



Automatic Instant Messaging Dialogue using Statistical Models and Dialogue Acts

A thesis presented

by

Edward Ivanovic

to

The Department of Computer Science and Software Engineering

in total fulfillment of the requirements

for the degree of

Master of Computer Science and Software Engineering by Research

University of Melbourne

Melbourne, Australia

July 2008

©2008 - Edward Ivanovic

All rights reserved.

Automatic Instant Messaging Dialogue using Statistical Models and Dialogue Acts

Abstract

Instant messaging dialogue is used for communication by hundreds of millions of people worldwide, but has received relatively little attention in computational linguistics. We describe methods aimed at providing a shallow interpretation of messages sent via instant messaging. This is done by assigning labels known as *dialogue acts* to utterances within messages. Since messages may contain more than one utterance, we explore automatic message segmentation using combinations of parse trees and various statistical models to achieve high accuracy for both classification and segmentation tasks.

Finally, we gauge the immediate usefulness of dialogue acts in conversation management by presenting a dialogue simulation program that uses dialogue acts to predict utterances during a conversation. The predictions are evaluated via qualitative means where we obtain very encouraging results.

Declaration

This is to certify that:

1. the thesis comprises only my original work towards the Masters except where indicated in the Preface,
2. due acknowledgement has been made in the text to all other material used,
3. the thesis is approximately 30,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Signed,

Edward Ivanovic

Contents

Title Page	i
Abstract	iii
Table of Contents	v
List of Figures	viii
List of Tables	ix
Citations to Previously Published Work	xi
Acknowledgments	xii
1 Introduction	1
1.1 Background	3
1.2 Motivation	4
1.3 Goals	5
1.4 Methodological Outline	6
1.5 Thesis overview	8
2 Instant Messaging	10
2.1 Historical Development of Instant Messaging	10
2.2 Instant Messaging Clients	12
2.2.1 AOL's Instant Messenger	12
2.2.2 MSN Messenger	14
2.3 Commercial aspects of free Instant Messaging	16
2.3.1 Forecasted Trends in Instant Messaging Usage	18
2.4 Linguistic Features of Instant Messaging	21
2.5 Conclusion	27
3 Human-Computer Dialogue	29
3.1 Dialogue Management Techniques	29
3.2 Additional research areas for realistic dialogue	34
3.3 Differences between Dialogue and Monologue	36
3.3.1 Conversational Implicature	36
3.3.2 Grounding	38
3.3.3 Turns, Messages, and Utterances	43

3.4	Dialogue Acts	47
3.5	Dialogue Act Annotation Schemes	49
3.5.1	Dialogue Act Markup in Several Layers	49
3.5.2	Modifications and Extensions of DAMSL	55
3.6	Prior Work on Dialogue Act Classification	55
3.6.1	Selected Statistical Learning Methods	56
3.6.2	Selected Symbolic Learning Methods	58
3.7	Summary	60
4	Construction of an IM dialogue corpus	63
4.1	Dialogue Act Tag Set	64
4.2	Data Collection	67
4.3	Data Annotation	69
4.4	Data Preparation	69
4.4.1	Message Synchronisation	69
4.4.2	Utterance Segmentation	72
4.4.3	Lemmatisation	78
4.4.4	PoS Tagging and Chunking	80
4.5	Conclusion	83
5	Utterance Segmentation and Dialogue Act Classification	84
5.1	HMM Utterance Segmentation Method	85
5.2	Dialogue Act Classification	87
5.2.1	Naive Bayes Model	89
5.2.2	Vector Space Model	90
5.2.3	Support Vector Machine Model	92
5.2.4	Maximum Entropy Model	93
5.3	Segmentation and Classification using Parse Trees	95
5.4	Conclusion	98
6	Evaluation	100
6.1	Dialogue Act Classification	101
6.1.1	Naive Bayes Model	102
6.1.2	Error Analysis of Classification Task	106
6.2	Utterance Segmentation	112
6.2.1	Evaluating Utterance Segmentation	112
6.2.2	Experimental Results and Discussion	115
6.3	An Application: Assisted Customer Support	120
6.3.1	Using the Simulation Program	122
6.3.2	Evaluating the Customer Support Assistant	124
6.3.3	The Entropy of Utterances within Dialogue Acts	128
6.4	Conclusion	130

7	Conclusions and Future Work	133
7.1	Conclusions	133
7.2	Future Work	135
A	Dialogue Macrogame Theory	148
B	Computer-assisted conversation logs	153
C	Dialogue Act Bigram transition probabilities	174
D	Utterance Classifications with Dialogue Acts	177
E	Online Chat Support Services	180
E.1	Online Chat Support	180
E.2	Online Chat Support Software and Services	180
F	Penn Treebank Part of Speech tag set	182

List of Figures

2.1	AOL Instant Messenger (AIM) screen shot.	13
2.2	MSN Messenger screen shot.	15
3.1	Finite-state automaton	30
3.2	Tree diagram of instant messaging constituents	44
3.3	Decision tree for Agreement aspect in DAMSL	50
4.1	Example lemmatised sentence.	79
4.2	Sample PoS-tagged and chunked data from corpus	82
5.1	Sample training data used for HMM segmentation	86
5.2	Feature representation mapped from corpus data	88
5.3	RASP parse tree of a message showing utterances separated into sub-trees.	95
5.4	Proper analyses of a parse tree	96
6.1	Learning curve of naive Bayes models	106
6.2	Relation between word entropy and dialogue act frequencies	111
6.3	Reference and three hypothesised segmentations for different algorithms.	113
6.4	Frequency distribution of utterance and message lengths in words.	114
6.5	WindowDiff results of various models used	116
6.6	Erroneous parse tree produced by RASP	119
6.7	Parse tree produced by RASP	120
6.8	Screen shot of the dialogue simulation program	121

List of Tables

1.1	An example of the beginning of a dialogue in our corpus.	2
2.1	Networking Timeline	19
2.2	Sample of a many-to-one relationship between requests and a response.	20
2.3	Distinctions between Written and Spoken Language	23
2.4	Computer Mediated Communication Spectrum	24
2.5	Common abbreviations and emoticons used in instant messaging . . .	26
3.1	Examples of different types of dialogue systems	31
3.2	Different techniques for dialogue given task complexity	32
3.3	Grounding acts for discourse units	41
3.4	Example of unsynchronised messages in instant messaging	46
3.5	Forward- and backward-looking functions in DAMSL	54
3.6	A selection of studies presented in Stolcke <i>et al.</i> (2000) showing the number of dialogue act tokens, dialogue act tag set size, tag set used, and accuracy.	57
4.1	The 12 dialogue act labels with a description and examples of each. .	65
4.2	Example of the beginning of a dialogue in our corpus	68
4.3	Characteristics of the MSN-Shopping corpus.	68
4.4	Example of unsynchronised messages	70
4.5	Method of scoring pairwise inter-annotator agreement	76
4.6	Annotator segmentation agreement	78
5.1	Hypothesised segmentations from sub-tree node combinations	97
6.1	Results for all classification algorithms	102
6.2	Mean n -gram accuracy of labelling utterances with dialogue acts . . .	103
6.3	Example of a dialogue with utterance boundaries	104
6.4	Relative misclassification error rates	107
6.5	Perplexity values for dialogue act transitions using bigrams	108
6.6	Errors showing the impact of dialogue act adjacency pairs	109

6.7	Example from the dialogue simulation program	125
6.8	Dialogue act rankings within the set of predictions by running the shopping evaluation program. n is the n -best rankings of correct suggestions. Percentages are cumulative.	127
6.9	Manually clustered utterances to calculate entropy	129
6.10	Number of utterance clusters with entropy values	130
C.1	Bigram dialogue act transition probabilities.	176

Citations to Previously Published Work

Portions of this thesis have appeared in the following papers:

Edward Ivanovic. 2005 Dialogue act tagging for instant messaging chat sessions. In *Proceedings of the ACL Student Research Workshop*, pages 79–84, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Edward Ivanovic. 2005 Automatic Utterance Segmentation in Instant Messaging Dialogue. In *Proceedings of the Australasian Language Technology Workshop*, pages 241–249, Sydney, Australia, December. Australasian Language Technology Association.

Acknowledgments

I have benefited greatly from the support and help of many people during my time spent researching and writing this thesis. In particular, I would like to express my gratitude to Steven Bird, my first supervisor, who gave me the opportunity to study computational linguistics and provided inspiration and encouragement along the way. Without his guidance, I may not have been enlightened by so many fascinating theories within and beyond computational linguistics. I am also very grateful to Timothy Baldwin, my second supervisor, who was not only very patient and generally made my research more enjoyable, but offered to get his hands dirty by parsing the data (several times) that I used in this thesis. Further, both my supervisors and Lawrence Cavedon have reviewed various drafts of this thesis, for which I am extremely appreciative.

I would like to thank my colleagues, who gave me constructive feedback during our Writers Anonymous meetings (which were not really anonymous). Beyond these meetings, I would particularly like to thank Trevor Cohn, Phil Blunsom, and Nicola Stokes for their stimulating conversations and thoughtful comments on my work.

Several people kindly volunteered to do the laborious task of annotating the data used in this thesis; I am grateful to Paul Bongiorno, Marija Ivanovic, Jeremy Nicholson, Vanessa Smith, and Patrick Ye for their hard work.

More generally, I would like to thank my family and friends who have been supportive in so many ways, especially my sister, Marija; Paul Bongiorno for being a good friend and dragging me out of my apartment to occasionally collide with the world; and, of course, Mum for never failing to remind me to visit home for Sunday roast, lest I forget to eat.

Chapter 1

Introduction

Support services in many domains have traditionally been provided over the telephone: when customers have queries, they dial a support number and speak to a support representative. Recent developments, however, have seen support services additionally being provided over the Internet. Many companies have web sites with Frequently Asked Questions (FAQs), and also offer e-mail support. More recently, real-time support via *online chat sessions* is being offered where customers and support representatives type short messages to each other. Appendix E contains a selection of companies that provide online chat support and companies that provide the software for online chat support.

Chat sessions are conducted over a network, such as the Internet, where textual messages can be sent and received between interlocutors in real-time. Due to their real-time nature, these chat sessions are commonly referred to as *instant messaging*.

Instant Messaging is a relatively new form of communication made possible by the Internet. Hundreds of millions of people around the world currently use instant messaging to communicate, sending billions of messages per day. 1.6 billion messages are sent daily from AOL's Instant Messenger alone—just one of the instant messaging services available.¹

Support services that are conducted via instant messaging vary from being person-person dialogue, similar to traditional call centres, through to being entirely auto-

¹Source: http://www.corp.aol.com/products/brands_aim.shtml (November 11, 2005).

Speaker	Message
Agent	Hello Customer, thank you for contacting MSN Shopping. This is Sanders and I look forward to assisting you today
Agent	How are you doing today?
Customer	good, thanks
Agent	Please let me know how I can assist you on MSN Shopping today.
Customer	I'd like to buy a gift for my sister - it's her birthday in 2 weeks but I'm not sure what to get her
Agent	No problem, Customer.
Agent	Please be assured that I will do my best to assist you in finding a nice Gift for your sister.
Customer	thanks
Agent	You are welcome.
Agent	Could you please let me know her interests and hobbies so that I can look for an appropriate gift for her?

Table 1.1: An example of the beginning of a dialogue in our corpus.

mated where customers engage in dialogue with a computer program. Some support services are partially automated by suggesting responses to a human agent, which may then be accepted or overwritten.

The research we present in this thesis aims to provide a degree of natural language understanding to assist in automating task-oriented dialogue, as can be found in support services. In line with advances in statistical modelling and natural language understanding, we apply various probabilistic and linguistic methods to improve discourse modelling in the support services domain. We collected a small corpus consisting of a set of dialogues between customers and support representatives from the MSN Shopping online support service. Potential customers using this service are assisted with finding various items for sale on the MSN Shopping web site. A sample from one of the dialogues in this corpus is shown in Table 1.1.

This introductory chapter describes the domain and purpose of the research pre-

sented in this thesis. Section 1.1 provides some definitions and introduces dialogue acts and utterances. In order to motivate this research, Section 1.2 gives an overview and discussion of some common customer support systems. We state our goals further in Section 1.3, our methodology is then outlined in Section 1.4 before presenting an overview of the rest of this thesis in Section 1.5.

1.1 Background

The term *dialogue*, as used in this thesis, refers to the exchange of written messages between two parties. The two parties may be either human-human or human-computer. We limit our study to only two parties because that is how customer support is usually conducted. However, the methods we describe may be useful in dialogue with more than two participants.

The messages that are exchanged during the dialogue are not necessarily sentence-based in the traditional sense; they instead contain one or more *utterances*, which are sometimes called *non-sentential units*. In this thesis, an *utterance* represents a communicative act, called a *speech act* by Searle (1979). These terms are discussed in more detail in Section 3.3.3, but we introduce them here with the following example. The message *ok, I'll try that* contains a sequence of two utterances which are represented by the speech acts *acknowledgement* (*ok*) and *commissive* (*I'll try that*). The *acknowledgement* lets the other party know that their message was received and understood, whereas the *commissive* commits the speaker to a future action.

When used in dialogue, as opposed to monologue, speech acts tend to exhibit adjacency pairs. That is, one particular type of speech act commonly follows another, such as questions are typically followed by answers. Speech acts are thus extended in dialogue to model the *conversational* functions of an utterance. To avoid confusion between the speech acts as used in monologue, the term *dialogue acts* is used to refer to this extended version used in dialogue (Jurafsky and Martin 2000:729).

Dialogue acts are useful as they provide some semantic information about an utterance. If a system reliably tagged utterances in instant messaging with such dialogue acts, downstream tasks could be aided with this information. For example,

a dialogue system needs to know if it was just asked a question or ordered to do something. However, messages received via instant messaging may contain more than one utterance, as demonstrated with the example *ok, I'll try that*, above. If a textual message does indeed contain more than one utterance, that message must be segmented into its utterances before classification can ensue. Section 3.3.3 discusses segmentation in more detail and we address the problem in Section 6.2.

Dialogue models aim to account for the properties observed in dialogue, of which dialogue acts play an important part. We use the term *dialogue modelling* in this thesis to refer to the process of designing a dialogue model, whereas *dialogue management* serves to control and restrict the interaction.

1.2 Motivation

Most companies that provide customer support are continuously trying to balance expensive human support costs, such as call centres, with cheaper methods such as FAQ pages on the World Wide Web and Interactive Voice Response (IVR) telephone systems. From a customer's perspective, the advantages of human support are clear: communication is easy, leading to questions being answered efficiently and problems resolved quickly. However, human support is expensive for many companies, sometimes prohibitively so. The lower cost of providing online resources, such as web sites, is attractive, but such resources require customers to search for their specific problem, which can be a frustrating process and inadequate when questions are not listed or not easily locatable. The desire to reduce costs whilst offering satisfactory support is fuelling significant commercial activity and research in high-quality, automated support services—some companies and products related to this are listed in Appendix E.

Various approaches to providing lower-cost customer support have been developed in recent years: many companies have opted to out-source call centres in countries that offer cheaper labour; IVR systems now commonly support speech recognition in an attempt to increase efficiency and customer satisfaction; web sites provide search facilities to help a user navigate through sometimes hundreds of FAQs. However,

significant problems remain: cheaper off-shore labour is highly dependant on both the local and global economies and is still prohibitively expensive for many companies; speech recognition suffers from low accuracy when dealing with a large range of accents, speakers, and vocabulary; and, FAQs may not contain the right answer or the answer may be buried in long instructions.

Computerised methods must be able to adequately simulate the interaction and understanding provided by humans if they are to offer satisfactory customer support at low cost. Many companies now offer user support via instant messaging environments with both human-human and human-computer dialogue. The current state of the art for human-computer dialogue systems is very basic: customer messages are matched to a pre-programmed list of messages using various text-matching techniques. If a match is found, a hard-coded response is sent to the customer. This method amounts to little more than searching in FAQ pages.

Beyond support services, instant messaging is being adopted as an internal communication mechanism by corporations. According to a recent study by International Data Corporation (IDC), there are more than 28 million *business* users of instant messaging world-wide sending approximately 1 billion messages per day, and instant messaging products are reaching more mainstream users (Mahowald and Levitt 2005). These business users are a subset of the general instant messaging population and the trend in business and mainstream usage growth is expected to continue through to at least 2009. As instant messaging uptake continues, the number of people comfortable with using it for support services will also increase, leading to further demand for automated instant messaging services from companies wanting to reduce support costs. This further motivates the research presented here as we investigate methods that assist in attaining deeper levels of natural language understanding in instant messaging.

1.3 Goals

The purpose of this research is to investigate ways of automatically identifying the semantic information in messages sent via instant messaging. The semantic informa-

tion should be rich enough to significantly assist instant messaging support services. For the purposes of this study, the semantic information is represented by a set of labels such as STATEMENT, THANKING, and OPEN-QUESTION, known as *dialogue acts*. The dialogue acts are assigned to segments of messages, known as *utterances*. Since messages may contain more than one utterance, we also explore automatically segmenting messages into utterances before classifying the utterances into dialogue acts.

A further part of this research is to investigate the use of both linguistic and probabilistic approaches to segmentation. The linguistic approach relies on parse trees of messages from instant messaging.

Only communication pertaining to customer support will be explored in this study, as opposed to social conversations. There is a fundamental difference between social and task-oriented dialogue. Social dialogue seeks to maintain a relationship between the interlocutors, whereas the primary purpose of task-oriented dialogue is to perform a known task.

Task-oriented conversations, such as customer support, tend to exhibit more structured discourse as participants intend to resolve some specific problem or elicit specific information. Socially-based conversations, on the other hand, often consist of less structured, ephemeral utterances that are typically motivated by reasons more subtle than task-based dialogue, on which we elaborate in Chapter 2.

1.4 Methodological Outline

There are two major parts of this study: segmenting messages into utterances, and classifying utterances into the dialogue acts, which represent their semantic content. We hypothesise that the segmentation and classification tasks can be accomplished using a combination of linguistic and statistical methods. This hypothesis is supported in part by previous work such as Stolcke *et al.* (2000) where utterances were automatically classified into dialogue acts using statistical techniques in an attempt to improve on speech recognition accuracy. Our corpus was obtained by holding several conversations with an instant messaging support service, which provided us with

real-world dialogue transcripts.

The motivation for using linguistic methods for segmentation comes from the observation that utterances in our corpus occur at major syntactic boundaries, which parse trees effectively represent as will be described in detail in Section 5.3. Therefore, marking and only searching at syntactic boundaries in a message reduces the search space when performing segmentation, which means there is less chance for error.

To test the accuracy of our segmentation and classification processes, we first produced a gold standard version of our corpus by manually segmenting and classifying utterances into dialogue acts. The dialogue act classification task was performed using several statistical methods, including naive Bayes, support vector machine, vector space, and maximum entropy models, which are all detailed in Section 5.2. The results were evaluated using basic accuracy $\frac{c}{a}$ where c is the number of correct classifications and a is the total number of classifications.

The segmentation task was evaluated using the WindowDiff metric rather than the more typical recall and precision metrics used in information retrieval (Pevzner and Hearst 2002). WindowDiff was selected as it gives partial credit to hypothesised segment boundaries that are near, but not on, reference segment boundaries. The advantage to this is that different segmentation algorithms may be compared under the assumption that the algorithm that is closest to the reference segmentation is the best. Recall and precision, on the other hand, reward hypothesised segmentations if and only if they exactly match the reference segmentation. We describe the WindowDiff metric further in Section 6.2.1.

Finally, we test the hypothesis that the semantic information represented by dialogue acts can assist in customer support services. To do this, we developed simulation software used to conduct conversations with the aim of replicating each of the dialogues in our corpus. During the conversation, the simulation program makes suggested responses to the customer service representative, which may be accepted or overridden. At the end of the simulation, information is presented showing how accurate the suggestions were.

1.5 Thesis overview

Chapter 2 begins with an introduction to Instant Messaging including an historical account of its popularity. It discusses current and forecasted usage patterns and the technological methods employed by commercial services set up to address growing customer support needs.

Chapter 3 goes on to discuss the differences between dialogue and monologue, highlighting various problems that must be overcome to provide a system capable of engaging in realistic dialogue. We describe the use of dialogue-acts as an aid to understanding a discourse in detail followed by a review of efforts to deal with dialogue automation.

Chapter 4 discusses the domain of study, data collection, and methods of data annotation and preparation. This chapter includes the manual segmentation and classification steps to create a gold standard. Part of the preparation includes message synchronisation, which seems to be unique to instant messaging dialogue. *Asynchronous messages* occur when users send messages at the same time rather than taking turns. This sometimes results from a message being mistaken as the end of the other person's turn. Asynchronous messages disrupt the logical flow of a dialogue and potentially result in dialogue act pairs being out of order, which causes problems with statistical discourse models. We tackle this problem as part of the preparation discussed in Section 4.4.

Chapter 5 introduces two models used to segment utterances and a further four models to classify dialogue acts. The segmentation methods begin with a discussion of using a statistical approach to segmentation based on hidden Markov models (HMMs), which achieve high accuracy, but are outperformed by the linguistically-based parse-tree model. The classification task is performed using naive Bayes, vector space, maximum entropy, and support vector machine models.

All of the aforementioned models are evaluated in Chapter 6, where the Window-Diff metric is discussed and presented as the preferred evaluation metric for the segmentation task as it overcomes problems associated with using recall and precision. As part of the evaluation and to demonstrate an example of how dialogue act classifi-

cation can be used to assist customer support, we developed a program that uses the methods presented here to suggest and rank utterances during a simulated dialogue between a customer and a support representative.

We show that the naive Bayes and support vector machine models achieve over 80% accuracy when classifying utterances into dialogue acts. Our comparison of segmentation algorithms shows that the parse-tree model we use, which is linguistically informed, outperforms the HMM-based models.

The conclusions and discussion for future research are drawn in Chapter 7, where we highlight the need for careful knowledge engineering in the early stages of a project such as this. In particular, we discuss the importance of identifying dialogue acts to create an appropriate tag set, which influences the accuracy obtained and usefulness of the resulting semantic tags.

Chapter 2

Instant Messaging

This chapter sets the context for instant messaging and charts its development, rise, and forecasted trends. The discussion begins with the development of early instant messaging systems within local and wide area networks. The advent of the Internet is then discussed and we explain how it allowed instant messaging to proliferate so that hundreds of millions of users world-wide currently use it.

2.1 Historical Development of Instant Messaging

Instant messaging was used in the early 1970s in the program *Term Talk*, which was part of a computer-assisted teaching system called PLATO, an acronym for *Programmed Logic for Automatic Teaching Operation*. *Term Talk* allowed users at terminals to send text messages to each other in real time via a common server. A decade later, in the 1980s, *talk* was developed for Unix systems allowing wide access instant messaging for Unix users connected to the same network, which at that time was largely university- and research-based. By 1984 the Internet emerged from the combination of various networks, including NSFNet (the National Science Foundation's university backbone), Usenet (a computer network established in 1980 primarily supporting newsgroups), Bitnet (a computer network of educational institutions), and others, while being made available to commercial interests.

The World Wide Web (WWW) was developed in 1990 to provide a way to access

and manage information on the Internet. The WWW allowed new information to be published and made accessible by anybody with an Internet connection and a web browser, which provided the incentive for home users to connect to the Internet, extending the user base far beyond university and government users. The increasing prevalence of the WWW, and therefore the Internet, provided the communications framework for other services, such as instant messaging, to become a practical and convenient method of communication.

The amalgamation of the disparate networks which together formed the Internet required that some software be modified or replaced to work on the new, larger network. One such software program was MUT, for *MultiUser Talk*, which was primarily used on smaller Bulletin Board Systems (BBSs). MUT allowed users to type messages to each other in real time and was replaced by *Internet Relay Chat*, released in 1988 and commonly known by its acronym, IRC. IRC quickly grew from its origins in Finland to connect to various universities around the world, limited only by the speed of Internet adoption.

IRC was primarily used by university staff and students and was run on many IRC servers all connected via the Internet. The servers were administered by *IRC Operators*, who were responsible for enforcing the network rules and able to drop users, or entire servers, from the larger IRC network. Debates amongst operators relating to which servers should be allowed to connect to the IRC network caused disagreement and resulted in various servers being disconnected from the IRC network. As a result, users regarded IRC as an unreliable service, providing the impetus for other third-party chat clients to be developed.

The first program generally available that addressed many of the shortcomings of IRC was ICQ, a play on the phrase *I seek you*. Client programs were freely available for download via the Internet and users would register to be assigned a unique ICQ number, similar to a telephone number. Users could then connect to a central server using their ICQ number and password. The server used a proprietary protocol and was administered by the company that created ICQ, Mirabilis.

Shortly after ICQ was released, several other instant messaging services became available, some of which are AOL's Instant Messenger (AIM), MSN Messenger, and

Yahoo Messenger. All of these programs addressed the problems with IRC in various ways: each service used a centralised server, which meant there was no need for multiple operators connecting servers and users; user identifiers were unique and protected with passwords; the client programs were generally easier to use with user interfaces simpler than IRC. The following section describes some of the current popular instant messaging clients.

2.2 Instant Messaging Clients

This section gives an overview of AOL's Instant Messenger (AIM) and MSN Messenger, which are two of the most popular instant messaging clients. Both programs work by sending units of messages rather than characters. That is, a user types some message and presses the Enter key to relay that message, as opposed to each character being sent automatically as it is typed. Other common features are file sending, where users may send files to each other; speech communication, where the computer's microphone and speaker are used to allow users to talk to each other similar to a telephone; and video communication, where a camera attached to the computer is used to allow users to see each other while speaking or typing to each other. Other instant messaging clients offer similar features and work in much the same way.

2.2.1 AOL's Instant Messenger

AOL's Instant Messenger (AIM) was released in 1997 and was one of the first popular instant messaging services. Users of AIM must register to receive a unique user name—called a *screenname* in AIM terminology. Figure 2.1 displays some of the main features available in AIM. On the left side of the figure is the *Buddy List*, which allows the user to add screennames of other users. The buddy list shows the status of each user on the service: whether they are online and available, online and not available, online and idle, or not online.

Advertising provides an important revenue stream for AOL and the buddy list displays various advertisements that change periodically. If a user clicks on an adver-

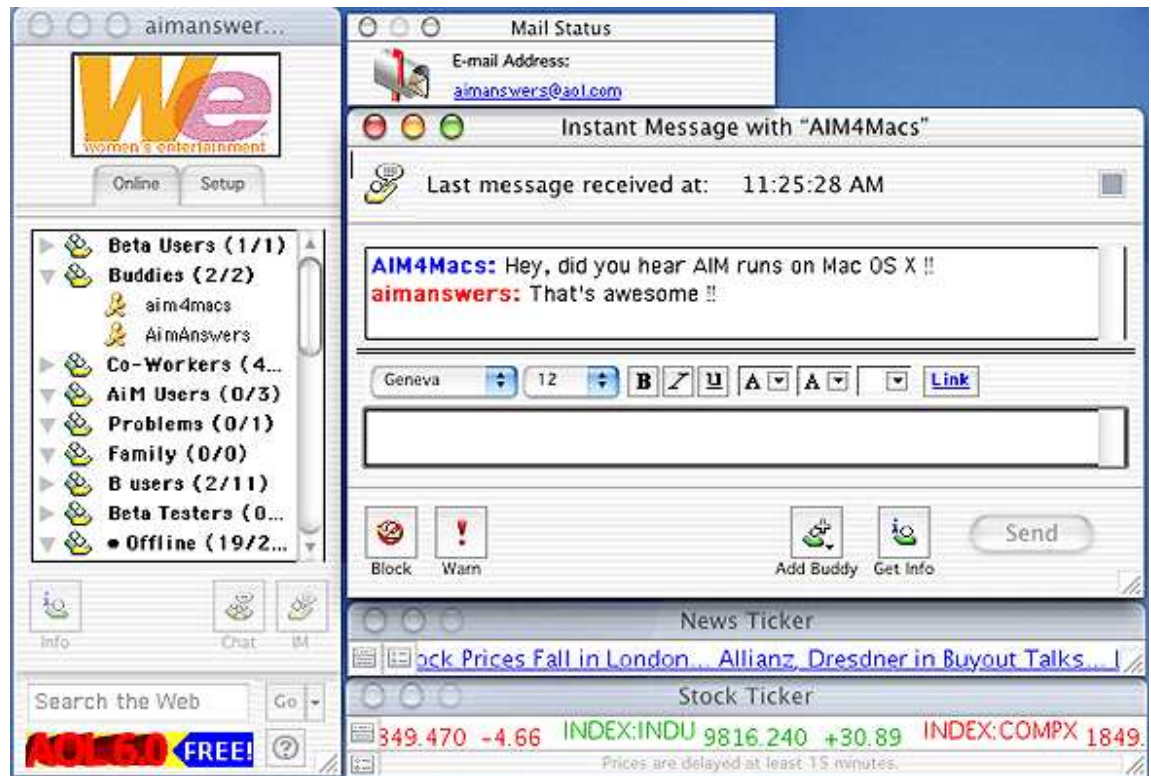


Figure 2.1: AOL Instant Messenger (AIM) screen shot.

tisement, a web browser is opened to display more information about the advertisement.

Communication takes place in the *Instant Message* window, shown in the middle-right of the figure. A message may be entered in the bottom portion of the window which is relayed to the other user when either the Enter key is pressed or the “Send” button is pushed. The message history is shown in the top portion of the window.

AIM users may be *warned* by pressing the “Warn” button on the *Instant Message* window. Warning a user displays a *Warning level* beside a user’s screenname in every other user’s buddy list that has that user listed, which has the effect of making *warnings* publicly known. The more warnings a user receives the higher the warning level, which slowly decreases back to zero over several hours. AIM users may also *block* other users, which means no more communication or buddy list status information will be allowed between the two parties.

Other features that are integrated with the AIM client are: AOL e-mail, shown on the top-right part of the figure; a news ticker showing news headlines; and a stock ticker that shows prices for stock market listings as can be seen in the bottom-right part of Figure 2.1.

2.2.2 MSN Messenger

MSN Messenger was released by Microsoft in 1999. Figure 2.2 shows the user’s list of contacts on the left and a chat window on the right. Users of MSN Messenger use an e-mail address as their unique user name. Once signed into the service, users name change their displayed *nickname*, but not their e-mail address.

Messages are sent to other users via the chat window, which can be opened by double-clicking on a name in their contact list, typing a message in the bottom part of the window shown in Figure 2.2 and pushing the “Send” button or the Enter key. When a message is sent the other party’s chat window automatically opens if it is not already open and displays the message in the chat history at the top portion of the window.

In 2005, Microsoft and Yahoo! announced an interoperability agreement between

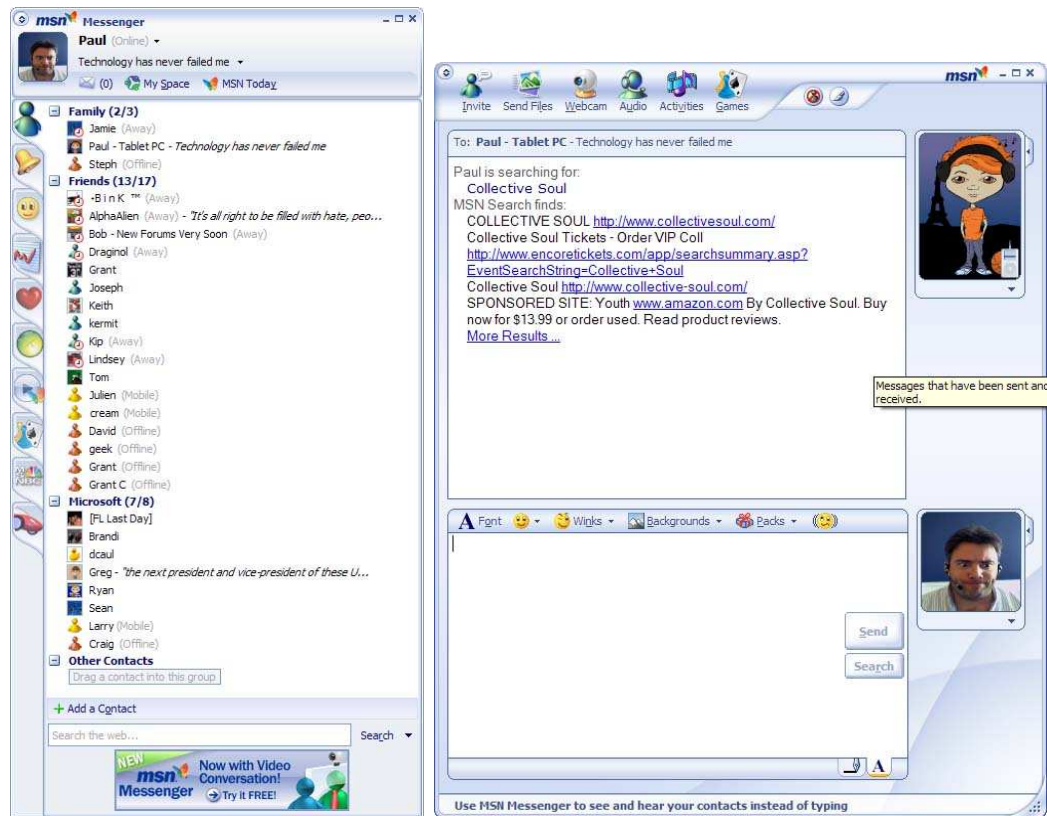


Figure 2.2: MSN Messenger screen shot.

their instant messaging services, allowing Yahoo! Messenger user to send messages to MSN Messenger users and vice-versa.

