account, we enumerated the following four features (D9

through D12):

- (D9) **Same keyword count in user utterances:** The number of times a concept corresponding to the current value appears in the previous user utterances. For example, when the current value is Tokyo, we count the number of times the word “Tokyo” or the concept “place=Tokyo” appears in the user utterance history.
- (D10) **Different keyword count in user utterances:** The number of times concepts not corresponding to the current value appear in the previous user utterances. For example, when the current value is Tokyo, we count the number of times non-Tokyo place names appear in the user utterance history.
- (D11) **Same keyword count in system utterances:** The number of times a concept corresponding to the current value appears in the previous system utterances. For example, when the current value is Tokyo, we count the number of times the word “Tokyo” or the concept “place=Tokyo” appears in the system utterance history.
- (D12) **Different keyword count in system utterances:** The number of times concepts not corresponding to the current value appear in the previous system utterances. For example, when the current value is Tokyo, we count the number of times non-Tokyo place names appear in the system utterance history.

3.6 Experiment

3.6.1 System

We prepared a telephone-based spoken dialogue system in the weather information service domain. The system provides Japan-wide weather information. Users specify a prefecture name or a city name, a date, and an information type (weather, temperature, and precipitation) to obtain the desired information.

The speech recognition engine is Julius (Lee et al., 2001) with its attached acoustic model, and the speech synthesis engine is FinalFluet (Takano et al.,

2001). The system has a vocabulary of 1,652 words. The language model is a trigram trained from transcriptions obtained from our previous dialogue data collection in the same domain (Higashinaka et al., 2003a).

The system uses the 1-best speech recognition hypothesis for language understanding. We realized our understanding grammar as a weighted finite state transducer (WFST) in a manner similar to (Potamianos and Kuo, 2000). We first prepared a set of transcribed utterances labeled with dialogue acts and concepts. An utterance is assumed to have a single dialogue act with zero or more concepts. Then, we converted the utterances into a WFST. An utterance corresponds to a path, which has one dialogue act and related concepts on its path. The whole grammar is a union of such paths. The resulting WFST maps a sequence of words into a scored list of dialogue acts augmented with concepts. For example, the user utterance “Tell me the weather for tomorrow” would derive “refer-info-date” as a dialogue act with “info=weather” and “date=tomorrow” as its concepts. Compared to keyword spotting, this can be seen as imposing lexical constraints using surrounding words. The scoring for the WFST was tuned to derive as few dialogue acts and as many concepts as possible from an utterance. Since an utterance may contain several dialogue acts, we made an epsilon transition from the end of the path to the start, enabling the recursion of the dialogue acts. There are 47 dialogue acts in our grammar.

The system maintains three slots for the intention recognition result; namely, the place slot, the date slot, and the information type slot. The intention recognition results are updated by the discourse understanding rules, which update the intention recognition results using the incoming dialogue acts and concepts. The system also holds a *grounding flag* for each slot to indicate if the value of a slot has been acknowledged by the user. For example, when the system confirms by asking “Are you interested in the weather in Tokyo?” and the user says “Yes,” then the grounding flags for the information type slot and the place slot are set to *true*. We call the slots that have been acknowledged by the user the *grounded slots*.

For discourse understanding, we prepared 47 discourse understanding rules. Each rule is responsible for the processing of a particular dialogue act and its related concepts. For example, in the case of the dialogue act “refer-info-date” with

concepts “info=weather” and “date=tomorrow,” a rule corresponding to “refer-info-date” is invoked, which allocates the concepts to the appropriate slots. Currently, our crude rules put every concept they encounter into the associated slots without consulting the dialogue history. Since only a single value is permitted to fill a slot, previous slot-fillers are always overwritten by the new ones.

There are other rules that deal with dialogue acts that do not have associated concepts, such as acknowledgments and denials. In these cases, corresponding rules are fired to set grounding flags to particular slots or erase particular values from them. Currently, all slots which are associated with the concepts included in the previous system confirmation request are grounded or erased by the succeeding acknowledgment or denial by the user. The system also has several rules that erase the values of particular slots. For example, the user utterance “the place is wrong” yields a dialogue act “erase-place,” which erases the value of the place slot. Our grammar allows two slots to be deleted at a time. Users cannot reject some values while simultaneously accepting others. The rules also handle closing remarks such as “good-bye” and the restart commands that initialize all values of the slots.

For response generation, the dialogue manager first determines whether or not the system should utter a back-channel (e.g., “uh-huh”). If the user’s previous dialogue act is not of a type explicitly requesting a response from the system, and no more than one slot is filled, the system assumes that the user has not completed his/her request and utters a back-channel. If the system decides not to utter a back-channel, it then checks how many slots have been filled and grounded.

If the system finds slots that are filled but ungrounded, the system confirms these slots in one utterance. For example, when slots for place and information type have been filled with “Tokyo” and “weather” and have not been grounded, the system would utter “Are you interested in the weather in Tokyo?” Similarly, if there is only one slot that is filled and ungrounded, it only confirms that one value. An example of confirmation requests would be “Did you say Tokyo?” The system does not use an implicit confirmation strategy.

If all the slots have been filled and grounded, the system sends a query to the weather database, retrieves the weather information, formulates it into a sentence, and utters it to the user. The current version of our system erases and

resets all the slots upon delivering the weather information. If none of the above conditions match, which is the case when the user explicitly requests a response with no slots filled or two or fewer slots grounded, the system asks the user to fill the missing slots one at a time in the order of place, information type, and date. An example of the system's utterances is "Tell me the area you are interested in." All the responses are generated by templates. There are 17 templates in all, including the ones for greetings and back-channels. The templates have forms such as "Did you say [place=X]?" and "Are you interested in the [info=X] in [place=Y]?" where X and Y are taken from slot values.

3.6.2 Data Collection

Eighteen subjects used the system over the telephone over a period of six days; three subjects per day. Each subject was given a task sheet listing the information to be requested. Each task demanded the user to ask about just one combination of a place, an information type, and a date. Therefore, if the user succeeds in the task, each dialogue in our collected data should contain one delivery of weather information from the system at the end of the dialogue. The subjects were instructed to complete the tasks one-by-one. Each subject engaged in 16 dialogues, for a total of 288 dialogues collected. Dialogues that took more than three minutes were aborted and regarded as failures. We separated the data into six groupings corresponding to the data for the six experiment dates.

The overall word error rate (WER) was 40.16%. The task completion rate was 95.83% (276/288). Figure 3.4 illustrates the number of turns required to complete the tasks in each grouping. Three is the minimum number of turns necessary to complete the tasks (a user's request, an acknowledgment of the system's confirmation, and a closing remark), and five out of six groupings had three as their mode value. Overall, the median number of turns is four, and the mode value is three.

The WER may seem high, but considering the nature of human-computer dialogues in which bad speech recognition prolongs dialogues, it is reasonable. We recorded the system and user utterances and the intention recognition results after each user utterance. All user utterances were transcribed.

We briefly ran an analysis of the slot samples we collected and found that most

of the errors were caused by speech recognition errors. This is because neither the speech understanding component nor discourse understanding component could override slot choices provided by the speech recognizer’s 1-best hypothesis. In addition, user utterances contained very little ambiguity needing resolution through language understanding and discourse understanding. For example, the system did not have to choose between place names for arrivals and departures as in the ATIS domain. Other than speech recognition errors, we noticed a small number of cases (15 cases) where our grammar could not output any parse for the input.

3.6.3 Data Screening

Before training confidence models, we screened the data. Since we do not deal with slots that do not have values, we discarded the data for such slots. Then, we removed the data for slots that had a single value in the slot value sequence. The data removed here are of two types: (1) data for slots that had just been filled and (2) data for slots having the same value consecutively all along the dialogue.

The data corresponding to type (1) were removed because we consider that there is little discourse information available for these slot values. The data corresponding to type (2) were removed because we consider it difficult to differentiate (a) the cases in which values do not change because of repeated misrecognitions from (b) those in which the recognizer keeps recognizing the correct values. This is because during the data collection, users frequently repeated the same keywords/phrases for emphases and implicit confirmations. For such data, we recommend using non-discourse features as in conventional methods.

In addition, we did not use the data of *grounded* slots, since it is natural to consider that slots that have been grounded are basically correct.

There were 4812 slot value samples in all, and after screening, 777 samples remained (362 positive samples and 415 negative samples).

Table 3.1 shows the breakdown of the slot value samples for each grouping, where *null*, *single*, *grounded*, and *single & grounded* denote the number of vacant slots, slots having a single value in the slot value sequence, the grounded slots, slots that have a single value in the slot value sequence and are grounded at the same time, respectively. Here, *error* indicates that the samples were not

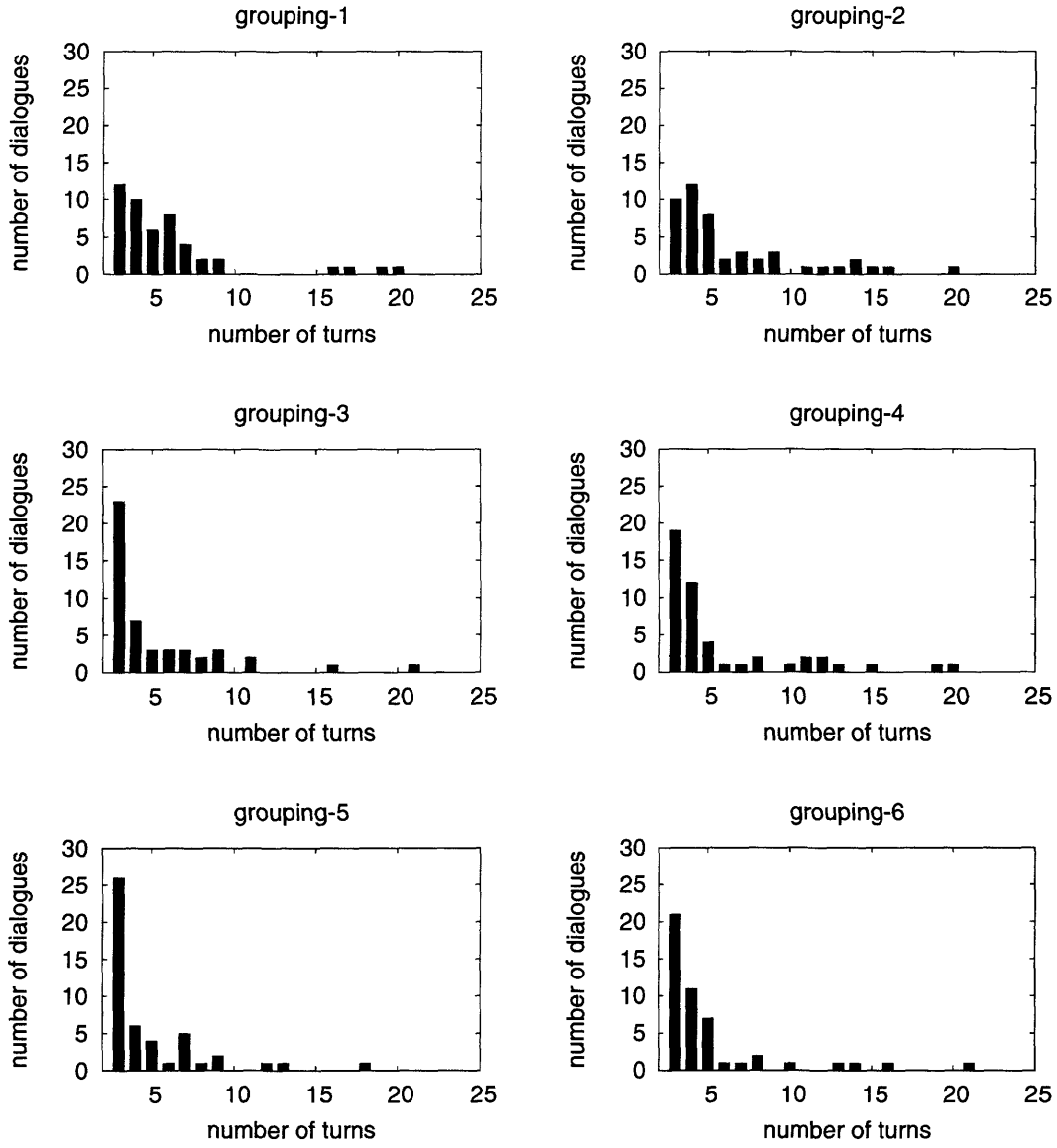


Figure 3.4: The number of turns required to complete the tasks in each grouping.

Table 3.1: Breakdown of the slot value samples for each grouping.

	<i>slots</i>	<i>null</i>	<i>single</i>	<i>grounded</i>	<i>single & grounded</i>	<i>error</i>	<i>selected as training sample</i>
grouping-1	864	267	278	41	131	2	145
grouping-2	927	215	327	33	107	0	245
grouping-3	759	246	281	28	133	0	71
grouping-4	831	262	247	39	158	0	125
grouping-5	696	236	213	29	124	0	94
grouping-6	735	245	248	32	113	0	97
Total	4,812	1,471	1,594	202	766	2	777

used because acoustic and language model features could not be retrieved for them because of defects in the recorded speech files. The numbers are mutually exclusive in the table.

3.6.4 Feature Extraction and Labeling

We extracted the acoustic and language model features and discourse features for all 777 slot value samples. As the acoustic and language model features, we used the same features that Hazen et al. used in (Hazen et al., 2002) (called word-level features) with some modifications. Modifications had to be made because of the differences in speech recognizers. In addition, since the utterance score in word-level features (W14) is derived from various features of whole utterances (utterance-level features), we combined the word-level features and the utterance-level features to create a single feature vector instead of using the utterance score, making the total number of our acoustic and language model features 27. We used 10-best speech recognition results for extracting the features.

Tables 3.2 and 3.3 show the acoustic and language model features we used with marks showing where the modifications were made. The label *<not available>*

¹Frame purity is conceptually the same as the N-best purity, with the focus on phonemes instead of words.

Table 3.2: List of word-level features. Labels *⟨not available⟩*, *⟨not used⟩*, and *⟨new⟩* indicate the modifications we made to the features used in (Hazen et al., 2002).

(W1)	Mean acoustic score
(W2)	Mean acoustic likelihood score <i>⟨not available⟩</i>
(W3)	Minimum acoustic score
(W4)	Maximum acoustic score <i>⟨new⟩</i>
(W5)	Acoustic score standard deviation
(W6)	Mean difference from maximum score
(W7)	Minimum difference from maximum score <i>⟨new⟩</i>
(W8)	Maximum difference from maximum score <i>⟨new⟩</i>
(W9)	Standard deviation of difference from maximum score
(W10)	Mean catch-all score <i>⟨not available⟩</i>
(W11)	Number of acoustic observations
(W12)	N-best purity
(W13)	Number of N-best <i>⟨not used⟩</i>
(W14)	Utterance score <i>⟨not used⟩</i> (utterance level features were used instead)
(W15)	Mean frame purity ¹ <i>⟨new⟩</i>
(W16)	Minimum frame purity <i>⟨new⟩</i>
(W17)	Maximum frame purity <i>⟨new⟩</i>

Table 3.3: List of utterance-level features. Labels *⟨not available⟩*, *⟨not used⟩*, and *⟨new⟩* indicate the modifications we made to the features used in (Hazen et al., 2002).