2.3 Commercial aspects of free Instant Messaging

All of the instant messaging clients we mention in this thesis are provided free of charge, including their usage. However, advertising and the purchase of *extensions* to the client program mean that there is also a lucrative business model behind the free products and services. Most of the popular client programs display advertisements which may be sold to advertisers as “real-estate” on users’ screens. Advertisers may also pay a fee every time a user clicks on an advertisement. Another revenue stream comes from the sale of third-party extensions, such as new emoticon images, background themes or games played within the instant messaging framework.

As with commercial radio and television advertising, more users translate to more advertising revenue, so companies have a strong incentive to obtain and maintain the largest share of instant messaging users. However, each company typically uses its own proprietary communication protocol, which is not compatible with any other instant messaging service. Existing users are therefore reluctant to change to a new service if that means they can no longer talk to their friends on the original service.

When Microsoft released its first version of MSN Messenger in 1999, users of MSN Messenger were able to communicate with users of AIM. This interoperability was not supported by AOL and was achieved by Microsoft analysing and replicating the communication protocol used by AIM, a technique known as reverse-engineering. This interoperability made it easy for AIM users to start using MSN Messenger with their AIM accounts. AOL perceived this threat to its advertising revenue stream and altered their protocol to break the interoperability. Microsoft proceeded to reverse-engineer the new protocol, again replicating it to mimic AIM. This continued through several iterations until Microsoft eventually gave up attempting to provide interoperability and instead focussed on building their own user-base. At the same time, Microsoft called for an open-standard instant messaging protocol, where any client could communicate with any other client.

As part of Microsoft's initiative to promote an open-standard instant messaging protocol, they released their MSN Messenger protocol specifications as an Internet Engineering Task Force (IETF) draft in 1999 for the Instant Messaging and Presence Protocol (IMPP) working group. In a press release dated August 18, 1999, Microsoft state the following:¹

“Our goal is to help people enjoy the benefits of free and open communication on the Internet, and we are pleased that this is resonating with so many consumers,” said Brad Chase, senior vice president, Consumer and Commerce Group, Microsoft. “Since 1997, Microsoft has actively worked with the industry on open standards for interoperability among messaging systems, and documenting our protocol is a significant step in this direction.”

Microsoft press release (August 18, 1999)

Also during this time, AOL and Time Warner announced their intention to merge. Time Warner's existing cable infrastructure, capable of delivering high-speed broadband Internet access, together with AOL's status as the dominant Internet Service Provider in America, meant that the merger required approval from America's Federal Trade Commission (FTC) as a potential monopoly existed. Microsoft took the opportunity to champion open interoperability by pointing out the potential for an instant messaging monopoly and lobbied the FTC to force AOL to open its instant messaging protocol as a condition of the merger. The merger was approved in December 2000 and AOL was not forced to open its protocol.²

However, in the six years that have since passed, MSN Messenger's protocol changed significantly and the new specifications were never again released. MSN Messenger's popularity increased from 1.3 million users to over 180 million users and Microsoft have resorted to altering the protocol to block out other third-party instant messaging clients that manage to reverse-engineer the protocol, repeating the original Microsoft versus AOL instant messaging experience.

It is not entirely ironic that Microsoft should abandon its push for open interoperability in favour of a proprietary protocol once their user base grew. Advertising

¹Available at <http://www.microsoft.com/presspass/press/1999/aug99/Protocolpr.mspx>

²Source: <http://money.cnn.com/2000/12/14/deals/aoltimewarner/>

revenue from instant messaging services is highly profitable as we see from Time Warner's (owner of AOL) reported earnings of over US\$1 billion from AOL advertising alone.³ With advertising revenue streams so large, businesses with a popular instant messaging client have great incentive not to risk losing their users by sharing protocols. Conversely, those with a small user-base have great incentive to entice users to switch to their own client programs and reverse-engineer the popular protocols.

One way to obtain more users is to simply buy the competition, which is what happened in 1998 when AOL acquired Mirabilis, the creator of ICQ. Alternatively, merging services may achieve a similar result, such as in 2005 where Microsoft and Yahoo! announced interoperability between their instant messaging chat clients.⁴

Regardless of which is the most popular client, there are currently hundreds of millions of people worldwide who use instant messaging. With the intense competition for users and large revenue streams, instant messaging shows signs of becoming more popular and has grown from something of a novelty to a mature form of communication. Table 2.1 (adapted from Baron (2003)) shows a time-line of the development of the Internet and instant messaging.

2.3.1 Forecasted Trends in Instant Messaging Usage

As the Internet's popularity increases, the number of people who use instant messaging is also expected to increase. New generations are as familiar and comfortable conducting their communication over IM, for both business and social purposes, as previous generations were with using a telephone.

During the late 1990s, when IM was experiencing growth surpassing that of Internet uptake, the largest growth was found to be amongst teenagers. Today, almost a decade later, that same user base, now adult, is comfortable using IM for customer support services instead of phone support. An early adopter of support via instant messaging was *Lands' End*, an online clothing web site. In 1999, *Lands' End* intro-

³Source: Time Warner Inc. annual report for the fiscal year ended December 31, 2004. Business segment results: AOL

⁴Source: <http://www.microsoft.com/presspass/press/2005/oct05/10-12MSNYahooMessengerPR.msp>

1968–1969	ARPANET (US Department of Defense Advanced Research Projects Agency Network) Used FTP (file transfer protocol) for sending documents, data or computer programs
1969	Telnet Allowed remote log-ins to ARPANET
early 1970s	PLATO (Programmed Logic for Automatic Teaching Operation) Included TermTalk to allow instant messaging between users
1973	TELENET Commercial packet-switching service; offshoot of ARPANET
late 1970s, early 1980s	BBSs (Bulletin Board Systems using telephone dial-up) e.g., Fido, The Well (Whole Earth 'Lectronic Link)
1979–1980	USENET (UNIX Users Network) Used different networking protocols than ARPANET; Distributed online forums called newsgroups
early 1980s	talk Provided instant messaging to UNIX users
1981	BITNET (Because It's There/Time Network) Cooperative network among IBM systems; Primarily used for electronic mail
1983	Internet ARPANET split into MILNET (for non-classified military information) and ARPANET (for computer research community); Old ARPANET became Internet
1988	IRC (Internet Relay Chat)
1990	WWW (World Wide Web) Created by Tim Berners-Lee
early 1990s	Gopher (Name of University of Minnesota mascot) System for locating documents on the Internet (arranged hierarchically by topic)
1993	Mosaic Web browser developed at the University of Illinois; System for locating information on the Internet (used word search)
1994	Netscape Commercial version of Mosaic
1996	ICQ (I seek you)
1997	AIM (AOL Instant Messenger)
1998	AOL acquires Mirabilis (creators of ICQ) AIM and ICQ interoperability ensures AOL has largest world-wide user-base
1999	MSN Messenger; Yahoo! Messenger
2005	MSN and Yahoo Messenger interoperability agreement

Table 2.1: Networking Timeline (main sources: Baron (2003); Microsoft press release: <http://www.microsoft.com/presspass/press/2005/oct05/10-12MSNYahooMessengerPR.msp>)

Request	Response
1: How's the weather in Melbourne? 2: How warm is it in Melbourne? 3: What's the weather like in Melbourne? 4: Is it raining in Melbourne?	The weather in Melbourne will be fine and sunny with a maximum temperature of 25C. It is currently 21C.

Table 2.2: Sample of a many-to-one relationship between requests and a response.

duced its *Lands' End Live* online customer service option to its web site. Potential customers browsing the site were given the option of clicking a button to chat live to a service agent. Online sales for the company increased from \$61 million in 1998 to \$162 million in 1999 after the service was introduced. Although it is impossible to know how much of this increase in sales is directly attributable to the live chat service, *Lands' End* continues to offer the service seven years on.

The benefits of IM are not only constrained to human-human communication. Many companies have recently adopted IM bots to assist their customers in support tasks, such as computer and configuration problems. In 2003, *Comcast*, a communication, media, and entertainment company with over 21 million customers, deployed their IM *AskComcast* service, which handles hundreds of thousands of customer interactions per month.⁵ The company claims large cost savings as a result of the automated IM service. Many other companies are deploying bots as a cost-effective way to promote company presence, marketing, sales or support assistance. However, to date, the techniques employed in these bots are rather primitive.

Current bots typically work by implementing a many-to-one list of requests to responses, such as shown in Table 2.2. Other systems expand this basic technique by using regular expressions in the request list, such as *.* weather .* Melbourne*, where the character sequence *.** matches any number of characters, hence matching requests 1 and 3 in Table 2.2. Some systems contain groups of semantically-equivalent responses and randomly select a response from the group, which makes the system seem less predictable. However, these techniques are severely limited in their ability

⁵Source: <http://www.conversagent.com/customers/comcast.html> (20 April 2006)

to maintain cohesion in a conversation as they are generally not designed to keep track of context in dialogue.

Automated IM support is, in several ways, cheaper, and more reliable than automated Interactive Voice Response (IVR) telephone systems: IM works via the Internet, which immediately provides global customer support for anyone with an Internet connection. There is no need for voice recognition as the input is already textual. The problem is instead shifted to automatically correcting spelling errors and making grammatical assumptions before processing a response.

To provide automated IM support, it is important to be aware of differences in language usage in IM compared to other forms of computer mediated communication, such as e-mail and web pages. IM often exhibits characteristics in language usage that are not common in other forms of writing. These differences are discussed in the following section.

2.4 Linguistic Features of Instant Messaging

There are significant differences between written and spoken language. Even though we consider the language, say English, to be the same, the speaker has relatively little time to decide on lexical choices and syntactical constructions when compared to the writer. Speakers make false starts, self-corrections, repetitions, and other dysfluencies that all contribute to various errors (Huddleston and Pullum 2003:12). Error rates are thus much higher in spontaneous speech than those in writing primarily due to the rapid production of speech. Speaking at several words per second leaves little time for reflection on construction choices and planning sentence structure. The speech that is ultimately uttered often tends to be more grammatically marked than writing. Writers, by contrast, have much more time to make corrections thus ensuring fewer grammatical errors.

To get a sense of where instant messaging fits with respect to traditional writing and speech, Table 2.3 shows a dichotomy between the two. It also defines some basic parameters which can be used to compare their linguistic features (Baron 2003). The variables are separated into two major categories: *form* and *content*. *Form* refers to

how the linguistic act is constructed or perceived, while *content* refers to the substance of the linguistic message.

Using these definitions as a guide, instant messaging is much more akin to face-to-face speech than traditional writing. This is evident when comparing it to some of the attributes of Face-to-Face Speech in Table 2.3: instant messaging is dialogue; communication is in real-time; the kinesic cues of face-to-face speech are replaced by emoticons and other expressions of emotions (explained later in this section); the writing is informal and spontaneous; and, it contains little or no editing.

To continue our categorisation of instant messaging, we turn to Computer Mediated Communication (CMC), which is loosely defined as any natural language messaging that is transmitted or received via a computer connection (Baron 2003). Table 2.4, adapted from Baron (2003:75), shows a classification of various CMC methods, forming a spectrum from *product*, such as an academic paper or book, through to *process*, such as one-to-one dialogue and IM. The writing styles classified at the top of the spectrum, labelled *product*, resemble traditional forms of writing that have existed for centuries. These types of writings are generally undertaken to produce some item that is read only when completed.

At the other end of the spectrum is one-to-one dialogue, labelled *process*, which closely resembles speech and IM. These types of writing are undertaken for the purpose of communication where the outcome is determined by both parties. There is generally no final product per se; even though e-mails may be saved and the IM conversations logged, the objective is defined by the conversation and any records that may subsequently exist are a byproduct.

Of the three items that are most process-like (e-mail, IM, and SMS), only IM is synchronous, meaning the participants have the potential to interact real-time, whereas e-mail and SMS are asynchronous. With asynchronous CMC, there is no assumption that the recipient will be at the device (computer, mobile phone, etc.) or that the message will be relayed immediately, for example, network congestion and other technical problems may delay e-mail servers. Writers using asynchronous CMC, such as e-mail, therefore have more time to proof-read their writing and correct grammar and spelling before sending their message. By this, we do not suggest that e-

	Variable	Traditional Writing	Face-to-Face Speech
Form	participants	monologue (no immediate feedback)	dialogue (commonly incorporates feedback)
	time issues	time-independent, durable	real-time, ephemeral
	accessibility	scannable (linear or random access)	only linearly accessible
	structural accoutrements	document formatting (e.g., text layout, punctuation)	prosody (e.g., intonation, volume, pauses)
Content	functions	heavily informational and documentary	heavily social (e.g., phatic, conveys attitudes)
	extralinguistic cues	minimal (e.g., handwriting, stationary choice)	kinesic cues (e.g., facial expression, posture)
	formality	more formal (e.g., no contradictions, proper subject-verb agreement and antecedent-pronoun agreement are important)	less formal (e.g., “There’s ten points here” and “A person should follow their own dream” are acceptable)
	internal structuring	more planning, more structured, more syntactically complex	more spontaneous, less structured, less syntactically complex
	editing	more likely to be edited (overall structure, grammar, spelling, punctuation)	little or no editing

Table 2.3: Distinctions between Written and Spoken Language (Baron 2003).

<div style="display: flex; align-items: center; justify-content: center;"> <div style="text-align: center; margin-right: 10px;"> PRODUCT (monologue) </div> <div style="text-align: center; margin-right: 10px;"> ↑ ↓ </div> <div style="text-align: center;"> PROCESS (dialogue) </div> </div>	Category	Examples	Comments
	completed works	academic papers, business reports	available through self-archiving or attachments
	Web sites	Web pages, Web logs (blogs)	increasing options for comments and interaction
	anonymous dialogue	newsgroups, MUDs, chat (including IRC)	some forums insist on vetting participants
	one-to-one dialogue (identified interlocutor)	e-mail, IM, SMS	e-mail and IM may have multiple recipients

Table 2.4: Computer Mediated Communication (CMC) Spectrum. Adapted from (Baron 2003:75).

mail correspondence is always grammatical with correct spelling. However, compared to instant messaging, where messages are sent immediately and are typically much shorter, users of e-mail have more time to edit their messages before sending them. As a result, e-mail messages tend to be longer and less casual than IM messages (Baron 2003).

The casual and time-constrained nature of instant messaging has led to some novel language usage. Somewhere in between speech and e-mail communication, instant messaging requires users to type their messages in real time, often under a self-imposed time restriction due to the knowledge that the other participant is waiting for a response. Users often augment their messages to include signs representing an emotional state or to signify that a particular message is to be taken seriously, sarcastically, facetiously or some other way. This results in conversations with many forms of ellipses, contractions, abbreviations, and acronyms, along with various sequences of special characters to represent a type of *meta-language*. This meta-language compensates for the communicative functions normally found in the prosody and facial expressions during face-to-face speech, but lost in writing. As an example, laughter may be represented with *haha*, yell using capital letters, and others as shown in Table 2.5. These characteristics place language usage in instant messaging much closer to speech than is commonly found in writing (Baron 1998; Crystal 2001). Part of a dialogue in our corpus, shown in example (1), illustrates

some of these features.

- (1) A: You are most welcome and you too have a great week. :)
B: hehe. ur awesome!

Example (1) shows three interesting and common features of instant messaging. The first is the *emoticon*, sometimes referred to as a *smiley*, at the end of A's utterance, " :) ". Turned ninety degrees clockwise, the colon represents a pair of eyes and the right-parenthesis a smiling mouth. This depiction of a smile indicates A's mood, somewhat compensating for the extralinguistic information that would otherwise be communicated via facial expressions or prosody.

The word *emoticon* is derived from *emotion* + *icon* and the concept was created by Scott Fahlman in 1982 who facetiously wrote at the time: "I propose . . . the following character sequences for joke markers: :-). Read it sideways. Actually, it is probably more economical to mark things that are NOT jokes, given current trends. For this use :-(" (Baron 2003:94).

The second interesting aspect of (1) is B's laughter represented by "hehe". Variations of laughter often appear as "haha" followed by any number of "ha"s after it—the more sequences of "ha"s indicate greater laughter. Acronyms such as "lol" and "rofl" which stand for "laughing out loud" and "rolling on [the] floor laughing", respectively, may also appear along with a multitude of others. The third feature of the example is B's contraction "ur", which stands for "you're" from the phonetic similarity derived from "u" for "you" and "r" for "are". Table 2.5 shows some more examples of abbreviations and emoticons from possibly hundreds in common usage.

There is no standard lexicon for the abbreviations, acronyms, and emoticons used in instant messaging. Hundreds of variations exist to express a broad range of emotions; many new ones are invented, and many less-popular ones fade out of usage altogether. Furthermore, emoticons vary widely amongst cultural and social boundaries. In Japan, a different emoticon tradition has emerged, known as *kaomoji*, which literally means "face marks". Katsuno and Yano (2002) explain that while emoticons in most Western cultures are meant to be read sideways, as shown in Table 2.5, *kao-*

lol	laughing out loud	brb	be right back	kinda	kind of
btw	by the way	l8r	later	plz	please
cuz	because	ur	your/you're	da	the
ttlyl	talk to you later	omg	O my God	idk	I don't know
:-)	smile	:-(frown	;-)	wink
:-o	surprised	:-\	confused	:-D	large grin

Table 2.5: Some common abbreviations and emoticons used in instant messaging.

moji are read right-side up. A smiling face in *kaomoji* is represented by the characters “^_^”; the focus of emoticon usage in Japan is on the eyes instead of the mouth.

Most popular IM client applications convert known emoticons into graphical images for display purposes. This is a convenient feature, particularly for people who may not be familiar with the character sequences. Many new emoticons have been invented to take advantage of this feature, quite often bearing no obvious resemblance to the image the characters represent. For example, many of the over 200 icons offered in the instant messaging client *Trillian* go beyond representing emotions, such as (&) for a dog's head, (~) for a reel of film, and (@) for a cat's head. This also creates some incompatibility amongst various IM clients; for instance, sending the character sequence “(nah)” in *Trillian* shows a smiling face, whereas MSN Messenger displays a goat.

A robust automated instant messaging dialogue system may need to address these characteristics when interpreting messages if it intends to represent the semantic content of emotional or attitudinal features in a message. In certain contexts, a smile as an emoticon may represent an acknowledgement of the previous message in much the same way as saying *ok*. The research we present in this thesis takes into account some of these linguistic features.

2.5 Conclusion

This chapter discussed the evolution of instant messaging (IM) and its proliferation supported by the Internet. With hundreds of millions of people using IM worldwide, many IM client programs have been developed in a race to obtain the most users. We showed that great incentives exist for companies to have a large user-base, particularly as advertising revenue is proportional to the number of users, accounting for over US\$1 billion for AOL alone in 2004.

Being a new communication medium, we showed how users supplement the extralinguistic information used in conversation, such as laughter and sighs, with new words and *emoticons*. Additionally, many abbreviations are used to convey common words or expressions more quickly. Because this convention is continually changing, IM bots should be able to handle at least the more common and important of these features as an aid to understanding an utterance; for instance, laughter or a smile may indicate an acknowledgement in some contexts.

With such a large IM user-base and the expectation for it to further increase, many companies are providing support services via IM in addition to their traditional telephone, IVR, and web-based support. IM support services are currently being conducted by human-human and human-computer dialogue.

An increasing IM user-base gives further justification for companies, particularly online merchants, to provide online support via IM since it is relatively cost-effective. This growth will motivate the development of better dialogue understanding in an attempt to automate more sophisticated support services that are capable of handling a greater range of problems and dialogue styles.

In many cases, IM is more convenient to use than traditional telephone support. Free-call 800 numbers, common in many countries, are free for the caller, but not for the receiver. They are also only accessible inside the country where the 800 number is hosted. By contrast, IM is available all hours and free to use globally over the Internet. These reasons are particularly important for international customers who would otherwise be making long-distance telephone calls and may not be able to reach customer support during standard business hours due to time-zone differences.

So far in this thesis, we have introduced IM and its emerging popularity in support services—the automation of which provides the impetus for this research. The next chapter discusses various techniques used to implement automatic dialogue systems. It then describes some common problems in natural language understanding, how they relate to dialogue, and differences between dialogue and monologue. Finally, we describe *dialogue acts* and a related scheme for annotating dialogue.

Chapter 3

Human-Computer Dialogue

The research presented in this thesis is solely focussed on task-oriented dialogue, which tends to exhibit a more predictable structure than social dialogue. Section 3.1 introduces some basic concepts in dialogue modelling with an overview of some common techniques for both simple and increasingly sophisticated dialogue systems. Providing realistic dialogue automation requires some degree of natural language understanding. We give an overview of some problems in natural language understanding with a focus on dialogue in Section 3.2. The differences between dialogue and monologue are then described in Section 3.3 where we describe *turns*, *messages*, and *utterances* in instant messaging and how they relate to each other. The theory of Dialogue Acts is described in Section 3.4 and dialogue act annotation, particularly DAMSL, described in Section 3.5.

3.1 Dialogue Management Techniques

A dialogue system is a system that mediates dialogue between a human user and a computer program. Dialogue systems are designed to handle events, typically utterances from the user. The events trigger modules that modify the state of the dialogue. Various levels of sophistication within this basic model allow for systems to handle relatively complex interactions.

Dialogue systems can either be highly directed, where the system prompts for

Module	Output	Events & Triggers
start		$\rightarrow m_1$
m_1	“Hello, would you like to leave someone a message?”	“yes” $\rightarrow m_2$; “no” $\rightarrow m_5$
m_2	“Enter the recipient’s name or ‘cancel’”	“cancel” $\rightarrow m_5$, else $\rightarrow m_3$
m_3	“Now type your message or ‘cancel’”	“cancel” $\rightarrow m_5$, else $\rightarrow m_4$
m_4	“Your message has been recorded.”	
m_5	“Good bye”	

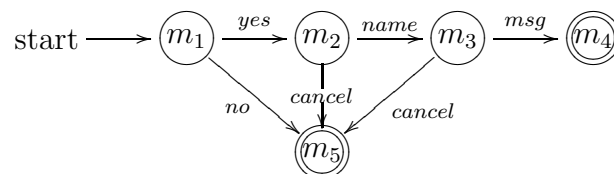


Figure 3.1: A simple finite-state automaton for a message relay service.

	Task-based	Non-task-based
Directed	<i>MovieFone</i> (AIM)	<i>Eliza</i>
Mixed-initiative	Present study	<i>SmarterChild</i> (AIM)

Table 3.1: Matrix showing different types of dialogue systems with examples of some popular bots and the present study. “AIM” is AOL’s Instant Messenger.

and only accepts particular information, or they can support mixed-initiative, where the user can provide information in any order and can change the goal or topic of the dialogue. Directed dialogue systems may be implemented by cascading modules which create a finite-state automaton such as in Figure 3.1. Directed dialogue systems require very little discourse modelling and are thus easier to implement and more common than their mixed-initiative counterparts. This simplicity, however, results in restrictive dialogue that can be frustrating to use.

Table 3.1 shows a matrix with task-based, non-task-based, directed, and non-directed dialogue systems. The present study concerns mixed-initiative, task-oriented dialogue, although the techniques presented here may also be applied to non-task-oriented systems. Dialogue systems that are highly directed may be implemented with finite-state scripts and thus do not require the use of dialogue acts.

From a customer support perspective, the complexity of a dialogue model increases with the scope of automated problem resolution. Mixed-initiative systems generally resort to becoming more directed when they encounter an utterance that can not be handled. If falling back to directed dialogue is still unsuccessful, the system typically transfers the customer to a live support representative. However, the cost of live support representatives is prohibitive for many smaller organisations.

The need for robust dialogue understanding is a clear necessity for automated dialogue support systems. Robust mixed-initiative dialogue systems are more accommodating, and hence successful at providing support in various domains. Some example domains include travel planning, computer-aided online shopping, various support services, and even human-computer interaction for obtaining help about how to use or access a certain application program feature.


Technique Used	Example Task	Task Complexity	Dialogue Phenomena Handled
Finite-state script	Long-distance calling	Least complex	User answers questions
Frame-based	Getting train arrival and departure information		User asks questions, simple clarification by system
Sets of contexts	Travel booking agent		Shifts between predetermined topics
Plan-based models	Kitchen design consultant		Dynamically generated topic structures, collaborative negotiation sub-dialogues
Agent-based models	Disaster relief management	Most complex	Different modalities (e.g., planned world and actual world)

Table 3.2: Dialogue and task complexity showing which techniques may be used for different levels of complexity.

Table 3.2 shows a classification of tasks that are suitable for dialogue-based interfaces (Allen *et al.* 2001). The tasks are ranked in order of complexity showing that as the complexity of the task increases, so too must the sophistication of dialogue management techniques.

Finite-state scripts, as described above, are the simplest way of implementing automated dialogue systems. Using this technique, a user can not provide answers to multiple questions in one turn as in “I want to travel from Melbourne to Sydney at 2pm”. Instead, the dialogue for such information would likely consist of three question-answer pairs, one for each of the departure city, destination city, and time of travel. The discussion of events and modules earlier shows how finite-state scripting techniques can be extended to allow some degree of user initiative.

Somewhat more complex, but also more convenient for the user, are *frame-based* approaches, which are able to retrieve multiple items of data from a single utterance. Frame-based systems maintain a set of parameters that need to be assigned values and then pattern-matching techniques are used to identify parameters using templates such as “from [CITY-NAME]” and “at [TIME]”, where “CITY-NAME” would match

with a list of city names and “TIME” would match a time pattern. There is no particular order that must be followed by the user when providing the information. What makes this approach significantly more complex than the system-directed finite-state technique is that the same information from the user can be expressed in many different ways. If the system is unable to retrieve any of the information, it can ask the user for what is missing, resorting to a system-directed, finite-state script if necessary.

The *sets of contexts* approach is an extension of the frame-based technique. Frames are associated within contexts where each context handles a specific task. A system designed to handle multiple tasks, such as flight, hotel, and car reservations, could represent each task as a context. Each context contains the frames necessary to obtain the required information from the user. Systems must allow the user to switch between contexts, giving information about the flight, hotel, and car reservation in any order. Recognising when a user has switched contexts can be a challenging task in itself, as in the case of a user providing a check-in time at a hotel versus the departure time for a returning flight.

The next two levels of complexity, *plan-based* and *agent-based* models, require systems to maintain models of the tasks and be able to reason about them. These tasks are too complicated to be modelled using contexts and frame-based techniques. Plan-based systems instead interact with the user to construct a plan. A fundamental requirement of these systems is to recognise the user’s intention during each step in the dialogue. Dialogue act recognition is a part of the broader intention recognition task, as described later in this thesis. The complexity and scope of tasks is generally much broader in plan- and agent-based models than in the simpler dialogue models. As a result, users’ intentions are more difficult to predict. Dialogue acts, described in Section 3.4, may be used to represent the intentions of the speaker; for example, knowing that the user has made a request rather than asked a question helps to at least narrow the scope of possible interpretations.

Agent-based models are similar to plan-based models, but also allow for the execution and monitoring of operations in a dynamically-changing world. The monitoring keeps track of a changing environment and takes initiative to inform the user of

pertinent events that may affect tasks, which may then lead to plans being modified.

Allen *et al.* (2001) suggest that the classifications in Table 3.2 probably cover all potential applications of human-computer interaction focussed on concrete tasks, as opposed to other functions of human conversation, such as social dialogue. Although any of these dialogue models can be applied to instant messaging systems, the work presented in this thesis is primarily relevant to the plan- and agent-based models because of dialogue acts being directly relevant to intention recognition, which is an approach used in Allen *et al.* (2001).

McTear (2002) also provide a thorough description of spoken dialogue systems that they classify into three main types:

1. finite state-based systems;
2. frame-based systems; and
3. agent-based systems.

These three types are analogous to those shown in Table 3.2, with *sets of contexts* and *plan-based models* considered more elaborate versions of *frame-based systems*.

As dialogue systems become more complex, their natural language understanding requirements necessarily increase. Some of the complexities of natural language understanding that affect dialogue systems are discussed in Section 3.2.

3.2 Additional research areas for realistic dialogue

Before discussing the differences between dialogue and monologue, we first give an overview of some challenges in natural language understanding that are independent of the communication methods used. Addressing these challenges is outside the scope of the present research, but this serves to highlight the complexity of a few tasks that are effortlessly resolved by people.

A robust dialogue system requires discourse modelling and a degree of natural language understanding, which minimally encompasses lexical semantics, anaphora

resolution, and word sense disambiguation. A discourse model, as discussed by Webber (1978), represents a dialogue participant's mental model of the ongoing discourse. Entities that are introduced in the dialogue are stored and may be subsequently referenced, which introduces the topic of *anaphora resolution*.

Anaphora resolution is a large research area in its own right and discussed in depth by Mitkov (2002). The problem, in short, is to match anaphoric references, such as pronouns, with their antecedents as in example (1) taken from our instant messaging corpus.

- (1) A₂₅: I am sending you the page that will pop up in a new window on your screen.
A₂₆: please have a look at it

In example (1), *it* is ambiguous and can refer to *page*, *window* or *screen*. Some level of reasoning must be performed to find the correct referent. This reasoning must take contextual information into account rather than rely on syntax rules alone.

There are many types of references possible, such as Indefinite Noun Phrases, Definite Noun Phrases, Demonstratives, One Anaphora, and others. A full discussion of this topic is outside the scope of the present research and we intend only to introduce the problem. A full treatment of anaphora resolution is given by Mitkov (2002).

Word sense disambiguation is concerned with choosing the correct sense, or definition, of a word where many senses exist (Hirst 1987; Resnik 1997; Resnik 1998). Example (2) shows an instance of this with the words *page* and *window*.¹

- (2) A₁₆: I am sending you a page that lists dress which might interest you. The page will open in a new window.

Depending on the context, *page* in (2) could mean a piece of paper being sent via postal mail or a URL address referring to a web page on the Internet. Likewise, *window* could refer to the glass window in the customer's home, or a browser window on the customer's computer screen. In this context, we would not consider that

¹All examples taken from our corpus of assisted e-shopping dialogues are cited verbatim.

the speaker intends to somehow send a piece of paper to us which will appear in a new glass window in the room, but discounting the possibility requires reasoning and certain assumptions to be made.

The extent to which each of these components are implemented directly affects the quality of the resulting dialogue system. The work on dialogue acts presented in this thesis will form an integral constituent assisting in the semantic interpretation of utterances. The utterances may then be augmented by processes which solve the problems mentioned in this section. Solving these problems, which are all areas of active research, must take into account aspects that are unique to dialogue and do not occur in other communication such as monologue. These differences are discussed in the following section with respect to the work presented in this thesis.

3.3 Differences between Dialogue and Monologue

In this section, we discuss a number of the differences between dialogue and monologue. Although the two have many similarities, dialogue has the added complexity of multiple speakers or writers referring to topics and items introduced into the conversation by another participant. The dialogue characteristics we cover here exist in both spoken and written dialogue, but spoken dialogue exhibits further characteristics related to the nature of spontaneous speech, which we do not address in this thesis.

The dialogue we consider in this thesis contains contributions from two participants taking turns to add information. This turn-taking behaviour typically creates a structure of adjacent pairs of utterances, such as questions and answers. The function of each utterance, whether it is a question, answer, et cetera, is represented by a dialogue act tag. Dialogue acts, their purpose, and rationale will be described further in Section 3.4.

3.3.1 Conversational Implicature

Participants in conversation share several assumptions about what is and is not being said. In the following excerpt from our corpus, the customer, C, desires to

purchase a dress as a gift from the Agent, A:

(3) A₆: Do you want to go for dress?

C₆: sure

A₇: Please have a look at this page . . .

A₈: Was it helpful?

C₇: They seem like halloween costumes.

Notice that the customer does not directly answer the agent's question at A₈. On the surface, C's response at C₇ is a statement of observation, yet one can clearly infer that C is not satisfied with the items on the web page that was sent.

Inferences such as that illustrated in (3) are common in dialogue and are examined in the work of Grice (1975, 1978) as part of his theory of conversational implicature. In this theory, Grice proposed that the inferences made by conversational participants engaged in *co-operative dialogue* are guided by a set of four maxims (Jurafsky and Martin 2000:727):

Maxim of Quantity: Be exactly as informative as is required:

1. Make your contribution as informative as is required (for the current purpose of the exchange).
2. Do not make your contribution more informative than is required.

Maxim of Quality Try to make your contribution one that is true:

1. Do not say what you believe to be false.
2. Do not say that for which you lack adequate evidence.

Maxim of Relevance Be relevant.

Maxim of Manner Be perspicuous:

1. Avoid obscurity of expression.
2. Avoid ambiguity.

3. Be brief (avoid unnecessary prolixity).
4. Be orderly.

These maxims are not always followed in practice and there is an element of subjectiveness within them: what is considered perspicuous by a speaker may not be to a hearer and vice-versa, the same applies for ambiguity and prolixity. However, Grice's theory states that conversational participants assume that these maxims will be followed in a co-operative discourse. Returning to example (3), the agent can infer that C's answer in C_7 is a *no* because of the Maxim of Relevance. The agent assumes that items resembling Halloween costumes are not appropriate when the desired item is a dress for a teenage girl.

Task-based dialogue exhibits these maxims since it is co-operative dialogue. Therefore, reasoning in task-based dialogue may rely on these maxims when interpreting utterances which helps to recognise intentions, among other tasks.

3.3.2 Grounding

In Table 2.4, we saw that information in monologue is disseminated as a *product* that does not alter regardless of the audience. This is obviously true for printed matter and generally true for a speech or performance, such as a play, where an actor does not alter the performance depending on the audience. Unlike monologue, dialogue participants negotiate the course of the dialogue during the discourse, contributing to the discourse in an orderly way. Clark and Schaefer (1989) argue that, for each utterance, discourse participants try to establish the mutual belief that the hearer has understood the speaker sufficiently for current purposes. This process of establishing *common ground* is accomplished collectively amongst the dialogue participants, resulting in conversational units called *contributions*.

Stalnaker (1978) also discusses this phenomenon and points out that speakers and hearers must constantly establish common ground as a form of synchronising their mutual beliefs in the context of the conversation. If a speaker is not confident that the hearer has understood the utterance, then the utterance may be reiterated or an explicit request for acknowledgment be made, such as *do you know what I mean?*

Jefferson (1984) identifies two types of grounding, or “acknowledgement tokens”, represented by saying *yeah* and *mm hmm*. Saying *yeah* in conversation indicates that the listener is prepared to shift from *recipient* to *speaker*, while *mm hmm* tends to exhibit a *passive reciprocity*: where the listener indicates that the speaker “is still in the midst of some course of talk, and shall go on talking” (Jefferson 1984:4)

Grounding is established by various methods by a speaker during a conversation. Clark and Schaefer (1989) propose a *contribution* model for grounding, which models the way mutual understanding is achieved by active participation from multiple participants. They categorise five kinds of *evidence of understanding* ranging from ‘weak’ visual indications through to ‘strong’ verbal or behavioural indications as shown in the following list, roughly graded from weakest to strongest:

1. *Continued attention*. B shows he is continuing to attend and therefore remains satisfied with A’s presentation.
2. *Initiation of relevant next contribution*. B starts in on the next contribution that would be relevant at a level as high as the current one.
3. *Acknowledgement*. B nods or says “uh huh”, “yeah” or the like.
4. *Demonstration*. B demonstrates all or part of what he has understood A to mean.
5. *Display*. B displays verbatim all or part of A’s presentation.

Each contribution is divided into two phases for participants **A** and **B** (Clark and Schaefer 1989:265):

Presentation Phase: **A** presents utterance *u* for **B** to consider. He does so on the assumption that, if **B** gives evidence *e* or stronger, he can believe that **B** understands what **A** means by *u*.

Acceptance Phase: **B** accepts utterance *u* by giving evidence *e'* that he believes he understands what **A** means by *u*. He does so on the assumption that, once **A** registers evidence *e'*, he will also believe that **B** understands.

Clark and Schaefer claim that if the *presenting* by **A** and the *accepting* by **B** is “done right”, then both participants will believe that **B** understands **A**, hence establishing common ground.

As audio and visual information is not available in instant messaging, *continued attention* is difficult to judge and at times unnecessary as the addressee may walk away for a short time then reply upon his return without having a negative impact on the conversation. These short breaks are sometimes advertised with the acronym *brb* for *[I will] be right back*. *Acknowledgements* in instant messaging are common with utterances such as *yea*, *wow*, *hmm* or *ok*.

We can see in examples (4) and (5) from our customer support corpus that *demonstration* is very common, particularly at the beginning of most of the dialogues as the agent seeks to confirm the customer’s aim. Example (4) shows an instance where the agent demonstrates he has understood the customer’s request. Likewise, in example (5), the agent demonstrates understanding by reformulating C’s utterance in C₄ with A₄.

- (4) C₃: i see. well, what would you recommend for the person who has everything?
 A₅: I would like to confirm that if you are looking for a gift?
 C₄: yep
 A₆: Thank you for confirming that. Is there a special occasion for this gift?
- (5) A₃: ...How may I help you today with MSN Shopping site?
 C₄: I want to buy a gift for my brother
 A₄: Thank you for the information, Customer. I understand that you want to buy a gift for your brother. Is that correct?
 C₅: yes ...

Displaying verbatim all or part of an interlocutor’s utterance is unusual in instant messaging. Participants assume utterances will be displayed and remain on the other participant’s screen during the entire discourse. The last message might be written back verbatim if technical problems occur, such as temporary network outage, but this is to assess whether any messages were lost during the outage rather than showing

Label	Description
Initiate	Begin new discourse unit (DU), content separate from previous uncompleted DUs
Continue	Same agent adds related content to open DU
Acknowledge	Demonstrate of claim understanding of previous material by other agent
Repair	Correct (potential) misunderstanding of DU content
Request Repair	Signal lack of understanding
Request Ack	Signal for other to acknowledge
Cancel	Stop work on DU, leaving it ungrounded and ungroundable

Table 3.3: Grounding acts for discourse units (DUs) from Traum (1999).

understanding.

Traum (1999) argues that Clark and Schaefer’s (1989) contribution model is insufficient to monitor on-line conversations. In regards to considering a particular utterance part of the presentation phase or the acceptance phase in the contribution model, Traum states that it is often necessary “to look at large segments of the conversation, both before and [after an utterance] before deciding how a particular utterance fits in” (Traum 1999:126). He further argues that “there is no easy way to tell the ‘state’ of the current contribution while engaged in a conversation”, as required for on-line processing (Traum 1999:126).²

To address these deficiencies, Traum (1999) presents an “on-line reformulation of the contribution model”, called the *grounding acts* model. This model does not require lookahead to track the progress of each communication and consists of the seven grounding acts shown in Table 3.3.

Units of grounded content are called *discourse units* rather than *contributions*. The main difference between the two is that *discourse units* only apply to the current utterance, whereas *contributions* can cover several turns in a dialogue. The grounding

²“State”, in this context, refers loosely to whether an utterance is awaiting acceptance or has been accepted.

acts are assigned to utterances denoting the function an utterance plays towards the achievement of common ground. An utterance may also correspond to more than one grounding act. The *acknowledge* grounding act in Traum’s grounding acts model covers the entire set of Clark and Schaefer’s (1989) *evidence of understanding*, which loses some granularity, but is easier to apply. A thorough analysis of issues in grounding can be found in Traum (1994).

Clark and Schaefer (1989) point out that regardless of how much detail is given to the grounding in dialogue, practically all discourse models make the following three assumptions:

1. *Common ground*: The participants in a discourse presuppose a certain common ground.
2. *Accumulation*: In the course of a discourse, the participants try to add to their common ground.
3. *Unilateral action*: The principal means by which the participants add to their common ground is by the speaker uttering the right sentence at the right time.

These assumptions are held for discourse models originating in various fields including philosophy, such as Lewis (1979), Stalnaker (1978), and in discourse representation theory (Kamp 1981; Kamp and Reyle 1991); linguistics, as in Heim (1983); artificial intelligence, such as Grosz and Sidner (1986); and psychology as in Johnson-Laird (1983) and van Dijk and Kintsch (1983).

Instant messaging discourse models also make these assumptions. We see in our corpus that common ground is often established, such as in message A₅ in (4). The customer service agent also assumes that the customer knows enough about desktop computing and terminology to understand “the page will open in a new window” refers to a web page and window on the customer’s computer screen; information accumulates and grounding occurs often, as shown in (5). Although our discourse model does not directly represents these features, *unilateral actions* do lead to certain sequences of dialogue acts which are captured by our model as will be discussed in Sections 3.4 and 5.2.1.

Further reading concerning co-operative dialogue and grounding can be found in Allwood (1976) and Allwood *et al.* (2000).

3.3.3 Turns, Messages, and Utterances

Unlike monologue, interlocutors in dialogue take turns when communicating. This is clearly evident when two people correspond by writing letters to each other: person A writes a letter to B, person B responds with a letter to A, A responds to B, et cetera. The medium in this example, postal mail, is conducive to maintaining turns rather than overlapping: the delay between sending and receiving letters tends to impose turn waiting. E-mail is similar where, although it may be relayed within a matter of seconds, the sender generally does not expect an immediate reply as no assumption can usually be made as to when the recipient will read the mail. Conversely, spoken dialogue exhibits frequent turn overlapping, which is partly due to the immediate nature of message delivery and response time. Instant Messaging dialogue sits somewhere in between these two extremes and also experiences overlapping turns. Overlapping turns are problematic when modelling discourse as the dialogue may seem less coherent, which we will address in this section.

There are two main kinds of instant messaging clients: *character relay* and *message relay* clients. *Character relay* clients transmit individual characters as they are typed. Pressing the backspace key deletes the previous character on both client programs. *Message relay* clients only send complete messages once written. A user will typically type a message then press the Enter key when they are satisfied with the message. We consider only message relay clients, rather than character relay clients, in this thesis as they are the most common amongst current popular instant messaging services.

Instant messaging dialogue has three main levels: *Turns*, *Messages*, and *Utterances*. Sequences of words are grouped hierarchically as characterised graphically in Figure 3.2.

The first level in Figure 3.2 is a *Turn*, consisting of at least one *Message*, which consists of at least one *Utterance*, defined as follows:

Message: A message is defined as a group of words that are sent from one dialogue

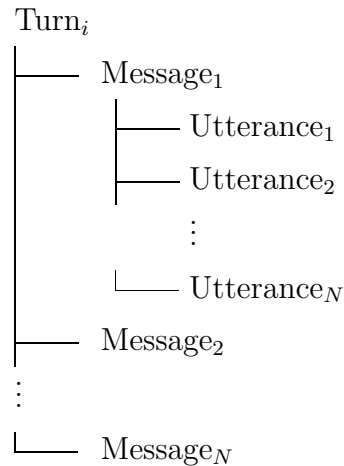


Figure 3.2: Tree diagram of the various constituents in instant messaging chat sessions. Turns may consist of many messages and messages may consist of many utterances.

participant to the other as a single unit. For the purposes of this study, this unit of text is physically constrained by *message relay* clients, which relay text in *message* units.

Turn: A dialogue participant normally writes one or more messages then waits for the other participant to respond, hence taking *turns* in writing messages. A *turn* always consists of at least one message.

Utterance: An utterance is a communicative act; each message in our corpus contains at least one utterance. As an example, the message “ok, I’ll try that” from our corpus contains two utterances: “ok” and “I’ll try that”. Each is a communicative act that can be taken independently to relate an *acknowledgement* (“ok”) and *commissive* (“I’ll try that”), which ostensibly commits the speaker to a future action. Utterances are often not grammatical sentences in the traditional sense. Where a sentence is a grammatical unit of one or more words abiding to some syntax, utterances are occasionally not even words, such as with *emoticons*; for example, “:-)” indicates a smile, which may represent an acknowledgement depending on the context. Many single-word utterances also exist, such as “ok”, “huh”, “hehe”, et cetera. Example utterances from

our corpus are enclosed within brackets in Table 4.2. For the purposes of this study, we do not regard cue phrases as utterances and dialogue acts can not span multiple utterances; that is, each utterance is assigned one and only one dialogue act.

Interlocutors sometimes send multiple messages in one turn; that is, a message is sent so that the other party may begin reading while another message is written by the same person within the same turn. Splitting a turn into multiple messages makes it easier to read than receiving one long message. However, when turns are split, the end of a turn is not always obvious: the person reading the first message might not realise another message is coming.³ This ambiguity partly results from messages in our corpus ending at utterance boundaries, so a complete communicative act has been received. As a result, accidental turn interruptions occur, which cause messages to become unsynchronised.

In the case of unsynchronised messages, each participant tends to respond to an earlier message than the immediately previous one, making the conversation seem somewhat incoherent when read as a transcript. An example of such a case is taken from our corpus and shown in Table 3.4.

In Table 3.4, Customer replied to message 10 with message 12 while Agent was still completing turn 6 with message 11. If the resulting discourse is read sequentially, it would seem that the customer ignored the information provided in message 11. This instant messaging phenomenon can lead to problems for systems that try to understand sequences of messages. The statistical models we use rely on sequences of dialogue acts and are thus affected by this lack of synchronicity. We discuss the problem of unsynchronised messages further in Section 3.4, which we then tackle in Section 4.4.1.

This section covered many topics in order to highlight some important considerations for natural language understanding in dialogue. Grounding is particularly

³Some message relay clients indicate that the other party is typing a message. However, this indication is no guarantee that a message will indeed be sent and is uncommon with web-based chat clients. Consequently, we ignore the effects of this feature in the present study.

Turn	Msg	Sec	Speaker	Message
5	8	18	Customer	[i was talking to mike and my browser crashed] ^{U₈} - [can you transfer me to him again?] ^{U₉}
5	9	7	Customer	[he found a gift i wanted] ^{U₁₀}
6	10	35	Agent	[I will try my best to help you find the gift,] ^{U₁₁} [please let me know the request] ^{U₁₂}
6	11	9	Agent	[Mike is not available at this point of time] ^{U₁₃}
7	12	1	Customer	[but mike already found it] ^{U₁₄} [isn't he there?] ^{U₁₅}
8	13	8	Customer	[it was a remote control car] ^{U₁₆}
9	14	2	Agent	[Mike is not available right now.] ^{U₁₇} [I am here to assist you.] ^{U₁₈}
10	15	28	Agent	[Sure Customer,] ^{U₁₉} [I will search for the remote control car.] ^{U₂₀}

Table 3.4: An example of unsynchronised messages occurring when a user prematurely assumes a turn is finished. Here, message (“Msg”) 12 is actually in response to 10, not 11 since turn 6 was sent as 2 messages: 10 and 11. Utterance boundaries are enclosed in brackets and denoted by U_n where n is the utterance number.

important as it influences or describes much of the flow of a dialogue, which in turn creates *dialogue act* adjacency pairs as will be described in the next section. We also described turns, messages, and utterances, which are a fundamental part of the research presented in this thesis as we investigate message segmentation and utterance classification into *dialogue acts*.

3.4 Dialogue Acts

In Section 3.3.2, we mentioned that discourse models assume *accumulation*, referring to participants trying to add to their common ground. Likewise, an interlocutor in dialogue intends to bring about some change in the hearer's information state. *Information state* refers to any information possessed by the hearer, such as the speaker's opinion on a matter, sharing a belief, rejecting a proposal, et cetera (Traum *et al.* 1999; Larsson and Traum 2000), which encompasses *accumulation* or adding to common ground. Utterances are exchanged when adding to common ground and the acts that utterances can play are referred to as *speech acts*, initially proposed by Austin (1962).

We earlier gave a simple example of speech acts that described the illocutionary force of an utterance. There are three types of acts that Austin identified:

locutionary act: The act of speaking an utterance.

illocutionary act: The surface meaning of the utterance, such as asking, answering, and greeting.

perlocutionary act: The effect the utterance has on the hearer.

As an example of each of these acts, consider the following sentence:

(6) It's cold in here.

Assuming that the utterance in (6) was spoken rather than written, the locutionary act was the act of actually speaking it, that is, producing the words that made

up the utterance. The illocutionary act is *Statement*: the speaker has stated an opinion. The perlocutionary act may be that the hearer does something to increase the temperature.

Searle (1979) extended the work of Austin to identify five categories into which every illocutionary act could be classified. In Searle's work, speech acts refer to Austin's illocutionary acts. Accordingly, any speech act could be classified into one of the following five categories:

Assertives e.g. *suggesting, boasting, concluding*.

Directives e.g. *asking, ordering, requesting, inviting*.

Commissives e.g. *promising, planning, betting, opposing*.

Expressives e.g. *thanking, vowing, deploring*.

Declarations e.g. *declaring war, announcing marriage, resigning*.

This development was seen as a first attempt at providing a complete set of high-level speech act tags. Searle's tag set has since been expanded to handle a wider range of functions that utterances can play in conversation. The expanded speech acts are called dialogue acts and include acts that represent *adjacency pairs* (Schegloff 1968). Adjacency pairs refers to moves in dialogue that are usually followed by complementary moves, such as an answer normally follows a question or a downplay after thanking.

Adjacency pairs are defined in terms of forward-looking and backward-looking functions. Forward-looking functions are similar to the speech acts already discussed; they characterise the effect an utterance has on the dialogue. Backward-looking functions are special speech acts that indicate how the current utterance relates to the previous one, thus forming a pair of adjacent dialogue acts.

Dialogue acts are a useful level of analysis for describing discourse structure and have been used to benefit tasks such as machine translation (Tanaka and Yokoo 1999) and the automatic detection of dialogue games (Levin *et al.* 1999), which are higher-level dialogue structures. Several other theories and projects derived from or

extending Speech Act Theory attempt to model dialogue at a deeper level. Two popular schemes, each serving a different purpose, are Dialogue Act Markup in Several Layers (DAMSL) and Dialogue Macrogame Theory (DMT). The following section describes DASML in some detail as our own dialogue act tag set, described in Section 4.1, is derived from DAMSL. DMT is briefly described in this chapter with more detail available in Appendix A.

3.5 Dialogue Act Annotation Schemes

Discourse and conversational knowledge can be represented at a number of different levels. A deep level of discourse structure represents things like the plans, specific intentions, and the goals of the interlocutors. More shallow levels of discourse structure represent dialogue acts, expected responses by the dialogue participants, and communication management aspects that serve to maintain the conversation, such as acknowledging an utterance by saying *okay*.

In this section, we describe Dialogue Act Markup in Several Layers (DAMSL), which has the ability to represent both the shallow and deep levels exhibited in dialogue. Our own dialogue act tag set is indirectly based on DAMSL as described in Section 3.5.2.

3.5.1 Dialogue Act Markup in Several Layers

Dialogue Act Markup in Several Layers (DAMSL) was an initiative to provide a general-purpose, high-level framework for dialogue acts. It allows tagging at several levels in order to represent the multiple and orthogonal functions of utterances as shown in the following example adapted from Allen and Core (1997):

Action-directive	A ₁ : take the train to Corning
Info-request, Hold(A ₁)	B ₁ : should we go through Dansville or Bath
Assert, Answer(B ₁)	A ₂ : Dansville

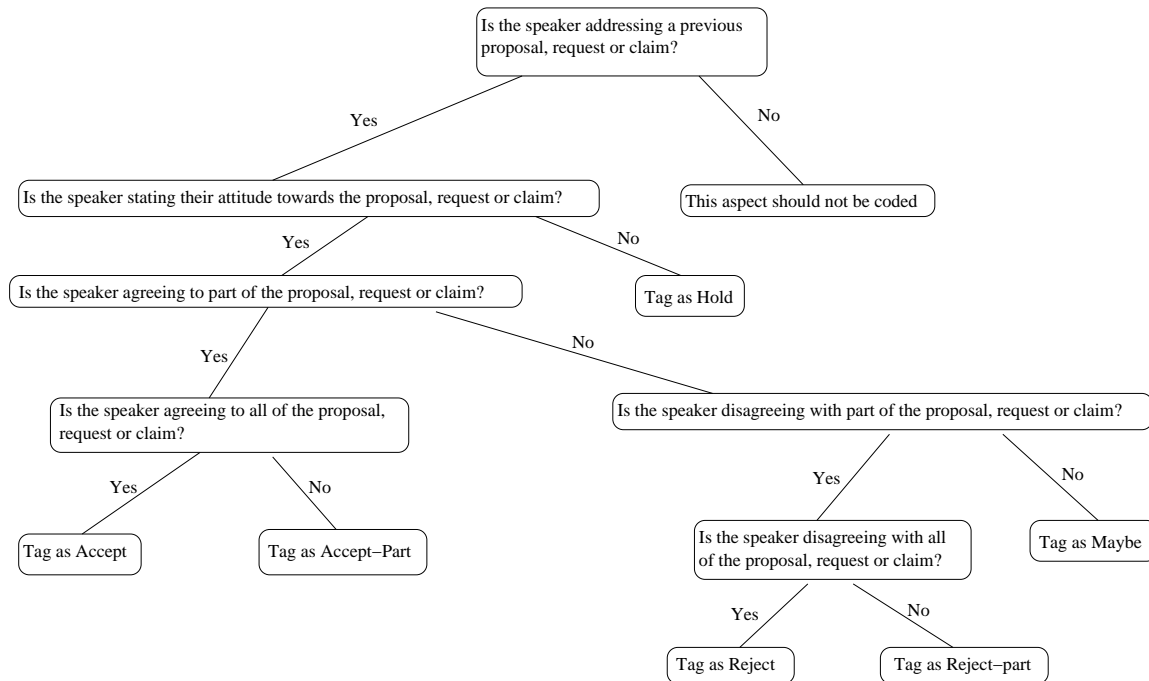


Figure 3.3: Decision tree for Agreement aspect in DAMSL. Taken from Allen and Core (1997)

In this example, speaker A’s request was vague since there were two possible paths for the train to take: Dansville and Bath. This prompted B to ask for clarification, which is labelled as both *Info-request* and *Hold*, indicating that B is not proposing an option. The final utterance is marked as both *Assert*, showing that A is making an assertion, and *Answer*, which shows that the utterance is an answer to B₁ in parentheses. The indented utterances indicate Allen and Core (1997) also provide decision trees for the DAMSL labels, such as the one in Figure 3.3.

The DAMSL framework was intended to allow greater sharing of data labelled with dialogue acts to aid in various research tasks. As described by Allen and Core (1997), utterance tags in DAMSL can be classified into the following four major categories:

Communicative Status: Records whether the utterance is intelligible and whether it was successfully completed.

Information Level: A characterization of the semantic content of the utterance.

Forward Looking Functions: How the current utterance constrains the future beliefs and actions of the participants, and how it affects the discourse.

Backward Looking Functions: How the current utterance relates to the previous discourse.

Most of the features in each of these categories can be applied directly to instant messaging dialogue. Section 4.1 describes our dialogue act tag set, which is based on a tag set by Stolcke *et al.* (2000). The tag set in Stolcke *et al.* (2000) was in turn derived from DAMSL. Because our tag set is related to DAMSL, we now turn to the four major DAMSL categories and their appropriation to instant messaging.

Communicative Status

The features for communicative status are UNINTERPRETABLE, ABANDONED, and SELF-TALK. Uninterpretable relates to an utterance that is not comprehensible. This can be caused by bad grammar, pronunciation or typing, but no instances existed in our corpus. It is also possible that an instant messaging interlocutor would deliberately send a message consisting of random characters which may be considered ‘uninterpretable’, but such behaviour would go against the maxims of Grice’s theory of conversational implicature discussed in Section 3.3.1, so we assume that interlocutors will not produce such utterances.

In speech, abandoned utterances may occur for a number of reasons, such as when a speaker makes an error or decides to say something different midway through the utterance. Utterances in DAMSL are only considered abandoned if they do not contribute to the dialogue. As discussed earlier, when messages are sent, they contain at least one complete utterance and therefore contribute to the conversation.

With instant messaging clients, messages are sent when the user presses the ‘Enter’ key or ‘Send’ button on the client application and can not be taken back or *unsent*. The closest a user can get to abandoning a message is by sending a new message asking the other participant to ignore the message. However, this creates new messages and so the original message is not *abandoned* in the same sense as in speech, but rather ignored by mutual agreement of the interlocutors, as in (7) below:

- (7) C₇: how many people are there where you're working?
A₁₁: There are many people working with MSN Shopping.
C₈: over 100?
A₁₂: Customer, I am sending you the page that lists remote control cars ...

The final question C₈ is not answered by A and C no longer pursues the topic, but nevertheless, the utterance itself was not abandoned as it was posed as a question that C presumably wanted answered. This contributes to the dialogue since A is aware that C would like particular information, assuming that A actually read the utterance. This interaction would be better represented using dialogue games, which are higher-level dialogue structures that may consist of many messages. The *games* represent sequences of utterances that serve various purposes, such as sharing particular information. If we classify A's attempts in (7) to have his question answered an *information seeking game*, then we can consider the *game* abandoned, but not the utterance. More information about dialogue games with examples are contained in Appendix A.

Self-talk occurs when a speaker talks to himself: in spoken dialogue, the speaker does not appear to intend to communicate what he is saying. Genuine self-talk in speech may occur in situations such as where the speaker has a habit of "thinking aloud" during a conversation. We do not consider this a possibility in instant messaging since anything sent to the other party is obviously intentional. Note that even if an interlocutor writes a message such as "I wonder if I should buy that", which is phrased as self-talk, the intention is to communicate the state of the interlocutor's mind. In this case, the presentation of the information is just a writing style rather than genuine self-talk. Utterances of this type should be tagged according to their illocutionary force, just like any other utterance. Therefore, the utterance "I wonder if I should buy that" could be tagged as either a question or statement, depending on the context.

Information Level

The information level in DAMSL represents the purpose of an utterance as it directly relates to the task or to its function in maintaining the conversation (Allen and Core 1997). Utterances on this level fall into one of four categories:

- **TASK:** Represents utterances that directly advance the goals of the conversation. This includes questions about the state of the world pertaining to the goals, suggestions on how the goals may be accomplished, and utterances about the general properties of the domain, such as eliciting details about the domain or problem.
- **TASK-MANAGEMENT:** Used for utterances that pertain to talking *about* the task as opposed to *accomplishing* it. These are utterances related to the problem solving process, such as ensuring the time is being kept by asking “are you keeping track of the time?” This includes utterances that ask for the status of a problem, asking for help on the procedures, and coordinating activities.
- **COMMUNICATION-MANAGEMENT:** Represents utterances that maintain the conversation, such as greetings and closings; acknowledgements like “Okay” and “uh-huh”; clarification requests like “sorry?” and “huh?”; trying to establish a communication channel, for example “are you there?”, and explicitly managing delays as in “wait a minute” and “brb” (be right back).
- **OTHER-LEVEL:** Utterances that do not neatly fall into any of the above categories even though they may be relevant to the dialogue. Examples are jokes and small-talk.

Each of these categories are relevant to instant messaging dialogue and can be used directly.

Forward- and Backward-looking Functions

Forward- and backward-looking functions in DAMSL represent the effect an utterance has on the subsequent utterance. The effect is not always evident from an utter-

Forward-looking functions	Backward-looking functions
Statement <ul style="list-style-type: none"> • Assert • Reassert • Other-statement 	Agreement <ul style="list-style-type: none"> • Accept • Accept-part • Maybe • Reject-part • Reject • Hold
Influencing-addressee-future-action <ul style="list-style-type: none"> • Open-option • Action-directive 	
Info-request	Understanding
Committing-speaker-future-action <ul style="list-style-type: none"> • Offer • Commit 	<ul style="list-style-type: none"> • Signal-non-understanding • Signal-understanding <ul style="list-style-type: none"> – Acknowledge – Repeat-rephrase – Completion • Correct-misspeaking
Conventional Opening Closing	
Explicit-performative	
Exclamation	
Other-forward-function	Answer
	Information-Relations

Table 3.5: Forward- and backward-looking functions in DAMSL. Functions do not necessarily match in displayed row order.

ance in isolation and may not even be what the interlocutor had intended. Therefore, it is sometimes necessary to look ahead in the dialogue to see how it was interpreted before being able to perform accurate labelling. This is obviously not possible in real-time dialogue as we can not look ahead during a conversation.

Forward-looking functions are often, but not always, matched by backward-looking functions, such as the question “would you like to buy a gift?” being followed by either a “yes” or “no”. Although possible, it is much less likely that it would be followed by another question, such as “Why would you think that?”. This level forms the basis for the dialogue-act tags presented and used in the present study. The forward-looking functions offered in DAMSL are shown in Table 3.5.

Unlike the dialogue-act labelling discussed in this thesis, multiple forward-looking

function tags in DAMSL may be assigned to a single utterance in order to represent the multiple effects an utterance can achieve. For example, the utterance “How are you?” may serve as both a Conventional Opening and an Info-request. The backward-looking functions in Table 3.5 follow the forward-looking functions in the same table, although not necessarily in the order displayed. Integrity is maintained with adjacency pairs by assigning a “response-to” tag to all backward-looking functions.

3.5.2 Modifications and Extensions of DAMSL

The DAMSL tags are the basis for our own tag set, which we discuss further in Section 4.1. More specifically, we based our tag set on the work of Stolcke *et al.* (2000), who modified the DAMSL tag set in several ways to make it more relevant to “task-free” domains, which suited the Switchboard corpus they were analysing. This was accomplished by using a multidimensional tag set called SWBD-DAMSL, which consisted of approximately 50 basic tags, such as QUESTION and STATEMENT. The tags could be combined to represent the different layers encoded within DAMSL, such as Information-Level annotation. This resulted in approximately 220 unique combinations that were used by the eight coders and were then collapsed into the 42 classes, mainly by merging many of the original infrequent classes. As reported by Stolcke *et al.* (2000), merging the tags was necessary in order to obtain higher interlabeller agreement as well as having enough data per class for statistical modelling purposes.

3.6 Prior Work on Dialogue Act Classification

Now that dialogue acts have been introduced, this section describes various methods that have been used to automatically classify utterances. Most of the studies we present deal with textual corpora, including transcribed spoken dialogue, with the remainder using recorded sound waves directly. We begin with a discussion of statistical methods, then discuss rule-based approaches that have been applied to dialogue act prediction.

3.6.1 Selected Statistical Learning Methods

Stolcke *et al.* (2000) report on thirteen studies, including their own, that predict dialogue acts from various corpora—both task-oriented and spontaneous human-to-human speech. The most commonly used corpora in the studies are Map Task, VERBMOBIL, and the ATR Conference corpus.

The Map Task corpus (Anderson *et al.* 1991) consists of pairs of speakers with slightly different maps of some imaginary terrain. They are tasked with reproducing a route drawn on only one of the maps without seeing the other speaker's map. The VERBMOBIL corpus consists of human-to-human dialogues scheduling appointments. The ATR Conference corpus consists of simulated dialogues between a secretary and a questioner at conferences. The Switchboard corpus contains recordings of human-to-human spontaneous speech (Godfrey *et al.* 1992).

Stolcke *et al.* (2000) aim to improve conversational speech recognition by integrating a probabilistic dialogue model with speech recognition. Their model predicted dialogue acts based on lexical, collocational, and prosodic cues, while also including constraints on likely dialogue act sequences by using dialogue act n -grams. The discourse structure is treated as a hidden Markov model with the individual dialogue acts treated as observations. The model was trained and evaluated using a hand-labelled database of 1,155 conversations from the Switchboard corpus.

The dialogue act recognition accuracy was 65% using errorful, automatically recognised words and prosody. A baseline for the experiments was obtained with a standard backoff trigram language model which was estimated from all the available training data. When evaluating the word recognition accuracy, all of the models, except for the Oracle language model, achieved a non-significant decrease in the word error rate compared to the baseline. The Oracle model chose the language model that corresponded to the hand-labelled dialogue act for each utterance, which gave a small, but statistically highly significant, improvement of 2.2% in the word error rate.

The results of the study were somewhat disappointing and were attributed to two main factors: the uneven distribution of the number of words and word types across dialogue act types, and; poor dialogue act recognition from spoken dialogue.

Source	Tokens	DAs/Tag Set	Accuracy
Nagata and Morimoto (1994)	2,450	15 / ATR	39.7%
Reithinger <i>et al.</i> (1996)	6,494	18 / VERBMOBIL	$\approx 40\%$
Mast <i>et al.</i> (1996)	6,494	18 / VERBMOBIL	59.7%
Reithinger and Klesen (1997)	2,701	18 / VERBMOBIL	74.7%
Chu-Carroll (1998)	915	15	49.71%
Wright (1998)	3,276	12 / Map Task	64%
Taylor <i>et al.</i> (1998)	9,272	12 / Map Task	47%
Samuel <i>et al.</i> (1998)	2,701	18 / VERBMOBIL	75.12%
Stolcke <i>et al.</i> (2000)	198,000	42 / SWBD-DAMSL	65%

Table 3.6: A selection of studies presented in Stolcke *et al.* (2000) showing the number of dialogue act tokens, dialogue act tag set size, tag set used, and accuracy.

Table 3.6, which has been adapted from Stolcke *et al.* (2000), shows key information comprising the number of dialogue act types and tokens, corpus, and accuracy from nine of the thirteen studies that were more closely related to the present work. It is important to note that depending on the actual corpus and dialogue act tag set used, results can vary widely. Therefore, we do not directly compare the results presented in Table 3.6.

The work in Table 3.6 uses various kinds of statistical models. The most common is using word n -grams to predict the next dialogue-act. Nagata and Morimoto (1994) used bigrams and and trigrams conditioned on preceding dialogue acts. Reithinger *et al.* (1996) varied this method by using deleted interpolation to smooth the dialogue n -grams. Chu-Carroll (1998) chooses which dialogue acts to use as conditions for predicting the next dialogue act by selectively skipping previous dialogue acts. This is made possible by using knowledge of sub-dialogue structure. The other studies all use variants of backoff, interpolated, or class n -gram language models to predict dialogue act likelihoods, except for Mast *et al.* (1996), who use semantic classification trees—a kind of decision tree conditioned on word patterns as features.

3.6.2 Selected Symbolic Learning Methods

Rules-based approaches can be broadly categorised in three main classes: decision trees, transformation-based learning, and memory-based learning. We discuss each in turn below.

Decision Trees

Decision trees can work with either discrete or continuous inputs. Discrete decision trees use *classification* learning, whereas learning continuous functions is called *regression*. We only discuss classification decision trees as they are appropriate for dialogue act classification.

A decision tree takes as input an object, which in this case is an utterance, and returns a decision—the predicted dialogue act classification. It does this by learning a sequence of tests during training that, when applied, lead to a leaf in the tree representing the decision. Each internal node in the tree represents a test. The branches from the node represent the possible values of the test. Each leaf node in the tree specifies the dialogue act that is to be assigned to the given utterance.

One clear advantage to decision trees is that they seem very natural to humans; for example, the format of many “How To” manuals is a single decision tree spanning hundreds of pages (Russell and Norvig 2003:653).