(U1)	Top-choice total score
(U2)	Top-choice average score
(U3)	Top-choice total N-gram score
(U4)	Top-choice average N-gram score
(U5)	Top-choice total acoustic score
(U6)	Top-choice average acoustic score
(U7)	Total score drop
(U8)	Acoustic score drop
(U9)	Lexical score drop
(U10)	Top-choice average N-best purity
(U11)	Top-choice high N-best purity
(U12)	Average N-best purity
(U13)	High N-best purity
(U14)	Number of N-best hypotheses <i>⟨not used⟩</i>
(U15)	Top-choice number of words

means that the feature was used in (Hazen et al., 2002), but not available for our speech recognizers, whereas *<new>* indicates that the feature was available, allowing us to incorporate it to our list of features. The label *<not used>* indicates that the feature was available, but not used as one of our features. The *<not used>* is only given to the number of N-best (W13 and U14) that always had a fixed value of ten in our setting. For a detailed description of the features, see (Hazen et al., 2002).

As the discourse features, we used all the discourse features except D6. The feature D6 was excluded by a process of backward-elimination using the F-measure as a criterion. We used the same experimental procedure as described in Section 3.6.7 to find features that are not contributing to the classification performance. (Refer to Section 3.6.6 for the derivation of the F-measure.) The exclusion of D6 may be attributable to the inter-dependency among the features. High correlation among features is likely to hinder the training of confidence models, making it difficult to allocate appropriate weights to them.

We first hand-labeled the reference intention recognition results after each user utterance using the transcriptions, and then automatically labeled slot values as correct or incorrect. This process took several hours for our data.

3.6.5 Confidence Model Training

We trained six confidence models for intention recognition results, taking every five of the six groupings as training data and making the remaining grouping the test data for the evaluation. For comparison, we also created, in the same way, six confidence models that only use the acoustic and language model features for training. Hereafter, we call the models trained by acoustic and language model features the *conventional models*, and the models trained by the acoustic and language model features plus the discourse features the *proposed models*.

We adopted the confidence model training method from Hazen et al. (2002). The method produces probabilistic confidence scores as log-likelihood ratios of posterior probabilities, using a weighted linear combination of the confidence feature vectors. The multi-dimensional feature vector \vec{f} is reduced to the raw score r by a linear combination with a projection vector \vec{p} such that

$$r = \vec{p}^T \vec{f} \quad (3.1)$$

We trained the projection vector \vec{p} in the same manner as Hazen et al. (2002), i.e., by initializing \vec{p} using a Fisher linear discriminant analysis and then updating each element of \vec{p} using a hill-climbing algorithm (Powell, 1964) to minimize the classification errors in the training data.

Using r , probabilistic confidence score c is calculated as follows:

$$c = \log \left(\frac{p(r|correct)P(correct)}{p(r|incorrect)P(incorrect)} \right) - t, \quad (3.2)$$

where $P(correct)$ and $P(incorrect)$ are *a priori* probabilities of correct and incorrect samples in the training data, and $p(r|correct)$ and $p(r|incorrect)$ are posterior probabilities for r for correct and incorrect samples, which were modeled with Gaussian density functions in this experiment. The t is a decision threshold.

Although we employed the simple linear projection model, it may also be possible to use other classification techniques, such as non-linear support vector machines and multi-layered perceptrons. However, since this work is particularly focused on discourse features and their effect on confidence scoring, we leave investigating the use of different classifiers as future work.

3.6.6 Evaluation

Table 3.4 shows the F-measure (harmonic mean of the precision and recall) for the conventional and proposed models when each grouping was used as the test data. The result for the method that uses posterior probability of words corresponding to concepts filling the slots is also shown for reference.

The posterior probability was calculated on the N-best list in a similar manner to the N-best posterior probability (Wessel et al., 2001). We used 10 for N, and the scaling factor α was set to 0.03, which was found to be the best in our pilot test with 982 utterances. The utterances here were those randomly selected from the collected data. The decision threshold used for each grouping was determined to achieve minimum classification errors within the training data. Although we acknowledge that increasing N improves the calculation of the posterior probab-

Table 3.4: F-measure for the method that uses posterior probability, the conventional and proposed models.

Test data	F-measure		
	posterior prob.	conv.	prop.
grouping-1	0.711	0.809	0.803
grouping-2	0.706	0.670	0.821
grouping-3	0.704	0.645	0.689
grouping-4	0.617	0.726	0.800
grouping-5	0.747	0.753	0.833
grouping-6	0.590	0.600	0.709
total	0.685	0.710	0.791

ity (Wessel et al., 2001), we considered 10 to be reasonable considering the fact that the calculation has to be performed in real-time in spoken dialogue systems.

The precision, recall, and F-measure are calculated as follows:

$$\text{Precision} = \frac{\# \text{ of slots correctly classified as correct}}{\# \text{ of slots classified as correct}} \quad (3.3)$$

$$\text{Recall} = \frac{\# \text{ of slots correctly classified as correct}}{\# \text{ of correct slots}} \quad (3.4)$$

$$\text{F-measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (3.5)$$

It is clear from Table 3.4 that the proposed models perform better than the conventional models overall. The method that uses posterior probability is the worst performing method among the three.

Figure 3.5 shows the FAR-FRR curves for the three methods. The figure clearly illustrates their difference in classification performance.

The FAR and FRR are calculated as follows:

$$\text{FAR} = \frac{\# \text{ of slots incorrectly classified as correct}}{\# \text{ of incorrect slots}} \quad (3.6)$$

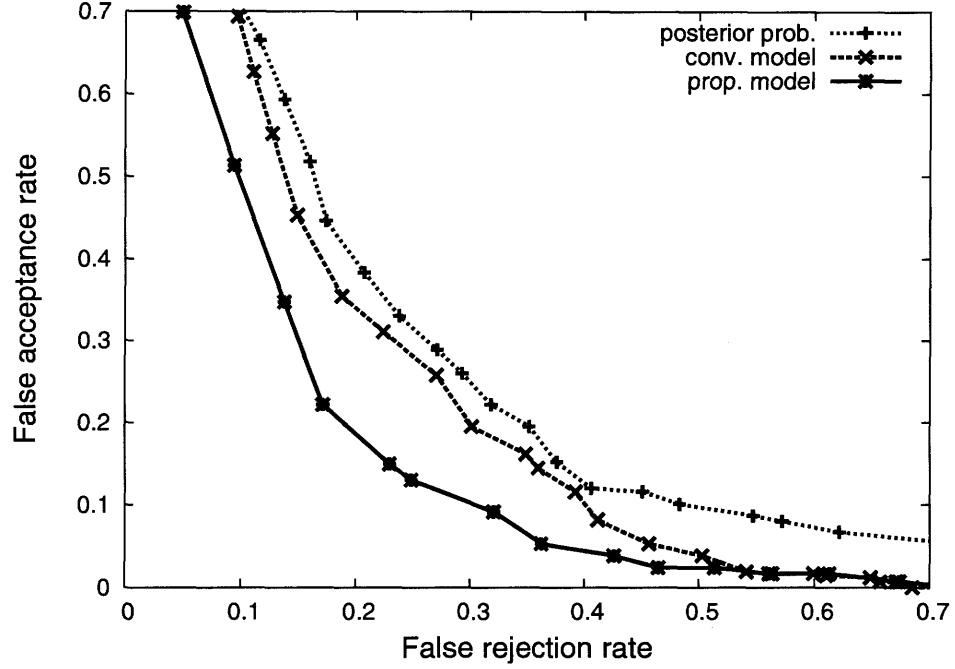


Figure 3.5: False acceptance rate (FAR)–false rejection rate (FRR) curves, and for the method that uses posterior probability, the conventional and proposed models.

$$\text{FRR} = \frac{\# \text{ of slots incorrectly classified as incorrect}}{\# \text{ of correct slots}} \quad (3.7)$$

The FAR is the rate at which the model incorrectly classifies negative samples as positives, and the FRR the rate at which the model incorrectly classifies positives as negatives.

Table 3.5 shows the matrix of counts of correct and incorrect items for the conventional and proposed models. Among all the samples, there were 83 that only the proposed models classified correctly, and 37 that only the conventional models classified correctly. From a statistical test [McNemar’s test (Gillick and Cox, 1989)], it was found that the two models have a statistically significant difference in terms of classification performance ($p = 3.99 \cdot 10^{-5}$), which verifies the effectiveness of the discourse features.

Table 3.5: Matrix of counts of correct and incorrect items for the conventional (conv.) and proposed (prop.) models.

	prop. correct	prop. incorrect
conv. correct	550	37
conv. incorrect	83	107

Table 3.6: F-measure for models each trained without D6 and one of the remaining discourse features.

Confidence models	F-measure	Drop in F-measure
prop. (All w/o D6)	0.791	0.000
w/o D6, D1	0.707	0.084
w/o D6, D2	0.756	0.035
w/o D6, D3	0.776	0.015
w/o D6, D4	0.750	0.041
w/o D6, D5	0.754	0.036
w/o D6, D7	0.751	0.040
w/o D6, D8	0.763	0.027
w/o D6, D9	0.758	0.032
w/o D6, D10	0.771	0.019
w/o D6, D11	0.765	0.025
w/o D6, D12	0.778	0.013

3.6.7 Impact of the Discourse Features

We investigated how each of the discourse features affects the classification results. Table 3.6 shows the F-measure for the models, each of which was trained without D6 and one of the remaining discourse features.

The row indexed by **prop. (All w/o D6)** represents the proposed models and the third column (Drop in F-measure) shows the drop of the F-measure from the proposed models. From the table, one can see that the same keyword pair count (D1) has a relatively large drop value, indicating that it may be more important than other features. On the other hand, the top slot purity (D3) and

3.6. Experiment

Table 3.7: Weights assigned to each of the discourse features in the six obtained confidence models. Averages and standard deviations of the weights are shown in the last column.

	model-1	model-2	model-3	model-4	model-5	model-6	avg. (sd.)
D1	-2.212	-2.525	-1.711	-1.879	-2.074	-1.659	-2.010 (0.3286)
D2	11.802	5.966	4.809	4.040	5.436	8.130	6.697 (2.8591)
D3	0.145	0.025	-2.304	-2.362	-1.415	-0.844	-1.126 (1.0971)
D4	0.139	-0.267	-0.258	-0.425	-0.265	-0.020	-0.183 (0.2043)
D5	-0.290	0.060	0.500	-0.449	0.500	0.540	0.143 (0.4377)
D7	0.589	0.120	0.055	-0.117	0.051	0.254	0.159 (0.2428)
D8	0.414	0.262	0.275	0.255	0.243	0.227	0.279 (0.0680)
D9	1.266	1.032	1.126	0.899	1.046	1.030	1.066 (0.1219)
D10	-0.156	-0.060	-0.111	-0.143	-0.479	-0.028	-0.163 (0.1622)
D11	-0.083	-0.113	-0.107	-0.028	-0.076	-0.462	-0.145 (0.1581)
D12	-0.018	0.103	-0.021	0.028	0.020	0.033	0.024 (0.0448)

the different keyword count in system utterances (D12) have small drop values, indicating their possible small contribution to the classification performance.

Our finding that the same key pair count (D1) is important may suggest that Grice’s maxim of quantity may be more useful than the others in terms of detecting errors in a dialogue. When we look at the weights of D1 in the confidence models, we find that the values are negative; that is, the larger the same keyword pair count, the lower the confidence. The small drop values of the top slot purity (D3) and the different keyword count in system utterance (D12) suggest that however many times different values occupy slot value sequences or system utterances, the confidence of slot values may not necessarily be affected.

Table 3.7 shows the weights assigned to each of the discourse features in the six obtained confidence models.² Notice that some of the features, such as D1 and D8, have very steady values for their weights compared to other features, suggesting that they play similar roles across all models. Large standard deviations in the weights of some of the features suggest that their effect on confidence scoring is

²Model-1 to model-6 are the models trained with all data except grouping-1 to grouping-6, respectively.

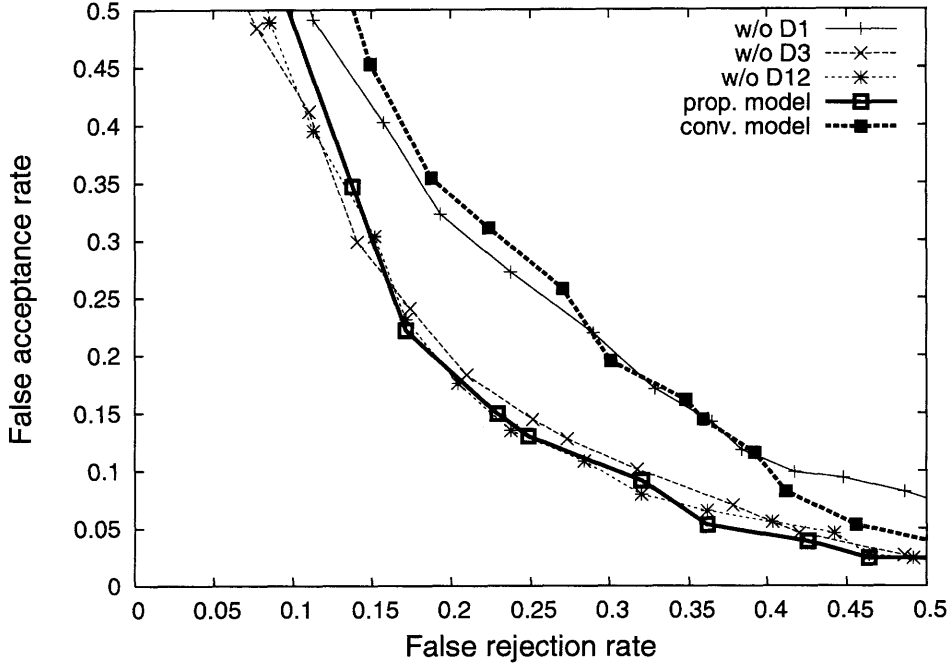


Figure 3.6: False acceptance rate (FAR)–false rejection rate (FRR) curves for the proposed and conventional models and for the models that do not use D1, D3, and D12 as discourse features.

likely to vary depending on the training data.

Weights that are larger than others do not necessarily reflect their importance of their associated features because of the ranges that each feature could take. For example, since the slot purity takes a value between 0 and 1, even with a very large weight, the overall effect of this feature will be limited.

Figure 3.6 shows the FAR-FRR curves for the models without D1, D3, and D12 along with those for the proposed models and the conventional models. It can be seen clearly that the models without D1 are close to the curve for the conventional models, and the models without D3 and D12 are almost on the curve for the proposed models.

3.6.8 Analysis of Successful Cases

We analyzed the successful 83 cases and found that there are mainly three patterns when our method succeeds:

1. **Slots that have a small slot purity and a large slot variety were successfully classified as incorrect.** We found 22 samples matching this pattern. This pattern suggests the rather obvious fact that if there are many different values in the slot value sequence, the slot value becomes dubious, indicating inconsistency in user utterances.
2. **Slots that have a large slot purity and a small slot variety were successfully classified as correct.** We found 28 samples matching this pattern. This pattern can be seen as the counterpart of the first pattern: the user's being consistent about a certain value adds confidence to that value.

By looking into the dialogue data, we noticed that dialogues in which this successful pattern was found contained the following interaction: (1) the user fills a slot relatively easily with X, (2) the slot is accidentally filled by some other value Y, and (3) the user fills the slot again with X. The conventional method was likely to find X incorrect, whereas the proposed method was likely to take X as correct. In a way, our method is using X's reliable past to boost X's confidence, overcoming the possible low acoustic and linguistic score of X.

3. **Slots that have a small slot purity, a large slot variety, and a large same keyword count were successfully classified as correct.** We found 22 samples matching this pattern. This can be seen as a special case of the first pattern, where samples the first pattern may classify as incorrect are rescued. Here, the same keyword count is acting as a booster of the confidence.

The pattern was found in dialogues where the following type of user utterances was frequently observed: "X's weather Y," where X and Y are both associated with the same slot and X is correct and Y is wrong (misrecognition). An example would be "Tokyo's weather Kyoto," which corresponds to two dialogue acts and concepts: "refer-place-info place=Tokyo info=weather" and "refer-place place=Kyoto."

Since our discourse understanding component handles dialogue acts sequentially, after this kind of utterance, the slot value can only be Y, which makes

X's slot purity very small. The same keyword count complements this small slot purity, suggesting X's potentially large slot purity.

There are 11 other samples that we could not categorize into patterns, partly because they were classified correctly by a combination of the patterns and partly because the weights for particular features were sometimes in an opposite polarity depending on the training data. Although we found two samples where the same keyword pair count was seemingly acting as a strong indicator of incorrectness, we did not categorize them as a pattern for the lack of samples. It is surprising that the importance of the same keyword pair count was not evident in the successful samples considering the drop in the F-measure when we did not use the feature. Investigating this issue remains as future work.

3.7 Summary and Future Work

We proposed a confidence scoring method for intention recognition results in spoken dialogue systems. Our method utilizes both discourse-related features and the acoustic and language model features of the speech recognition results to train confidence models for slot values. Experimental results show that the proposed method significantly improves the confidence scoring, indicating the effectiveness of the discourse features.

The results also indicate the usefulness of using Grice's maxims of cooperativeness to detect errors in spoken dialogue interactions. In addition, the analysis of the successful cases have revealed that the confidence model training process was capturing useful patterns to detect errors in the slot values, making the patterns possible decision rules.

As future work, firstly, we plan to perform experiments using different systems in order to verify our approach in different settings, including domains and dialogue strategies. Secondly, we would like to explore other discourse features since the discourse features presented in this work may not sufficiently characterize the slot values. For example, we are planning to incorporate features that represent relationships and constraints among the slots because slot values tend to have dependencies in certain situations. The use of other classification techniques for confidence model training, including non-linear classification methods,

should also be considered in this connection.

Thirdly, we would like to evaluate our method using workable dialogue systems. In this work, we performed an off-line evaluation, which is based on the assumption that a corpus collected with a certain system is similar to one collected by the improved version of the system. However, in the case of interactive systems, this is not necessarily the case. Therefore, to fully verify the proposed method, an on-line (interactive) evaluation is necessary.

Finally, since Grice's maxims have been found useful for the confidence scoring of intention recognition results, we would also like to investigate the possibility of using Grice's maxims for improving the understanding component in spoken dialogue systems.

Although future work remains, the results of our experiments suggest that our approach is promising. As a final remark, we point out that the discourse features we introduced can be easily obtained as long as the system follows our assumptions about spoken dialogue systems, which facilitates application of our method to other systems.

CHAPTER 4

Ranking Multiple Dialogue States by Corpus Statistics to Improve Discourse Understanding

4.1 Introduction

In spoken dialogue systems, since a speech recognizer usually outputs multiple speech recognition hypotheses and the syntactic and semantic analysis normally produces multiple parses, the discourse understanding component of a system typically receives multiple dialogue acts to update a dialogue state. Many systems use the best dialogue act candidate from the best parse of the best speech recognition hypothesis to update the dialogue state. However, this could lead to inaccuracy because the dialogue act is selected independently of the current dialogue state.

Recent work considers all the combinations of dialogue acts and the current dialogue state to create multiple dialogue states to be ranked with regard to the context so that the best dialogue state can be selected (Wutiwiwatchai and Furui, 2004). In addition, since it is sometimes difficult to decide on a single dialogue state due to ambiguity in user utterances, an approach to keep multiple dialogue states and resolve the ambiguity using succeeding user utterances has also been proposed (Nakano et al., 1999; Miyazaki et al., 2005). In this way, the correct dialogue state, which was not incidentally selected as the best interpretation in the previous turn, could survive until the next turn. Although this approach

has proven to improve discourse understanding accuracy (Miyazaki et al., 2005), the approach requires hand-crafted rules to accurately rank the dialogue states, which is costly and difficult to maintain and port to other domains. An automatic method would make the development of spoken dialogue systems scalable.

This chapter proposes automatically ranking multiple dialogue states using statistical information derived from the corpora of dialogues conducted between a system and users. We hypothesize that a dialogue state that has seen the most likely sequence of dialogue acts and updates is the most probable dialogue state and use the sequential probability of dialogue act types and dialogue state updates to rank dialogue states. Although there are approaches that aim to automatically rank several conflicting understanding candidates within a dialogue state (Bohus and Rudnicky, 2006; Bohus, 2007), our approach is different in that we rank dialogue states which represent the system’s interpretations of a whole dialogue.

In the next section, we describe the discourse understanding process using multiple dialogue states in spoken dialogue systems. In Section 4.3, we describe previous work, and in Section 4.4, we explain our approach in detail. In Section 4.5, we describe the experiments we performed to verify our approach in the train ticket reservation and weather information service domains. In the last section, we summarize the chapter and mention future work.

4.2 Discourse Understanding Using Multiple Dialogue States

Figure 4.1 shows how the architecture in Figure 1.2 changes when a spoken dialogue system uses multiple dialogue states for discourse understanding.

In this architecture, the multiple dialogue acts that can be derived for an utterance due to ambiguity in speech understanding are received by the discourse understanding component to update its multiple dialogue states. The multiple dialogue states are ranked according to their priority with regards to the context. In utterance generation, the dialogue manager can choose to use only the highest ranked dialogue state or take into account several highly ranked dialogue states to generate its responses, such as ‘A or B’ type confirmation requests when there are some competing dialogue states.

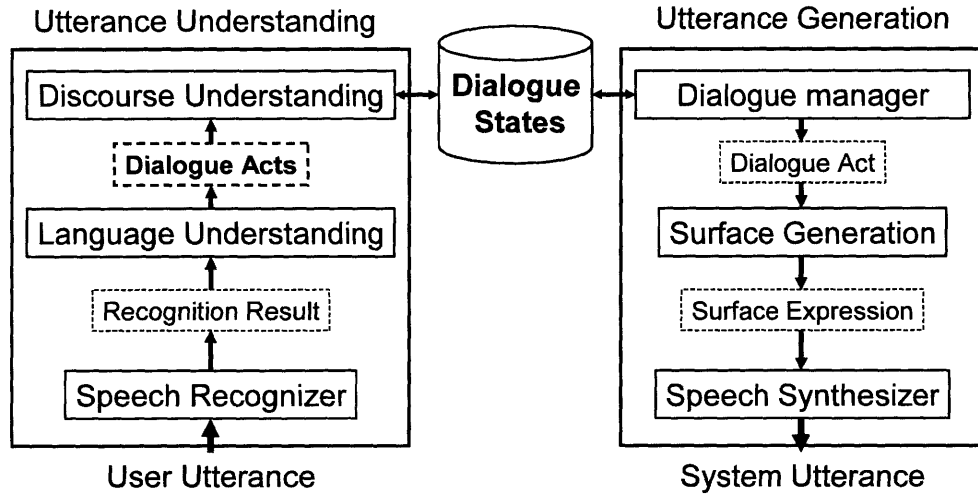


Figure 4.1: Architecture of a spoken dialogue system that utilizes multiple dialogue states for discourse understanding.

This chapter concerns a spoken dialogue system that uses multiple dialogue states for discourse understanding and focuses on the method of ranking the multiple dialogue states. Here, the objective of discourse understanding is to obtain the best ranking of the dialogue states, not to output a single dialogue state. Compared to a system that uses only a single dialogue state, holding multiple dialogue states makes it possible for the system to resolve the ambiguity of previous user utterances with succeeding ones.

Figure 4.2 illustrates a piece of dialogue in the train ticket reservation domain in which the user says “From Tokyo” (U1) and “From” in the utterance is inaudible and not recognized by the system. This utterance creates two dialogue acts; namely (*refer-origin place=Tokyo*) (filling the origin slot with “Tokyo”) and (*refer-dest place=Tokyo*) (filling the destination slot with “Tokyo”). As a result, two dialogue states (DS1 and DS2) are created from DS0. In the figure, we only show the intention recognition result (frame) of a dialogue state for conciseness (See Figs. 4.8 and 4.10 for a more detailed illustration of a dialogue state).

Suppose that, after the system’s back-channel (S2), the user says “To Kyoto” (U2) which corresponds to a dialogue act (*refer-dest place=Tokyo*). This act creates two new dialogue states (DS3 and DS4) from DS1 and DS2, respectively. A system with a single dialogue state may choose DS2 as the best dialogue state

User and System Utterances

S1: May I help you? U1: From Tokyo S2: Uh-huh U2: To Kyoto S3: From Tokyo to Kyoto?

'From' was inaudible

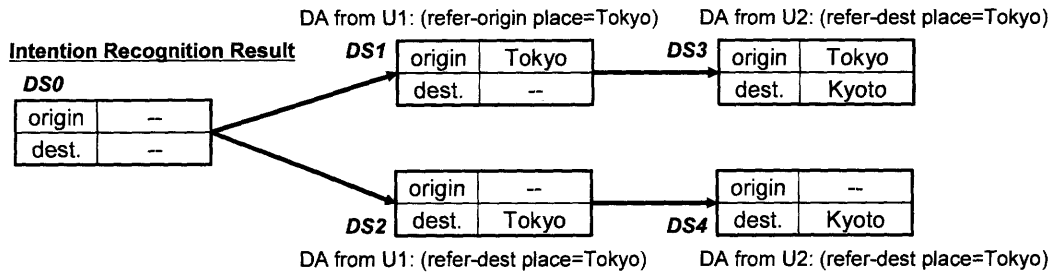


Figure 4.2: Example of discourse understanding using multiple dialogue states. (S, U, DA, and DS stand for a system utterance, a user utterance, a dialogue act, and a dialogue state, respectively.)

and discard DS1, making it impossible for the system to reach DS3 after the user's next utterance U2. By having both DS1 and DS2, the system can wait for the next user utterance to decide which dialogue state (DS1 or DS2) was actually true after U1. In this example, the system successfully chooses DS3 as the best dialogue state and makes an appropriate confirmation request (S3).

Figure 4.3 shows another example, in which the speech recognition result outputs two hypotheses ("To Tokyo" and "To Kyoto") for the user utterance "To Kyoto" (U1). These hypotheses create two dialogue acts that result in two dialogue states (DS1 and DS2) from DS0. Suppose that the system chooses DS1 as the best dialogue state and generates a confirmation request (S2), which is denied by the user (U2). By having multiple dialogue states, the system can reconsider that DS2 was actually correct and make an appropriate confirmation request "Did you say Kyoto?" (S3) on the basis of the correct dialogue state (DS4). Note that the value of the destination slot in DS4 (Kyoto) is preserved from DS3 because the exchange of utterances "To Tokyo" (S2) and "No" (U2) does not negate the fact that the destination is Kyoto.

In both examples, the ambiguity of a user utterance, originating from parsing and speech recognition, is preserved in the form of multiple dialogue states and then correctly disambiguated by correctly ranking the dialogue states using the

4.3. Previous Work

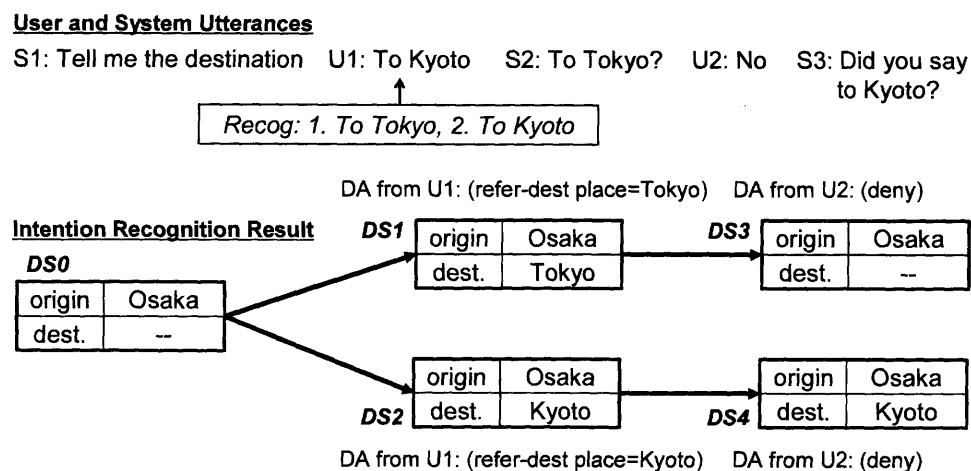


Figure 4.3: Example of discourse understanding using multiple dialogue states.

succeeding utterances, making it possible for the system to obtain the user's correct intention more accurately and efficiently. Although holding multiple dialogue states has such advantages, the problem is how to achieve appropriate ranking of the dialogue states.

4.3 Previous Work

In previous work, Nakano et al. (1999) proposed holding multiple dialogue states to deal with utterances that convey meaning over several speech intervals and the inability to determine the understanding result at each interval end. Multiple dialogue states are used to represent the ambiguity of whether the user has completed his/her utterance as well as the ambiguity arising from the existence of the multiple applicable interpretation rules. Dialogue states are scored based on which interpretation rules have been applied and the scoring is based on a system developer's intuition. Miyazaki et al. (2005) augmented Nakano et al.'s method to deal with n-best recognition hypotheses and reported improvement in discourse understanding accuracy. However, hand-crafted rules are also used in ranking dialogue states.

The reliance on hand-crafted rules to rank multiple dialogue states is problematic because, when the number of dialogue states becomes large, it becomes

difficult to design rules to obtain reasonable ranking results. Although only a small number of dialogue states are considered in the examples in Section 4.2, in a more realistic setting, the system has to consider the much larger number of dialogue states that can be created from N-best recognition hypotheses with N typically larger than one or two. Since the number of dialogue states grows exponentially as the dialogue progresses, the ranking by hand can easily become intractable. Another problem is that creating rules requires expertise in dialogue system development, which hinders rapid development of systems.

In a theoretical work on spoken dialogue processing, Williams and Young (2007) proposed having a probability distribution over dialogue states (user intentions) in order to model the understanding process of a spoken dialogue system as a partially observable Markov decision process (POMDP) and to learn the best policy for a dialogue manager by reinforcement learning using dialogue simulations. Here, the updating of the distribution is similar to ranking multiple dialogue states. Although they offer a good framework to estimate the distribution over dialogue states from various evidences and observations in a dialogue, their current use of contextual information is limited to the previous user and system dialogue act types; e.g., they do not consider N-grams of dialogue act types as in our approach. Since our aim is to find out what kind of discourse-level information is useful in ranking dialogue states, we believe our work is complementary to theirs.

There is also a body of work that aims to automatically estimate the confidence of slot values (Pradhan and Ward, 2002; Hazen et al., 2002; Higashinaka et al., 2006). However, these studies do not consider keeping multiple slot value candidates. One exception is Bohus (2007), who propose keeping multiple slot value candidates and ranking them using a confidence scoring function that takes into account various features of a dialogue, including speech recognition confidence scores for the words/concepts filling the slots, as well as discourse-level information such as whether the confirmation request regarding the slot value has been implicitly or explicitly confirmed by the user. However, the method does not focus on ranking multiple dialogue states and the features they use are specifically designed to deal with slot values. We emphasize that ranking slot values and ranking dialogue states are fundamentally different in that a dialogue

state represents the system's interpretation of what has happened so far in a dialogue, making the task of ranking multiple dialogue states similar to ranking possible worlds, which is in close connection with the multi-world model (Nagao, 1989) applied to processing a written discourse. This work aims to apply a similar model to understanding a spoken dialogue.