Wright (1998) applies decision trees to a subset of the DCIEM Maptask corpus (Bard *et al.* 1996). The corpus consists of 25 dialogues of fully spontaneous speech, comprising 4787 dialogue moves which they classified into 12 move types. After training on 20 dialogues and testing on 5, they achieve 44% classification accuracy using a unigram model of dialogues moves whereas using a 4-gram model of dialogue moves results in 63% accuracy.

Transformation-based Learning

Transformation-based learning, introduced by Brill (1995), uses a set of rules that classify the tokens—being utterances in this case. Each rule is instantiated from a preset group of templates, such as “if current tag is A, it is preceded by tag B and/or

the word C is present in one of the preceding N utterances, change the current tag to D.”

The rules are generated during supervised training by instantiating all possible rules given a corpus, then iteratively evaluating them and selecting the best set. The iterative selection process applies each rule to the corpus and calculates the precision. The rule that adds the maximum precision is selected and the process is repeated with the next rule. A common stopping criterion is to end when the remaining rules do not increase the accuracy beyond some threshold—often zero. However, since the number of rules with this method may be extremely large and computationally impractical, various modifications to this general algorithm are used to optimise training times. One such method is the Monte-Carlo pruning method, which follows the same process, except that the initial pool of rules are randomly selected from all possible rules.

Samuel *et al.* (1998) apply transformation-based learning with Monte-Carlo optimisation to the VerbMobil corpus and obtain a classification precision of 75%.

Memory-based Learning

Memory-based learning is a supervised classification technique that stores solved examples of given problems, then using similarity-based reasoning to solve new problems on the basis of the stored examples. Comparing the new problems with the stored ones can be done by several methods. One common method is the k -nearest-neighbours classification. This method takes examples labelled with classes and represents them as points in an example space. New examples are then also transformed into points on the example space and are assigned a class based on the k nearest examples using some distance measure between them.

More information about memory-based learning can be found in Daelemans and van den Bosch (2005), which also describes a software package that can be used to apply memory-based learning called TiMBL, for the Tilburg Memory-Based Learner. Levin *et al.* (2003) use TiMBL to apply a set of 70 speech acts to spoken task-oriented dialogue. Their corpus is the NESPOLE! travel and tourism database, which consists of over 14,000 tagged sentences in English, Italian, and German. They use 20-fold

cross-validation to achieve 69.82% accuracy for English speech-act classification and 67.57% accuracy for German.

Lendvai *et al.* (2003) report dialogue act classification also using TiMBL to classify a corpus of 3,738 question-answer pairs from 441 dialogues involving more than 400 different speakers. Their tag-set comprises 94 tags, such as *suggest*, *request*, and *reject*. They used 10-fold cross-validation and achieve an overall accuracy of 73.5%.

3.7 Summary

This chapter began with a discussion of various techniques used in the automatic processing of task-oriented dialogue. The techniques ranged from simple finite-state scripts through to the more complex agent-based models. We described how the simpler and most common finite-state automata can be implemented. Handling user input in the simpler systems is typically accomplished by scanning the user's message for text patterns, such as *my name is*. Designing conversational agents that can understand natural language at a more sophisticated level than the string pattern-matching used in frame-based models is an active area of research. More sophisticated techniques are necessary as pattern-matching techniques do not capture enough of the subtlety that people depend on for communication.

A fundamental goal of understanding dialogue is intention recognition, which dialogue acts are able to represent to a degree. Allen *et al.* (2001) point out that reasoning is unavoidable in determining the intentions of the user, however, statistical techniques are useful for certain subproblems such as parsing.

There are many aspects of natural language understanding that may seem trivial to native speakers of a language, but are in fact rather complex. We highlighted anaphora resolution and word sense disambiguation as examples and showed how they can occur in instant messaging dialogue. Anaphora resolution is concerned with matching referents to their antecedents, such as matching the pronoun *her* with a particular female. Word sense disambiguation occurs when one sense of a word must be chosen from multiple senses in a dictionary, such as *window* referring to the glass pane installed on a house, whereas in computer terminology it refers to the way

information may be presented on a computer screen. We then expanded on these topics by discussing the differences between monologue and dialogue and we showed how *grounding* is essential for dialogue understanding.

In grounding, which is specific to dialogue, interlocutors share information during a discourse as a way of acknowledging that they understand each others' utterances. This occurs in several ways throughout the dialogue. We discussed the differences in grounding with instant messaging compared to face-to-face conversations and showed how interlocutors in instant messaging compensate for the lack of visual information when grounding.

To prepare for further discussions about instant messaging in this thesis, we defined three levels of conceptual word groupings in instant messaging: utterances, messages, and turns. Our dialogue acts correspond to utterances, but recognising an utterance first requires message segmentation as messages may contain multiple utterances. We explained how turns may consist of multiple messages which allows for inadvertent interruptions, causing problems when interpreting messages sequentially.

Finally, we described dialogue acts, their relation to speech acts, and Dialogue-Act Markup in Several Layers (DAMSL) as a precursor to describing our own dialogue act tag set in Section 4.1. Deeper levels of discourse understanding are represented by theories such as Dialogue Macrogame Theory (DMT), providing for more robust dialogue models. Although outside the scope of this research, assigning dialogue acts to utterances is a starting point to implementing DMT.⁴

Our research concerns utterance segmentation and dialogue act classification and their immediate use in improving discourse understanding. The following chapters describe the process of collecting and preparing our corpus for the segmentation and classification tasks. Chapter 4 begins with a description of the design of our dialogue act tag set, which we use for the classification task. The methods we used to collect our data are presented in Section 4.2, which is then annotated using the process and format described in Section 4.3. Data preparation comprises of synchronising

⁴Appendix A contains more information about DMT.

messages, segmenting messages into utterances, then classifying the utterances into dialogue acts, which are all explained in Section 4.4. Chapter 5 details our experiments in segmentation and classification followed by the results in Chapter 6.

Chapter 4

Construction of an IM dialogue corpus

Large corpora are widely available for both spoken language (Godfrey *et al.* 1992; Burnard 2000; Anderson *et al.* 1991) and written language (Burnard 2000; Marcus *et al.* 1993; Lewis *et al.* 2004). Spoken and written grammars have been extensively studied as these forms of communication have been used for thousands of years. However, instant messaging has received very little attention by comparison due to its relatively recent introduction. As a result, obtaining a corpus adequate for research is more difficult for instant messaging than other genres of language.

In Section 2.4, we discussed the differences between instant messaging and other forms of communication such as speech and writing. Because of the differences in language usage and the unavailability of an appropriate instant messaging corpus, we constructed a corpus from an online e-shopping assistance service. The service provides help to potential customers in finding items for purchase via an instant messaging, browser-based interface.

This chapter describes the process employed to collect and prepare the data used in this research. We begin by describing the creation of our dialogue act tag set in Section 4.1, which we used to classify utterances. The creation of our e-shopping corpus is described in Section 4.2, followed by the annotation scheme we used for marking up the corpus with dialogue acts in Section 4.3. Finally, the data preparation

necessary for our segmentation models is described in Section 4.4, with a summary and conclusion presented in Section 4.5.

4.1 Dialogue Act Tag Set

To create our dialogue act tag set, we manually labelled our corpus using a subset from Stolcke *et al.*'s (2000) list of 42 tags. Some tags, such as UNINTERPRETABLE and SELF-TALK, were eliminated as they were not relevant for our corpus. Tags that were semantically similar were collapsed into one tag; for example, NO ANSWERS, REJECT, and NEGATIVE NON-NO ANSWERS were all represented by NO-ANSWER in our tag set. The main reason for this was to avoid sparse data problems given that our corpus size was substantially smaller than the Switchboard corpus used by Stolcke *et al.* (2000). Additionally, this deviation was appropriate since the primary purpose for Stolcke *et al.* (2000) was to enable computational dialogue act modelling for *conversational speech* with an aim to improve conversational speech recognition. As a result, their tag set was biased towards categories that were lexically or syntactically distinct and could be reliably identified. Our dialogue act tag set was created by one annotator.

Inter-annotator agreement is an important factor when creating any labelled corpus. The Kappa statistic is used to compare inter-annotator agreement normalised for chance resulting in values ranging from 0 (random labelling) to 1 (perfect agreement) as described in Siegel and Castellan (1988). Carletta (1996) argues that Kappa values of 0.8 or higher indicate acceptable reliability of inter-annotator agreement. Stolcke *et al.* (2000) report 84% inter-annotator agreement with their set of 42 tags and a Kappa statistic of 0.8. For the present study, labelling was carried out by three computational linguistics graduate students who achieved 89% dialogue act classification agreement, resulting in a Kappa value of 0.87. These results allow us to confidently use our tag set in automated classification.

The complete list of the 12 DAs we used is shown in Table 4.1 along with a description and examples of each DA in our corpus. Although most of the tags are self-explanatory, there are some subtle differences when tagging dialogue questions.

Tag	Description	Example	%
STATEMENT	Used for assertions that may state a belief or commit the speaker to doing something.	I am sending you the page which will pop up in a new window on your screen.	36.0
THANKING	Conventional thanking	Thank you for contacting us	14.7
YES-NO-QUESTION	A closed question which can be answered by either a ‘yes’ or ‘no’.	Did you receive the page, Customer?	13.9
RESPONSE-ACK	A backward-looking acknowledgement to the previous message. Used to confirm that the previous utterance was received/accepted.	Sure	7.2
REQUEST	Used to express a speaker’s desire that the hearer do something - either performing some action or simply waiting.	Please let me know how I can assist you on MSN Shopping today	5.9
OPEN-QUESTION	A question that can not be answered with only a ‘yes’ or ‘no’. The answer is usually some form of explanation or statement.	how do I use the international version?	5.3
YES-ANSWER	A backward-looking label being an affirmative response to a YES-NO-QUESTION	yes, yeah	5.1
CONVENTIONAL-CLOSING	Various ways of ending a conversation.	Bye Bye	2.9
NO-ANSWER	A backward-looking label being a negative response to a YES-NO-QUESTION	no, nope	2.5
CONVENTIONAL-OPENING	Greetings and other ways of starting a conversation.	Hello Customer	2.3
EXPRESSIVE	Can serve as both an acknowledgement of a previous utterance and an indication of the speaker’s mood; e.g. sending “haha” acknowledges the previous utterance while expressing amusement. This category includes emoticons.	haha, :-), wow	2.3
DOWNPLAYER	A backward-looking act often used after THANKING to minimise the significance of the THANKING.	You are welcome, my pleasure	1.9

Table 4.1: The 12 dialogue act labels with a description and examples of each.

There are two types of *question* tags in our tag set: OPEN-QUESTION and YES-NO-QUESTION. It is sometimes unclear how to label a question such as:

- (1) May I know the price range please?

Although it is phrased as a YES-NO-QUESTION, the hearer infers that this is an indirect request with the same illocutionary force as saying:

- (2) Please tell me the price range

These types of utterances are therefore loosely referred to as *indirect speech acts* (Perrault and Allen 1980). Of all the YES-NO-QUESTION utterances in our corpus, 29% are considered ambiguous. This includes questions similar to (1) such as:

- (3) is there a price range you would like to stay within?

We previously mentioned that dialogue acts are based on the illocutionary force of an utterance, however, in these ambiguous cases we labelled utterances with the base form of the question without looking ahead to see how it was interpreted. The rationale for this was to narrow the options and leave the final decision of whether to treat such an utterance as a YES-NO-QUESTION, OPEN-QUESTION or REQUEST to a more sophisticated system that could overwrite the surface form dialogue act if necessary.

Perrault and Allen (1980) propose a method of deciding whether to interpret a dialogue act¹ directly or indirectly under the assumption that dialogue participants cooperate and infer the goals being sought. Using various *plausible plan inference rules*, dialogue participants construct a set of questions based on assumptions of cooperative dialogue to decide how to reply to a question. If S and A are dialogue participants and ACT is some action, then one such plan inference rule is:

“If S believes that A wants to do ACT then it is plausible that S believes that A wants to achieve the effects of ACT.”

¹Perrault and Allen (1980) use the term “Speech Act” to mean the equivalent of dialogue act as used here. We use the term “dialogue act” for consistency.

This rule serves to justify the subsequent re-tagging of sentence (1) from YES-NO-QUESTION to REQUEST. Although the research presented in this thesis does not implement any such rules, we mention it here only to show how the task may be approached.

4.2 Data Collection

Task-based conversations were obtained using an online support service designed to assist potential customers in finding items for purchase. Eight conversations were conducted using the MSN Shopping service, made up of approximately 550 utterances and 4,500 words. This data forms an appropriate corpus for the research presented in this thesis since it is task-based and real-world dialogue. The data was gathered by five volunteers who acted as customers for the MSN Shopping service. Each volunteer was assigned the task of finding some item to purchase and was given a number of hypothetical situations as examples. The hypothetical situations included the following:

- Buying a birthday gift for a sibling without having any particular item in mind;
- Buying specific items for personal use;
- Buying a specific item for someone living in another country.

Although the number of dialogues that were ultimately gathered was rather small by most corpora standards, the corpus contained approximately 550 utterances, which was sufficient for our experimentation purposes.

The volunteers contacted the MSN Shopping service through the normal channels on the MSN Shopping web site. The web site redirects customers to a browser-based instant messaging web page consisting of a small text entry field at the bottom of the window for the user to type messages and a larger window above it containing the conversation history. Each volunteer proceeded to ask questions and otherwise engage in a dialogue in order to complete the task. An e-mail of the dialogue transcript was sent by the MSN Shopping web site to the customer at the end of the session. Table

Speaker	Message
Sanders	[Hello Customer] ^{CONVENTIONAL-OPENING} , [thank you for contacting MSN Shopping] ^{THANKING} . [This is Sanders and I look forward to assisting you today] ^{STATEMENT}
Sanders	[How are you doing today?] ^{OPEN-QUESTION}
Customer	[good] ^{STATEMENT} , [thanks] ^{THANKING}
Sanders	[How may I help you today?] ^{OPEN-QUESTION}

Table 4.2: An example of the beginning of a dialogue in our corpus. Utterance boundaries are shown in brackets and dialogue-act tags as superscripts.

Feature	Example	Prevalence
Spelling	whcih re,ote poeple	2.7%
Capitalisation	french i i'm india	2.4%
Emoticon	:- (:-) :	0.7%
Abbreviation	ooh haha hehe ur cya	2.5%
Other	“merchants.Thank” “working working”	1%
Total		$\approx 9.3\%$

Table 4.3: Characteristics of the MSN-Shopping corpus.

4.2 shows the beginning of one of the dialogues along with utterance boundaries and dialogue acts.

Questions were not limited only to buying the gifts; general questions were asked about the service and some of the dialogues contain emoticons and other forms of expressive statements such as laughter. Some characteristics of the dialogues are shown in Table 4.3. The distribution of the features between agent and customer was fairly even and there was no evidence of errors being corrected during the course of the dialogues.

The emoticons and abbreviations in the corpus are common in instant messaging dialogue, many of which were shown in Table 2.5 in Section 2.4. Neither utterances nor sentences were split over multiple messages and no synchronisation was necessary

for the dialogues.

4.3 Data Annotation

The MSN Shopping data was manually segmented into utterances and labelled with the dialogue acts by an expert coder as described in Section 4.1. We used a simple XML-based encoding scheme for marking up the data as shown in the example below:

```
Sanders <tag:Conventional-Opening>Hello Customer</tag>,  
<tag:Thanking>thank you for contacting MSN Shopping</tag>.  
<tag:Statement>This is Sanders and I look forward to assisting you  
today</tag>.
```

To make the annotation task quicker, we wrote a program that allows part of a line to be selected and tagged with any of our dialogue acts using a simple graphical user interface. The labelled data was then used to train our statistical models.

4.4 Data Preparation

To train the segmentation and classification models, we first had to prepare our corpus for each of the models described in Chapter 5. The first step in preparing the data was to replace the customer’s actual name with the word “Customer” in the corpus for anonymity. Appendix B contains the dialogue transcripts.

This section describes the rest of the data preparation that was necessary for each model.

4.4.1 Message Synchronisation

In our description of turns, messages, and utterances in Section 3.3.3, we briefly discussed how the end of a turn is not always obvious in instant messaging dialogue. Users often divide turns into multiple messages, usually at clause or utterance boundaries, which can result in the end of a message being mistaken as the end of the turn.

Turn	Msg	Sec	Speaker	Message
5	8	18	Customer	[i was talking to mike and my browser crashed] ^{U₈} - [can you transfer me to him again?] ^{U₉}
5	9	7	Customer	[he found a gift i wanted] ^{U₁₀}
6	10	35	Agent	[I will try my best to help you find the gift,] ^{U₁₁} [please let me know the request] ^{U₁₂}
6	11	9	Agent	[Mike is not available at this point of time] ^{U₁₃}
7	12	1	Customer	[but mike already found it] ^{U₁₄} [isn't he there?] ^{U₁₅}
8	13	8	Customer	[it was a remote control car] ^{U₁₆}
9	14	2	Agent	[Mike is not available right now.] ^{U₁₇} [I am here to assist you.] ^{U₁₈}
10	15	28	Agent	[Sure Customer,] ^{U₁₉} [I will search for the remote control car.] ^{U₂₀}

Table 4.4: An example of unsynchronised messages occurring when a user prematurely assumes a turn is finished. Here, message (“Msg”) 12 is actually in response to 10, not 11 since turn 6 was sent as 2 messages: 10 and 11. Utterance boundaries are enclosed in brackets and denoted by U_n where n is the utterance number.

This ambiguity can lead to accidental turn interruptions which cause messages to become unsynchronised.

When interruptions occur, each participant tends to respond to an earlier message than the immediately previous one, making the conversation seem somewhat incoherent when read as a transcript. An example of such a case is shown in Table 4.4 in which Customer replied to message 10 with message 12 while Sally was still completing turn 6 with message 11. That is, her turn was split into two messages, 10 and 11. If the resulting discourse is read sequentially it would seem that the customer ignored the information provided in message 11. The time between messages shows that only 1 second elapsed between messages 11 and 12, so message 12 must in fact be in response to message 10.

Message M_i is defined to be *dependent* on message M_d if the user wrote M_i having already seen and presumably considered M_d . The importance of unsynchronised messages is that they result in the dialogue acts also being out of order, which is problematic when using Markov language models. Therefore, we re-synchronise the messages before training and classification.

After manually marking message dependencies in the dialogue included in Appendix A, we found that 20.0% of the messages were not synchronised. We developed a method using the typing rate, defined as the elapsed time between two messages divided by the number of characters in a message, to determine message dependencies. Mathematically, we calculate the typing rate with the formula $\frac{time(M_i) - time(M_d)}{length(M_i)}$, where M_i is the current message and M_d is the dependent message. The dependent message may be the immediately preceding message such that $d = i - 1$ or any earlier message where $0 < d < i$ with the first message being M_1 . The algorithm to find dependant messages finds message M_d such that the typing rate of M_i is greater than or equal to the typing threshold using the elapsed time between M_d and M_i . This is shown in Algorithm 1.

The *typing_threshold* in Algorithm 1 was calculated by taking the 90th percentile of all observed typing rates that had their dependent messages manually labelled, which resulted in a typing rate of 5 characters per second. The baseline accuracy of detecting turn-taking message dependencies is 80%. We arrive at this baseline figure

Algorithm 1 Find message dependency for message M_i

 $d \leftarrow i$
repeat
 $d \leftarrow d - 1$
 $typing_rate \leftarrow \frac{time(M_i) - time(M_d)}{length(M_i)}$
until $typing_rate \geq typing_threshold$ or $d = 1$ or $speaker(M_i) = speaker(M_d)$

by deducting the 20% of messages that are unsynchronised assuming that $M_d = M_{i-1}$, as mentioned earlier. Using Algorithm 1, we achieved a correct dependency detection accuracy of 92.5%.

The MSN-Shopping assistance dialogues were conducted via a web browser front-end, which caused delays of a few seconds or so per message, whereas most instant messaging clients have only sub-second delays. The relatively long delay resulted in only few unsynchronised messages being sent; however, for faster mediums, such as most instant messaging clients, we expect the proportion of unsynchronised messages to be closer to that discussed in this section and demonstrated in Appendix A. Because this dialogue is not a central part of the corpus studied in this thesis, we do not present a linguistic analysis of it, but include it as a marked-up appendix for the reader to peruse.

4.4.2 Utterance Segmentation

Utterance segmentation is concerned with identifying utterances, typically being communicative acts, within a larger sequence of words. To date, much research into segmentation has focussed on transcribed speech, such as the work by Stolcke and Shriberg (1996), Gavaldà *et al.* (1997), Walker and Whittaker (1990), and Tetreault *et al.* (2004). However, speech segmentation and utterance segmentation have different aims: large-vocabulary speech recognisers segment speech into *acoustic* segments for more efficient processing. This is accomplished using criteria such as non-speech intervals and turn boundaries in dialogue. These methods are not appropriate for instant messaging utterance segmentation because the acoustic segmentation meth-

ods rely on prosodic markers such as intonation and pauses in speech, which do not exist in written dialogue. Furthermore, *utterances* are larger syntactical units than the word-boundary segments typically used in speech recognition tasks. The segmentation methods we use are therefore based on both statistically- and linguistically-motivated approaches.

Utterance segmentation is inherently subjective and it is therefore important to collect inter-annotator agreement statistics before relying on any segmentation data. As an example, the message *yes, ok* may be regarded either as one ACKNOWLEDGEMENT or a YES-ANSWER followed by an ACKNOWLEDGEMENT. Deciding which segmentation is appropriate depends largely on how the utterances will be used in downstream tasks and which dialogue acts are available in the tag set.

When creating a dialogue act tag-set and tagging utterances, it is important to be aware of the inter-dependance between the two tasks: utterances must be segmented so that they are assigned only one dialogue act, yet dialogue acts must exist which adequately describe the communicative acts represented by the utterances. To illustrate this point, the earlier example with the message *yes, ok* could only be tagged as one ACKNOWLEDGEMENT act if YES-ANSWER did not exist in the tag-set. Similarly, if YES-ANSWER was available, but ACKNOWLEDGEMENT was not, then the entire message could only be tagged as a YES-ANSWER. However, the existence of both tags allows a more accurate, albeit more subjective, representation of the communicative acts in the message by allowing $[yes]^{\text{YES-ANSWER}}$, $[ok]^{\text{ACKNOWLEDGEMENT}}$.

Another case of subjectivity is where a message is segmented into two or more adjacent utterances with the same dialogue-act rather than just one segment; for example, (4) could just be marked as one utterance.

(4) $[\text{Goodbye}]^{\text{CONVENTIONAL-CLOSING}}$ and $[\text{take care}]^{\text{CONVENTIONAL-CLOSING}}$

Gold-Standard data

A gold-standard version of our corpus was created by an expert coder manually performing utterance segmentation and dialogue act classification. To produce this gold standard, messages were first segmented independently of the tag-set, followed

by dialogue act classification. While classifying the utterances, we ensured that appropriate dialogue act tags were available. If no appropriate dialogue act tag existed in our tag set, one was created. An example of a dialogue act tag that was created in this way is EXPRESSIVE, which is used to represent emoticons such as :-).

Not every word was included in an utterance in our gold-standard data, such as the conjunction *and* in (4). Our corpus contains 31 instances of words not included in any utterance. Of these, 24 were instances where the word *customer* was omitted as in $[No\ problem]^{\text{RESPONSE-ACK}}, Customer$. The remaining 7 omissions were conjunctions. Although our segmentation models assume that every word belongs to some utterance, we ignore this discrepancy with the gold standard data because the occurrence of words omitted from utterances is very small.

Inter-Annotator Agreement on Segmentation

To gauge the subjectivity of the segmentation task, we asked three subjects with no formal training in linguistics to segment the messages in our corpus into utterances. The subjects were briefed on what constitutes an utterance before beginning the task. On average, the coders marked an utterance boundary approximately every 6.9 words, which is close to the gold standard mean utterance length of 7.6 words.

Two different measures of inter-annotator agreement were used: Kappa (κ) and mean inter-annotator agreement. The first measure, κ , factors out expected agreement and is calculated as $\frac{P(A)-P(E)}{1-P(E)}$, where $P(A)$ is the observed agreement amongst the coders and $P(E)$ is the expected agreement; that is, the probability that coders agree by chance (Carletta 1996; Di Eugenio and Glass 2004). The second measure is the mean inter-annotator agreement amongst the three codes, which is computed by considering each boundary marked by any annotator and taking the mean of the inter-annotator agreement at that boundary. This also included messages that were judged to be a single utterance and were thus not segmented.

(5) $ok, \overset{A}{|}$ I'll try that $\overset{A,B,C}{|}$

In (5) above, annotator *A* has marked a segment after the word *ok*, breaking the

message into two utterances, whereas annotators B and C regard the entire message as only one utterance. In this example, we assign the scores $\frac{1}{3}$ for the first utterance and $\frac{2}{3}$ for the second, resulting in a final inter-annotator agreement of 0.5. This method of scoring answers the question *where do utterance boundaries occur within a message?* as opposed to *where are the utterances within a message?* There is a subtle difference between the questions: the latter question would result in the probabilities $\frac{1}{3}$, $\frac{1}{3}$, and $\frac{2}{3}$, representing A 's segmentation *[ok,][I'll try that]*, of which each utterance is assigned $\frac{1}{3}$ agreement, and B 's and C 's decision to keep the entire message as one utterance. This second scoring method results in a mean agreement score for the message of $\frac{4}{9} \approx 0.44$. The difference between the two methods is that a $\frac{1}{3}$ score is applied to A 's second utterance, which has the effect of penalising the same disagreement twice: there is no way that A can achieve agreement with B and C having already differed in their decision to keep the message as one utterance. Further, the final segmentation at the end of the message is not a decision any of the coders make, and so should not be given judgement. In other words, the score should be based on the decisions the coders make, which in this case is A deciding to segment the message after the first word and B and C deciding not to segment it. Because of these reasons, we use the first method of scoring, answering *where do utterance boundaries occur within a message?* which does not penalise decisions twice and recognises the implied agreement at the end of a message. This method thus represents inter-annotator agreement more accurately.

Table 4.5 shows some more examples of a message hypothetically segmented in different ways by three coders A , B , and C and the score for each segmentation. The end of each message contains an implicit unanimous agreement. Although it may seem counter-intuitive that examples 4 and 5 have the same score given C 's decision to segment the message into two utterances in 4, this is the intended result: in 4, C 's decision to segment after *try* is given a score of $\frac{1}{3}$ and no other annotator is in agreement. At that point, C no longer gets any score for deciding to leave the entire message as one utterance. In 5, on the other hand, a score of $\frac{1}{3}$ is assigned for C deciding not to segment the message and keep it as one utterance, again, no other annotator is in agreement. If either A or B agreed with C in 4, the score would

	Message	Score
1	ok, I'll try that	$\frac{3}{3} = 1$
2	ok, ^A I'll try that	$\frac{1}{2} \times (\frac{1}{3} + \frac{2}{3}) = \frac{1}{2}$
3	ok, ^A I'll ^{A,B,C} try ^B that	$\frac{1}{3} \times (\frac{1}{3} + \frac{3}{3} + \frac{1}{3}) = \frac{5}{9} \approx 0.55$
4	ok, ^A I'll ^{A,B} try ^C that	$\frac{1}{3} \times (\frac{1}{3} + \frac{2}{3} + \frac{1}{3}) = \frac{4}{9} \approx 0.44$
5	ok, ^A I'll ^{A,B} try that	$\frac{1}{3} \times (\frac{1}{3} + \frac{2}{3} + \frac{1}{3}) = \frac{4}{9} \approx 0.44$
6	ok, ^A I'll ^{A,B} try ^B that	$\frac{1}{4} \times (\frac{1}{3} + \frac{2}{3} + \frac{1}{3} + \frac{1}{3}) = \frac{5}{12} \approx 0.42$

Table 4.5: Method of scoring pairwise inter-annotator agreement for some hypothetical message segmentations by three coders *A*, *B*, and *C*. Segments are marked with ‘|’ followed by the coders who marked that location.

have been increased accordingly. Similarly in example 5, *B*, instead of *C*, decides to segment after *try* and the score is reduced to $\frac{5}{12}$, which is the intended result as *B* no longer agrees with *A* on the second utterance. The algorithm for computing these scores on a per-message basis is shown in Algorithm 2, which is the method we use to score segmentation agreements.

To calculate κ , we used words as our unit to determine the probability of observed inter-annotator agreement $P(A)$ and expected agreement $P(E)$. κ reached 0.8 with $N = 4252$ (words) and $k = 3$ (number of annotators). A score of 0.8 indicates that the segmentation task is sufficiently objective to attempt to automate. Mean inter-annotator agreement on locations where any coder had marked a boundary was 84% with $N = 618$.

Table 4.6 shows the annotator agreement for the segmentation task. Most of the inter-annotator disagreements, shown on the left side of the table, resulted from annotators not marking a similar number of segments as opposed to utterance boundary disagreements. Of all the segments, annotator A marked 37.8%, B marked 33.2%, and C 29%.

Pairwise agreement between each annotator and the gold-standard is shown on the right side of the table. We described earlier the discrepancy in annotation styles between the gold-standard, where not all words were included in an utterance, and the annotator segments, where every word had to be included in an utterance. When

Algorithm 2 Returns a score for the pairwise inter-annotator agreement for any given message without double-penalising segmentation disagreements.

```

procedure SCOREMESSAGE(message)                                ▷ Input message to score
  scores ← [ ]                                                    ▷ Initialise empty array
  annot ← [ ]                                                    ▷ Empty array to store a list of annotators
  for all word in message do
    n ← getSegmentMarksAt(word)
    if n > 0 and not last word then
      scores.insert(n)
      annot.insert(getAnnotsAt(word))    ▷ Annotators that marked word
    else if last word then                                ▷ Implied agreement at end of message
      numAgreed ← 0
      for all a in Annotators do
        if a not in annot then
          numAgreed ← numAgreed + 1
        end if
      end for
      if numAgreed > 0 then
        scores.insert(numAgreed)
      end if
    end if
  end for
  result ←  $\frac{\text{sum}(\text{scores})}{\text{len}(\text{Annotators}) \times \text{len}(\text{scores})}$ 
  return result                                                ▷ result is in the interval [0, 1]
end procedure

```

Annotators	Agreement	Annotator	GS-Agreement
$A \cap B \cap C$	84.2%	A	85%
$A \cap B \cap C'$	6.4%	B	82%
$A \cap C \cap B'$	3.0%	C	87%
$A \cap B' \cap C'$	4.2%		
$B \cap A' \cap C'$	1.8%		
$C \cap A' \cap B'$	0.2%		

Table 4.6: Annotator segmentation agreement for the three annotators. The left side of the table shows inter-annotator agreement. The right side shows pairwise annotator agreement with the gold-standard (GS) produced by an expert coder.

comparing the annotator segmentations with the gold-standard, we handled this difference by ignoring orphaned words (those that were not included in an utterance) and accepting matches where annotator segment markers separated the gold-standard utterances. As an example, both (6-a) and (6-b) would be accepted as a match, but not (6-c). Gold-standard segments are enclosed within brackets “[]” and the annotator segment markers are denoted with pipes “|”.

- (6) a. [Goodbye] | and [take care]
b. [Goodbye] and | [take care]
c. [Goodbye] and [take | care]

As can be seen in Table 4.6, the annotators all agreed with over 80% of the gold-standard annotations; C matched 87%, A 85%, and B matched with 82% of the gold-standard utterance segments.

4.4.3 Lemmatisation

Lemmatisation is a method of mapping a word form to its *lemma* or *lexeme root*, which is based on inflectional morphology rather than derivational morphology. For instance, *running* maps to *run*, but *stocking* only maps to *stock* as a verb, and not when used as a noun. An advantage of lemmatising the data is that it helps to

Original sentence:

This_DT is_VBZ Sanders_NNP and_CC I_PRP look_VBP forward_RB to_TO
 assisting_VBG you_PRP today_NN

Analysed sentence:

this_DT be_VBZ Sanders_NNP and_CC I_PRP look_VBP forward_RB to_TO
 assist+ing_VBG you_PRP today_NN

Figure 4.1: Example lemmatised sentence.

overcome data sparseness problems by reducing the number of unique word tokens while not over-generalising.

Our corpus was lemmatised using the morphological tools described in Minnen *et al.* (2001), which are based on finite-state techniques and implemented using the *flex* Unix utility. The morphological analyser from this tool-set comprises a set of morphological generalisations coupled with a list of specific (irregular) word forms. Using generalisations, or rules, rather than an explicit lexicon means that it is possible to lemmatise new vocabulary items or unknown words with regular morphology. The rules were acquired semi-automatically from several large corpora and machine-readable dictionaries such as the CELEX lexical database of English (version 2.5) (Baayen *et al.* 1995) and the British National Corpus (BNC) (Burnard 2000).

The analyser expects to receive input in the form *wordform_label* where *wordform* specifies the word form to be analysed and *label* is an optional PoS tag for the word form, using the symbol `_` as a delimiter. An example input and output sentence from our corpus is shown in Figure 4.1.

An example generalisation rule is shown in (7):

(7) $\{A\}+\{C\}$ ‘‘ied’’ $\{\text{return(lemma(3, ‘‘y’’, ‘‘ed’’))};\}$

The left-hand side of the rule is a regular expression with $\{A\}+$ denoting at least one occurrence of an element within the predefined character set A, which represents the letters of the Latin alphabet, and C the set of consonants. The “ied” is a string

literal. The regular expression in (7) will match the word *carried*, for instance. The right-hand side is the rule to execute upon a match, which in this case calls the *lemma* function to replace the last three characters of the word by a *y*, followed by the delimiter *+* and the inflection type *ed*. The output is therefore *carry+ed*.