In the context of plan-based utterance understanding (Allen and Perrault, 1980; Carberry, 1990), when there is ambiguity in the understanding result of a user utterance, an interpretation best suited to the estimated plan should be selected. In addition, the system must choose the most plausible plans from multiple possible candidates. Although we do not adopt plan-based representation of dialogue states, this problem is close to what we are dealing with. However, no systematic ways to score the candidates for disambiguation have been proposed.

4.4 Approach

We propose automatically ranking multiple dialogue states using statistical information that can be derived from dialogue corpora. Since a dialogue state is a result of (a) a sequence of dialogue acts by the user and system and (b) the updates by them, we hypothesize that a dialogue state that has seen the most likely sequence of dialogue acts and updates is the most probable dialogue state. For this purpose, we derive two kinds of statistical information from a corpus: (1) the N-gram probability of a dialogue act type sequence and (2) the occurrence probability of a dialogue state update pattern. We use these two probabilities to assign scores to the dialogue states for ranking.

Figure 4.4 shows an example of a dialogue corpus that we need in order to extract the statistical information. The corpus contains speech recognition results for each user utterance, dialogue acts for each user and system utterances, and the transition of dialogue states. In the example, the sequence from hyp-DS1 to hyp-DS4 ('hyp' stands for hypothesis) indicates the transition of the system's intention recognition result when the dialogue took place. The corpus also contains correct dialogue acts and dialogue states ('ref' stands for reference) that can be labeled later by an annotator. Here, a correct dialogue state means the dialogue state that a human overhearing the conversation would think the system should have

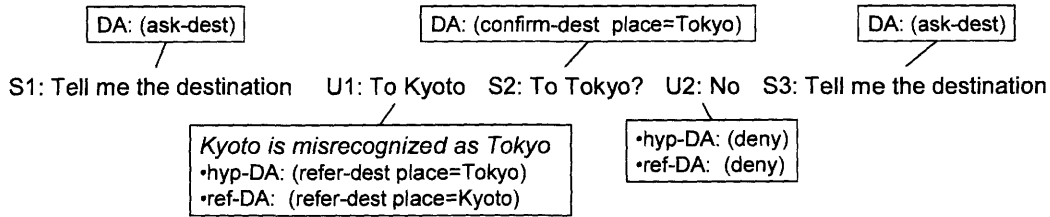
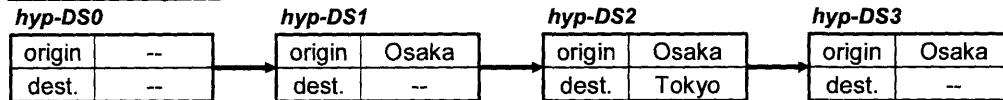
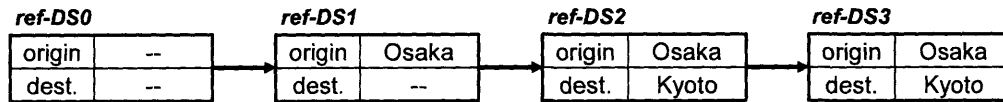
User and System Utterances**Sequence of intention recognition results of the system****Correct sequence of intention recognition results labeled later**

Figure 4.4: Example of a dialogue corpus.

possessed.

From such a corpus, we can obtain sequences of ref-DAs and ref-DSs, which can be used to calculate probabilities (1) and (2). If a dialogue act type sequence such as *refer-origin refer-dest* is a probable one, DS3 would be chosen over DS4 in Fig. 4.2, and if a dialogue state update pattern such as from ref-DS2 to ref-DS3 is found likely to occur, the system would be able to correctly choose DS4 as the best dialogue state after U2 in Fig. 4.3.

4.4.1 Statistical Information

4.4.1.1 N-gram Probability of a Dialogue Act Type Sequence

We employ the N-gram probability for the probability of a dialogue act type sequence. Here, a dialogue act type sequence means a sequence of dialogue act types of both user and system utterances. N-gram probability of dialogue act types has been used to statistically estimate the next dialogue act type in disambiguating speech understanding results (Nagata and Morimoto, 1994; Reithinger and Maier, 1995). It has also been used in finding problematic dialogues in a

1. Whether slots asked previously by the system are changed.
2. Whether slots being confirmed are changed.
3. Whether slots already confirmed (grounded) are changed.
4. Whether slots that do not have values are filled.
5. Whether slots that have values are overwritten.
6. When 5 is true, whether slot values do not change as a result.
7. Whether the system's previous utterance is an open prompt.

Figure 4.5: Seven binary attributes to classify a dialogue state update.

tutoring domain by detecting an unlikely sequence (Forbes-Riley and Litman, 2005). Using the same idea, we collect dialogue act type sequences from the dialogue corpus and create an N-gram language model to calculate the N-gram probability.

4.4.1.2 Occurrence Probability of a Dialogue State Update Pattern

We use the occurrence probability of a dialogue state update pattern for the probability of a dialogue state update. The simple bigram of dialogue states would not be sufficient due to the complexity of the data that a dialogue state possesses, which can cause data sparseness problems.

We first classify the ways that a dialogue state is updated into 96 classes characterized by seven binary attributes (Fig. 4.5), and then compute the occurrence probability of each class in a corpus. Note that the number of classes is not 128 (2^7) because attribute 6 is dependent on attribute 5. In the classification, an update after an open prompt is treated separately by having attribute 7, because such a prompt would lead to an unrestricted user utterance, leading to its own update pattern. Contrary to the N-gram probability of dialogue act types that represents a brief flow of a dialogue, the probability of a dialogue state update represents a more detailed flow of a dialogue, focusing mainly on the intention recognition result.

4.4.2 Scoring of Dialogue States Using the Statistical Information

Using the two probabilities, we define the score of a dialogue state S_{t+1} as

$$S_{t+1} = S_t + \alpha \cdot s_{act} + \beta \cdot s_{ngram} + \gamma \cdot s_{update} \quad (4.1)$$

where S_t is the score of a dialogue state just before the update (initially set to zero), s_{act} the score of a dialogue act, s_{ngram} the score concerning the N-gram probability of a dialogue act type sequence, s_{update} the score concerning the occurrence probability of a dialogue state update pattern, and α , β , and γ are the weighting factors. The dialogue act score (s_{act}) is introduced to prioritize dialogue states updated by dialogue acts derived from reliable speech recognition and parsing results. For s_{act} , speech recognition confidence or linguistic and acoustic scores of speech recognition results from which the dialogue act originates can be used.

Using Eq. 4.1, $m \times l$ new dialogue states created from l dialogue acts and m dialogue states are scored and ranked. Since the number of dialogue states grows exponentially, we consider that the maximum number of dialogue states has to be set in order to drop low-score dialogue states and thereby perform the operation in real time. This dropping process can be considered as a beam search in view of the entire discourse process; thus, we name the maximum number of dialogue states *the dialogue state beam width*.

4.5 Experiment

We performed experiments to verify our approach. We first collected dialogue data using two systems in different domains and annotated the dialogues with reference dialogue acts and dialogue states so as to extract the statistical information. We then evaluated the usefulness of the statistical information by performing a dialogue-state-ranking experiment.

4.5.1 Systems

4.5.1.1 Train Ticket Reservation Domain

We prepared a Japanese spoken dialogue system in the train ticket reservation domain (hereafter the train domain). Using the system, users reserve a train seat by specifying a place of departure, destination, train type, train number, and date. The system works on the phone. The speech recognition engine is Julius (Lee et al., 2001) with its attached acoustic model trained for telephony, and the speech synthesis engine is FinalFluet (Takano et al., 2001). The system has a vocabulary of 193 words. For the language model, we used a trigram trained from randomly generated texts of acceptable phrases.

The system uses the 1-best speech recognition hypothesis for language understanding. We realized our understanding grammar as a weighted finite state transducer (WFST) in a manner similar to (Potamianos and Kuo, 2000). We first defined sequences of words and concepts that derive dialogue acts in a manner similar to SPEECHBUILDER (Glass and Weinstein, 2001), and from them compiled a WFST that maps a sequence of words into a scored list of dialogue acts augmented with concepts. For example, we define a sequence “I would like to go to *Tokyo*” as one corresponding to a dialogue act type *refer-dest*. Here, ‘Tokyo’ is marked as a place concept. When the system parses the same sentence, the WFST outputs a dialogue act *refer-dest* together with “place=Tokyo” as its concept. The scoring for the WFST was tuned to derive as few dialogue acts and as many concepts as possible from an input. The output order of the dialogue acts with the same score depends on how they are found in the search process. The top ranked dialogue act is passed on to the discourse understanding component to update the dialogue state. There are 12 dialogue acts in our grammar.

The system simply uses a single dialogue state for discourse understanding because the aim of this system is to collect dialogue data to extract the statistical information. The dialogue state has five slots for the intention recognition result: *origin*, *destination*, *train type*, *train number*, and *date*. The intention recognition result is updated by manually created discourse understanding rules. We have 15 rules for this domain. For example, there is a rule to process a dialogue act *refer-dest* which fills the destination slot with the place name in the dialogue act.

Currently, our crude rules put every concept they encounter into the associated slots without consulting the dialogue history. Since only a single value is permitted to fill a slot, previous slot-fillers are always overwritten by the new ones. For each slot, the system also holds a *grounding flag* that indicates if the value of a slot has been acknowledged by the user.

For response generation, the dialogue manager first determines whether or not the system should utter a back-channel (e.g., “uh-huh”). Note that, in Japanese spoken dialogue, back-channels are very frequently observed. If the user’s previous dialogue act is not of a type explicitly requesting a response from the system or a filler and no more than three slots are filled, the system assumes that the user has not completed his/her request and utters a back-channel. If the system decides not to utter a back-channel, it then checks how many slots have been filled and grounded. If the system finds slots that are filled but ungrounded, the system confirms these slots in one utterance. Similarly, if there is only one slot that is filled and ungrounded, it only confirms that one value. The system does not use an implicit confirmation strategy. If all the slots have been filled and grounded, the system tells the user that it has completed the reservation.

If none of the above conditions hold, which is the case when the user explicitly requests a response with no slots filled or three or fewer slots grounded, the system asks the user to fill the missing slots one at a time in the order of the place of departure, destination, date, train type, and train number. All the responses are generated by templates. There are 19 templates in all, including the ones for greetings and back-channels. The templates have forms such as “Do you want to go to [destination=X] from [origin=Y]?”, where X and Y are taken from the destination slot and origin slot, respectively.

4.5.1.2 Weather Information Service Domain

Another system was developed in the same way in the weather information service domain (hereafter the weather domain). The system is capable of delivering Japan-wide weather information from a weather database updated regularly. The system has a vocabulary of 839 words, covering most principal cities and all prefectures in Japan. The system uses the 1-best speech recognition hypothesis for language understanding. It uses a WFST constructed from 15 dialogue act

definitions for parsing. The system has a single dialogue state and the intention recognition result has three slots: *place*, *date*, and *information type* (general weather, probability of precipitation, and warning). The dialogue state is updated by 14 discourse understanding rules. The system uses the same back-channeling and confirmation strategies as the train domain. The system has 17 templates for utterance generation.

4.5.2 Data Collection

Using the two systems, we collected dialogue data using human subjects. We recruited 15 subjects (9 males and 6 females), and each subject performed 16 dialogues (8 dialogues per system) by calling the systems on the phone. On the basis of scenarios that we prepared in advance, they were instructed to reserve certain train seats or to retrieve weather information. In the train domain, the subjects reserved a single seat in a single session, whereas weather information for multiple places and dates were sought in the weather domain in a single session because some scenarios instructed them to compare the general weather or temperature of several places. Note that the weather system cannot deliver weather information for multiple places or dates at the same time. They must be separately elicited.

We collected 120 dialogues for each domain. We recorded all speech recognition results (10-best hypotheses, although the systems used only 1-best hypotheses in the data collection), dialogue acts (parsing results of the 10-best speech recognition hypotheses), system’s utterances, start and end times of user’s utterances, and dialogue states before and after the user utterance. The user’s voice and the system’s voice were also recorded. We transcribed all user utterances. There are 1,815 and 2,090 utterances in the train and weather domains, respectively.

Dialogues that took more than three minutes were regarded as failures. The task completion rates were 88.33% (106/120) and 78.33% (94/120) in the train and weather domains, respectively. Compared to the train domain, the task success of the weather domain was lower, perhaps because of the complexity of the assignments. The word error rates (WERs) were 42.08% and 48.06% in the train and weather domains, respectively. The speech recognition accuracy

was rather low, probably because the input was telephone speech, the language models were not created from transcriptions of real user utterances but from artificially generated ones, and there were many phonologically similar place names in the lexicons especially in the weather domain. However, considering the reasonable task success rates and that misrecognition triggers further misrecognition in human-computer dialogues, we consider this speech recognition performance to be tolerable.

4.5.3 Annotating Reference Dialogue Acts and Dialogue States

On the basis of the transcriptions, reference dialogue acts were annotated by hand for each user utterance in the collected dialogue data. For annotation, we used the dialogue act set defined for the data collection systems. Sometimes, there were utterances that were difficult to annotate using the dialogue act set. In such cases, the most appropriate dialogue act with regards to the domain was used for annotation. For an utterance “I would like to go to Yokohama in Kanagawa prefecture” in the train domain, since there is no dialogue act that includes an elaboration of a place “in Kanagawa prefecture”, we annotated it with (*refer-dest place=Yokohama*), which is appropriate in terms of this domain. If none of the predefined dialogue acts could be annotated for an utterance (e.g., an out-of-domain utterance), *filler* was assigned.

Using the reference dialogue acts, we automatically annotated the reference dialogue states. We made each system used in the data collection update its dialogue state by the reference dialogue acts as input. We recorded the dialogue states after the processing of each reference dialogue act as reference dialogue states. Note that the discourse understanding rules of the systems were designed to correctly update a dialogue state as long as the input is a correct dialogue act. We call the collected dialogue data with these annotations *the corpus*.

Table 4.1: Examples of dialogue act type sequences and their per word trigram probabilities in the train ticket reservation domain.

Dialogue Act Type Sequence	per word trigram probability
refer-origin back-channel refer-origin-dest	0.140
refer-origin back-channel refer-origin	0.195
refer-origin back-channel refer-date	0.114
refer-origin back-channel refer-dest	0.303
refer-origin back-channel refer-train	0.114
refer-origin request confirm-origin	0.176
refer-origin confirm-origin-dest filler	0.131
refer-origin confirm-origin-dest acknowledge	0.131
refer-origin confirm-origin-dest refer-origin-dest	0.131
refer-dest back-channel refer-dest	0.062

4.5.4 Deriving Statistical Information

4.5.4.1 Trigram Probability of a Dialogue Act Type Sequence

From the sequences of reference dialogue acts in the corpus, we created an N-gram language model of dialogue act types for each domain using the CMU-Cambridge Toolkit (Clarkson and Rosenfeld, 1997). We chose three as N (trigram) and used Good-Turing discounting.

Table 4.1 shows examples of dialogue act type sequences and their per word trigram probabilities in the train domain calculated using the trigram language model. Here, per word trigram probability means the average probability of all trigrams in a sequence. The trigram probabilities of the first two tokens are represented by their unigram and bigram probabilities. It can be seen from the table that the sequence *refer-origin back-channel refer-dest* is much more probable, with the probability of 0.303, than *refer-dest back-channel refer-dest*, with 0.062. It seems very unlikely that a user would mention the destination again after the system’s back-channel, which would prioritize DS3 over DS4 in Fig. 4.2.

Table 4.2 shows examples of dialogue act type sequences and their per word tri-

Table 4.2: Examples of dialogue act type sequences and their per word trigram probabilities in the weather information service domain.

Dialogue Act Type Sequence	per word trigram probability
confirm-place acknowledge tell-info	0.378
confirm-place acknowledge request	0.120
confirm-place refer-city back-channel	0.106
confirm-place refer-city confirm-place	0.335
confirm-place refer-info request	0.255
confirm-place refer-pref request	0.096
confirm-place refer-pref confirm-place	0.180
confirm-place refer-date refer-pref	0.255
confirm-place deny back-channel	0.145
confirm-place deny refer-city	0.114
confirm-place deny ask-place	0.243
confirm-place deny refer-pref	0.117

gram probabilities in the weather domain. It can be seen from the table that after the system’s confirmation of a place, the user is likely to acknowledge the confirmation request followed by the system’s delivery of weather information (*confirm-place acknowledge tell-info*). The sequence *confirm-place refer-city confirm-place*, in which the user corrects the system by referring to a city name followed by the system’s next confirmation request, seems equally likely.

4.5.4.2 Occurrence Probability of a Dialogue State Update Pattern

From all consecutive pairs of reference dialogue states in the collected data, we obtained the occurrence probability of each dialogue state update pattern using the classification scheme in Fig. 4.5.

Table 4.3 shows all the patterns in the corpus in the train domain. The seven binary values in the table indicate the conformity to attributes 1-7 from left to right. The patterns are ordered by the magnitude of occurrence probability. The pattern in bold font indicates that it is not observed in the weather domain (See Table 4.4 for comparison).

4.5. Experiment

Table 4.3: The 18 dialogue state update patterns and their occurrence probabilities in the train ticket reservation domain. See Fig. 4.5 for the details of the binary attributes. Attributes 1-7 are ordered from left to right.

# Pat.	Attributes (1-7)	Prob	# Pat.	Attributes (1-7)	Prob
1	0 0 0 0 1 1 0	0.3218	10	0 0 0 1 1 1 0	0.0050
2	0 0 0 0 0 0 0	0.2964	11	0 1 0 0 1 1 0	0.0044
3	0 0 0 1 0 0 0	0.1256	12	0 0 0 0 0 0 1	0.0017
4	0 0 0 1 0 0 1	0.0645	13	0 1 0 1 1 1 0	0.0017
5	0 0 1 0 0 0 0	0.0623	14	0 1 0 0 1 0 0	0.0011
6	0 1 0 0 0 0 0	0.0474	15	1 0 0 1 1 1 0	0.0011
7	1 0 0 1 0 0 0	0.0452	16	0 1 1 0 1 0 0	0.0006
8	0 1 0 1 0 0 0	0.0138	17	0 0 1 0 1 0 0	0.0006
9	0 1 1 0 0 0 0	0.0066	18	0 0 0 1 1 0 0	0.0006

Out of 96 possible patterns, we observed 18 patterns. It can be seen from the table that there are two dominating patterns: one in which the slot values are overwritten to the same values, and another in which there is no change to the slot values. This leads us to believe that the transition from DS2 to DS4 would be more probable than DS1 to DS3 in Fig. 4.3 because the former corresponds to pattern 2 with the probability of 0.2964 (no change in the slots) and the latter matches pattern 6 with the probability of 0.0623 (deleting the value of a slot being confirmed). It is intuitive that no change in the slots was very likely because, considering the transition of reference dialogue states, once the true intention of the user has been recognized, slot values should not change as long as the intention of the user is consistent, which is the case in our scenario-based dialogues.

Table 4.4 shows the update patterns found in the corpus of the weather domain. We found 23 patterns in all. Overall, the tendency of the observed patterns and their probabilities are similar to that in the train domain. The patterns that are unique in this domain are mostly the ones that conform to attribute 5; i.e., whether slots that have values are overwritten. As mentioned, the user's intention is not supposed to change during a dialogue; however, since the subjects often had to seek information for multiple places/dates in the scenarios of this domain, they sometimes changed their intentions in the middle of a dialogue largely for

Table 4.4: The 23 dialogue state update patterns and their occurrence probabilities in the weather information service domain.

# Pat.	Attributes (1-7)	Prob	# Pat.	Attributes (1-7)	Prob
1	0 0 0 0 1 1 0	0.3096	13	0 0 0 0 0 0 1	0.0072
2	0 0 0 0 0 0 0	0.2962	14	0 1 1 0 0 0 0	0.0062
3	0 0 0 1 0 0 0	0.0880	15	0 1 0 0 1 0 0	0.0053
4	0 1 0 0 0 0 0	0.0766	16	0 0 0 0 1 0 0	0.0038
5	0 0 0 1 0 0 1	0.0679	17	1 0 0 1 1 1 0	0.0033
6	0 0 0 1 1 1 0	0.0244	18	1 0 0 0 1 0 0	0.0019
7	0 1 0 0 1 1 0	0.0244	19	0 0 0 1 1 0 0	0.0014
8	0 0 1 0 0 0 0	0.0239	20	1 0 1 0 0 0 0	0.0010
9	0 1 0 1 0 0 0	0.0230	21	0 0 1 0 1 0 0	0.0010
10	1 0 0 1 0 0 0	0.0134	22	0 1 1 0 1 1 0	0.0005
11	0 1 0 1 1 1 0	0.0124	23	0 0 1 1 0 0 0	0.0005
12	0 0 1 0 1 1 0	0.0081			

the purpose of avoiding repeated misrecognition.

4.5.5 Evaluation

4.5.5.1 Offline Discourse Understanding

We propose evaluating our approach by *offline discourse understanding*, in which we make the discourse understanding component that we want to evaluate process sequences of user and system utterances as they are recorded in the corpus. The discourse understanding performance is evaluated by the ranking accuracy of the dialogue states that the component outputs after each user utterance. Although we naturally believe that an online evaluation, in which dialogue experiments are performed by human subjects using a system based on our approach, is preferable, considering that a dialogue state is a system's interpretation of what has happened in a dialogue, we consider it reasonable to evaluate discourse understanding by how accurately the system makes sense of a recorded sequence of user and system utterances.

Suppose that the corpus has a dialogue as illustrated in Fig. 4.2. In offline discourse understanding, the initial dialogue state (DS0) is updated by the system prompt (S1). Then, the user utterance U1 is taken from the corpus to update the dialogue state using the dialogue acts for U1. Here, the dialogue acts can be those recorded in the corpus, re-parsing results of the recorded speech recognition hypotheses, or those newly created by re-understanding the utterance from the recorded voice. If the system derives m dialogue acts for the utterance, the system would create m dialogue states (List-1). After the m dialogue states are updated by the system's back-channeling act (S2), n dialogue acts for U2 update the dialogue states to create $m \times n$ dialogue states (List-2). Finally, we evaluate the ranking accuracy of List-1 and List-2.

Although offline discourse understanding does not require human subjects, it can still be computationally expensive when the number of utterances to process is large because all utterances in the corpus have to be sequentially processed to create the lists of dialogue states. Even if the dialogue state beam width is set to 100, when the system uses 10-best speech recognition hypotheses for language understanding, as many as 1,000 dialogue states can be created at a time. This computational cost especially hinders experiments with varying parameters. Our idea for coping with this shortcoming is to prepare in advance lists of possible dialogue states after each user utterance using a default set of parameters and just re-rank the stored lists of dialogue states when we need to evaluate with different parameters. The lists of dialogue states would be different when other sets of parameters are employed because of the accumulative nature of the score of a dialogue state (cf. Eq. 4.1). However, such lists would still be useful for comparing the performance of different discourse understanding methods because we can evaluate them by their relative ranking performance; i.e., the method that ranks dialogue states in the most suitable order can be considered to be better than others.

We prepared lists of dialogue states after each user utterance in the corpus. We modified the discourse understanding components of the systems used in the data collection to handle multiple dialogue states and made them sequentially understand the utterances in the corpus. Here, the utterances mean the dialogue acts as they are stored in the corpus. A user utterance is represented by the

dialogue acts for the 10-best speech recognition hypotheses.

For the ranking of dialogue states, we used Eq. 4.1 with the weighting factors $\alpha = 1$, $\beta = 0$, and $\gamma = 0$, which means that the ranking was purely based on the score of dialogue acts. For s_{act} , we used the common logarithm of the posterior probability of a speech recognition hypothesis from which the dialogue act originates. The posterior probability is estimated from acoustic and language model scores as described in (Bouwman et al., 1999). We stored the top-100 dialogue states after each utterance. In the train domain, we have 1,815 lists of dialogue states corresponding to the number of the utterances. In the weather domain, we have 2,090 such lists.

4.5.5.2 Ranking Experiment

We re-ranked each list of the stored dialogue states using Eq. 4.1 with different sets of weighting factors (α , β , and γ). We assumed that the weighting factors take either 1 or 0 in this experiment because we wanted to investigate how the use of the statistical information affects discourse understanding. We do not consider the case where all the weighting factors are 0 because the ranking is impossible.

For s_{act} , we use the common logarithm of the posterior probability of the speech recognition hypothesis for a dialogue act. For s_{ngram} , we use the common logarithm of the per word trigram probability for the dialogue act type sequence corresponding to the current user utterance, the previous system utterance, and the user’s previous utterance. Since an utterances may correspond to multiple dialogue acts, the sequence would include three or more dialogue act types. For example, an utterance “No, I’d like to go to Tokyo” would correspond to (*deny*) and (*refer-dest place=Tokyo*). For s_{update} , we use the common logarithm of the occurrence probability of the dialogue state update pattern.

4.5.5.3 Evaluation Criteria

Mean Reciprocal Rank (MRR) The MRR is a metric for evaluating ranking performance. It is the inverse of the rank of the first correct answer in an answer candidate list. This metric is commonly used in information retrieval and question answering systems research, where the ranking of the answer candidates

is important. Since we also evaluate ranking, we consider it an appropriate measure. To calculate the MRR, we only focus on the lists where there is a correct dialogue state because we are interested in whether a correct dialogue state can be ranked higher using the statistical information.

Slot Accuracy Even though good ranking would mean better discourse understanding, it is also desirable that the slot values of the top ranked dialogue state be accurate. Therefore, we calculate the slot accuracy of the top ranked dialogue states. The slot accuracy is calculated by dividing the number of correctly filled slots over the number of slots.

Concept Error Rate (CER) We also calculate the CER of the slot values of the top ranked dialogue states because the CER is commonly used in speech understanding research. The CER is calculated by dividing the number of incorrect slots by the number of filled slots. Although Higashinaka et al. (2004) proposed creating a measure for discourse understanding by finding a measure that correlates closely with the performance of a dialogue system, the measure assumes that the system holds a single dialogue state. The best measure they propose is based on the precision of the update of a dialogue state (called update precision), which is difficult to calculate when a system has multiple dialogue states because it is not clear whether the sequence of dialogue states with a different understanding history can be used to calculate the update precision. We leave it our future work to find an appropriate measure for discourse understanding based on multiple dialogue states.

4.5.5.4 Results

For evaluation, we split the corpus into five sets and performed a five-fold cross validation, extracting the statistical information from four sets and evaluating with the remaining set in a round-robin fashion. For the calculation of the MRR, we used 850 and 878 lists that contained correct dialogue states in the train and weather domains, respectively. We found a large number of dialogue state lists that did not contain correct dialogue states because of many speech recognition failures. For example, if all the speech recognition hypotheses contain

Table 4.5: Evaluation results for each combination of the weighting factors in the train ticket reservation (TRAIN) and weather information service (WEATHER) domains.

Weights			TRAIN			WEATHER		
α	β	γ	MRR	ACC	CER	MRR	ACC	CER
1	0	0	0.676	0.769	0.405	0.748	0.686	0.508
0	0	1	0.678**	0.767	0.393	0.769**	0.695*	0.495*
0	1	0	0.738**	0.787*	0.362*	0.693	0.689	0.441*
0	1	1	0.734**	0.789*	0.350**	0.735*	0.692*	0.451**
1	0	1	0.691**	0.767	0.393	0.771**	0.695	0.495
1	1	0	0.747**	0.791**	0.359**	0.722	0.697	0.433**
1	1	1	0.745**	0.792**	0.344**	0.750**	0.697**	0.446**

** Statistical significance ($p < 0.01$) over baseline ($\alpha = 1, \beta = 0, \gamma = 0$)

* Statistical significance ($p < 0.05$) over baseline ($\alpha = 1, \beta = 0, \gamma = 0$)

words/concepts that would fill the slots in a wrong way, all resulting dialogue states would be incorrect. For the slot accuracy and the CER, we used all top ranked dialogue states in the whole lists of dialogue states.

Table 4.5 shows the evaluation results for the combinations of the weighting factors in the train and weather domains. Here, the baseline is the one that solely relies on the speech recognition confidence ($\alpha = 1, \beta = 0, \gamma = 0$). For the statistical comparison of the MRRs, we performed a sign test that compares the number of times a combination of weighting factors ranks the correct dialogue state higher than the baseline. For the slot accuracy and the CER, we calculated the mean of the slot accuracy and the CER for each dialogue and compared the number of dialogues that had the better mean of the slot accuracy or the CER.

It can be seen from the table that when we use the statistical information together with the speech recognition confidence ($\alpha = 1, \beta = 1, \gamma = 1$), we achieve the best performance with all evaluation criteria significantly better than the baseline in both domains, showing the effectiveness of using both the probability of a dialogue act type sequence and the dialogue state update pattern. In other combinations of weighting factors, the effects of the statistical information seems

to be limited, making improvements in some of the evaluation criteria with the exception of $[\alpha = 1, \beta = 1, \gamma = 0]$ in the train domain, which shows comparable performance to $[\alpha = 1, \beta = 1, \gamma = 1]$.

Overall, the train domain benefited more from the use of the statistical information. We consider this is due to the ambiguity of dialogue acts that resides in the train domain; i.e., utterances with bare place names would yield multiple dialogue acts referring to places of departure and destinations. On the other hand, the ambiguity of utterances solely come from the multiple speech recognition hypotheses in the weather domain. This is demonstrated by the fact that the use of the trigram probability of a dialogue act type sequence ($\beta = 1$) does not affect the results as much as the probability of a dialogue state update pattern does ($\gamma = 1$) in the weather domain; that is, there seems to be less need to disambiguate dialogue acts. In some cases, understanding based on only the statistical information ($\alpha = 0$) is better than relying only on the speech recognition confidence, probably due to numerous speech recognition errors.

4.5.5.5 Impact of the Dialogue State Beam Width

We calculated the MRR by placing a limit on the maximum number of dialogue states to hold (dialogue state beam width). For example, when we set the dialogue state beam width to N , only the dialogue states within the top- N are considered in the re-ranking process. By changing the dialogue state beam width, it is possible to examine how the number of dialogue states could affect ranking accuracy.

Figure 4.6 shows the performance changes in the MRR with different dialogue state beam widths. We used $\alpha = 1$, $\beta = 1$, and $\gamma = 1$ as the weighting factors. It can be seen that as the system holds more dialogue states, the MRR improves. However, the improvement begins to saturate when the dialogue state beam width is around 10, indicating that this could be the sufficient number of dialogue states. This result re-confirms that having multiple dialogue states can actually improve discourse understanding and that the system with multiple dialogue states could work in real time because the number of dialogue states to hold could be as few as 10, which does not impose a computational problem.