Of course, not every word can be handled by the general rule in (7). Exceptions to the generalisation rules are handled by other rules as in (8), which handles the verb *boogied* by returning *boogie+ed* rather than *boogy+ed*.

```
(8)    ‘‘boogied’’      {return(lemma(1,‘’,‘‘ed’’));}
```

The first parameter in the `lemma` command identifies the last letter in *boogied*, which is then replaced with the empty string `‘’` in the second parameter—effectively deleting it. Finally, the third parameter specifies the string to add to the result, forming *boogie+ed*.

Rules (7) and (8) do not need to use the PoS tags that are available to them. However, there are cases where the PoS tags are necessary for accurate lemmatisation. An example is words with consonant doubling, such as with the past tense inflection of the verb *submit*, which is *submitted*. These are handled with a rule that takes the PoS tags into account as in (9).

```
(9)    {A}+‘‘tted_V’’   {return(lemma(3, ‘’, ‘‘ed’’));}
```

Rule (9) converts the verb *submitted* to *submit+ed* using the *lemma* function to replace the last three characters with an empty string and adding *+ed*. The morphological analyser maintains a list of verbs that undergo consonant doubling to ensure the rule is only applied when necessary.

4.4.4 PoS Tagging and Chunking

Part of speech (PoS) tagging is the task of assigning part of speech tags, such as nouns, verbs, adjectives, etc., to words. We used the Penn Treebank Project PoS tag set, which consists of 48 tags such as NNS (Noun, plural), NN (Noun, singular or

mass), NNP (Proper noun, singular), et cetera (Marcus *et al.* 1993). PoS tags can then be used in tasks such as chunking and parsing. Our data was assigned PoS tags via the fnTBL Toolkit (Ngai and Florian 2001), which is an efficient implementation of Brill’s (1995) transformation-based learning (TBL) algorithm.

The TBL algorithm by Brill (1995) is a rule-based approach to automated learning. For PoS-tagging, the system is trained using the Penn Treebank Wall Street Journal corpus (Marcus *et al.* 1993). Training on this data is not ideal for instant messaging dialogue since, as already discussed, instant messaging contains many abbreviations, acronyms, and spelling mistakes not found in newspaper text. However, because our corpus was from customer support, the language usage was not as casual as more socially-oriented instant messaging.

The learning process used by TBL iterates through a set of transformation rules, assigning PoS tags to each word in the corpus and computes the accuracy after each iteration. If the resulting accuracy is significantly more accurate than what existed before applying the rule, that rule is saved in an ordered list and the process repeats with the next rule. When no further significant improvement in accuracy is made through additional transformations, the learning process ends and the ordered list of resulting transformations becomes the rule set with which to tag new data.

Brill (1995) showed that decision trees are a special case of transformation lists and that transformation lists have certain advantages over decision trees, particularly in respect to being more resistant to sparse data problems. The transformation rules come from a list of eight templates, three of which are in the following example:

Change tag **a** to tag **b** when:

1. The preceding (following) word is *w*.
2. The word two before (after) is *w*.
3. One of the two preceding (following) words is *w*.

The reported PoS tagging accuracy in Brill (1995) is 96.6% on the Penn Treebank Wall Street Journal corpus. The fnTBL Toolkit optimises the training time of the

A: [INTJ hello_{UH}] [NP customer_{NN}] ,O <s> [VP thank_{VB}] [NP you_{PRP}] [PP for_{IN}] [VP contact_{VBG}] [NP Msn_{NNP} Shopping_{NNP}] .O <s> [NP this_{DT}] [VP be_{VBZ}] [NP Sanders_{NNP}] [O and_{CC}] [NP I_{PRP}] [VP look_{VB}] [ADVP forward_{RB}] [PP to_{TO}] [VP assist_{VBG}] [NP you_{PRP}] [NP today_{NN}] .O

A: [ADVP how_{WRB}] [O be_{VB}] [NP you_{PRP}] [VP do_{VBG}] [NP today_{NN}] ?O

B: [ADJP good_{JJ}] ,O <s> [NP thanks_{NNS}]

Figure 4.2: Sample PoS-tagged and chunked data from the MSN Shopping corpus. Utterance boundaries are marked by <s>, chunks are enclosed in brackets, chunk tags are the first tag within each chunk, and each word is followed by its PoS tag.

original TBL process.

Chunking consists of grouping words into phrases based on some head word. Typically, chunks are made up of a single content word, the *head word*, with other function words. Chunks were initially proposed by Abney (1991) as a precursor to creating parse trees in order to improve performance and was the Conference on Computational Natural Language Learning (CoNLL) shared task in 2000.

The chunking task was modelled as a classification task and performed using the same algorithm we used for PoS tagging, the fnTBL toolkit, and trained on the Penn Treebank (Marcus *et al.* 1993).

Sample Data

Chunks in our corpus are in the following format: [INTJ hello_{UH}]. The chunk is enclosed in brackets, the first label (INTJ) is the chunk’s label, in this case indicating an *interjection* chunk. This chunk contains only one word, *hello*, followed by its part of speech tag (UH). Referring to the Penn Treebank tagset in Appendix F, we can see that UH indicates an interjection.

Figure 4.2 illustrates some more characteristics of our data after PoS tagging, lemmatisation, and chunking. Utterance boundaries are marked by <s> tags, chunks are enclosed within brackets, and each word is followed by its PoS tag. The actual chunks in the data use IOB tags similar to those described in Ramshaw and Marcus

(1995).

4.5 Conclusion

This chapter began with a description of our dialogue act tag set, which we used to classify utterances in our corpus. We then discussed the process used to collect a suitable instant messaging corpus for our task. Eight dialogues were collected using the *MSN Shopping* assistance web site. The conversations were held with the aim of finding items to buy from online merchants. The *MSN Shopping* data is appropriate for study since it is task-based and conducted via an instant messaging interface. We use this data for the classification and segmentation tasks described in this thesis.

We described the data preparation required for the classification and segmentation tasks. The problem of message synchronisation was discussed, where users may accidentally interrupt each other by mistaking the end of a multi-message turn. We presented an approach to the problem that re-synchronises messages achieving 92.5% accuracy.

Segmentation of our *MSN Shopping* corpus was performed manually by an expert coder to produce a gold standard. We also asked three volunteers to segment messages so that we could calculate mean inter-annotator agreement and the Kappa coefficient, κ . A κ calculation resulted in a score of 0.8, which is evidence that the segmentation task can be accomplished reliably and used for training and evaluation.

Finally, we described lemmatisation, part of speech tagging, and chunking, which are used as input for our segmentation models discussed in the next chapter.

Having prepared our corpus, we are ready to train and evaluate our segmentation and classification models. The next chapter discusses the segmentation and classification tasks in detail.

Chapter 5

Utterance Segmentation and Dialogue Act Classification

When messages are received via instant messaging, they may contain more than one communicative act, or utterance. In the example “ok, I’ll try that”, we have the two utterances: “ok” and “I’ll try that”. We may represent the role that each utterance plays by assigning it a tag such as RESPONSE-ACKNOWLEDGEMENT for “ok” and STATEMENT for “I’ll try that”. However, we must first break the message into two utterances before assigning the dialogue act tags to them.

Since dialogue acts work at the utterance level, messages must first be segmented into utterances before the resulting utterances can be classified into dialogue acts. Several techniques were used to automatically segment messages into utterances using two models. The first was a Hidden Markov Model (HMM) with various features, and the second was based on the parse tree of each message. The approaches used with the HMM were based on sequences of lemmas, part of speech tags, and head words of chunks.

Our first goal was to determine which features obtained from IM transcripts would be useful in detecting utterance segments within messages. The data available from IM chat transcripts are the speaker, message text, and time stamp of each message. We could not draw from existing research on using features for non-sentential utterance segmentation from spoken dialogue, such as Lleida and Rose (2000), Johnson

(1997), and Ladefoged (1996), since the data available to us are very different to that in recorded speech, which includes prosody information.

We observe that utterances in our data do not cross message boundaries; that is, users send messages only after utterances are fully typed. This allows us to process each message in isolation rather than merging words from adjacent messages to form complete utterances, which would be the case if utterances were split across messages. The average message length in our corpus is 10.2 words. Two approaches are used for the segmentation task: Hidden Markov Models (HMMs) and a probabilistic model based on parse trees. The HMM method is described in this section as it only does segmentation, which we consider a data preparation task for classification. The Parse Tree method is described in Chapter 5 as it also performs dialogue act classification with the segmentation.

This chapter turns to the methods used to segment and classify our instant messaging corpus. A description of the dialogue act classification methods we evaluated is first presented, followed by techniques to perform utterance segmentation.

5.1 HMM Utterance Segmentation Method

In the absence of reliable punctuation cues, we looked at approaches based on the available lexical information of a message. One such approach was to use a hidden Markov model (HMM) to find the most likely segment boundaries. HMMs are stochastic models that extend the theory of Markov chains to predict *hidden states* based on *observations*. We experimented with three kinds of input, or observations, for the HMM, which were sequences of: (i) lemmas, (ii) part of speech (PoS) tags, and (iii) head words of chunks, as shown in Figure 5.1.

Each observation is paired with a hidden state, being a subsequent *segment* or *non-segment* marker. Figure 5.1 shows the message “ok, I’ll try that” represented in the different observations and states we use. More detail about HMMs can be found in Rabiner (1990).

The rationale behind using chunks in the segmentation task is to reduce the number of possible segments as we observe that utterance boundaries do not lie within

Message: “[ok,] [I’ll try that]”

Lemma version: “ok (S) I (N) will (N) try (N) that (S)”

PoS tag version: “NN (S) PRP (N) MD (N) VB (N) IN (S)”

Chunk version: “ok (S) I (N) try (N) that (S)”

Figure 5.1: Data used to segment the message “ok, I’ll try that” using lemmatised data, PoS tags, and chunked data with an HMM. (S) and (N) represent the state of the preceding token being a segment or non-segment marker respectively.

chunks and chunks may consist of multiple words.

We first trained an n -gram language model with add-one smoothing and Katz backoff (Katz 1987) to hypothesize the most probable locations of utterance boundaries for each individual message. The resulting segmentations were then evaluated using the WindowDiff metric as described in Section 6.2.1.

Elements used to represent the segments were lemmas, PoS tags, and chunks. Segment beginnings in our training data were marked with a <s> tag. This allowed each element to be in one of two states: S or NO-S depending on whether or not it had a <s> tag before it. We build two probability distributions P_S and P_{NO-S} representing the probability that token t_k is at the beginning of a segment or not, respectively. Using this state information permits us to use an HMM with the following forward computation for the likelihoods of the states at each position k . Stolcke and Shriberg (1996) used the following formula for segmentation only considering word tokens, whereas we use t to represent lemma, PoS, or chunk tokens:

$$\begin{aligned}
 P_{NO-S}(t_1 \dots t_k) &= P_{NO-S}(t_1 \dots t_{k-1})p(t_k | t_{k-2} t_{k-1}) \\
 &\quad + P_S(t_1 \dots t_{k-1})p(t_k | \text{<s>} t_{k-1}) \\
 P_S(t_1 \dots t_k) &= P_{NO-S}(t_1 \dots t_{k-1})p(\text{<s>} | t_{k-2} t_{k-1})p(t_k | \text{<s>}) \\
 &\quad + P_S(t_1 \dots t_{k-1})p(\text{<s>} | \text{<s>} t_{k-1})p(t_k | \text{<s>})
 \end{aligned}$$

The Viterbi algorithm (Jurafsky and Martin 2000:177) is then used to find the most likely sequence of S and NO-S states given the observations.

5.2 Dialogue Act Classification

Given all available evidence E about a dialogue, we aim to find the dialogue act sequence S with the highest posterior probability $P(S|E)$ given that evidence. We refer to the process of identifying the optimal S for a given E as utterance segmentations. We apply four statistical models to the task of utterance segmentation, namely: a naive Bayes model, vector space model, a support vector machine (SVM), and a maximum entropy (maxent) model, as will be described in this section. We developed our own learners for the naive Bayes and vector space models, and used Chang and Lin’s (2001) implementation for the SVM learner and Le’s (2001) implementation for the maxent learner.

For training, these models require the messages to be marked up with dialogue acts, while the dialogue act boundaries serve to indicate the utterance boundaries within each message. Therefore, no further data preparation was required. The SVM and maxent models, however, required that our corpus be converted into the *sparse matrix format* for training and testing.

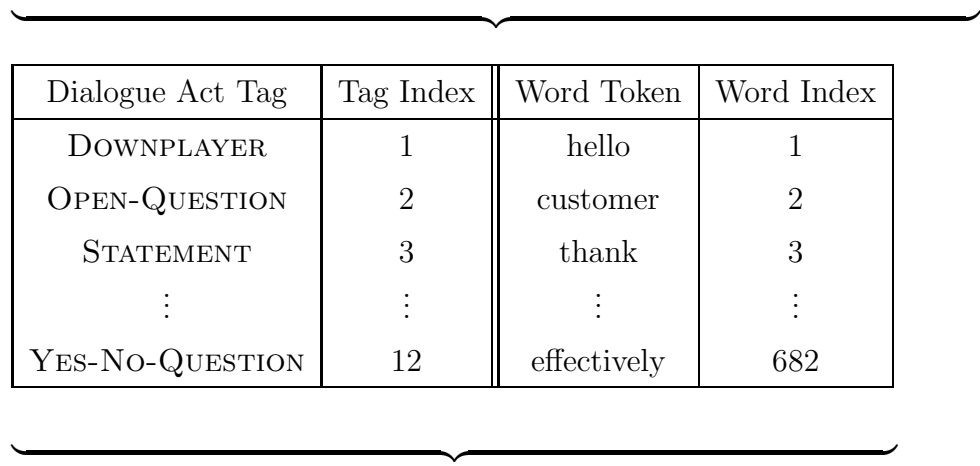
The sparse matrix format consists of one line per data instance, which in our case is an utterance. An example illustrating the mapping of dialogue data in our corpus to the sparse matrix feature representation is shown in Figure 5.2. Each data instance begins with a class label and followed by feature attributes in the format $\mathbf{f}:\mathbf{c}$, where \mathbf{f} is the feature and \mathbf{c} is the number of times that feature occurs. For our purposes, the dialogue act is the class label and the word tokens within the utterance are features. Since we regard each word token as a feature, the count \mathbf{c} is always 1. The dialogue acts and word tokens were represented by indexes into two dictionaries of all the dialogue acts and words tokens in our corpus, respectively.

The first number of each line is the index into the set of dialogue acts used, corresponding to the dialogue act for the utterance: in the sparse matrix data format in Figure 5.2, 5 is CONVENTIONAL-OPENING, 7 is THANKING, and 3 is STATEMENT. There is no specific ordering to the list of dialogue acts, but the indexing must be consistent across the training and testing data. Similarly, the numbers following the dialogue act index are also indexes into a dictionary of all the words in our corpus.

CONVENTIONAL-OPENING Hello Customer

THANKING thank you for contacting MSN Shopping

STATEMENT This is Sanders and I look forward to assisting you today



Dialogue Act Tag	Tag Index	Word Token	Word Index
DOWNPLAYER	1	hello	1
OPEN-QUESTION	2	customer	2
STATEMENT	3	thank	3
⋮	⋮	⋮	⋮
YES-NO-QUESTION	12	effectively	682

5 1:1 2:1

7 3:1 4:1 5:1 6:1 7:1 8:1

3 9:1 10:1 11:1 12:1 13:1 14:1 15:1 16:1 17:1 4:1 18:1

Figure 5.2: Feature representation mapped from corpus data. The format of the data is DIALOGUE-ACT Word-Index:1

The example in Figure 5.2 shows the first three lines from the first dialogue, so the word-index numbers increment by one for each position since every word is unique until the second-last word in the third utterance, ‘to’, which happens to also occur as the fourth word in the corpus—second word in the second utterance. The word index ordering in subsequent utterances becomes much less predictable.

We now discuss each of these models in turn.

5.2.1 Naive Bayes Model

The naive Bayes model was implemented using a *bag-of-words* feature representation such that the most probable dialogue act \hat{d} given a bag-of-words input vector \bar{v} is taken to be:

$$\hat{d} = \arg \max_{d \in D} \frac{P(\bar{v}|d)P(d)}{P(\bar{v})} \quad (5.1)$$

$$P(\bar{v}|d) \approx \prod_{j=1}^n P(v_j|d) \quad (5.2)$$

$$\hat{d} = \arg \max_{d \in D} P(d) \prod_{j=1}^n P(v_j|d) \quad (5.3)$$

where v_j is the j th element in \bar{v} , D denotes the set of all dialogue acts and $P(\bar{v})$ is constant for all $d \in D$.

The use of $P(d)$ in Equation 5.3 assumes that dialogue acts are independent of one another. However, we intuitively know that if someone asks a YES-NO-QUESTION then the response is more likely to be a YES-ANSWER rather than, say, CONVENTIONAL-CLOSING. This intuition is reflected in the bigram transition probabilities obtained from our corpus as shown in Appendix C.

To capture this dialogue act relationship we trained standard n -gram models of dialogue act history with add-one smoothing for the calculation of $P(v_j|d)$. The bigram model uses the posterior probability $P(d|h)$ rather than the prior probability $P(d)$ in Equation 5.3, where h is the history represented by an n -gram context vector containing the previous dialogue act or previous 2 dialogue acts in the case of the

trigram model.

We expect adjacent dialogue acts to be different depending on whether they come from the same or different speaker. When speaker A asks a YES-NO-QUESTION, for example, speaker B will likely respond with either a YES-ANSWER or a NO-ANSWER. However, if speaker A says two utterances in a row, we expect the adjacency pairs to be different. An example of such a sequence appears at the beginning of most of our dialogues, where the opening usually begins with the following sequence of dialogue acts from the agent: CONVENTIONAL-OPENING, THANKING, STATEMENT as shown in the first message in Table 4.2. To model these differences, we experimented with including speaker turns with the posterior probability in the bigram and trigram models, arriving at the following equation:

$$\hat{d} = \arg \max_{d \in D} P(d, s|h) \prod_{j=1}^n P(v_j|d) \quad (5.4)$$

where s is a binary variable signifying whether dialogue act d_i is by the same or different speaker as the previous dialogue act d_{i-1} and denoted by “same” or “diff”. The results from these models are presented and discussed in Section 6.1.1.

5.2.2 Vector Space Model

The vector space model maps documents and queries into vectors of features, then uses the cosine similarity function to match a query vector with the most similar document vector (Salton 1971). Documents were created by grouping utterances across our corpus into their respective dialogue acts. We thus produced 12 documents, one for each of the dialogue acts shown in Table 4.1. The utterances to be classified were taken as the queries.

The features we use in this model take the bag-of-words approach in a similar manner to that of the naive Bayes model described earlier.

The vectors are represented as follows:

$$\begin{aligned}\vec{d}_j &= (w_{1,j}, w_{2,j}, w_{3,j}, \dots, w_{n,j}) \\ \vec{q}_k &= (w_{1,k}, w_{2,k}, w_{3,k}, \dots, w_{n,k})\end{aligned}$$

where \vec{d}_j denotes the vector for document j , \vec{q}_k denotes the vector for query k , $w_{i,j}$ represents the weight of term i in document j with n weights in the document collection. From this, we can create a term-by-document matrix where the columns represent the documents, and the rows represent the terms.

Document weighting was done using the *inverse document frequency* method originally described by Sparck Jones (1972). Using this method, we assign higher weights to terms that appear in fewer documents, since it is the infrequent terms that play a greater role in differentiating a document compared to more common terms such as *the* and *a*, which we would expect to occur in most or all documents. We use the fraction $\frac{N}{n_i}$ to inversely weight the frequency of a term, where N is the total number of documents, 12 in our case, and n_i is the number of documents that contain term i . Where document collections are very large, the measure $idf_i = \log(\frac{N}{n_i})$ is commonly used to reduce the resulting range. However, with only 12 documents in our model, the log function was deemed unnecessary and we simply use $idf_i = \frac{N}{n_i}$ instead. Experimenting with the log function led to slightly worse results.

The idf_i is combined with the term frequency tf to obtain the final $tf \cdot idf$ weighting as follows:

$$w_{i,j} = tf_{i,j} \cdot idf_i \quad (5.5)$$

where $w_{i,j}$ is the weight of term i in document j .

Weighting queries is performed using a different method: by normalising the term weights within the query using the formula:

$$w_{i,k} = \frac{\sum_{i=1}^N tf_{i,k}}{\sqrt{\sum_{i=1}^N tf_{i,k}^2}} \quad (5.6)$$

where $w_{i,k}$ is the weight of term i in query k and $tf_{i,k}$ is the raw term frequency of i in query k .

Since both query and document vectors are normalized, we assess the similarity between them by calculating the dot product, which is equivalent to the cosine of the angles between \vec{q}_k and \vec{d}_j , with the following equation:

$$\text{sim}(\vec{q}_k, \vec{d}_j) = \vec{q}_k \cdot \vec{d}_j = \sum_{i=1}^N w_{i,k} \times w_{i,j} \quad (5.7)$$

5.2.3 Support Vector Machine Model

Support Vector Machines (SVMs), introduced by Boser *et al.* (1992), are binary classification algorithms based on kernel methods. Kernels are similarity measures from a representation of *patterns*, where a pattern represents some data from the *domain* \mathcal{X} , which is some nonempty set. The domain in our case is the set of utterances in our corpus.

The goal of SVMs is to predict a class given only a set of features. Training takes a set of data instances where each instance consists of a set of features and a class label. Given empirical data $(x_1, y_1), \dots, (x_m, y_m) \in \mathcal{X} \times \{\pm 1\}$, where x_i are taken from the domain \mathcal{X} and y_i are labels, we want to classify some new pattern $x \in \mathcal{X}$ into its corresponding label $y \in \{\pm 1\}$. However, to do so requires some similarity measure with which to compare the new x with the instances from the training set. Note that the labels y are restricted to the set $\{-1, +1\}$, making this a binary classification problem. We will shortly describe a technique for enabling SVMs to perform multi-class classification.

A similarity measure is of the form:

$$\begin{aligned} k : \mathcal{X} \times \mathcal{X} &\rightarrow \mathbb{R} \\ (x, x') &\mapsto k(x, x'), \end{aligned}$$

where $x, x' \in \mathcal{X}$ (Schölkopf and Smola 2001). x and x' are two patterns that are compared by function k , the *kernel*, which returns a real number characterising their similarity. The similarity measure is based on a geometric interpretation of the features,

such as the dot product in 5.8, also shown in 5.7 using slightly different notation.

$$(\mathbf{x} \cdot \mathbf{x}') := \sum_{i=1}^N (\mathbf{x})_i (\mathbf{x}')_i \quad (5.8)$$

where $(\mathbf{x})_i$ denotes the i th entry of \mathbf{x} .

In order to perform such geometrical comparisons of the features, the patterns must first be mapped into some dot product space \mathcal{H} , also called a *feature space*. This is accomplished using a map:

$$\begin{aligned} \Phi : \mathcal{X} &\rightarrow \mathcal{H} \\ x &\mapsto \mathbf{x} \end{aligned}$$

so we can now define the similarity measure from the dot product in \mathcal{H} as

$$k(x, x') := (\mathbf{x} \cdot \mathbf{x}') = (\Phi(x) \cdot \Phi(x')) \quad (5.9)$$

Using Φ allows for a choice of mapping functions to obtain the best results given the type of data. SVMs can thus map the training vectors into a higher dimensional space for nonlinear pattern matching. We used the SVM classification tools by Chang and Lin (2001) with the radial basis function kernel, which is a non-linear kernel.

The binary classification using $y \in \{\pm 1\}$ has been extended to support multi-class classification using several techniques. One of these techniques is the *one-against-one* approach. This approach trains a binary SVM for each combination of the labels and obtains a decision function. A k -class problem will thus have ${}^kC_2 = k(k-1)/2$ decision functions. In our case, we have 12 dialogue acts, which results in ${}^{12}C_2 = 66$ decision functions. A voting strategy is used where each SVM essentially casts one vote and the class with the most votes is ultimately chosen (Schölkopf and Smola 2001).

Extensive literature about SVMs is available, such as Vapnik (1995) and Müller *et al.* (2001).

5.2.4 Maximum Entropy Model

Maximum entropy (maxent) models are statistical models that can estimate a probability distribution given the sparse evidence typically used as training data. A

probability model $p(a, b)$ is selected using the *Principle of Maximum Entropy*, which states that the selection of the correct probability distribution $p(a, b)$ must be one which maximises entropy, or “uncertainty”, subject to the constraints imposed by the training data. This principle ensures that the model is unbiased as only the training data constrains the distribution and no arbitrary assumptions are made about the model. Formally, if \mathcal{A} denotes the set of possible classes, and \mathcal{B} denotes the set of possible contexts, p should maximise the entropy:

$$H(p) = - \sum_{x \in \mathcal{E}} p(x) \log p(x) \quad (5.10)$$

where $x = (a, b)$, $a \in \mathcal{A}$, $b \in \mathcal{B}$, and $\mathcal{E} = \mathcal{A} \times \mathcal{B}$, and should remain consistent with the evidence seen in the training data (Ratnaparkhi 1997).¹

From Berger *et al.* (1996), Equation 5.10 can be re-written to work with conditional probabilities rather than joint probabilities as:

$$H(p) = - \sum_{a,b} \tilde{p}(a)p(b|a) \log p(b|a) \quad (5.11)$$

where \tilde{p} is the empirical probability distribution based on the training sample.

In our case, \mathcal{A} is the dialogue act tag set and \mathcal{B} is the set of possible words and emoticons that may exist in any utterance, represented as *features*. The utterances in our training data are encoded as features using the sparse data format described in Section 5.2. Each feature is assigned a weight to control the extent to which it contributes towards $p(a, b)$. The weights are determined by any of a number of algorithms that maximise entropy over the observed data during training, a process known as *parameter estimation*. The weights are then multiplied by the features from the testing data during classification to obtain the probability of each label being assigned to each feature, with the maximum probability ultimately selected.

We used the maxent tool provided by Le (2001) to build our classifier and the *Generalized Iterative Scaling* algorithm for parameter estimation. Malouf (2002) describes and compares several parameter estimation algorithms, including *Generalized*

¹A more common notation for entropy is $H(X)$ where X is a random variable with probability distribution $p(x)$. We use the alternate notation $H(p)$ here to be consistent with the cited references, which emphasise the dependency of the entropy on the probability distribution p .

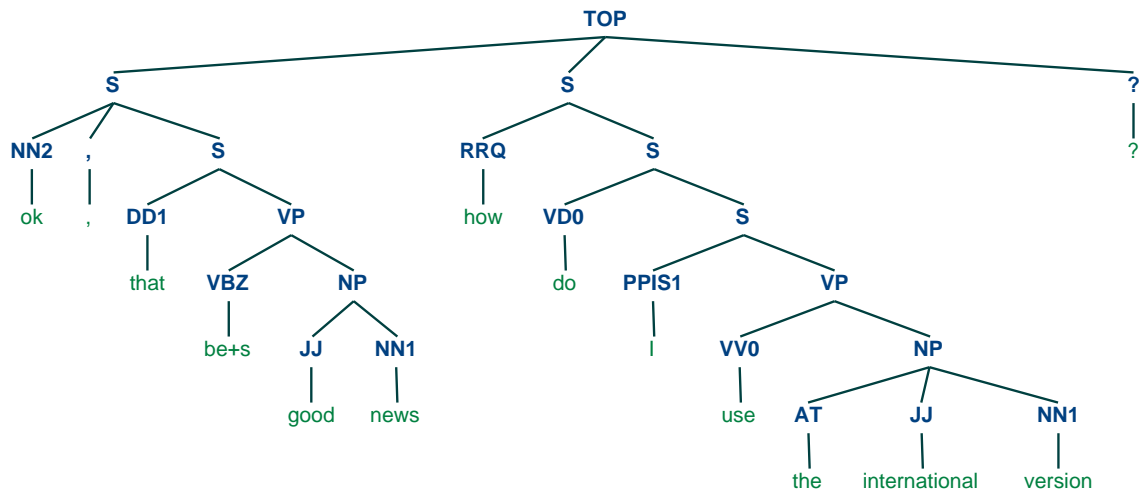


Figure 5.3: RASP parse tree of a message showing utterances separated into sub-trees.

Iterative Scaling. An introduction and more information on maxent models can be found in Ratnaparkhi (1997).

5.3 Segmentation and Classification using Parse Trees

Parse tree nodes generally contain phrases as illustrated in Figure 5.3. The parse tree segmentation method assumes that, based on observation in our corpus, utterance boundaries only occur at major syntactic boundaries. Taking advantage of this characteristic, we search for utterance boundaries at various phrase boundaries identified by the parser. This is a similar approach to the use of chunks described in Section 5.1, where we hypothesise that a segment boundary exists before each chunk. With the parse tree method, the notion of a chunk changes to represent phrases within a parse tree. Using nodes in parse trees to hypothesise segment boundaries significantly reduces the possibility of obtaining false-positives when compared with hypothesising segments at word or chunk boundaries in Section 5.1. This is because a sub-tree generally represents multiple words and utterance segments may only occur between sub-trees. If the sub-trees group a larger number of words than the chunks

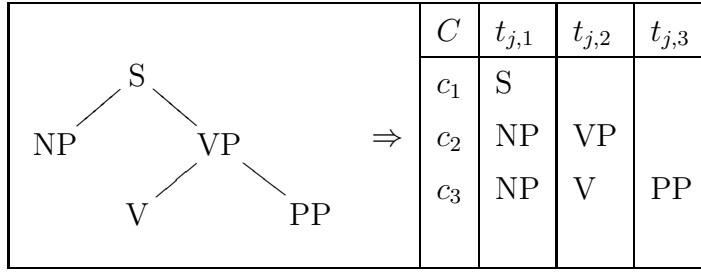


Figure 5.4: Proper analyses, C_1^3 from the illustrated parse tree.

used in the HMM method, our search space is reduced since there are fewer possible segment boundaries. Of course, this is only true if the parse trees are correct.

We parsed the messages in our *MSN-Shopping* corpus using RASP, which stands for Robust Accurate Statistical Parsing (Briscoe and Carroll 2002). RASP is a domain-independent statistical parser that accepts unannotated text as input and performs tokenisation, PoS tagging, lemmatisation, and parsing using separate modules for each task. Both statistical and finite-state methods are used depending on the module.

Message boundaries were taken to be hard boundaries and RASP was free to perform sentence tokenisation within messages. After a manual inspection of the resulting parse trees, we observed that the utterance segments always occurred within a maximum depth of 2 nodes from the parse tree root. We used this depth limit in building a table of possible “cuts” through the tree. These cuts, or proper analyses (Chomsky and Miller 1963), contain every combination of sub-trees, as illustrated in Figure 5.4, resulting in a sequence C of nodes:

$$C = c_1, c_2, c_3, \dots, c_h \quad (5.12)$$

where c_i is a sequence of tree nodes such that:

$$c_i = t_1, t_2, t_3, \dots, t_n \quad (5.13)$$

where the leaves, or words, of each sub-tree t_i represent a possible utterance. An example of these sub-tree combinations is shown in Table 5.1.

We then calculate the most likely dialogue act for the leaves (words) within each node using the naive Bayes classifier described in Section 5.2.1, which returns the

C	Nodes t	Utterances
c_1	[TOP]	[ok that be+s good news how do I use the international version]
c_2	[S][S]	[ok that be+s good news] [how do I use the international version]
c_3	[S][RRQ][S]	[ok that be+s good news] [how] [do I use the international version]
c_4	[NN2][S][S]	[ok] [that be+s good news] [how do I use the international version]
c_5	[NN2][S][RRQ][S]	[ok] [that be+s good news] [how] [do I use the international version]

Table 5.1: Hypothesised segmentations from sub-tree node combinations $C = c_1, c_2, \dots, c_5$ for the parse tree in Figure 5.3.

most likely dialogue act and its corresponding probability in the interval 0 to 1. The resulting dialogue act and its corresponding probability are stored with the node t_i . Next, we calculate the probability of a correct sequence of utterances based on dialogue act bigrams and the product of the dialogue-act classification probabilities, using the following formulae:

$$\langle c^*, \mathbf{d}^* \rangle = \arg \max_{\langle c \in C, \mathbf{d} \rangle} \prod_{t_i \in c} \hat{P}(d_i | t_i, d_{i-1}) \quad (5.14)$$

$$\hat{P}(d_i | t_i, d_{i-1}) = P(d_i | d_{i-1}) \prod_{w \in \text{leaves}(t_i)} P(w | d_i) \quad (5.15)$$

where c^* is the best node combination (or sequence of segments), \mathbf{d} is the sequence of dialogue acts, C is the set of proper analyses at depth ≤ 2 , $P(d_i | t_i)$ is the probability of node $t_i \in c$ being dialogue-act d based on its leaves (words), d_{i-1} is the previously assigned dialogue-act (using bigrams), and w is a word in node t_i .

Using this method has the effect of evaluating the classification and segmentation tasks at the same time, taking the most probable combination. Algorithm 3 shows the process used to find the best proper analysis in C . The `classify` method returns

the highest probability of all dialogue acts given the words in node n using the naive Bayes method. It also returns the corresponding dialogue act which is then stored with the respective node n .