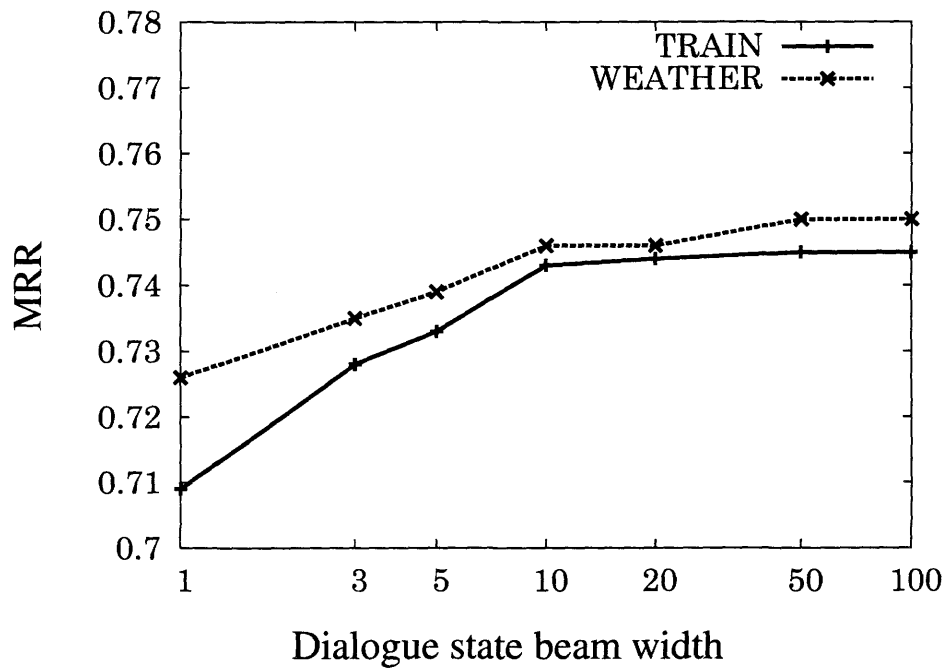


Figure 4.6: Performance changes in the Mean Reciprocal Rank (MRR) with different dialogue state beam widths in the train ticket reservation (TRAIN) and weather information service (WEATHER) domains. The x-axis is on the log scale.

4.5.5.6 Analysis of Successful Cases and Failures

We examined the instances of the ranking results to see what improvement had been made by the statistical information and whether further improvement can be made.

Figures 4.7 and 4.8 show a case where the statistical information is successfully exploited. In this piece of dialogue, the user said “*Hikari hyaku nijû kyû gô* (Hikari express number 129)”, whose best speech recognition hypothesis was “*Hikari nihyaku nijû kyû gô* (Hikari express number 229)”. After this utterance, the correct dialogue state which was updated using the third speech recognition hypothesis “Hikari express number 129” was ranked second in the dialogue state list. The top dialogue state was the one updated with the best speech recognition hypothesis. Then, the next system utterance “*Hikari nihyau nijû kyû gô desune?* (Did you say Hikari express number 229?)” and the next user utterance “*Chigaimasu* (No)” were processed to update the dialogue states. As a result, the dialogue state that was updated from the second-ranked dialogue state came to the top. This is achieved by the high probability of the dialogue act type sequence and the dialogue state update pattern, showing that the understanding example in Fig. 4.3 actually took place.

Figures 4.9 and 4.10 show a typical case where the statistical information has an adverse effect. In this piece of dialogue, the user asked for the probability of precipitation in Aomori prefecture. Since the speech recognition for this utterance was good, the best dialogue state that solely relies on the speech recognition confidence is correct. However, when we use the statistical information, the best dialogue state is the one updated with the third speech recognition hypothesis. This happened because the previous user utterance referred to a city and the third hypothesis also refers to a city, which boosted the probability of a dialogue act type sequence. We found a number of such cases where the correct speech recognition does not yield the best dialogue state because of some dialogue act sequences that are unlikely but do happen in actual dialogue. In many cases of failure, the contextual information is too strong and hinders straightforward understanding of the correct speech recognition results. We suspect that optimizing or dynamically changing α , β , and γ would lead to better ranking, but this remains future work.

User and System Utterances:

U1 : Hai (Yes)
S1 : Dono ressha desuka? (Which train type?)
U2 : Hikari hyaku nijû kyû gô (Hikari express number 129)
S2 : Hikari nihyaku nijû kyû gô desune?
(Did you say Hikari express number 229?)
U3 : Chigaimasu (No)

Speech recognition Hypotheses for U2:

1. *Hikari nihyaku nijû kyû gô* (Hikari express number 229)
[recognition confidence: 0.1091]
2. *Chigau hyaku nijû kyû gô* (No, number 129)
[recognition confidence: 0.1072]
3. *Hikari hyaku nijû kyû gô* (Hikari express number 129)
[recognition confidence: 0.1067]

Figure 4.7: A typical case where the statistical information is successfully exploited. See also Fig. 4.8 for how dialogue states are updated in this piece of dialogue. Recognition confidence means the posterior probability of a speech recognition hypothesis estimated from acoustic and language model scores (Bouwman et al., 1999).

<The 2nd-ranked dialogue state after U2 ($\alpha = 1, \beta = 1, \gamma = 1$)>:

Dialogue State ID: 3400 (Parent Dialogue State ID: 1900)

Accumulated Score: -13.185 $\left(\begin{array}{l} \text{Score: } -2.741 \\ S_{act} : -0.972 \ S_{ngram} : -1.393 \ S_{update} : -0.376 \end{array} \right)$

Dialogue Act History: [(acknowledge)] \longrightarrow [(ask-type)] \longrightarrow [(refer-type type=Hikari)(refer-number number=129)]

Previous Frame: grounded

origin	Shin-Kobe	yes
dest.	Okayama	yes
date	(1 7)	yes
type	--	no
number	129	no

Current Frame: grounded

origin	Shin-Kobe	yes
dest.	Okayama	yes
date	(1 7)	yes
type	Hikari	no
number	129	no

<The top-ranked dialogue state after U3 ($\alpha = 1, \beta = 1, \gamma = 1$)>:

Dialogue State ID: 4292 (Parent Dialogue State ID: 3400)

Accumulated Score: -15.08 $\left(\begin{array}{l} \text{Score: } -1.895 \\ S_{act} : -0.890 \ S_{ngram} : -0.526 \ S_{update} : -0.479 \end{array} \right)$

Dialogue Act History: [(refer-type type=Hikari)(refer-number number=129)]
 \longrightarrow [(confirm-type-number type=Hikari number=229)] \longrightarrow [(deny)]

Previous Frame: grounded

origin	Shin-Kobe	yes
dest.	Okayama	yes
date	(1 7)	yes
type	Hikari	no
number	129	no

Current Frame: grounded

origin	Shin-Kobe	yes
dest.	Okayama	yes
date	(1 7)	yes
type	Hikari	no
number	129	no

Figure 4.8: The 2nd-ranked dialogue state after U2 and the top-ranked dialogue state after U3 in the piece of dialogue shown in Fig. 4.7.

User Utterance:

Aomori-ken no kôsuï kakuritsu wo oshiete kudasai
(please tell me the probability of precipitation in Aomori prefecture)

Speech recognition Hypotheses:

1. *Aomori-ken no kôsuï kakuritsu oshiete kudasai*
(Please tell me the probability of precipitation Aomori prefecture)
[recognition confidence: 0.1031]
2. *Aomori-ken no kôsuï kakuritsu wo oshiete kudasai*
(Please tell me the probability of precipitation in Aomori prefecture)
[recognition confidence: 0.1012]
3. *Aomori-shi no kôsuï kakuritsu oshiete kudasai*
(Please tell me the probability of precipitation Aomori city)
[recognition confidence: 0.1003]

Figure 4.9: A typical case where the statistical information has an adverse effect because of strong contextual constraints. See also Fig. 4.10 for how dialogue states are updated in this piece of dialogue.

<The top-ranked dialogue state ($\alpha = 1, \beta = 0, \gamma = 0$)>:

Dialogue State ID: 262 (Parent Dialogue State ID: 32)

Accumulated Score: -2.852 **Score:** -0.987
Dialogue Act History: $\left(\begin{array}{l} S_{act} : -0.987 \ S_{ngram} : -2.745 \ S_{update} : -0.745 \end{array} \right)$

[(refer-city place=Chitose city)(refer-city place=Habikino city)] → [(Back-channel)]
→ [(refer-pref place=Aomori pref.)(refer-info type=precipitation)(request)]

Previous Frame: grounded

place	Habikino City	no
info	--	no
date	--	no

Current Frame: grounded

place	Aomori pref.	no
info	precipitation	no
date	--	no

<The top-ranked dialogue state ($\alpha = 1, \beta = 1, \gamma = 1$)>:

Dialogue State ID: 422 (Parent Dialogue State ID: 32)

Accumulated Score: -12.668 **Score:** -4.402
Dialogue Act History: $\left(\begin{array}{l} S_{act} : -0.999 \ S_{ngram} : -2.745 \ S_{update} : -0.658 \end{array} \right)$

[(refer-city place=Chitose city)(refer-city place=Habikino city)] → [(Back-channel)]
→ [(refer-pref place=Aomori city)(refer-info type=precipitation)(request)]

Previous Frame: grounded

place	Habikino city	no
info	--	no
date	--	no

Current Frame: grounded

place	Aomori city	no
info	precipitation	no
date	--	no

Figure 4.10: The top-ranked dialogue states with two different parameter sets; namely, $[\alpha = 1, \beta = 0, \gamma = 0]$ and $[\alpha = 1, \beta = 1, \gamma = 1]$ after the user utterance shown in Fig. 4.9. Aomori city was incorrectly preferred to Aomori prefecture to fill the place slot because the user's previous utterance was recognized to have a city name due to speech recognition errors.

4.6 Summary and Future Work

We proposed a new discourse understanding method that ranks multiple dialogue states using statistical information obtained from dialogue corpora. The method uses the trigram probability of dialogue act types and the probability of dialogue state update pattern in addition to the speech recognition confidence of a dialogue act to score a dialogue state. Experimental results in the train ticket reservation domain and the weather information service domain show that our approach can significantly improve the ranking of the dialogue states. We also confirmed that it is effective to hold multiple dialogue states for discourse understanding and that the sufficient number of dialogue states to hold could be as few as 10.

There still remain several issues that we need to explore. These include the use of statistical information other than the probability of a dialogue act type sequence and the probability of a dialogue state update pattern, the optimization of the weighting factors (α , β , and γ), and more experiments in larger domains. It should also be noted that the experiment we performed was an offline evaluation. An online evaluation would be desirable for a more accurate evaluation. Despite these issues, the present results show that our approach is promising.

CHAPTER 5

Conclusions and Future Directions

In our research, we seek to improve discourse understanding in spoken dialogue systems so that systems and users can interact more naturally using speech. Discourse understanding refers to a system's ability to understand user utterances based on the context of a dialogue. Until now, discourse understanding has been performed in a heuristic manner because the primary importance has been place on developing workable systems. However, as systems become increasingly complex, systematic methods are needed if we are to achieve continuous progress in the field and also reduce the cost of system development. This thesis proposed three such methods/approaches to tackle three problems in discourse understanding in spoken dialogue systems. In what follows, we summarize the contributions of this thesis and mention future work.

5.1 Summary of Contributions

5.1.1 Establishment of an Evaluation Measure for Discourse Understanding

We successfully created suitable evaluation measures for discourse understanding in spoken dialogue systems. We enumerated possible metric candidates for the evaluation of discourse understanding and calculated their correlation with the system's performance through dialogue experiments. As a result, the update recall, frame match rate, and update precision were shown to have relatively good correlation with system performance, suggesting they are appropriate as evalua-

tion measures. We also found that the F-measure, which is the harmonic mean of update recall and update precision, is also a good measure. By using these measures, we can now safely compare discourse understanding components of various systems and thereby advance discourse understanding research. Although we have not yet put the evaluation measures we created into practice, we plan to use the measures to evaluate our prospective systems.

5.1.2 Improvement in Confidence Scoring of Intention Recognition Results Using Discourse Information

We successfully showed that discourse information is useful for improving confidence scoring of intention recognition results. Although confidence scoring itself does not directly improve discourse understanding performance, accurate confidence scoring would make it possible for the discourse understanding component to automatically reject erroneous slot values and also to ground certain slot values without explicit confirmations requested by the dialogue manager, which would result in better discourse understanding performance.

In our approach, we used Grice's maxims of cooperativeness, which are the norms that should be followed in a collaborative conversation, to derive discourse features for slot values. We utilize these discourse features together with the acoustic and language model features of the speech recognition results to train confidence models for slot values. Experimental results showed that the proposed method significantly improves the confidence scoring, indicating the effectiveness of the discourse features.

5.1.3 Automatic Disambiguation of User Intentions Using Corpus Statistics

We successfully showed that the statistical information that can be derived from dialogue corpora are useful for disambiguating possible interpretations of user intentions. Due to the multiple speech recognition hypotheses created by speech recognizers and the ambiguity in natural language, a spoken dialogue system has to choose the best interpretation from multiple possible interpretations. This corresponds to ranking multiple dialogue states. Previously, the ranking has

been performed by hand-crafted rules, which are expensive to make and require expertise in system development.

We assumed that the best dialogue state is the one that has seen the most likely sequence of dialogue acts and dialogue state updates, and derived an N-gram language model of dialogue act types and the transition probability of dialogue state updates from dialogue corpora. We used these two kinds of statistical information to score dialogue states for ranking. Experimental results showed that the statistical information can improve the ranking performance. Although the ranking performance achieved by our approach may not be superior to that achieved by carefully created hand-crafted rules, our approach offers a promising first step for automatically disambiguating user intentions in spoken dialogue systems.

5.2 Future Directions

5.2.1 Incorporating Confidence Scoring into Ranking Dialogue States

Although the purposes may be different (error detection vs. disambiguation), it is noticeable that the work of confidence scoring of intention recognition results described in Chapter 3 and that of ranking multiple dialogue states described in Chapter 4 bear similarity because they both evaluate the plausibility of the system's understanding results. Therefore, it may be possible to use the confidence scores of intention recognition results for the ranking of dialogue states and also to use the corpus statistics for the confidence scoring of intention recognition results.

However, we need to take into account several differences in the two approaches before such integration can take place. First of all, the former focuses on intention recognition results (slot values) and the latter on dialogue states. Although an intention recognition result forms an important part of a dialogue state, a dialogue state contains many other pieces of information, such as the history of user and system utterances. Therefore, it may not be possible to represent a dialogue state with the small number of features used for the confidence

scoring of intention recognition results. Similarly, considering the complexity of values that a dialogue state may take, we may need a large amount of data if we want to train confidence models for dialogue states because of the data sparseness problems.

Another difference lies in the criteria that we used for scoring. For the confidence scoring of intention recognition results, Grice's maxims, which provide intuitive principles in a cooperative dialogue, are used. On the other hand, corpus statistics are used for the scoring of dialogue states. Since Grice's maxims offer a fixed criterion based on a theory in pragmatics and corpus statistics derive criteria empirically from data, the former can be regarded as taking a top-down approach and the latter as taking a bottom-up one. Although Grice's maxims and corpus statistics can be naively integrated, since they are derived from different perspectives, it would be necessary to examine their characteristics, similarities, and differences as a first step. In addition, the understanding models are different in the two approaches. The system is assumed to have a single dialogue state in Chapter 3, whereas multiple dialogue states are assumed in Chapter 4. It is not clear whether the features based on Grice's maxims can be derived successfully for multiple dialogue states. It is likely that sequences of slot values may differ greatly in the two approaches.