Algorithm 3 Find best utterance segmentations $c \in C$. The **classify** method also returns the best dialogue acts and probabilities which are stored with their nodes n .

```

1:  $max_p \leftarrow 0$  ▷ stores the best probability
2:  $max_c \leftarrow \text{None}$  ▷ stores the best tree node combination
3: for all  $c$  in  $C$  do
4:    $p \leftarrow 1$ 
5:   for all  $n$  in  $c$  do
6:      $p \leftarrow p \times \text{classify}(\text{leaves}(n))$ 
7:   end for
8:   if  $p > max_p$  then
9:      $max_p \leftarrow p$ 
10:     $max_c \leftarrow c$ 
11:   end if
12: end for

```

Importantly, our implementation of the naive Bayes algorithm uses a bag-of-words as its features, multiplying the probability of each word in the dialogue act with the probability of each other word using Equation 5.3. This allows the product in line 6 of Algorithm 3 to be used as a ranking score amongst the proper analyses even though the number of nodes n in c may vary within C . If a different classification algorithm were used, then line 6 may have to be modified to preserve mathematical integrity.

5.4 Conclusion

In this chapter, we have described the models used to segment messages into utterances and classify utterances into dialogue acts. For the classification task, we trained naive Bayes, vector space, SVM, and maxent classifiers. We described the

sparse matrix format, which was required by the SVM and maxent models used for the dialogue act classification task.

For the segmentation task, we used HMMs and a linguistically-informed parse tree model. Some of the segmentation models were designed to take advantage of the observation that utterances have a greater probability of occurring at certain syntactic boundaries. We use this knowledge to reduce the search space by only hypothesising utterance boundaries at plausible syntactic boundaries rather than at every word.

Chapter 6 evaluates and compares the results of each of these models. We also present a mock customer support system which uses the models described in this thesis to simulate a dialogue and suggest responses to the customer support agent.

Chapter 6

Evaluation

The previous chapter discussed the problem of message segmentation and utterance classification: a user might send the message “ok, I’ll try that”, which must be segmented into the two utterances “ok” and “I’ll try that” before each of the utterances are then classified into the dialogue acts `RESPONSE-ACKNOWLEDGEMENT` and `STATEMENT`, respectively. We described the four models we use to perform the classification task, namely: naive Bayes, vector space, support vector machine, and maximum entropy models. Finally, we presented a parse tree model that combines the segmentation and classification tasks to find the best combination of segments and dialogue acts for any given message. Each of the segmentation and classification methods described in the previous chapter are evaluated here.

This chapter begins with a discussion of the dialogue act classification task, presents results for each of the models used, then moves on to discuss utterance segmentation. The topics are presented in this order as the segmentation task we use relies on the utterance classification models, so it helps to appreciate the classification task first. Section 6.1.1 discusses the naive Bayes model in more detail as it is the main model we use. Section 6.1.2 provides an error analysis of the classification task showing where common misclassification errors occurred and why.

In Section 6.2, we discuss the utterance segmentation task and the difficulties with providing a segmentation evaluation metric. We then describe an evaluation metric that rewards near-missed boundaries and is able to meaningfully rank different seg-

mentation models from best to worst. Using this metric, we present the segmentation results from our models. To complete the research, we bring the various modules together as part of a proof-of-concept program which assists a human agent in support services. It does this by making suggestions as to what the next utterance should be based on the dialogue history. The program is evaluated by analysing how accurate its suggestions were compared to what the human agent wanted to say.

6.1 Dialogue Act Classification

Section 5.2 described the four dialogue classification models we evaluated: the naive Bayes model using bigrams, vector space model, SVM model (Chang and Lin 2001), and Maxent (Le 2001) model. Each of the models was evaluated using 8-fold cross validation with the gold standard manually segmented data in our dialogue corpus.

k -fold cross validation is a process by which a corpus, or data, is divided into k subsets. The model being evaluated is then trained on $k - 1$ of the sets and the remaining set of data is used for testing. The process is repeated k times where each iteration uses a different set for testing. This ensures that every data point is in a test set exactly once and in a training set exactly $k - 1$ times. The k results are then averaged, which provides fairer results and reduces the effects of sparse data. We treat each of our eight dialogues as a subset of the corpus, thus we use $k = 8$.

Table 6.1 shows the results for each model. Each model was evaluated using the pre-segmented version of our corpus as discussed in Chapter 4. The Minimum, Maximum, and Mean columns are obtained from the cross-validation technique used for evaluation. The baseline used for this task was to assign the most frequently observed dialogue act to each utterance, namely, STATEMENT.

We decided to use the naive Bayes model as our main model because it is a relatively simple model that consistently produced very good results; even the more complex SVM and Maxent models produced only marginally better classification result. The simplicity of the naive Bayes model makes it easy to integrate with the segmentation task without significantly sacrificing accuracy. Since the naive Bayes

Model	Minimum	Maximum	Mean
Baseline	—	—	36.0%
Naive Bayes	75.0%	92.4%	81.8%
Vector Space	67.2%	80.8%	73.0%
SVM	67.4%	91.1%	81.7%
Maxent	73.3%	93.3%	82.8%

Table 6.1: Results for all classification algorithms using 8-fold cross validation showing minimum, maximum, and mean results obtained during cross validation evaluation.

model was used as the main classification model in the segmentation task, it is discussed in more detail below.

6.1.1 Naive Bayes Model

Evaluation of the model was conducted via 8-fold cross validation across the eight dialogues in our corpus. Table 6.2 shows the results of running the experiment with various models replacing the prior probability $P(d)$ in Equation 6.2 with $P(d|h)$ where h contains a fixed size history of dialogue acts as described in Section 5.2.1. Equations 6.2 and 6.1 are repeated here from Section 5.2.1 for convenience.

$$P(\bar{v}|d) \approx \prod_{j=1}^n P(v_j|d) \quad (6.1)$$

$$\hat{d} = \arg \max_{d \in D} P(d) \prod_{j=1}^n P(v_j|d) \quad (6.2)$$

where v_j is the j th element in \bar{v} (the words in the current utterance) and D denotes the set of all dialogue acts.

We tested the accuracy of the bag-of-words approach by using only the likelihood in Equation 6.1, which omits $P(d)$ from Equation 6.2. The likelihood resulted in a mean accuracy of 80.1%, which was quite high given the dialogue-act n -gram results in Table 6.2. The high accuracy obtained with only the likelihood reflects the high dependency between dialogue acts and the actual words used in utterances. This

Model	Minimum	Maximum	Mean	Hit %	Perplexity
Baseline	—	—	36.0%	—	—
Likelihood	72.3%	90.5%	80.1%	—	—
Unigram	74.7%	90.5%	80.6%	100	7.7
Bigram	75.0%	92.4%	81.8%	97	4.7
Trigram	69.5%	94.1%	80.9%	88	3.3

Table 6.2: Mean accuracy of labelling utterances with dialogue acts using n -gram models; shown with hit-rate results and perplexities.

dependency is represented well using the bag-of-words approach. Including $P(d)$ to arrive at Equation 6.2 yields a slight increase in accuracy to 80.6%.

The bigram model obtains the best result with 81.8% mean accuracy. This result is due to more accurate predictions with $P(d|h)$ as shown in Equation 6.3. The trigram model resulted in slightly lower accuracy, which was partly due to a lack of training data and to dialogue act adjacency pairs not being dependent on dialogue acts further removed, as discussed in Section 3.4. The bigram model uses d_{i-1} for the dialogue act history h , whereas the trigram model uses d_{i-1}, d_{i-2} .

$$\hat{d} = \arg \max_{d \in D} P(d|h) \prod_{j=1}^n P(v_j|d) \quad (6.3)$$

In order to gauge the effectiveness of the bigram and trigram models in view of the small amount of training data, hit-rate statistics were collected during testing. These statistics, presented in Table 6.2, show the percentage of conditions that existed in the various models. Conditions that did not exist were not counted in the accuracy measure during evaluation.

The perplexity values for the various n -gram models are shown in Table 6.2. Perplexity is related to entropy and can be thought of as a way to assess the complexity of a language model (Cover and Thomas 1991). In our case, the perplexity represents the number of dialogue acts that could follow the dialogue act context vector in any particular language model (or no context vector for the unigram model). The perplexity PP is based on the entropy H of a discrete random variable D that ranges

Speaker	Message
Sanders	[Hello Customer] ^{CONVENTIONAL-OPENING,diff} , [thank you for contacting MSN Shopping] ^{THANKING,same} . [This is Sanders and I look forward to assisting you today] ^{STATEMENT,same}
Sanders	[How are you doing today?] ^{OPEN-QUESTION,same}
Customer	[good] ^{STATEMENT,diff} , [thanks] ^{THANKING,same}
Sanders	[How may I help you today?] ^{OPEN-QUESTION,diff}

Table 6.3: An example of the beginning of a dialogue in our corpus showing utterance boundaries in brackets and dialogue-act tags and speaker transitions in superscript.

over all dialogue acts δ and probability mass function $p(d) = Pr\{D = d\}, d \in \delta$ using the following formulae:

$$H(D) = - \sum_{d \in \delta} p(d) \log p(d) \quad (6.4)$$

$$PP(D) = 2^{H(D)} \quad (6.5)$$

To calculate the conditional entropy for the bigram and trigram models, we use

$$H(Y|D) = \sum_{d \in \delta} p(d) H(Y|D = d) \quad (6.6)$$

$$= - \sum_{d \in \delta} p(d) \sum_{y \in \psi} p(y|d) \log p(y|d) \quad (6.7)$$

$$PP(Y|D) = 2^{H(Y|D)} \quad (6.8)$$

As expected, the biggest improvement, indicated by a decrease in perplexity, comes when moving from the unigram to bigram model.

The models evaluated in Table 6.2 did not model speaker transitions, meaning that the source of a dialogue act, the actual speaker, is ignored. However, we would intuitively expect that the dialogue acts for two adjacent utterances from the same speaker, say A_1, A_2 would be different to the dialogue acts for A_1, B_1 . Section 5.2.1 described this scenario in more detail.

We modified the naive Bayes models to include speaker transitions, which assigned an attribute *same* or *diff* to each utterance depending on whether it was from the same

or different speaker, respectively, to the previous utterance. This version of the model is represented mathematically by Equation 5.4, which is repeated in Equation 6.9 for convenience.

$$\hat{d} = \arg \max_{d \in D} P(d, s|h) \prod_{j=1}^n P(v_j|d) \quad (6.9)$$

where s is a binary variable signifying whether dialogue act d_i is by the same or different speaker as the previous dialogue act d_{i-1} and denoted by “same” or “diff”. Table 6.3 shows an example of the first few utterances from a dialogue marked with the *same* and *diff* speaker transition attributes.

Using speaker transitions in the model did not provide significantly different results. The scores were almost identical to the n -gram models that did not include speaker transitions, which is somewhat counter-intuitive. To help determine whether these results were attributable to sparse training data, we produced learning curves for each model by running cross validation tests using 1 to 8 dialogue conversations with both models. We also gathered *test data* accuracy results by training and testing on the same set of dialogues, rather than splitting the data into training and testing sets. Using the same data for training and testing gives an indication as to whether the model over-fits and how it well it generalises to with increased data. The resulting learning curves are shown superimposed in Figure 6.1.

As can be seen in Figure 6.1, the results of the cross-validation models, denoted with “XV”, are identical except for the last sample using all the dialogues in our corpus where the mean accuracy result was 1 per cent higher when modelling speaker transitions. The “Test Data” models, which test on the same data used for their training, remain relatively constant and plateau at approximately 93% accuracy as the number of dialogues increases. This indicates that more data may increase the cross-validation accuracy results. However, modelling speaker transitions does not seem to make a significant difference in either of the models as the number of dialogues is increased.

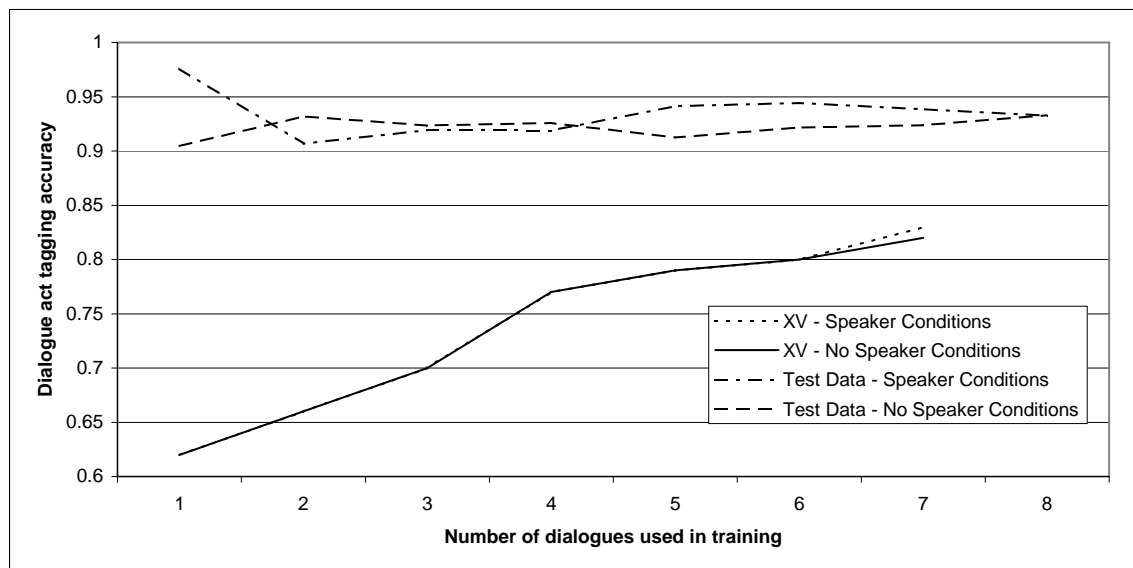


Figure 6.1: Learning curve of naive Bayes models with and without speaker transition conditions. (XV = cross-validated, Test Data = trained and tested over the entire dataset).

6.1.2 Error Analysis of Classification Task

Table 6.4 shows the classification errors that occurred using the naive Bayes classifier with bigram dialogue act modelling. The error rates displayed are the relative percentage of errors within each dialogue act. EXPRESSIVE, which is the least frequent dialogue act in our corpus, has a 100 per cent error rate, which is simply due to sparse data. The EXPRESSIVE utterances in our corpus include *lol*, *:)*, *haha*, *:-)*, *wow*, *hehe*, *hm*, and *hmm*. Each of these utterances appeared only once except for *lol* and *haha* which each appeared in two dialogues. Thus, the models were rarely able to find EXPRESSIVE tokens in the bag-of-words approach used and had to rely instead on the bigram transition probabilities.

Most of the transitions to EXPRESSIVE begin from STATEMENT. However, we see from Appendix C, which contains the bigram transition probabilities for all dialogue acts, that $P(\text{THANKING}|\text{STATEMENT})$ is the second most probable transition and far more likely than $P(\text{EXPRESSIVE}|\text{STATEMENT})$. The most probable transition from STATEMENT is another STATEMENT, but this is only slightly higher than the

Correct Dialogue Act (Error Rate)	Misclassification	%
EXPRESSIVE (100%)	THANKING	91.7%
	DOWNPLAYER	8.3%
OPEN-QUESTION (69.6%)	YES-NO-QUESTION	65.2%
	STATEMENT	4.3%
DOWNPLAYER (66.7%)	STATEMENT	50.0%
	RESPONSE-ACK	8.3%
	NO-ANSWER	8.3%
RESPONSE-ACK (44.4%)	STATEMENT	27.8%
	THANKING	11.1%
	YES-NO-QUESTION	2.8%
	DOWNPLAYER	2.8%
REQUEST (34.5%)	STATEMENT	31.0%
	YES-NO-QUESTION	3.4%
NO-ANSWER (33.3%)	STATEMENT	13.3%
	RESPONSE-ACK	6.7%
	YES-ANSWER	6.7%
	DOWNPLAYER	6.7%
CONVENTIONAL-OPENING (16.7%)	THANKING	16.7%
STATEMENT (9.8%)	YES-ANSWER	3.1%
	YES-NO-QUESTION	2.6%
	THANKING	1.0%
	REQUEST	1.0%
	RESPONSE-ACK	1.0%
	CONVENTIONAL-CLOSING	0.5%
	DOWNPLAYER	0.5%
CONVENTIONAL-CLOSING (7.7%)	DOWNPLAYER	7.7%
YES-ANSWER (7.1%)	STATEMENT	3.6%
	YES-NO-QUESTION	3.6%
YES-NO-QUESTION (4.9%)	STATEMENT	3.7%
	REQUEST	1.2%
THANKING (1.4%)	STATEMENT	1.4%

Table 6.4: Relative misclassification error rates; shows the correct dialogue act according to the gold standard and the incorrectly chosen dialogue acts.

Dialogue Act	PP
STATEMENT	0.16
THANKING	0.19
EXPRESSIVE	0.19
REQUEST	0.22
YES-NO-QUESTION	0.23
YES-ANSWER	0.23
DOWNPLAYER	0.23
RESPONSE-ACK	0.24
CONVENTIONAL-CLOSING	0.24
NO-ANSWER	0.29
CONVENTIONAL-OPENING	0.40
OPEN-QUESTION	0.50

Table 6.5: Perplexity values (PP) for dialogue act transitions based on the bigram model.

transition to THANKING and since THANKING has far fewer words in the bag-of-words, the add-one smoothing probability gives the unseen EXPRESSIVE tokens a higher probability compared to the add-one smoothing for STATEMENT. The combination of these factors explains why EXPRESSIVE is being misclassified as THANKING.

Table 6.5 shows the perplexity for each dialogue act transition using bigrams. The perplexity values are calculated from the transition probabilities shown in Appendix C. Dialogue acts with higher transition perplexity are less able to rely on the n -gram discourse model and must instead rely on keywords in the bag-of-words likelihood.

Table 6.6 shows part of a dialogue where STATEMENT was repeatedly misclassified as YES-ANSWER because of the combination of the effects of the bigram dialogue act transition probabilities, the phenomenon of phrasing an OPEN-QUESTION as a YES-NO-QUESTION, and sparse training data. From the dialogue act transition probabilities presented in Appendix C, we can see that $P(\text{YES-ANSWER}|\text{YES-NO-QUESTION}) \approx$

Utterance	Correct DA	Predicted DA
Who is the gift item for?	OPEN-QUESTION	YES-NO-QUESTION
a successful business woman	STATEMENT	YES-ANSWER
Do you have a price range in mind for this?	YES-NO-QUESTION	YES-NO-QUESTION
let say less than \$150	STATEMENT	YES-ANSWER
May I know few of her interests?	YES-NO-QUESTION	YES-NO-QUESTION
travel, fine dining, active sports like running, yoga	STATEMENT	YES-ANSWER
Could you please let me know her interests and hobbies so that I can look for an appropriate gift for her?	OPEN-QUESTION	YES-NO-QUESTION
May I know the price range please?	OPEN-QUESTION	YES-NO-QUESTION
how do I use the international version?	OPEN-QUESTION	YES-NO-QUESTION

Table 6.6: Some errors showing the impact of dialogue act adjacency pairs in the model.

0.34 and $P(\text{STATEMENT}|\text{YES-NO-QUESTION}) \approx 0.36$, which are very similar. With the posterior probability of these two dialogue acts being so close, the likelihood computation using the bag-of-words in the naive Bayes model (that is, $\arg \max_{d \in D} P(d|W)$, where d is any dialogue act in the set D and W represents the words in the utterance being classified) will largely determine the classification.

In Table 6.6, the likelihood of the utterances being YES-ANSWER is greater than being a STATEMENT. At first glance, this result seems counter-intuitive as the utterances seem to contain rather general words as opposed to words that would normally indicate a YES-ANSWER, such as *yes*. On further analysis, we can see that the reason for the misclassification is due to sparse training data. Words like *successful*, *business*, and *woman* do not occur in either STATEMENT or YES-ANSWER and only occur in one of the dialogues, which is not used for training when being used for testing. In this instance, the dialogue fragment shown in (1) introduces *successful business woman* in Customer₆. This is not exactly a domain-specific phenomenon, since the customer is intending to buy a gift, but it is the only time in our corpus that we see

these words.

(1) Agent₇: I am trying to get a sense of what gift would be suitable. Who is the gift item for?

Customer₆: a successful business woman

Agent₈: Thank you. Do you have a price range in mind for this?

The model relies on the prior probability when an unknown word is encountered. In our case, the prior probability is assigned using add-one smoothing within each dialogue act for the likelihood. However, since the YES-ANSWER class contains far fewer words than the STATEMENT class, unknown words in YES-ANSWER are assigned a greater prior probability than STATEMENT simply because the add-one smoothing is calculated on a per-dialogue act class basis. Therefore:

$$\begin{aligned} P(\text{YES-ANSWER} | a \text{ successful business woman}) &= 1.889e^{-3} \times (6.297e^{-4})^3 \\ &> P(\text{STATEMENT} | a \text{ successful business woman}) = 0.0252 \times (1.7e^{-4})^3 \end{aligned}$$

STATEMENT contains the word *a* more often than YES-ANSWER, hence the probability for that word being an order of magnitude larger than YES-ANSWER, but that difference is quickly reduced by the unknown word probabilities obtained via smoothing. This problem could be alleviated by using a different smoothing technique, but the larger problem of sparse data still remains.

In absolute terms, the most frequent single misclassification was OPEN-QUESTION being misclassified as a YES-NO-QUESTION, which accounted for 15.6% of all errors. The second most common misclassification errors are from EXPRESSIVE being classified as THANKING attributing to 11.5% of all errors, which we discussed above, and the third most common error was misclassifying RESPONSE-ACK as STATEMENT accounting for 10.4% of all errors. Examples of some of these errors are shown in Table 6.6. In these cases, the problems are related to the posterior probabilities of dialogue act transitions. Given that the previous utterance was a STATEMENT, $P(\text{OPEN-QUESTION} | \text{STATEMENT}) \approx 0.053$ whereas $P(\text{YES-NO-QUESTION} | \text{STATEMENT}) \approx 0.2$. This large difference places a rather heavy weighting on obtaining a YES-NO-QUESTION, which is not counteracted by the word likelihood probability since both

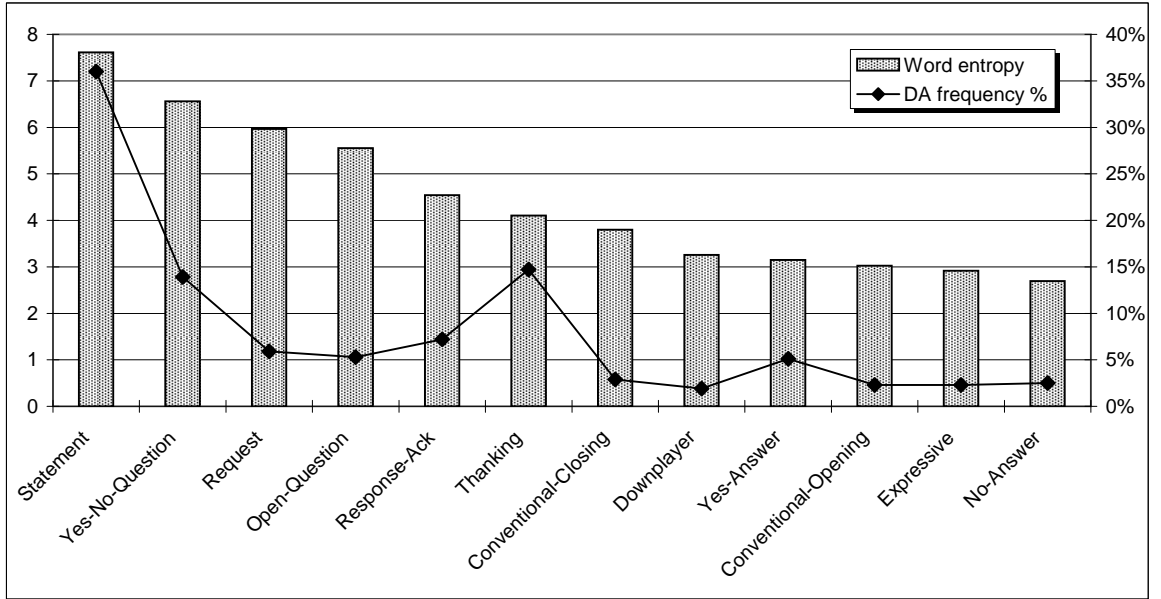


Figure 6.2: Graph showing the relation between word entropy and dialogue act frequencies. Dialogue act frequencies are given as a percentage of the total number of utterances in the overall corpus.

YES-NO-QUESTION and OPEN-QUESTION have a similarly high word entropy as shown in Figure 6.2.

The *business woman* example in Table 6.6 highlights a deficiency in the models used. The STATEMENT utterances are backward-looking dialogue acts in response to an OPEN-QUESTION. However, because we do not model forward- and backward-looking utterances as pairs, each utterance affects the following utterance regardless of whether or not they form an adjacency pair. We expect this deficiency to be alleviated by modelling speaker turns, forward- and backward-looking dialogue acts, and the dialogue games discussed in Appendix A. These additional features would form a richer discourse model which should largely eliminate the most common types of errors discussed here.

This section presented the results of the dialogue act classification task using the naive Bayes model, which is our primary classification model. We experimented with using speaker transition features in the model, but did not find a significant improvement. A detailed error analysis, with examples, revealed most misclassifications

were caused by sparse data, deficiencies in the way adjacency pairs are modelled or a combination of both.

6.2 Utterance Segmentation

Utterance segmentation is an inherently subjective task. The constituents that make up an utterance are not always clear; for example, the message *yes, ok* may be regarded as one ACKNOWLEDGEMENT or a YES-ANSWER followed by an ACKNOWLEDGEMENT. Deciding which segmentation should be considered correct depends largely on how the utterances will be used in downstream tasks. A similar case of subjectivity is where a message is segmented into two or more adjacent utterances with the same dialogue-act rather than just one segment; for example, [*Goodbye*]^{CONVENTIONAL-CLOSING} *and* [*take care*]^{CONVENTIONAL-CLOSING} could just be marked as one utterance.

In this section, we first discuss why the standard evaluation metrics of *recall* and *precision* are not appropriate for this type of segmentation, and then discuss the WindowDiff metric, which is used instead.

6.2.1 Evaluating Utterance Segmentation

The segmentation algorithms described in section 4.4.2 were evaluated via 8-fold cross-validation where seven of the chat sessions in our corpus were used for training and one for testing. This process was repeated for all dialogues and the mean result is presented. The results from these tests are based on the evaluation metrics as discussed in this section.

Using the Recall and Precision Metrics for Segmentation

The standard information retrieval evaluation metrics of *recall* and *precision* are not well-suited to evaluating segmentation tasks. Recall is the ratio of correctly hypothesised segment boundaries to the total number of actual boundaries. Precision is the ratio of correct boundaries detected to all hypothesised boundaries.

The main problem with using the recall and precision metrics for segmentation tasks is in handling near-boundary misses, that is, a false-positive that occurs near a true boundary. The recall and precision metrics will penalise a system equally, regardless of whether a hypothesised segment boundary is off by one or ten words. This makes it difficult to compare various segmentation algorithms since the ones that are very close to correct will be judged just as poorly as those that perform at the baseline.

Ref:	<table border="1"> <tr> <td>Hello Customer,</td> <td>how can I help you today?</td> </tr> </table>	Hello Customer,	how can I help you today?
Hello Customer,	how can I help you today?		
Hyp ₁ :	<table border="1"> <tr> <td>Hello Customer,</td> <td>how can I help you today?</td> </tr> </table>	Hello Customer,	how can I help you today?
Hello Customer,	how can I help you today?		
Hyp ₂ :	<table border="1"> <tr> <td>Hello</td> <td>Customer, how can I help you today?</td> </tr> </table>	Hello	Customer, how can I help you today?
Hello	Customer, how can I help you today?		
Hyp ₃ :	<table border="1"> <tr> <td>Hello Customer, how can I help you</td> <td>today?</td> </tr> </table>	Hello Customer, how can I help you	today?
Hello Customer, how can I help you	today?		

Figure 6.3: Reference and three hypothesised segmentations for different algorithms.

By way of illustration, Figure 6.3 shows an example reference segmentation and three hypothesised segmentations from different algorithms. The segmentation represented by Hyp₁ is a perfect match and obviously deserves to be ranked first. The segmentation by Hyp₂ differs from the reference by one word, whereas Hyp₃ is off by five words. Using the recall and precision metrics will result in a score of zero for both algorithms that produced Hyp₂ and Hyp₃, yet Hyp₂ is clearly a much closer match and deserves to be ranked above Hyp₃ but below Hyp₁. The WindowDiff metric addresses this issue as described below.

The WindowDiff Metric

Our manually segmented dialogue corpus is used as a ‘gold standard’ with which to compare hypothesised segmentations. The WindowDiff metric, proposed by Pevzner and Hearst (2002) as a segmentation metric, is an improvement over the recall and precision metrics by rewarding near-misses relative to their degree of divergence from the gold standard. The algorithm works by choosing a window size k , which is typically equal to half of the average segment length in a corpus. This k -sized window then slides over the hypothesised segmentation data and compares segment and non-

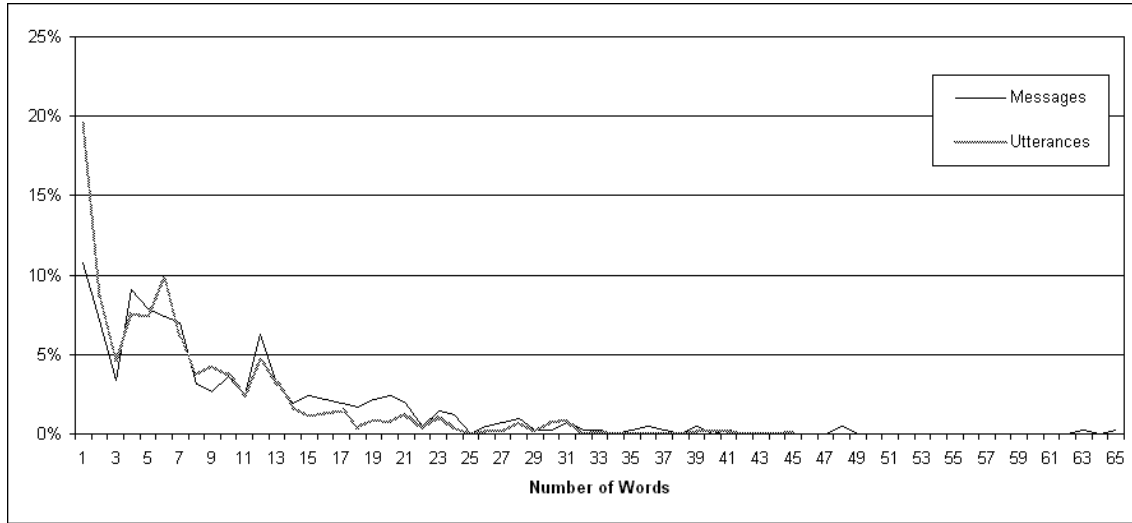


Figure 6.4: Frequency distribution of utterance and message lengths in words.

segment marks with the reference data. If the number of hypothesised and reference segments within the window size differ, a counter is incremented and the window continues to the next position. The final score is then divided by the number of positions evaluated. The WindowDiff formula from Pevzner and Hearst (2002) is shown below:

$$\text{WindowDiff}(ref, hyp) = \frac{1}{N - k} \sum_{i=1}^{N-k} (|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0) \quad (6.10)$$

where $b(i, j)$ represents the number of boundaries between positions i and j in the text and N represents the number of words in the message. The WindowDiff effectively measures the error rate, and thus a perfect system would receive a score of zero.

The WindowDiff metric gives partial credit to near-misses, which is useful for comparing the performance of various segmentation algorithms. This is based on the rationale that the best algorithm is the one that predicts hypothesised segments closest to the reference segments.

The window size k is typically equal to half of the average segment length in a corpus, which in our case is the utterance length. However, a fixing k in this manner is only appropriate with a reasonably uniform distribution of segment lengths. Both utterance and message lengths in our corpus exhibit an exponential distribution

as shown in Figure 6.4. Furthermore, changing k affects the WindowDiff results differently depending on the segmentation model. We are therefore unable to choose any one fixed value for k and be confident that the WindowDiff results will give a reliable indication of each model’s performance relative to the other models. To address this problem, we adjust the window size k from 1 to 20 for each message and take the mean result over each window size. For each message, the maximum allowable value of k is the message length. This technique provides a fair evaluation and shows the effect a varying window size has on the results.

The WindowDiff metric measures the error rate, which means that low valued results are preferred. The best possible value is 0, indicating a perfect match with the reference segments, whereas a value of 1 would indicate a 100% error rate. Small values for k provide fine-grained comparisons, which reward correct non-segment boundaries (true negatives) and segment boundaries (true positives). When $k = 1$, the WindowDiff metric acts as an accuracy metric evaluating segment and non-segment markers on a per-word basis. At the other extreme, when k is large, any discrepancy within the window will be penalised. At the maximum value $k = \text{length}(\text{message})$, any difference in segmentations between a hypothesised message and reference message results in a 100% error rate, or WindowDiff score of 1. Only perfect matches are rewarded in this case.

6.2.2 Experimental Results and Discussion

The WindowDiff results for the various models and window sizes are shown in Figure 6.5 along with the baseline scores. Because WindowDiff measures the error rate, a lower score indicates higher accuracy. It is important to point out that the WindowDiff metric, as used in this study, only evaluates hypothesised *utterance* boundaries *within* messages. This is in contrast to *message* boundaries, which are already known and not hypothesised. Therefore, a WindowDiff score is obtained over each *message* and the final score is the mean for all messages.

The best result was achieved by the parse tree method, while the worst was given by the HMM PoS tag model. All of the models exceeded the baseline. The relative

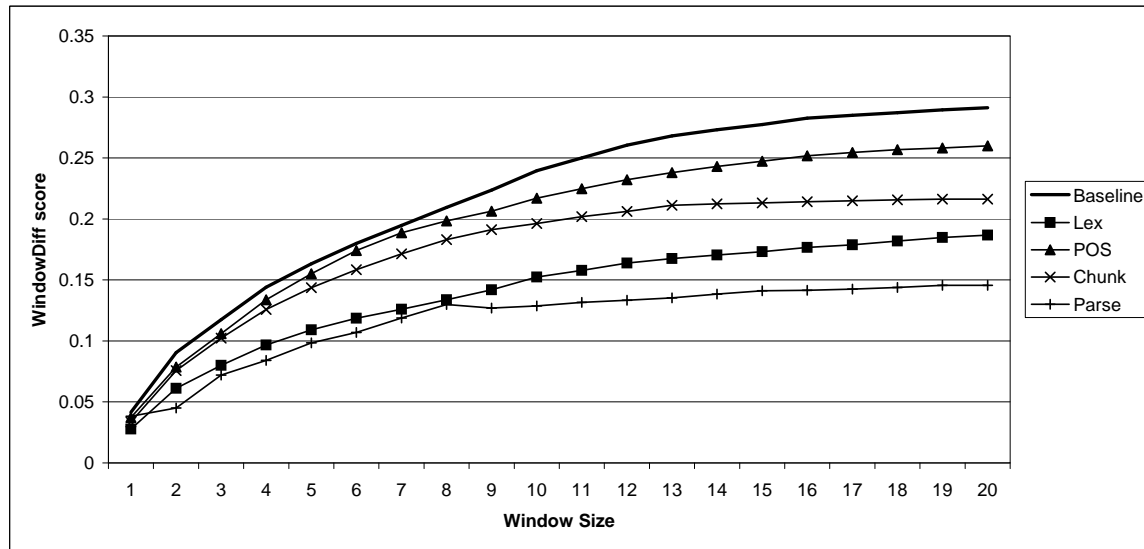


Figure 6.5: WindowDiff results of various models used and varying window size k from 1 to 20. A lower score indicates better accuracy.

difference between the models varies little as the window size changes. The WindowDiff score begins to taper off as k increases past 20 words, which is at approximately the 90th percentile of utterance lengths in our corpus. This plateau is due to the adjustment we make to k when k is greater than the message length: the maximum length of k is the length of the current message being evaluated. For example, if a message contains 12 words then k is set to 12 during the iterations of k from 12 to 20 for all evaluations with that message. Setting k to the message size means that there will only be one comparison of that WindowDiff algorithm since the window size spans the entire message. For practical purposes in these cases, $N - k = 1$ in Equation 6.10.

Small values of k result in lower WindowDiff scores than larger k values, which is due to the way the WindowDiff algorithm compares segments within the window. An equal penalty is applied regardless of whether there are five or two segments within a window that should only contain 1. Therefore, a window length spanning the entire message will at most return only one penalty if the hypothesised segments differ at all from the reference segments. Since the window spans the entire message, only one comparison is performed which results in the equivalent of a 100 per cent error rate.

Conversely, when k is small, the number of unequal windows between the reference and hypothesised segmentations will also be small since we have so few false positives, while the number of comparisons will be high. This leads to a low WindowDiff score, which is desirable.

A perfect score of 0 is never achieved since there are always some misaligned segments. Similarly, a score of 1, the worst possible score, does not occur since many of the single-utterance messages are accurately detected. None of the models approach the baseline as the window size increases, which indicates that some of the multi-utterance messages are also accurately detected. We know this because the baseline assumes that each message is one utterance. Therefore, if the multi-utterance messages were not being detected, or if messages were being over-segmented, the WindowsDiff error rate will approach or exceed that of the baseline.

Baseline

An analysis of our data revealed that messages contain up to four utterances. Of these messages, 65.7% contained only one utterance, 24.5% contained two utterances, 4.6% contained three utterances, 0.5% contained four utterances, and 4.8% did not contain any utterances as they were URL links pointing to web pages that contained the search results from the shopping service assistant.

The baseline is calculated by assuming that each message contains only one utterance since this is the majority class. Our corpus contains approximately 415 messages.

HMM Results

We used three types of features with the HMM: lemmas, POS tags, and the head word of chunked data. The POS tag model performs the worst, whereas the lemma model is the best of the HMM models. This indicates that cue words play a major role in determining utterance segment boundaries. Replacing the words with their respective POS tags obscures this information.

Using POS tags can sometimes help to overcome data sparseness problems as it has the effect of generalising words. However, in this case it over-generalises, resulting

in poorer performance.

The rationale behind using chunked data is that the number of possible boundaries is reduced as we hypothesise boundaries between chunks rather than words. Since utterance boundaries in our corpus do not lie within chunks, we hypothesised that this would increase the probability of correct segment boundary detection. However, the results show that the HMM benefits from using all words rather than only the chunks' head words.

The main types of errors produced by the HMM are false positives based on words that commonly occur at the start of an utterance, such as *what*, occurring mid-sentences as in (2):

(2) but I'm not sure what to get her

The reference data has this as one utterance, but the HMM detects a false positive starting at *what*.

Parse Tree Results

The parse tree method gives the best results. A qualitative evaluation of the DA classifications assigned to detected utterances gave an accuracy of 84%. The *accuracy* was measured by determining whether the automatically assigned dialogue act was appropriate for the given utterance, then dividing the number of tags deemed correct by the total number of utterances.

The most common type of error the parse tree method makes is to separate words near the root of a parse tree away from a deeper right node. Figure 6.6 shows a parse tree produced by RASP for (3) below:

(3) Thank you for approaching us. I would surely try to help you today

The parse tree for (3) is problematic. The first word, *thank*, is detached from the S node that contains the rest of the sentence. Our model treats (3) as a sequence of word tokens $W = w_1, w_2, w_3, \dots, w_{13}$ and finds that $P(\text{THANKING}|w_1) \times P(\text{STATEMENT}|W_2^{13}) > P(d|W)$, where d is any dialogue act. In this instance, the

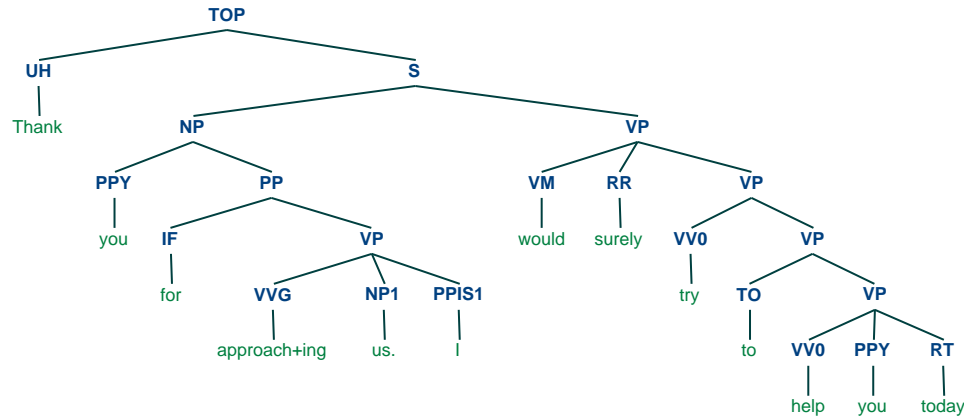


Figure 6.6: Erroneous parse tree of sentence (3) as produced by RASP.

problem of sentence tokenisation is quite apparent. Since IM contains somewhat unpredictable usage of punctuation, we did not rely on full-stops to indicate the end of a sentence. Additionally, the POS tagger mistakenly tagged “us.” as a proper noun (NP1) referring to *United States*. Consequently, RASP did not segment the two sentences, which prevented our model from evaluating the correct utterances. This illustrates the high dependency our model has on the quality of the parse trees, which may in turn rely on the quality of the POS tagger.

Another type of error arises when the model does not detect any segmentations within a message where there ought to be. An instance of this is in (4) below:

(4) right, but I do not know of any and do not speak/read french

The reference data has the word *right* segmented and tagged as RESPONSE-ACK and the rest of the message as one STATEMENT. However, our model does not evaluate that possibility as the corresponding parse tree in Figure 6.7 does not combine the words as would be required.

A small-scale analysis of 75 randomly-selected RASP parse trees from our corpus shows 89% parsing accuracy for the purposes of our model. This was evaluated by one

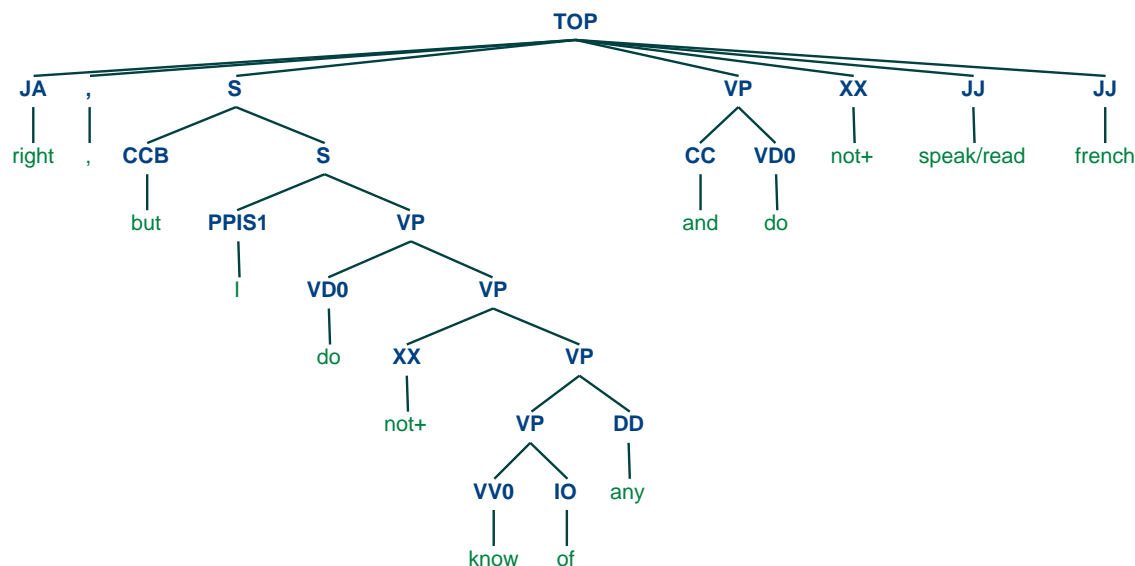


Figure 6.7: Parse tree of sentence (4) as produced by RASP.

person manually evaluating the parse trees to determine whether the utterances had the expected sub-tree structure—part of speech tags were ignored for this evaluation as our parse-tree segmentation model does not directly use them.

6.3 An Application: Assisted Customer Support

We mentioned earlier that the segmentation and classification methods presented can be used to assist a customer support representative during a conversation. Using the naive Bayes classification module discussed in this thesis, we developed a program that suggests responses during a live dialogue. The suggested utterances are grouped into their respective dialogue act tags and the user then selects a dialogue act that is appropriate. Once a dialogue act is selected, all utterances available for that dialogue act are shown. An utterance can then be selected from the list or a new one typed in.

Figure 6.8 shows the user interface, which displays a ranked list of suggested dialogue acts and utterances. The dialogue acts are ranked from highest to lowest

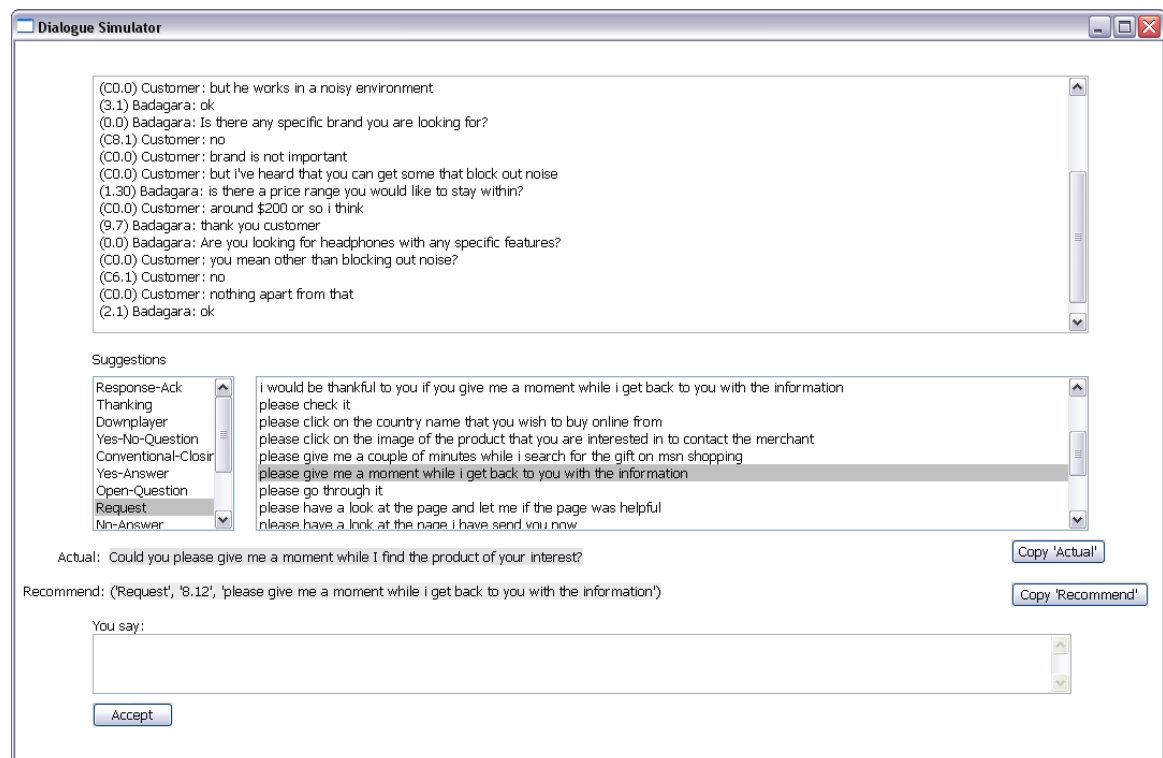


Figure 6.8: Example of the interface for automatically suggesting utterances. “Wants:” refers to next utterance in the dialogue that should be matched as closely as possible to achieve the same pragmatic result.

probability based on our naive Bayes model described in Section 5.2.1. The utterances within the dialogue acts are ranked by their frequency count during training. The evaluation is only focussed on the dialogue-act rankings, not the utterance rankings.

The system is first trained on all but one dialogue in our corpus. This training is identical to that preciously described in Section 6.2.1. Following training, a customer support scenario is simulated using the one dialogue that was not used for training, known as the *target dialogue*. The aim of the simulation is for the program to reproduce substantially all of the Agent’s utterances given the identical customer utterances. By following this process, we hope to answer the question “Had the program been used at the time the dialogues were originally taking place, how closely to human behaviour would the program have responded?”

Note that the customer utterances during the simulation are actually being played back from the target dialogue in our corpus; there is no human customer during the simulation. The process is repeated using each dialogue in our corpus as the target dialogue. Segmentation is also not performed during the evaluation. Multi-utterance responses are not generated during the simulation. Suggestions are made at the utterance level, which means that when the agent is required to send a multi-utterance message, the utterances are predicted and sent sequentially as individual messages. Messages from the simulated client are sent in one utterance at a time based on the segmented gold-standard version of the corpus, so further segmentation is not performed.

6.3.1 Using the Simulation Program

When the simulation begins, each utterance in the target dialogue must be reproduced with an identical or semantically-equivalent utterance and in the same order. The required utterance is based on the target dialogue and denoted by the label “Wants:”—this is the utterance that must be reproduced.

The program recommends an utterance from the list of all suggested utterances by highlighting it. The purpose of recommending an utterance is simply for convenience during the evaluation. The recommendation is chosen by selecting the utterance from

the entire suggestions list with the lowest word error rate compared with the “Wants:” utterance. The *word error rate* is calculated using

$$100 \times \frac{\text{minumum edit distance}}{\text{Total Words in Target Utterance}}$$

(Jurafsky and Martin 2000:271). The minimum word error rate with respect to the target utterance is the recommended utterance.

The user has three options at this point: 1) accept the recommended utterance by pressing the Copy ‘Recommend’ button; 2) take a different utterance from the suggestion list by double-clicking on an utterance in the list; or 3) typing an entirely new response. Pressing the Copy ‘Actual’ button simply copies the target utterance for convenience.

If a similar tool was used for an actual live conversation, the “Actual:” utterance and Copy ‘Actual’ button would obviously not be present as they rely on the simulated target dialogue. Likewise, recommending an utterance would need to use some method other than a word error rate comparison with the target utterance.

The customer’s utterances are excluded from the final evaluation results. The reason for this is that the system is designed to assist a support representative and it would therefore be unrealistic to evaluate utterances suggested for a customer.

The dialogue is printed in the history log at the top of the window. Each row contains the ranking of the dialogue act d in the suggestions list along with the ranking of the utterance u within the dialogue act in the format $(d.u)$, followed by the selected utterance. Customer utterances are flagged with a ‘C’ prefix on the ranking numbers. Therefore, ‘3.1’ signifies the first suggested utterance in the third dialogue act and ‘C8.1’ signifies a customer’s utterances being the first utterance in the eighth dialogue act and that the utterance. Utterances that were not chosen from the suggestion list have a ranking of ‘0.0’. Examples can be seen in Figure 6.8 and Table 6.7. The complete logs of the conversations are presented in Appendix B.

6.3.2 Evaluating the Customer Support Assistant

The simulation program maintains the history of the dialogue simulation. Recorded information includes the dialogue act predictions for each utterance, the list of suggested utterances, and the rankings of the utterances that were chosen by the user. An example of part of a completed simulation is shown in Table 6.7.

System evaluation was based on dialogue act rankings, not utterance rankings; that is, if an utterance was selected from the list of suggestions, the dialogue act rank from which it came is recorded. The idea is that utterances from highly-ranked dialogue acts, say the first three, can be selected quicker than typing them in, assuming an efficient user interface. Customer utterances were not included in the evaluation.

We did not include the STATEMENT dialogue act as part of the evaluation since it is far too general to predict using dialogue acts alone, requiring a level of semantic reasoning.

It is important to emphasise that a dialogue-act ranking was only included if the correct utterance was selected from that dialogue act. For example, if the system predicted that YES-NO-QUESTION is the most likely response, but did not provide the actual question in its list, no score was given.

Table 6.8 shows the n -best suggestion results, from 1 to 3, for each dialogue act. The cumulative percentage the correct utterance appeared in that rank is also displayed. Customer utterances were not included in these results.

Unsurprisingly, the best performance was with CONVENTIONAL-OPENING. This is simply a greeting that occurs at the start of every dialogue, welcoming the customer and introducing the agent.

The DOWNPLAYER dialogue act is the second most accurate with 67% correct in the first ranking and 100% correct by the third rank. This dialogue act has the advantage of having a low utterance entropy, as will be explained in Section 6.3.3 below. Briefly, this means that an appropriate utterance will be easy to find, providing the system correctly predicts that DOWNPLAYER is required.

The YES-NO-QUESTION dialogue act has the greatest proportion of failed predictions. This is not surprising as the system does none of the semantic reasoning and

Utt	Dialogue Act	Rank	Utterance
1	CONVENTIONAL-OPENING	1.4	hello customer
2	THANKING	2.78	thank you for contacting msn shopping
3	STATEMENT	1.41	i am mike and i will be assisting you today
4	OPEN-QUESTION	3.28	how are you doing today?
5	STATEMENT	C1.20	good
6	THANKING	C1.1	thanks
7	YES-NO-QUESTION	C0.0	are you a real person?
8	STATEMENT	2.6	i am a human
9	YES-NO-QUESTION	0.0	Please let me know how I can assist you on MSN Shopping today
10	STATEMENT	C1.79	i want to buy a gift for my brother
11	STATEMENT	C0.0	but i'm not sure what to get him
12	DOWNPLAYER	3.20	no problem
13	STATEMENT	1.100	i will help you in finding a great gift for your brother please be assured
14	THANKING	C1.1	thanks
15	DOWNPLAYER	1.2	you are welcome
16	YES-NO-QUESTION	0.0	Could you please let me know his interests and hobbies so that I can look for an appropriate gift for him?

Table 6.7: An example from the beginning of one of the dialogues produced with the automated response program. Ranks are in the form $n.m$ where n is the dialogue act ranking and m is the utterance ranking within that dialogue act. Those beginning with ‘C’ are customer responses and 0.0 denotes that no appropriate utterances were suggested.

natural language generation required to form an appropriate question. Utterances from YES-NO-QUESTION were correctly predicted approximately 66% of the time within the first three ranks, which was typically with the most common questions, such as *Do you have a price range?*, which is asked in every dialogue.

Similarly, REQUEST was correct only 11% of the time within the first rank. Again, this is a complex forward-looking dialogue act that requires semantic reasoning to predict accurately. However, it was correct 85% of the time within the top 3 dialogue act ranks, which indicates that the system is correctly predicting REQUEST with at least the same accuracy, and that most types of requests also occur in the other dialogues. Some examples of common REQUEST utterances are *please let me know your question*, *please give me a moment while I look into your request*, and *please have a look at this page*.

The results under *All Dialogue Acts* in Table 6.8 are the total number of non-customer utterances in the dialogue, excluding STATEMENT utterances. The results of the top-3 are high, with 81% of utterances being predicted. However, only 25% of utterances were available in the first dialogue act rank throughout the evaluation.

The results presented in Table 6.8 do not take into account the rankings of utterances within dialogue acts. These numbers have been deliberately omitted since they are highly dependant on the sorting algorithm used for utterances, which is outside the scope of the present research. Because there are many ways to express the same meaning behind an utterance, we calculated the entropy for the more predictable dialogue acts such as YES-ANSWER and THANKING, as described in Section 6.3.3 following.

During the course of evaluating the program, we discovered that a more fine-grained dialogue act tag set may have been helpful in the classification task and utterance prediction. The reason for this is that certain utterances were tagged as STATEMENT that, although not incorrect, performed a more specific role such as in the following instance in (5):

- (5) [Hello Customer,]^{CONVENTIONAL-OPENING} [thank you for contacting us.]^{THANKING} [I

DA	<i>n</i>	%	DA	<i>n</i>	%
DOWNPLAYER	1	67%	THANKING	1	16%
	2	78%		2	60%
	3	100%		3	92%
OPEN-QUESTION	1	36%	RESPONSE-ACK	1	0%
	2	55%		2	60%
	3	82%		3	100%
YES-NO-QUESTION	1	25%	YES-ANSWER	1	29%
	2	63%		2	57%
	3	66%		3	86%
CONVENTIONAL-OPENING	1	100%	CONVENTIONAL-CLOSING	1	25%
				2	63%
				3	88%
REQUEST	1	11%	NO-ANSWER	1	50%
	2	67%		2	50%
	3	85%		3	100%
All Dialogue Acts	1	25%			
	2	62%			
	3	81%			

Table 6.8: Dialogue act rankings within the set of predictions by running the shopping evaluation program. *n* is the *n*-best rankings of correct suggestions. Percentages are cumulative.

am Mike and I will be assisting you today.]]^{STATEMENT}

The first message of each dialogue has the same structure as the message shown in (5). The dialogue acts CONVENTIONAL-OPENING and THANKING appropriately represent the illocutionary force of their respective utterances, but although the utterance *I am Mike and I will be assisting you today* is correctly tagged STATEMENT, it would be better represented by an additional dialogue act such as INTRODUCTION—doing so would result in a more meaningful flow of the dialogue. Furthermore, the STATEMENT class, being very general, is the largest class and represents 36% of the utterances in our corpus. Classifying some of the utterances in STATEMENT into more specific classes would also help reduce the uneven distribution of utterances and word entropy that currently exists.

The differences between the likelihood and unigram and bigram models was statistically significant using a paired two-sample t-test ($p < 0.05$, two-tailed). However, the unigram and bigram models did not achieve statistical significance ($p \approx 0.31$ without speaker transitions and $p \approx 0.06$ with speaker transitions, two-tailed).

6.3.3 The Entropy of Utterances within Dialogue Acts

Entropy measures the randomness of some event or variable. In our case, we use entropy to measure how varied utterances are within a dialogue act. Utterance entropy gives an indication as to how usefully each dialogue act can be used as a basis for suggesting utterances. For example, a low entropy would indicate that very little searching or semantic analysis is required when choosing an utterance from a recommended dialogue act, such as NO-ANSWER. On the other hand, a high entropy would indicate that selecting an appropriate utterance may require further advanced semantic analysis, such as utterances in STATEMENT.

To calculate the entropy, we first manually clustered utterances within dialogue acts based on what was deemed to represent a similar meaning. An example of clusterings for the CONVENTIONAL-OPENING dialogue act is shown in Table 6.9. In this example, the utterances in class 1 are deemed equivalent because they all serve the

Utterance	Class c	$p(c)$
hello customer welcome to msn shopping	1	0.43
hi	2	0.28
hello	2	0.28
hello customer	1	0.43
hello customer welcome to msn shopping online support	1	0.43
hello krishna	3	0.14
hello?	4	0.14

Table 6.9: Manually clustered utterances in the CONVENTIONAL-OPENING dialogue act used to calculate entropy within dialogue acts.

purpose of welcoming the customer, whereas class 2 contains more general greetings, 3 is the customer greeting the agent, and 4 is somewhere between a greeting and question. When deciding on which utterance to use a response, utterances within a class are interchangeable. Therefore, the choice is reduced from choosing an individual utterance to choosing a class within a dialogue act.

The entropy of a random variable C is calculated by the formula:

$$H(C) = - \sum_{c \in \zeta} p(c) \log_2 p(c) \quad (6.11)$$

where C ranges over all the clusters within a dialogue act, the set of which we will call ζ , and c is any given cluster in ζ . This formula results in a value of 3.4 for the CONVENTIONAL-OPENING example shown in Table 6.9.

The entropy calculations for all of the dialogue acts that were computed are shown in Table 6.10. We did not calculate entropy values for STATEMENT, OPEN-QUESTION, YES-NO-QUESTION, and REQUEST because they are all complex forward-looking dialogue acts that require higher-level semantic and pragmatic modelling to predict usefully; it is thus rare to have repeat utterances within these dialogue acts.

The THANKING dialogue act had the most classes, whereas YES-ANSWER and NO-ANSWER had the fewest with only one class each. The complete utterance clusterings for each dialogue act are included in Appendix D.

Dialogue act	Classes	Entropy
THANKING	8	8.2
EXPRESSIVE	5	4.1
CONVENTIONAL-OPENING	4	3.4
DOWNPLAYER	2	3.4
CONVENTIONAL-CLOSING	2	2.9
RESPONSE-ACK	2	1.5
YES-ANSWER	1	0
NO-ANSWER	1	0

Table 6.10: Number of utterance clusters and entropy values for low-entropy dialogue acts.

Based on the entropy values in Table 6.10, we can see that YES-ANSWER and NO-ANSWER are the most useful dialogue acts used for suggesting a response. In these cases, an agent using a computer-assisted support program similar to our simulation program need only confirm that the correct dialogue act is selected rather than choosing an actual utterance.

As the entropy increases, so too does the semantic reasoning necessary to select an appropriate utterance. However, having dialogue acts automatically selected and ranked will at least help in responding more quickly in many cases. Furthermore, this framework can be enhanced to add semantic reasoning or more advanced searching capabilities for ranking and suggesting utterances within dialogue acts.

6.4 Conclusion

In this chapter, we have presented and compared the results of several models used for utterance segmentation and classification. Finding utterance boundaries in IM dialogue is a critical step for aiding utterance classification and downstream language processing modules such as dialogue response planning. In evaluating our results, we discussed the shortcomings of using the standard information retrieval metrics of recall and precision, using the WindowDiff metric as an alternative, which

has the advantage of more accurately comparing different segmentation algorithms.

The best result from our dialogue act classifier was obtained using a bigram discourse model resulting in an average tagging accuracy of 81.8% when not including speaker transitions as part of the posterior probabilities and 83% when they were included (see Table 6.2). Although this result is higher than the results from 13 studies presented by Stolcke *et al.* (2000) with accuracy ranging from $\approx 40\%$ to 81.2%, the tasks, data, and tag sets used were all quite different, so a direct comparison is not possible. One interesting observation is that the high classification accuracy results obtained by the likelihood in the naive Bayes model indicate a strong correlation between word tokens and dialogue acts.

The differences between the naive Bayes models with and without speaker transitions included in the model were not as large as we expected. After testing whether this was a result of insufficient training data by producing learning curves of the two models with a reduced number of dialogues being used for training, there was no evidence to conclude that sparse data was a problem as the learning curves were identical except for a small rise with the inclusion of the final dialogue. However, with approximately 550 utterances in our corpus, the number of utterance-speaker pairs is halved at 225, which may not be sufficient for a definitive conclusion on the matter.

In the segmentation task, we showed that the parse tree model performs best overall and has the advantage of combining both segmentation and classification in one step to give the optimal combined result. It uses the linguistic intuition that utterances are complete constituents, which are modelled well by parse trees. However, this heavy reliance on the quality of the parse trees is also a weakness; most of the errors obtained using the parse tree method may be attributed to the production of poor parse trees. Parser performance is further complicated due to the nature of IM text, which contains many spelling mistakes, unreliable use of punctuation, and non-sentential messages and utterances. These features confuse part of speech taggers and parsers which are trained on text from edited sources such as newspaper articles. That notwithstanding, the preliminary results using the RASP parser are very encouraging.

We demonstrated how the research discussed in this thesis can be used to assist a customer support agent in responding to customer utterances during a dialogue. We did this by writing a program that simulated an agent and customer dialogue and suggested each utterance during the course of the dialogue. The program was evaluated by recording the dialogue act rank of the user's selection of the *correct* utterance from a ranked list of suggested utterances.

Our dialogue simulation program showed that it is possible to significantly narrow plausible responses using dialogue acts; 81% of utterances, excluding STATEMENT, were correctly predicted within the top three ranked dialogue acts: 25% were in the first dialogue act and 62% in the second. The plausible responses can be used as suggestions to assist a human agent in customer support scenarios.

During error analysis, we found that the most common types of errors that occurred during classification were due to a combination of sparse training data, non-optimal smoothing in our naive Bayes model, and adjacency pair dialogue act probabilities being used on all utterances in a chain rather than modelling forward- and backward-looking dialogue acts and dialogue games. These are positive findings as they are addressable by gathering more data and working on more sophisticated discourse models, giving direction for future work.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

The increasing popularity of instant messaging (IM) and automated support services via IM presents many new challenges for research and development. Most work in automatic dialogue systems to date has been highly directed, relying on simple string pattern-matching techniques. As the natural language understanding features in support services become more ambitious, dialogue systems and discourse models must become more advanced to meet the higher expectations.

Dialogue via IM provides the ability for an automated agent to ask clarification questions to better meet the needs of a customer. This feature alone gives automated support services via IM a significant advantage over other forms such as FAQs and IVR systems. Appropriate dialogue modelling will help an automatic agent reliably distinguish the intention of utterances, such as those intended as questions rather than statements.

This thesis has investigated the use of dialogue acts in the growing field of IM and its use in both human-human and human-computer support services. We have demonstrated ways of applying new and existing techniques in a novel domain for dialogue modelling in IM chat sessions. We collected a corpus of task-oriented instant messaging dialogues for analysis and defined a set of twelve dialogue acts appropriate for the support-based domain we studied. Utterances in our corpus were manually

classified into dialogue-acts and used to train our segmentation and classification models. Since our models are statistically-based and automatically trained, they are thus applicable to other domains where labelled training data is available.

Key characteristics of IM dialogue that are not found in spoken or other forms of written dialogue were identified and analysed. In particular, message synchronisation and utterance segmentation seem to be peculiar to IM. The methods we described for synchronising messages, message segmentation, and dialogue act classification all achieve high accuracy and are essential parts for a deeper understanding of dialogue.

Using the kappa statistic to measure inter-annotator agreement, we conclude that our tag set can be used reliably for annotation tasks. However, analysis of our results indicate that specific dialogue acts that more closely represent the illocutionary force of utterances will benefit the classification and segmentation processes. Another major cause of errors was our model's reliance on the prior probabilities when unknown words were encountered. Specifically, the prior probabilities were computed using add-one smoothing on our naive Bayes model. More training data will reduce the chance that unseen words are encountered and hence the need to fall back on the prior probabilities during classification, which we expect to result in increased accuracy. That notwithstanding, our results are very encouraging and show that high accuracy is attainable for both the classification and segmentation tasks and can be improved further.

The parse-tree method we used for combining the segmentation and classification tasks obtained very good results especially considering that the part-of-speech (POS) tagger and parser we used were trained on newspaper text rather than an IM corpus. Our error analysis revealed that many of the errors were due to poor parse trees, which in turn were often due to poor POS tags. Spelling mistakes and ungrammatical sentences are common in IM, which confused the POS tagger. We therefore expect that pre-processing the IM data to remove its novelties or otherwise normalising it would result in better performance of downstream tasks.

Our dialogue support simulation program showed that using dialogue acts alone is enough to recommend appropriate responses within the first three dialogue act rank-

ings 61% of the time. As a result, the work presented here can be used immediately to benefit human-computer support services via IM. The techniques we used for the simulation program provide a framework that can be enhanced, such as adding more semantic processing. A first step for enhancement would be the address the issues mentions, such as IM POS tagging and parsing, some finer-grained dialogue acts, and more training data. Doing so would improve the suggestions and may ultimately lead to fully automated dialogue.