Taking these differences into account, we should investigate possible ways to integrate the two approaches for further improvement.

5.2.2 Handling Other Discourse Phenomena

The primary emphasis of this thesis is on understanding error-prone user utterances using discourse information. Such understanding capability is essential for workable systems because, according to (Lippmann, 1997; Mori, 2007), speech recognition errors are unlikely to be eliminated in the near future and can easily become a bottleneck for other processes in spoken dialogue systems. We believe more work has to be done to reduce understanding errors caused by misrecognition.

In addition to being able to understand utterances containing speech recognition errors, spoken dialogue systems are expected to possess other discourse understanding capabilities. One is the capability is to perform anaphora/reference

resolution. Successfully understanding anaphoric expressions, such as *it*, *he*, or *she*, would facilitate users in stating their requests in a concise and efficient manner, improving the usability of speech interfaces. Functions to handle ellipsis (omitting a part of a sequence) and zero anaphora (gaps in sentences that have an anaphoric function) would also be important for similar reasons. Although there is a body of research on reference resolution (Lappin and Leass, 1994; Iida et al., 2007; Isozaki and Hirao, 2003), the methods mainly focus on written discourse and are therefore unlikely to be directly applied to speech due to differences between texts and speech. In current spoken dialogue systems, reference resolution is done by hard-crafted rules (Filisko, 2002). However, as studies of text processing indicate (op den Akker et al., 2002), the accuracy can be limited. A systematic method is desired for improvement.

In discourse processing, it is important to find relationships between elements in a discourse. For example, in text processing, finding a relationship between sentences/paragraphs has been found useful in many NLP tasks, such as automatic summarization (Dang, 2006) and question answering (Dang et al., 2007). Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) is often used for representing a discourse structure, and automatic detection of RST relations has been widely studied (Marcu and Echihiabi, 2002; Soricut and Marcu, 2003).

In spoken dialogue processing, much work has been done on finding relationships between utterances, which corresponds to assigning dialogue act type tags to utterances. For example, in the switchboard corpus (Godfrey et al., 1992), utterances are tagged with a SWBD-DAMSL tag-set, which defines an utterance's relation to its previous one (Jurafsky et al., 1997). Recent spoken dialogue systems perform automatic dialogue act type detection to improve speech understanding (Nagata and Morimoto, 1994; Reithinger and Maier, 1995). In fact, our approach is one example of such detection in that it utilizes dialogue act type N-grams for ranking multiple dialogue states (see Chapter 4). However, due to difficulty in modeling discourse structures of spoken dialogue and defining dialogue acts, application of dialogue act type detection has been limited to a small portion of a discourse (at most a few utterances), and dialogue act types considered are mostly domain specific. With further studies, we will probably see the development of spoken dialogue systems that can fully understand the entire

structure of a discourse.

In this thesis, we did not cover plan recognition (Allen and Perrault, 1980; Carberry, 1990), which is essential for accurately recognizing user intentions. Currently, there are no well-established techniques for performing plan recognition under uncertain input. Recent reports have proposed statistically performing plan recognition in spoken dialogue systems (Allen et al., 2007). We will keep abreast of these studies and investigate ways ourselves for realizing robust plan recognition for spoken dialogue systems.

5.2.3 Towards More Conversational Spoken Dialogue Systems

This thesis focused on task-oriented spoken dialogue systems, and the tasks handled are limited to relatively small domains, such as meeting room reservation and weather information services. However, it would be desirable for systems to be able to handle a wide variety of domains simultaneously so that users would not need to select systems depending on their needs. Several attempts have been made to enable systems to handle multiple domains (Ikeda et al., 2007). However, it is still an emerging field. Since topics vary in human-human interaction, an ability to cope with multiple domains would make current systems more conversational and easier to interact with.

In addition, there is a good body of research in the field of question answering; that is, to answer questions in natural language by extracting relevant pieces of texts from text archives (Dang et al., 2007). Since many types of questions, from factoid questions that ask for named-entities, such as proper nouns to numerical expressions, to non-factoid ones that ask for reasons and procedures, can be answered with current techniques (Isozaki, 2005; Higashinaka and Isozaki, 2008), incorporating question answering capability into dialogue systems would allow systems to cover a wide variety of topics and could offer a promising direction towards more conversational systems. To date, there have been several systems equipped with a question answering functionality (Cincarek et al., 2007; Misu and Kawahara, 2007; van Schooten et al., 2007).

Spoken dialogue systems could serve many other purposes than just task ac-

complishment. For example, they are good media for social interaction, such as casual conversation or chats (Eggins and Slade, 1997). Realizing such interaction could make systems important partners in our daily lives, enabling exchanges of not just information but also emotion. We believe studies in psychology and sociology would play key roles in realizing systems that enable such an interaction. Recently, the field of persuasive computing is emerging (Fogg, 2003), in which machines are regarded as means to influence people's attitudes. Although little work has been done on the effects of dialogue systems on people's attitudes, we consider it would become one of the most important issues in the future.

Bibliography

- Sherif Abdou and Michael Scordilis. Integrating multiple knowledge sources for improved speech understanding. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1783–1786, 2001.
- Alicia Abella and Allan L. Gorin. Construct algebra: Analytical dialogue management. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 191–199, 1999.
- James Allen, George Ferguson, and Amanda Stent. An architecture for more realistic conversational systems. In *Proceedings of the 2001 International Conference on Intelligent User Interfaces*, pages 1–8, 2001.
- James F. Allen, Myroslava Dzikovska, Mehdi Manshadi, and Mary Swift. Deep linguistic processing for spoken dialogue systems. In *Proceedings of the ACL 2007 Workshop on Deep Linguistic Processing*, pages 49–56, 2007.
- James F. Allen and C. Raymond Perrault. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178, 1980.
- Egbert Ammicht, Alexandros Potamianos, and Eric Fosler-Lussier. Ambiguity representation and resolution in spoken dialogue systems. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2217–2220, 2001.
- Paolo Baggia and Claudio Rullent. Partial parsing as a robust parsing strategy. In *Proceedings of the 18th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 123–126, 1993.

- Frédéric Béchet, Allen L. Gorin, Jeremy H. Wright, and Dilek Hakkani-Tür. Detecting and extracting named entities from spontaneous speech in a mixed-initiative spoken dialogue context: How May I Help You? *Speech Communication*, 42:207–225, 2004.
- Daniel G. Bobrow, Ronald M. Kaplan, Martin Kay, Donald A. Norman, Henry Thompson, and Terry Winograd. GUS, a frame driven dialog system. *Artificial Intelligence*, 8:155–173, 1977.
- Dan Bohus. *Error Awareness and Recovery in Conversational Spoken Language Interfaces*. PhD thesis, Carnegie Mellon University, Pittsburgh, 2007.
- Dan Bohus and Alexander Rudnicky. A K hypotheses + other belief updating model. In *Proceedings of the AAAI Workshop on Statistical and Empirical Approaches to Spoken Dialogue Systems*, 2006.
- Johan Bos, Ewan Klein, Oliver Lemon, and Tetsushi Oka. DIPPER: Description and formalisation of an information-state update dialogue system architecture. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue (SIGdial)*, pages 115–124, 2003.
- Gies Bouwman, Janienke Sturm, and Lou Boves. Incorporating confidence measures in the Dutch train timetable information system developed in the ARISE project. In *Proceedings of the 24th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 493–496, 1999.
- Sandra Carberry. *Plan Recognition in Natural Language Dialogue*. MIT Press, 1990.
- Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Jennifer Chu-Carroll. MIMIC: An adaptive mixed initiative spoken dialogue system for information queries. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 97–104, 2000.
- Jennifer Chu-Carroll and Sandra Carberry. Collaborative response generation in planning dialogues. *Computational Linguistics*, 24(3):355–400, 1998.

- Jennifer Chu-Carroll and Bob Carpenter. Vector-based natural language call routing. *Computational Linguistics*, 25(3):361–388, 1999.
- Tobias Cincarek, Izumi Shindo, Tomoki Toda, Hiroshi Saruwatari, and Kiyohiro Shikano. Development of preschool children subsystem for ASR and Q&A in a real-environment speech-oriented guidance task. In *Proceedings of the INTERSPEECH*, pages 1469–1472, 2007.
- Philip Clarkson and Ronald Rosenfeld. Statistical language modeling using the CMU-Cambridge toolkit. In *Proceedings of the 5th European Conference on Speech Communication and Technology (EUROSPEECH)*, 1997.
- Anna Corazza, Renato De Mori, Roberto Gretter, and Giorgio Satta. Computation of probabilities for an island-driven parser. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):936–950, 1991.
- Marc G. Core and Lenhart K. Schubert. Dialog parsing in the TRAINS system. Technical Report 612, Computer Science Department, The University of Rochester, 1996.
- Hoa Trang Dang. Overview of DUC 2006. In *Proceedings of the 6th Document Understanding Conference (DUC)*, 2006.
- Hoa Trang Dang, Jimmy Lin, and Diane Kelly. Overview of the TREC 2007 question answering track. In *Proceedings of the 16th Text REtrieval Conference (TREC)*, 2007.
- Kohji Dohsaka, Norihito Yasuda, and Kiyoaki Aikawa. Efficient spoken dialogue control depending on the speech recognition rate and system’s database. In *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 657–660, 2003.
- Suzanne Eggins and Diana Slade. *Analyzing Casual Conversation*. Continuum Intl Pub Group, 1997.
- Taku Endo, Nigel Ward, and Minoru Terada. Can confidence scores help users post-editing speech recognizer output? In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, pages 1469–1472, 2002.

- Junlan Feng, Srihari Reddy, and Murat Saraçlar. Webtalk: mining websites for interactively answering questions. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2485–2488, 2005.
- Edward A. Filisko. A context resolution server for the galaxy conversational systems. Master’s thesis, Massachusetts Institute of Technology, 2002.
- B. J. Fogg. *Persuasive Technology*. Morgan Kaufmann Publishers, 2003.
- JT Foote, SJ Young, GJF Jones, and K. Spärck Jones. Unconstrained keyword spotting using phone lattices with application to spoken document retrieval. *Computer Speech and Language*, 11:207–224, 1997.
- Kate Forbes-Riley and Diane J. Litman. Using bigrams to identify relationships between student certainness states and tutor responses in a spoken dialogue corpus. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue (SIGdial)*, pages 87–96, 2005.
- Larry Gillick and Stephen Cox. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the 14th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 532–535, 1989.
- James Glass and Timothy J. Hazen. Telephone-Based Conversational Speech Recognition in the JUPITER Domain. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, 1998.
- James Glass, Joseph Polifroni, Stephanie Seneff, and Victor Zue. Data collection and performance evaluation of spoken dialogue systems: The MIT experience. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP)*, pages IV:1–4, 2000.
- James Glass and Eugene Weinstein. SPEECHBUILDER: Facilitating spoken dialogue system development. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1335–1338, 2001.

- David Goddeau, Eric Brill, James Glass, Christine Pao, Michael Phillips, Joseph Polifroni, Stephanie Seneff, and Victor Zue. GALAXY: A human-language interface to on-line travel information. In *Proceedings of the 3rd International Conference on Spoken Language Processing (ICSLP)*, pages 707–710, 1994.
- David Goddeau, Helen Meng, Joseph Polifroni, Stephanie Seneff, and Senis Busayapongchai. A form-based dialogue manager for spoken language applications. In *Proceedings of the 4th International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 701–704, 1996.
- John J. Godfrey, Edward E Holliman, and Jane McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 27th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 517–520, 1992.
- Allen L. Gorin, Giuseppe Riccardi, and Jeremy H. Wright. How may I help you? *Speech Communication*, 23(1–2):113–127, 1997.
- Herbert Paul Grice. Logic and conversation. In P. Cole and J. Morgan, editors, *Syntax and Semantics 3: Speech Acts*, pages 41–58. New York: Academic Press, 1975.
- Kadri Hacioglu and Wayne Ward. A concept graph based confidence measure. In *Proceedings of the 27th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 225–228, 2002.
- Timothy J. Hazen, Stephanie Seneff, and Joseph Polifroni. Recognition confidence scoring and its use in speech understanding systems. *Computer Speech and Language*, 16:49–67, January 2002.
- Paul Heisterkamp. Linguatronic: Product-level speech system for Mercedes-Benz car. In *Proceedings of the 1st International Conference on Human Language Technology Research (HLT)*, pages 1–2, 2001.
- Ryuichiro Higashinaka and Hideki Isozaki. Corpus-based question answering for why-questions. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP)*, volume 1, pages 418–425, 2008.

- Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. Evaluating discourse understanding in spoken dialogue systems. In *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1941–1944, 2003a.
- Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. Evaluating discourse understanding in spoken dialogue systems. *ACM Transactions on Speech and Language Processing*, 1:1–20, 2004.
- Ryuichiro Higashinaka, Mikio Nakano, and Kiyoaki Aikawa. Corpus-based discourse understanding in spoken dialogue systems. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 240–247, 2003b.
- Ryuichiro Higashinaka, Katsuhito Sudoh, and Mikio Nakano. Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems. *Speech Communication*, 48(3–4):417–436, 2006.
- Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. Extracting important sentences with support vector machines. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 342–348, 2002.
- Julia Hirschberg, Diane Litman, and Marc Swerts. Prosodic and other cues to speech recognition failures. *Speech Communication*, 43:155–175, 2004.
- Jing Huang, Geoffrey Zweig, and Mukund Padmanabhan. Information extraction from voicemail. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 290–297, 2001.
- Ryu Iida, Kentaro Inui, and Yuji Matsumoto. Zero-anaphora resolution by learning rich syntactic pattern features. *ACM Transactions on Asian Language Information Processing (TALIP)*, 6(4, Article 12), 2007.
- Satoshi Ikeda, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Topic estimation with domain extensibility for guiding user’s out-of-grammar utterance in multi-domain spoken dialogue systems. In *Proceedings of the INTERSPEECH*, pages 2561–2564, 2007.

- Hideki Isozaki. An analysis of a high-performance Japanese question answering system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(3):263–279, 2005.
- Hideki Isozaki and Tsutomu Hirao. Japanese zero pronoun resolution based on ranking rules and machine learning. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 184–191, 2003.
- Daniel Jurafsky, Elizabeth Shriberg, and Debra Biasca. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13, 1997. <http://www.stanford.edu/~jurafsky/ws97/manual.august1.html>.
- Kazunori Komatani and Tatsuya Kawahara. Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, volume 1, pages 467–473, 2000.
- Roland Kuhn and Renato De Mori. The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):449–460, 1995.
- Shalom Lappin and Herbert J. Leass. An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561, 1994.
- Steffan Larsson and David R. Traum. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering*, 6:323–340, 2000.
- Akinobu Lee, Tatsuya Kawahara, and Kiyohiro Shikano. Julius – an open source real-time large vocabulary recognition engine. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1691–1694, 2001.
- Yi-Chung Lin and Huei-Ming Wang. Probabilistic concept verification for language understanding in spoken dialogue systems. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1049–1052, 2001.

- Richard P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22:1–16, 1997.
- Diane. J. Litman and James. F. Allen. Discourse processing and common-sense plans. In Philip R. Cohen, Jerry Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, pages 365–388. MIT Press, Cambridge, Mass., 1990.
- Diane J. Litman and Scott Silliman. ITSPOKE: An intelligent tutoring spoken dialogue system. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 5–8, 2004. (Demonstration paper).
- Klaus Macherey, Franz Josef Och, and Hermann Ney. Natural language understanding using statistical machine translation. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2205–2208, 2001.
- William Mann and Sandra Thompson. Rhetorical structure theory: Toward a functional theory of text organization. In *Text*, volume 8, pages 243–281, 1988.
- Daniel Marcu and Abdessamad Echihabi. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 368–375, 2002.
- Teruhisa Misu and Tatsuya Kawahara. Speech-based interactive information guidance system using question-answering technique. In *Proceedings of the 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 145–148, 2007.
- Noboru Miyazaki, Mikio Nakano, , and Kiyooki Aikawa. Spoken dialogue understanding using and incremental speech understanding method. *Systems and Computers in Japan*, 36(12):75–84, 2005.
- Renato De Mori. Spoken language understanding: a survey. In *Proceedings of the 10th biannual IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 365–376., 2007.

- Katashi Nagao. Semantic interpretation based on the multi-world model. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1467–1472, 1989.
- Masaaki Nagata and Tsuyoshi Morimoto. First steps toward statistical modeling of dialogue to predict the speech act type of the next utterance. *Speech Communication*, 15:193–203, 1994.
- Mikio Nakano, Yasuhiro Minami, Stephanie Seneff, Timothy J. Hazen, D. Scott Cyphers, James Glass, Joseph Polifroni, and Victor Zue. Mokusei: A telephone-based Japanese conversational system in the weather domain. In *Proceedings of the 7th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1331–1334, 2001.
- Mikio Nakano, Noboru Miyazaki, Jun-ichi Hirasawa, Kohji Dohsaka, and Takeshi Kawabata. Understanding unsegmented user utterances in real-time spoken dialogue systems. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 200–207, 1999.
- Mikio Nakano, Noboru Miyazaki, Norihito Yasuda, Akira Sugiyama, Jun-ichi Hirasawa, Kohji Dohsaka, and Kiyoaki Aikawa. WIT: A toolkit for building robust and real-time spoken dialogue systems. In *Proceedings of the 1st SIGdial Workshop on Discourse and Dialogue (SIGdial)*, pages 150–159, 2000.
- Rieks op den Akker, Marjan Hospers, Erna Kroezen, Anton Nijholt, and Danny Lie. A rule-based reference resolution method for Dutch discourse analysis. In *Proceedings of the 2002 International Symposium on Reference Resolution for Natural Language Processing*, 2002.
- Bryan Pellom, Wayne Ward, and Sameer Pradhan. The CU Communicator: an architecture for dialogue systems. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 723–726., 2000.
- Roberto Pieraccini and David Lubensky. Spoken language communication with machines: The long and winding road from research to business. In *Proceedings*

of the 18th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/AIE), pages 6–15, 2005.

Alexandros Potamianos and Hong-Kwang J. Kuo. Statistical recursive finite state machine parsing for speech understanding. In *Proceedings of the 6th International Conference on Spoken Language Processing (ICSLP)*, volume 3, pages 510–513, 2000.

Michael. J. D. Powell. An efficient method of finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7 (2):155–162, 1964.

Sameer S. Pradhan and Wayne H. Ward. Estimating semantic confidence for spoken dialog systems. In *Proceedings of the 27th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 233–236, 2002.

Mazin G. Rahim, Chin-Hui Lee, and Biing-Hwang Juang. Discriminative utterance verification for connected digits recognition. *IEEE Transactions on Speech and Audio Processing*, 5(3):266–277, 1997.

Norbert Reithinger and Elisabeth Maier. Utilizing statistical dialogue act processing in Verbmobil. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 116–121, 1995.

Charles Rich, Candace Sidner, and Neal Lesh. COLLAGEN: Applying collaborative discourse theory. *AI Magazine*, 22(4):15–25, 2001.

Stephanie Seneff. Robust parsing for spoken language systems. In *Proceedings of the 17th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 23–26, 1992.

Stephanie Seneff. Response planning and generation in the MERCURY flight reservation system. *Computer Speech and Language*, 16(3–4):283–312, 2002.

Yusuke Shinyama, Takenobu Tokunaga, and Hozumi Tanaka. Kairai - software robots understanding natural language. In *Proceedings of the 3rd International Workshop on Human-Computer Conversation*, pages 158–163, 2000.

- Satinder Singh, Diane Litman, Michael Kearns, and Marilyn Walker. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Journal of Artificial Intelligence Research*, 16:105–133, 2002.
- Alex J. Smola and Bernhard Scholköpfung. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.
- Radu Soricut and Daniel Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, volume 1, pages 149–156, 2003.
- Randall Sparks, Lori Meiskey, and Hans Brunner. An object-oriented approach to dialogue management in spoken language systems. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pages 211–217, 1994.
- Satoshi Takano, Kimihito Tanaka, Hideyuki Mizuno, Masanobu Abe, and ShiN’ya Nakajima. A Japanese TTS system based on multi-form units and a speech modification algorithm with harmonics reconstruction. *IEEE Transactions on Speech and Audio Processing*, 9(1):3–10, 2001.
- David R. Traum and Steffan Larsson. *Information State Approach*. Kluwer Academic Publishers, 2003.
- Boris van Schooten, Sophie Rosset, Olivier Galibert, Aurélien Max, Rieks op den Akker, and Gabriel Illouz. Handling speech input in the Ritel QA dialogue system. In *Proceedings of the INTERSPEECH*, pages 126–129, 2007.
- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- Marilyn Walker, Candance Kamm, and Diane Litman. Towards developing general models of usability with PARADISE. *Natural Language Engineering: Special Issue on Best Practice in Spoken Dialogue Systems.*, 6:363–377, 2000.
- Marilyn Walker, Alex Rudnický, John Aberdeen, Elizabet Bratt, John Garofolo, Helen Hastie, Audrey Le, Bryan Pellom, Alex Potamianos, Rebecca Passonneau, Rashmi Prasad, Salim Roukos, Gregory Sanders, Stephanie Seneff, and

- Davi Stallard. DARPA Communicator evaluation: Progress from 2000 to 2001. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, pages 273–276, 2002.
- Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 271–280, 1997.
- Frank Wessel, Ralf Schlüter, Klaus Macherey, and Hermann Ney. Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3):288–298, 2001.
- Jason D. Williams and Steve Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422, 2007.
- Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.
- Chai Wutiwiwatchai and Sadaoki Furui. Belief-based nonlinear rescoring in Thai speech understanding. In *Proceedings of the INTERSPEECH*, pages 2129–2133, 2004.
- Yong Wang and Ian H. Witten. Inducing model trees for predicting continuous classes. In *Proceedings of the 9th European Conference on Machine Learning*, pages 128–137, 1997.
- Victor Zue, Stephanie Seneff, Joseph Polifroni, Mikio Nakano, Yasuhiro Minami, Timothy Hazen, and James Glass. From JUPITER to MOKUSEI: Multilingual conversational systems in the weather domain. In *Proceedings of the Workshop on Multilingual Speech Communications (MSC)*, pages 1–6, 2000.

List of Publications

Journal Articles

1. Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. Evaluating discourse understanding in spoken dialogue systems. *ACM Transactions on Speech and Language Processing*, 1:1–20, 2004.
2. Ryuichiro Higashinaka, Katsuhito Sudoh, and Mikio Nakano. Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems. *Speech Communication*, 48(3–4):417–436, 2006.
3. Ryuichiro Higashinaka, Marilyn Walker, and Rashmi Prasad. An unsupervised method for learning generation dictionaries for spoken dialogue systems by mining user reviews. *ACM Transactions on Speech and Language Processing*, 4(4, Article 8, 25 pages), 2007.

Conference Papers

1. Ryuichiro Higashinaka and Katashi Nagao. Interactive paraphrasing based on linguistic annotation. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 1218–1222, 2002.
2. Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. A method for evaluating incremental utterance understanding in spoken dialogue systems. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, pages 829–832, 2002.

3. Ryuichiro Higashinaka, Mikio Nakano, and Kiyoaki Aikawa. Corpus-based discourse understanding in spoken dialogue systems. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 240–247, 2003.
4. Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. Evaluating discourse understanding in spoken dialogue systems. In *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 1941–1944, 2003.
5. Ryuichiro Higashinaka, Katsuhito Sudoh, and Mikio Nakano. Incorporating discourse features into confidence scoring of intention recognition results in spoken dialogue systems. In *Proceedings of the 30th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 25–28, 2005.
6. Ryuichiro Higashinaka, Rashmi Prasad, and Marilyn Walker. Augmenting variation of system utterances using corpora in spoken dialogue systems. In *Proceedings of the 9th biannual IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 262–267, 2005.
7. Ryuichiro Higashinaka, Rashmi Prasad, and Marilyn Walker. Learning to generate naturalistic utterances using reviews in spoken dialogue systems. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 265–272, 2006.
8. Ryuichiro Higashinaka and Hideki Isozaki. NTT’s question answering system for NTCIR-6 QAC-4. In *Proceedings of the 6th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access (NTCIR)*, pages 460–463, 2007.
9. Ryuichiro Higashinaka, Kohji Dohsaka, and Hideki Isozaki. Learning to rank definitions to generate quizzes for interactive information presentation. In *Proceedings of the the 45th Annual Meeting of the Association for*

- Computational Linguistics (ACL)*, pages 117–120, 2007. (Poster Presentation).
10. Ryuichiro Higashinaka, Kohji Dohsaka, Shigeaki Amano, and Hideki Isozaki. Effects of quiz-style information presentation on user understanding. In *Proceedings of the INTERSPEECH*, pages 2725–2728, 2007.
 11. Ryuichiro Higashinaka and Hideki Isozaki. Corpus-based question answering for why-questions. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP)*, volume 1, pages 418–425, 2008.
 12. Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyoaki Aikawa. Learning decision trees to determine turn-taking by spoken dialogue systems. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, pages 861–864, 2002.
 13. Emma Barker, Ryuichiro Higashinaka, François Mairesse, Robert Gaizauskas, Marilyn Walker, and Jonathan Foster. Simulating Cub Reporter dialogues: The collection of naturalistic human-human dialogues for information access to text archives. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC)*, 2006.
 14. Yasuhiro Minami, Minako Sawaki, Kohji Dohsaka, Ryuichiro Higashinaka, Kentaro Ishizuka Hideki Isozaki, Tatsushi Matsubayashi, Masato Miyoshi, Atsushi Nakamura, Takanobu Oba, Hiroshi Sawada, Takeshi Yamada, and Eisaku Maeda. The world of Mushrooms: Human-computer interaction prototype systems for ambient intelligence. In *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI)*, pages 366–373, 2007.
 15. Minako Sawaki, Yasuhiro Minami, Ryuichiro Higashinaka, Kohji Dohsaka, Takeshi Yamada Tatsushi Matsubayashi, Hideki Isozaki, and Eisaku Maeda. Quizmaster Mushrooms: ‘Who is this’ quiz dialogue system. In *Proceedings of the 9th International Conference on Multimodal Interfaces (ICMI)*, 2007. (Demonstration).

16. Minako Sawaki, Yasuhiro Minami, Ryuichiro Higashinaka, Kohji Dohsaka, Takeshi Yamada Tatsushi Matsubayashi, Hideki Isozaki, and Eisaku Maeda. Quizmaster Mushrooms: ‘Who is this’ quiz dialogue system. In *Proceedings of the 10th biannual IEEE workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2007. (Demonstration).

Other Publications

1. Ryuichiro Higashinaka and Katashi Nagao. Intelligent document modification based on external annotations. Technical Report SIG-ICS-120, Information Processing Society of Japan, 2000. (In Japanese).
2. Ryuichiro Higashinaka and Katashi Nagao. Methods to facilitate web browsing using annotations. In *Proceedings of the 3rd Workshop on Internet Technology (WIT)*, 2000. (In Japanese).
3. Ryuichiro Higashinaka and Katashi Nagao. Interactive paraphrasing of on-line documents. In *Proceedings of the 7th Annual Meeting of the Association for Natural Language Processing*, 2001. (In Japanese).
4. Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyooki Aikawa. A method for evaluating incremental utterance understanding in spoken dialogue systems. Technical Report SIG-NL-149, Information Processing Society of Japan, 2002. (In Japanese).
5. Ryuichiro Higashinaka and Takehiko Ohno. Estimating document understanding based on eye movements. Technical Report SIG-HI-102, Information Processing Society of Japan, 2003. (In Japanese).
6. Ryuichiro Higashinaka, Mikio Nakano, and Kiyooki Aikawa. A statistical discourse understanding method for multiple-context-based spoken dialogue systems. Technical Report SIG-SLP-45, Information Processing Society of Japan, 2003. (In Japanese).
7. Ryuichiro Higashinaka, Mikio Nakano, and Kiyooki Aikawa. A statistical discourse understanding method for multiple-context-based spoken dialogue

- systems. In *Proceedings of the 7th Annual Meeting of the Association for Natural Language Processing*, pages 55–58, 2003. (In Japanese).
8. Ryuichiro Higashinaka and Takehiko Ohno. Gaze-based automatic dictionary lookup system. In *Proceedings of the 9th Annual Meeting of the Association for Natural Language Processing*, 2003. (In Japanese).
 9. Ryuichiro Higashinaka and Mikio Nakano. Offline evaluation of discourse understanding in spoken dialogue systems. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing*, pages 63–66, 2004. (In Japanese).
 10. Ryuichiro Higashinaka, Noboru Miyazaki, Mikio Nakano, and Kiyoaki Aikawa. Evaluating discourse understanding in spoken dialogue systems. Poster Presentation at *NTT Workshop on Communication Scene Analysis (CSA)*, 2004.
 11. Ryo Sato, Ryuichiro Higashinaka, Masafumi Tamoto, Mikio Nakano, and Kiyoaki Aikawa. Learning decision trees to determine turn-taking by spoken dialogue systems. Technical Report SIG-NL-149, Information Processing Society of Japan, 2002. (In Japanese).
 12. Katsuhito Sudoh, Ryuichiro Higashinaka, Mikio Nakano, and Kiyoaki Aikawa. Improving statistical language models using untranscribed dialogue data based on review-based confidence scoring. Technical Report SIG-SLP-45, Information Processing Society of Japan, 2003. (In Japanese).
 13. Mikio Nakano, Ryuichiro Higashinaka, Matthias Denecke, Katsuhito Sudoh, Noboru Miyazaki, and Kohji Dohsaka. A framework for interactive speech understanding systems that are trainable using dialogue corpora. In *Proceedings of the 10th Annual Meeting of the Association for Natural Language Processing*, pages 67–70, 2004. (In Japanese).
 14. Katashi Nagao and Ryuichiro Higashinaka. Wordlogue: Weblog classification and retrieval using data on word frequency and annotation. In *Proceedings of the 68th IPSJ Annual Conventions*, 2006. (In Japanese).

15. Eisaku Maeda, Yasuhiro Minami, Masato Miyoshi, Minako Sawaki, Hiroshi Sawada, Atsushi Nakamura, Junji Yamato, Takeshi Yamada, and Ryuichiro Higashinaka. The World of Mushrooms-transdisciplinary approach to human-computer interaction with ambient intelligence. *NTT Technical Review*, 4(12):17–25, 2006.
16. Yasuhiro Minami, Ryuichiro Higashinaka, Minako Sawaki, Kohji Dohsaka, Tatsushi Matsubayashi, Takeshi Yamada, Hideki Isozaki, and Eisaku Maeda. A quiz dialogue system evaluated in speech recognition. In *Proceedings of the Autumn Meeting of the Acoustical Society of Japan*, 2007. (In Japanese).