7.2 Future Work

There are several techniques worth investigating to improve the dialogue recognition accuracy reported in this thesis. The methods used to detect dialogue acts presented here do not take into account sentential structure and word ordering. The sentences in (1) would thus be treated equally with a bag-of-words approach.

- (1) a. john has been to london
- b. has john been to london

Without the punctuation (as is often the case with informal typed dialogue) the bag-of-words approach will not differentiate the sentences, whereas if we look at the ordering of even the first two words we can see that “john has ...” is likely to be a statement, whereas “has john ...” would be a question. It would be interesting to research other types of features, such as phrase structure or even looking at the order of the first few words and the parts of speech of words in an utterance to determine its dialogue act.

Aspects of dialogue macrogame theory (DMT) (Mann 2002) may also help to increase classification accuracy. Utterances in DMT, are grouped together to form a *game*, an example of which is in Appendix A. Games may be nested as in the following example:

- (2) A₁: May I know the price range please?
- B₁: In which currency?

A₂: \$US please

B₂: 200–300

In Example (2), speaker B has nested a clarification question which is required before providing the price range. The dialogue act bigram model we presented in this thesis will incorrectly model this interaction as the sequence YES-NO-QUESTION, OPEN-QUESTION, STATEMENT, STATEMENT, whereas if the model accounted for dialogue game structures, the question/answer pairs would be extracted, grouped, and represented appropriately. A flat dialogue act structure thus seems somewhat inferior by comparison, but is simpler to implement. In future work, we intend to investigate techniques for detecting and tagging dialogue macrogames and incorporating them with automatic dialogue systems.

Although other studies have attempted to automatically tag utterances with dialogue acts (Stolcke *et al.* 2000; Jurafsky *et al.* 1997; Kita *et al.* 1996) it is difficult to fairly compare results because the corpora used were significantly different, the domains were different (transcribed spoken dialogue versus typed dialogue), and the dialogue acts were also different ranging from a set of 9 (Kita *et al.* 1996) to 42 (Stolcke *et al.* 2000). It may be possible to use a standard set of dialogue acts for a particular domain, but inventing a set that could be used for all domains seems unlikely. This is primarily due to different labelling requirements in various applications. A superset of dialogue acts that covers all domains would necessarily comprise of a large number of tags (at least the 42 identified by Stolcke *et al.* (2000)) with many tags not being appropriate for other domains. Indeed, this was the impetus for the DAMSL project discussed in Section 3.5.1.

Our MSN Shopping corpus mostly followed the general grammar as used in newspaper text, which allowed RASP to perform well for most messages. However, spelling mistakes and typographical errors did occur and must be assumed for IM data from any domain. Furthermore, we expect that at least some constructions and abbreviations that are common in socially-based IM will make their way into task-oriented IM. As a result, having the ability to gather enough information about unseen words and constructions will allow a system to adapt to a changing language.

The effect of emoticons in dialogue is interesting because they usually give a strong indication of how the previous utterance should be interpreted, which may affect the classification of the previous dialogue act. Although there were no apparent examples of this causing a problem using our corpus, it may be an aspect to address using other corpora.

Gathering information is particularly convenient in dialogue since the system has the opportunity to ask questions to the user when an unknown word or phrase is encountered. The problem, of course, is knowing which questions are the *right* questions to ask. Existing work in acquiring information about unknown words has tackled the problem in static documents (not real-time dialogue), such as in Erbach (1990) and Barg and Walther (1998). More recently, work in acquiring unknown words in dialogue has also been explored. In van Schagen (2004), for example, a dialogue system initiates a sub-dialogue to elicit certain information when an unknown is encountered. The sub-dialogue presents the user with a number of example sentences which contain the new word. The user then either accepts or rejects individual sentences depending on whether or not the word usage was correct. The advantage of a system being able to ask questions about unknown words will provide a more robust approach to learning a language using reinforcement learning techniques (Singh *et al.* 2002; Henderson *et al.* 2005). Successfully incorporating such a technique could dramatically increase the usefulness of the dialogue simulator presented here to achieve fully automated dialogue for support services.

Another clear task for future work is gathering a larger task-based instant messaging corpus. The current study used a corpus of nine dialogues totalling approximately 550 utterances. Although this was sufficient for the purposes of the study, to further this research, a larger corpus will help to avoid problems with scarce data, particularly if more granular dialogue acts are to be used. It would also be interesting to evaluate the models presented here in real-life situations with many different users and agents. Doing so will provide more data with which to evaluate the models' usefulness in assisting customer support agents and verifying the observations made in the present study.

Obtaining more task-oriented and socially-based instant messaging data will allow us to empirically compare language usage between the two. Identifying the differences in language usage may allow for more accurate language models in automated systems depending on whether they are targeting task-oriented problems, socially-based dialogue or both. More accurate language models will also enable better performance for lemmatisation, part-of-speech tagging, and downstream tasks such as segmentation, classification, parsing, and discourse modelling. This is particularly important for our parse-tree model, which is highly dependent on the quality of the parse trees for accurate results.

With hundreds of millions of people worldwide using instant messaging, the incentive for understanding language through this medium is clear. We established that dialogue acts are a very useful form of semantic representation for utterances. The work we presented in this thesis showed how instant messaging communication can be processed to deliver immediate benefits for support services.

Bibliography

- ABNEY, STEVEN P. 1991. Parsing by Chunks. In *Principle-Based Parsing: Computation and Psycholinguistics*, ed. by R. C. Berwick, Steven P. Abney, and C. Tenny, 257–278. Dordrecht: Kluwer.
- ALLEN, JAMES, and MARK CORE, 1997. Draft of DAMSL: Dialog Act Markup in Several Layers. Unpublished manuscript.
- ALLEN, JAMES F., DONNA K. BYRON, MYROSLAVA DZIKOVSKA, GEORGE FERGUSON, LUCIAN GALESCU, and AMANDA STENT. 2001. Toward conversational human-computer interaction. *AI Magazine* 22.27–37.
- ALLWOOD, JENS. 1976. Linguistic Communication in Action and Co-Operation: A Study in Pragmatics. Technical Report 2, Gothenberg.
- , DAVID TRAUM, and KRISTIINA JOKINEN. 2000. Cooperation, dialogue and ethics. *International Journal of Human-Computer Studies* 53.871–914.
- ANDERSON, ANNE, MILES BADER, ELLEN GURMAN BARD, ELIZABETH BOYLE, GWYNETH DOHERTY, SIMON GARROD, STEPHEN ISARD, JACQUELINE KOWTKO, JAN MCALLISTER, JIM MILLER, CATHERINE SOTILLO, HENRY THOMPSON, and REGINA WEINERT. 1991. The HCRC Map Task Corpus. *Language and Speech* 34.351–366.
- AUSTIN, JOHN L. 1962. *How to do Things with Words*. Oxford, UK: Clarendon Press.
- BAAYEN, R. HARALD, RICHARD PIEPENBROCK, and LEON GULIKERS. 1995. *The CELEX Lexical Database. Release 2 (CD-ROM)*. Philadelphia, Pennsylvania, USA: Linguistic Data Consortium, University of Pennsylvania.
- BARD, E. G., C. SOTILLO, A. H. ANDERSON, H. S. THOMPSON, and M. M. TAYLOR. 1996. The DCIEM Map Task Corpus: spontaneous dialogue under sleep deprivation and drug treatment. *Speech Communication* 20.71–84.

- BARG, PETRA, and MARKUS WALTHER. 1998. Processing unknown words in HPSG. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, 91–95, Morristown, NJ, USA. Association for Computational Linguistics.
- BARON, NAOMI S. 1998. Letters by Phone or Speech by Other Means: The Linguistics of Email. *Language and Communication* 18.133–170.
- . 2003. The language of the Internet. In *Handbook for Language Engineers*, ed. by Ali Farghaly, chapter 3, 59–127. CSLI Publications.
- BERGER, ADAM L., VINCENT J. DELLA PIETRA, and STEPHEN A. DELLA PIETRA. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics* 22.39–71.
- BOSER, BERNHARD E., ISABELLE M. GUYON, and VLADIMIR N. VAPNIK. 1992. A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–152, New York, NY, USA. ACM Press.
- BRILL, ERIC. 1995. Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Computational Linguistics* 21.543–565.
- BRISCOE, TED, and JOHN CARROLL. 2002. Robust accurate statistical annotation of general text. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, 1499–1504, Las Palmas, Spain.
- BURNARD, LOU, 2000. *The British National Corpus Users Reference Guide*. Oxford University Computing Services, Oxford, UK.
- CARLETTA, JEAN. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics* 22.249–254.
- CHANG, CHIH-CHUNG, and CHIH-JEN LIN, 2001. *LIBSVM: a library for support vector machines*. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- CHOMSKY, NOAM, and GEORGE A. MILLER. 1963. Introduction to the Formal Analysis of Natural Languages. In *Handbook of Mathematical Psychology*, ed. by Robert Duncan Luce, R. R. Bush, and E. Galanter, volume 2, 269–321. New York, USA: Wiley and Sons.

- CHU-CARROLL, JENNIFER. 1998. A Statistical Model for Discourse Act Recognition in Dialogue Interactions. In *Applying Machine Learning to Discourse Processing. Papers from the 1998 AAAI Spring Symposium*. Tech. rep. SS-98-01, ed. by Jennifer Chu-Carroll and Nancy Green, 12–17. AAAI Press, Menlo Park, CA.
- CLARK, HERBERT H., and EDWARD F. SCHAEFER. 1989. Contributing to Discourse. *Cognitive Science* 13.259–294.
- COVER, THOMAS M., and JOY A. THOMAS. 1991. *Elements of Information Theory*. New York, USA: Wiley.
- CRYSTAL, DAVID. 2001. *Language and the Internet*. Cambridge: Cambridge University Press.
- DAELEMANS, WALTER, and ANTAL VAN DEN BOSCH. 2005. *Memory-based Language Processing*. Cambridge, UK: Cambridge University Press.
- DI EUGENIO, BARBARA, and MICHAEL GLASS. 2004. The kappa statistic: a second look. *Computational Linguistics* 30.95–101.
- ERBACH, GREGOR. 1990. Syntactic Processing of Unknown Words. In *Artificial Intelligence IV - methodology, systems, applications*, ed. by P. Jorrand and V. Sgurev, 371–382. Amsterdam: North-Holland.
- GAVALDÀ, MARSAL, KLAUS ZECHNER, and GREGORY AIST. 1997. High performance segmentation of spontaneous speech using part of speech and trigger word information. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 12–15, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- GODFREY, JOHN J., EDWARD C. HOLLIMAN, and JANE MCDANIEL. 1992. SWITCHBOARD: Telephone Speech Corpus for Research and Development. In *ICASSP*, volume 1, 517–520.
- GRICE, H. P. 1975. Logic and Conversation. In *Syntax and Semantics: Speech Acts*, ed. by Peter Cole and J. L. Morgan, volume 3, 41–58. New York, USA: Academic Press.
- 1978. Further Notes on Logic and Conversation. In *Pragmatics: Syntax and Semantics*, ed. by Peter Cole, volume 9, 113–127. New York, USA: Academic Press.
- GROSZ, BARBARA J., and CANDACE L. SIDNER. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics* 12.175–204.

- HEIM, IRENE. 1983. File Change Semantics and the Familiarity Theory of Definiteness. In *Meaning, Use, and Interpretation of Language*, ed. by Rainer Bäuerle, Christoph Schwarze, and Arnim von Stechow, 164–189. Berlin, Germany: Walter de Gruyter.
- HENDERSON, J., O. LEMON, and K. GEORGILA. 2005. Hybrid reinforcement/supervised learning for dialogue policies from COMMUNICATOR data. In *Proceedings of the IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Edinburgh, UK.
- HIRST, GRAEME. 1987. *Semantic interpretation and the resolution of ambiguity*. New York, NY, USA: Cambridge University Press.
- HUDDLESTON, and PULLUM. 2003. *The Cambridge Grammar Of The English Language*. Cambridge, UK: Cambridge University Press.
- JEFFERSON, GAIL. 1984. Notes on a systematic deployment of the acknowledgement tokens ‘yeah’ and ‘mm hm’. *Papers in Linguistics* 197–216.
- JOHNSON, KEITH. 1997. *Acoustic and Auditory Phonetics*. Oxford, UK; Cambridge, MA, USA: Blackwell.
- JOHNSON-LAIRD, P. N. 1983. *Mental models: towards a cognitive science of language, inference, and consciousness*. Cambridge, MA, USA: Harvard University Press.
- JURAFSKY, DANIEL, REBECCA BATES, NOAH COCCARO, RACHEL MARTIN, MARIE METEER, KLAUS RIES, ELIZABETH SHRIBERG, ANDREAS STOLCKE, PAUL TAYLOR, and CAROL VAN ESS-DYKEMA. 1997. Automatic detection of discourse structure for speech recognition and understanding. *Proceedings of the 1997 IEEE Workshop on Speech Recognition and Understanding* 88–95.
- , and JAMES H. MARTIN. 2000. *Speech and Language Processing*. New Jersey, USA: Prentice Hall.
- KAMP, HANS. 1981. A Theory of Truth and Semantic Representation. In *Formal Methods in the Study of Language*, ed. by J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof, volume 1, 277–322. Amsterdam: Mathematisch Centrum.
- , and UWE REYLE. 1991. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Reidel. Re-published by Kluwer, Studies in Linguistics and Philosophy, 42, 1993.
- KATSUNO, HIROFUMI, and CHRISTINE R. YANO. 2002. Face to face: on-line subjectivity in contemporary Japan. *Asian Studies Review* 26.205–232.

- KATZ, SLAVA M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35.400–401.
- KITA, KENJI, YOSHIKAZU FUKUI, MASAOKI NAGATA, and TSUYOSHI MORIMOTO. 1996. Automatic Acquisition of Probabilistic Dialogue Models. In *Proceedings of the Fourth International Conference on Spoken Language*, volume 1, 196–199, Philadelphia, PA, USA.
- LADEFOGED, PETER. 1996. *Elements of Acoustic Phonetics*. Chicago, IL, USA: University of Chicago. Second Edition.
- LARSSON, STAFFAN, and DAVID R. TRAUM. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Natural Language Engineering* 6.323–340.
- LE, ZHANG, 2001. *MaxEnt*. Software available at http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.
- LENDVAI, PIROSKA, ANTAL VAN DEN BOSCH, and EMIEL KRAHMER. 2003. Machine Learning for Shallow interpretation of User Utterances in Spoken Dialogue Systems. In *Proceedings of EACL-03 Workshop on Dialogue Systems: Interaction, Adaptation and Styles of Management*, 69–78, Budapest, Hungary.
- LEVIN, LORI, CHAD LANGLEY, ALON LAVIE, DONNA GATES, DORCAS WALLACE, and KAY PETERSON. 2003. Domain Specific Speech Acts for Spoken Language Translation. In *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue*, Sapporo, Japan.
- , KLAUS RIES, ANN THYME-GOBEL, and ALON LAVIE. 1999. Tagging of speech acts and dialogue games in Spanish Call Home. In *Towards Standards and Tools for Discourse Tagging (Proceedings of the ACL Workshop at ACL'99)*, 42–47, College Park, Maryland, USA.
- LEWIS, DAVID D., YIMING YANG, TONY G. ROSE, and FAN LI. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *The Journal of Machine Learning Research* 5.361–397.
- LEWIS, DAVID K. 1979. Scorekeeping in a Language Game. *Journal of Philosophical Logic* 8.339–359.
- LLEIDA, EDUARDO, and RICHARD ROSE. 2000. Utterance verification in continuous speech recognition: decoding and training procedures. *Speech and Audio Processing, IEEE Transactions on* 8.126–139.

- MAHOWALD, ROBERT P., and MARK LEVITT. 2005. Worldwide Enterprise Instant Messaging Applications 2005–2009 Forecast and 2004 Vendor Shares: Clearing the Decks for Substantial Growth. Technical report, IDC. Doc #34058. Press release at <http://www.itresearch.com/getdoc.jsp?containerId=prUS00246505>.
- MALOUF, ROBERT. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *COLING-02: proceedings of the 6th conference on Natural language learning*, 1–7, Morristown, NJ, USA. Association for Computational Linguistics.
- MANN, WILLIAM. 2002. Dialogue Macrogame Theory. In *Proceedings of the 3rd SIGdial Workshop on Discourse and Dialogue*, 129–141, Philadelphia, PA, USA.
- MARCUS, MITCHELL P., MARY ANN MARCINKIEWICZ, and BEATRICE SANTORINI. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics* 19.313–330.
- MAST, M., R. KOMPE, S. HARBECK, A. KIESSLING, H. NIEMANN, E. NÖTH, E. G. SCHUKAT-TALAMAZZINI, and V. WARKE. 1996. Dialog Act Classification With the Help of Prosody. In *ICSLP-96*, volume 3, 1732–1735, Philadelphia, PA.
- MCTEAR, MICHAEL F. 2002. Spoken dialogue technology: enabling the conversational user interface. *ACM Computing Surveys* 34.90–169.
- MINNEN, GUIDO, JOHN CARROLL, and DARREN PEARCE. 2001. Applied morphological processing of English. *Natural Language Engineering* 7.207–223.
- MITKOV, RUSLAN. 2002. *Anaphora Resolution*. London, UK: Pearson ESL, first edition.
- MÜLLER, KLAUS R., SEBASTIAN MIKA, GUNNAR RÄTSCH, KOJI TSUDA, and BERNHARD SCHÖLKOPF. 2001. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks* 12.181–202.
- NAGATA, MASAOKI, and TSUYOSHI MORIMOTO. 1994. First steps towards statistical modeling of dialogue to predict the speech act type of the next utterance. In *ISSD-93: Selected papers presented at the international symposium on Spoken dialogue*, 193–203, New York, NY, USA. Elsevier North-Holland, Inc.
- NGAI, GRACE, and RADU FLORIAN. 2001. Transformation-Based Learning in the Fast Lane. In *Proceedings of NAACL-2001*, 40–47, Pittsburgh, PA, USA.
- PERRAULT, C. RAYMOND, and JAMES F. ALLEN. 1980. A plan-based analysis of indirect speech acts. *Computational Linguistics* 6.167–182.

- PEVZNER, LEV, and MARTI A. HEARST. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics* 28.19–36.
- RABINER, LAWRENCE R. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. In *Readings in Speech Recognition*, ed. by Alex Waibel and Kai-Fu Lee, 267–296. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- RAMSHAW, LANCE, and MITCHELL P. MARCUS. 1995. Text Chunking Using Transformation-Based Learning. In *Proceedings of the Third Workshop on Very Large Corpora*, 82–94, Somerset, New Jersey. Association for Computational Linguistics.
- RATNAPARKHI, ADWAIT. 1997. A simple introduction to maximum entropy models for natural language processing. Institute for research in cognitive science, technical report 97–08, University of Pennsylvania.
- REITHINGER, NORBERT, RALF ENGEL, MICHAEL KIPP, and MARTIN KLESEN. 1996. Predicting Dialogue Acts for a Speech-To-Speech Translation System. In *ICSLP-96*, volume 2, 654–657, Philadelphia, PA.
- , and MARTIN KLESEN. 1997. Dialogue Act Classification Using Language Models. In *EUROSPEECH-97*, volume 4, 2235–2238.
- RESNIK, PHILIP. 1997. Selectional preference and sense disambiguation. In *Proceedings of ACL SIGLEX Workshop on Tagging Text with Lexical Semantics*, 52–57, Washington, D.C. Association for Computational Linguistics.
- . 1998. WordNet and Class-based Probabilities. In *WordNet: An Electronic Lexical Database*, ed. by Christiane Fellbaum. Cambridge, MA: MIT Press.
- RUSSELL, STUART, and PETER NORVIG. 2003. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition edition.
- SALTON, GERARD (ed.) 1971. *The SMART Retrieval System, Experiments in Automatic Document Processing*. Englewood Cliffs, NJ, USA: Prentice Hall.
- SAMUEL, KEN, SANDRA CARBERRY, and K. VIJAY-SHANKER. 1998. Dialogue act tagging with Transformation-Based Learning. In *Proceedings of the 17th international conference on Computational linguistics*, volume 2, 1150–1156, Morristown, NJ, USA. Association for Computational Linguistics.
- SCHEGLOFF, EMANUEL A. 1968. Sequencing in Conversational Openings. *American Anthropologist* 70.1075–1095.

- SCHÖLKOPF, BERNHARD, and ALEXANDER J. SMOLA. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.
- SEARLE, JOHN R. 1979. *Expression and Meaning: Studies in the Theory of Speech Acts*. Cambridge, UK: Cambridge University Press.
- SIEGEL, SIDNEY, and N. JOHN CASTELLAN, JR. 1988. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill, second edition.
- SINGH, SATINDER P., DIANE J. LITMAN, MICHAEL J. KEARNS, and MARILYN A. WALKER. 2002. Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System. *Journal of Artificial Intelligence Research (JAIR)* 16.105–133.
- SPARCK JONES, KAREN. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28.11–21.
- STALNAKER, R. C. 1978. Assertion. In *Pragmatics: Syntax and Semantics*, ed. by Peter Cole, volume 9, 315–332. New York, USA: Academic Press.
- STOLCKE, ANDREAS, NOAH COCCARO, REBECCA BATES, PAUL TAYLOR, CAROL VAN ESS-DYKEMA, KLAUS RIES, ELIZABETH SHRIBERG, DANIEL JURAFSKY, RACHEL MARTIN, and MARIE METEER. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics* 26.339–373.
- , and ELIZABETH SHRIBERG. 1996. Automatic Linguistic Segmentation of Conversational Speech. In *Proceedings, ICSLP 96. Fourth International Conference on Spoken Language*, volume 2, 1005–1008, Philadelphia, PA, Maryland. ICSLP.
- TANAKA, HIDEKI, and AKIO YOKOO. 1999. An efficient statistical speech act type tagging system for speech translation systems. In *Proceedings of the 37th conference on Association for Computational Linguistics*, 381–388. Association for Computational Linguistics.
- TAYLOR, PAUL, SIMON KING, STEPHEN ISARD, and HELEN WRIGHT. 1998. Intonation and Dialog Context as Constraints for Speech Recognition. *Language and Speech* 41.489–508.
- TETREAULT, JOEL, MARY SWIFT, PREETHUM PRITHVIRAJ, MYROSLAVA DZIKOVSKA, and JAMES ALLEN. 2004. Discourse annotation in the Monroe Corpus. In *ACL 2004 Workshop on Discourse Annotation*, 103–109, Barcelona, Spain. Association for Computational Linguistics.

- TRAUM, DAVID R., 1994. *A Computational Theory of Grounding in Natural Language Conversation*. Rochester, New York, USA: University of Rochester dissertation.
- 1999. Computational Models of Grounding in Collaborative Systems. In *Working Papers of the AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*, 124–131, Menlo Park, California. American Association for Artificial Intelligence.
- , JOHAN BOS, ROBIN COOPER, STAFFAN LARSSON, IAN LEWIN, COLIN MATHESON, and MASSIMO POESIO. 1999. A model of dialogue moves and information state revision. Technical report, TRINDI Task-oriented instructional dialogue. Trindi Project Deliverable D2.1.
- VAN DIJK, TEUN A., and WALTER KINTSCH. 1983. *Strategies of Discourse Comprehension*. New York, USA: Academic Press.
- VAN SCHAGEN, MAARTEN. 2004. Tauria: A tool for acquiring unknown words in a dialogue context. In *Proceedings of the Australasian Language Technology Workshop*, 131–138, Sydney, NSW, Australia. Australasian Language Technology Association.
- VAPNIK, VLADIMIR N. 1995. *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc.
- WALKER, MARILYN, and STEVE WHITTAKER. 1990. Mixed initiative in dialogue: an investigation into discourse segmentation. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, 70–78, Morristown, NJ, USA. Association for Computational Linguistics.
- WEBBER, BONNIE LYNN, 1978. *A Formal Approach to Discourse Anaphora*. Harvard University dissertation.
- WRIGHT, HELEN. 1998. Automatic utterance type detection using suprasegmental features. volume 4, 1403–1406, Sydney, Australia.

Appendix A

Dialogue Macrogame Theory

Dialogue Macrogame Theory (DMT) is a method for describing the organization of dialogue, wherein dialogue acts are used to encompass a complete exchange of information that may span multiple turns.¹ DMT, based on speech acts, is a successor to Dialogue Game Theory, developed in the 1970s and '80s by Bill Mann and others at USC-Information Sciences Institute. The aim of DMT is to account for the coherence of entire dialogues: Mann (2002) states that “a dialogue is said to be coherent if a person who has good access to the dialogue is left with the impression that every part of the dialogue contributed to the remainder, or equivalently that there are no parts whose presence is not easily explained” . Dialogue acts may be combined with DMT to model more sophisticated discourse structures, such as nested games.

Games in DMT are tagged based on their purpose and show a clear continuation from one to the other. Each utterance is part of a game and each game consists of three goals: the goal of the initiator, the goal of the responder, and a joint goal.

There are about 19 games currently defined in DMT and each must define the three goals listed above. For example, the *Information Offering Game* defines the following three goals:

1. Goal of the initiator: to provide particular information to recipient.
2. Goal of the responder: to identify and receive the particular information offered.

¹Although we do not use DMT in the the present study, this brief introduction serves as a reference for our discussions.

3. Joint goal: the responder comes to possess the particular information.

Because the course of a dialogue is generally under the joint control of both participants, DMT uses a negotiation metaphor whereby a game is bid by the initiator, and the responder accepts or rejects the bid. If a bid is accepted then the game is considered open. The eventual termination of the game is also a bid that can also either be accepted or rejected. Regardless of whether actions such as game acceptance and termination are explicit or implicit, they are always explicitly labelled in DMT.

Games are mutually recursive, that is, a game may include instances of other games. This is common in scenarios such as where an interlocutor seeks clarification to a question before being able to answer it as in the following example:

- (1) A₆: How much would you like to spend?
 B₅: for all the gifts or just this one?
 A₇: Just the dress.
 B₆: around \$80

Quite often, large portions of dialogue are not in pursuit of any particular goal. Actions such as politeness and acknowledgments do not involve joint goals and may occur either within or out of a game. These actions, generally confined to a single utterance, are called *Unilaterals* or *Unilateral acts* in DMT and include the following: *Action*, *Direct*, *Assess*, *Suggest*, *Tell*, *Media Management*, *Politeness*, and *Acknowledgement*.

Appendix A shows an example of a tagged discourse using DMT. Deciding which tag to assign to an utterance usually requires looking ahead to see how that utterance was interpreted by the other party, which is similar to assigning tags in DAMSL.

Not all game terminations are obvious. We often see games terminated implicitly, simply because they are no longer pursued and a new game is bid. The new game is usually a continuation of the same topic, taking into account the result of the previous game. DMT handles this by assigning multiple tags to a single utterance showing acceptance of the game termination (**at**) and the bidding of the new game (**bg**) as in utterance 53.

Grounding acts are not handled explicitly in DMT: they are either unilateral acts

or part of a game, depending on the type of grounding as listed in section 3.3.2. This is due to the scope of DMT, which is concerned with tagging games within a dialogue rather than the *flow* of the dialogue. Groundings, however, are usually independent of games and may exist either within or out of a game.

The following conversation is a task-based dialogue annotated using DMT as described by Mann (2002).

Elapsed Sec	Utt #	Spkr	Text	Game Acts & Unilaterals	Game Name	depth 3	depth 2	depth 1
	1	A	hey	MM		o	-	-
7	2	B	hey hey	MM		o	-	-
11	3	A	why r you up so late	bg	IS1	o	-	-
20	4	B	i'm working from home tomorrow and need to get some stuff out tonight	ag		o	-	-
21	5	B	margo's leaving me at 2:00 tomorrow			o	-	-
10	6	A	is that your wife?	bg	IS2	o	-	-
1	7	B	alone with jack for the weekend			o	-	-
9	8	A	jack's your son?	bg	IS3	o	-	-
16	9	B	margo's my pet squirrel and jack is my hamster	ag		o	-	-
9	10	A	:-D	ACK		o	-	-
10	11	A	hahaha	Tell		o	-	-
3	12	B	yeah... wife and son'	S-REP, bt		o	-	-
5	13	A	very funny	at, Tell		o	-	-
14	14	A	hey, what does PEG stand for?	bg	IS4	o	-	-
4	15	B	not compared to you	Tell		o	-	-
8	16	B	p/e to growth ratio	ag, bt		o	-	-
19	17	A	p/e = profit/earnings?	bg	CL1	o	-	-
21	18	B	p/e is price/(generally the fy1 eps)	ag		o	-	-
14	19	B	growth is long term growth rate			o	-	-
25	20	B	people believe the ratio should be 1			o	-	-
14	21	B	> the company is over valued			o	-	-
5	22	B	< undervalued			o	-	-
5	23	A	interesting	ACK, bt		o	-	-
20	24	B	kind of a goofy ratio	rt		o	-	-
13	25	A	is PEG always a single number or can it also be a time-series number?	bg		o	-	-
					CL2	o	-	-
1	26	B	it is best to compare vs other companies peg ratio	bt		o	-	-
25	27	A	good point. that's where our sectors come into it	at		o	-	-
26	28	B	i would think that since price is time series and lt growth is time series... peg would be	ag		o	-	-
7	29	B	it makes a neat graph	bt		o	-	-
23	30	A	but what does the graph indicate? how the company was valued over time?	rt, bg		o	-	-
					CL3	o	-	-
8	31	B	exactly	ag		o	-	-
40	32	B	the problem is that l-t growth is kind of a goofy number			o	-	-

Time	Speaker	Text	Topic	Category	Diagram
33	A	so it doesn't make sense to use time-series data and somehow come up with a single PEG		CL4	Diagram 1: A line graph showing a single upward trend from left to right.
15	A	unless you took the average or something I guess	S-REP		
6	B	a single PEG for each day?	bg	CL5	Diagram 2: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
24	A	normally you'd get 1 PEG per day, right?	bg	CL6	
4	B	correct	ag, bt		Diagram 3: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
35	A	so if I take 1 year of prices, etc, it wouldn't make sense to pop out a single PEG from all that data?	bg		
25	B	correct. it wouldn't make sense.	ag, bt	CL7	Diagram 4: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
30	A	the formula then, for PEG, is (price / eps) / 1-t growth?	bg	CL8	
17	B	right on	ag, bt		Diagram 5: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
23	A	cool - so here's my question...	ACK, MM		
5	B	shoot	MM		Diagram 6: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
18	A	doing this over, say, 1 year time-series data	Tell		
10	A	price varies daily, that's cool	Tell		Diagram 7: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
18	A	but what about eps and growth? how do they vary?	bg	IS5	
14	B	also daily	ag		Diagram 8: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
9	B	at least, potentially daily	S-REP, bt		
4	A	growth is defined as...?	bg	IS6	Diagram 9: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
24	B	analysts calc these based on estimates (5 years)	ag		
8	B	if an estimate changes			Diagram 10: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
10	B	then the L-T growth changes	bt		
2	A	so I can't calc either growth or eps myself, right?	at, bg	CL9	Diagram 11: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
32	B	you mean neither are fixed?	bg	CL10	
1	A	i'm just asking where the actual growth and eps numbers come from. do I have to get them based on what the analysts are entering, or can I calc them myself?	rg, bg		Diagram 12: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
53	B	l-t growth is (eps(fy5)/eps(fy0)) - 1 and eps is eps(fy0) from the database	ag	IS6	
9	B	wait	MM		Diagram 13: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
6	A	k	MM		
9	B	eps is eps(fy1) from the database	S-REP		Diagram 14: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
21	B	as far as graphing, david had a great idea	bt, bg	IO1	
2	A	is l-t growth correct as above?	bg	CL11	Diagram 15: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
8	B	yes	ag, bt		
4	A	idea?	ag		Diagram 16: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
40	B	look at eps as next 365 days eps			
15	B	goldman passed this along			Diagram 17: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
3	B	it would be			
51	B	(% year complete)*eps(fy+1) + (1-% year complete)*eps(fy+2)			Diagram 18: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
48	B	it produces a nice smooth line instead of a drop when fy2 becomes fy1	bt		
18	A	is any of this currently being graphed?	at, bg	IS7	Diagram 19: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
22	B	at goldman this is used (i guess)	ag		
18	B	i don't know if PEG is, to be honest	bt		Diagram 20: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
16	A	is PEG anywhere on our site currently?	at, bg	IS8	
4	B	i know that % of year completed formula is being used for p/e	bt		Diagram 21: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
45	B	don't think so	ag		
3	B	i'm checking	S-REP		Diagram 22: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
8	B	nope	bt		
23	A	do you invest seriously? do you believe in these methods and numbers?	at, bg	IS9	Diagram 23: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
32	B	peg is something people use in comparison, but i am not a fan	ag		
24	B	i don't invest seriously... not enough time			Diagram 24: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
51	B	oh yeah, and not enough money			
20	A	we have a lot of data coming in via [Company X] ([Data K], [Data L], [Data M], & not [Company Y]). I'd love to get your thoughts on what you believe the best way is to examine this data and, essentially, predict it...or choose stocks.	bg		Diagram 25: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
1	B	i didn't have the [Company Z] options	bt	IS10	
15	A	shame re. the [Company Z] options	rt, bt		Diagram 26: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
16	B	nah... i only had to work there for 9 months	rt, bt		
5	A	I mean "now [Company Y]" above	S-REP		Diagram 27: A line graph showing a single upward trend from left to right, with a small vertical line segment at the end.
46	A	or maybe that's what our web site is already trying to do	at, S-REP		

Appendix B

Computer-assisted conversation logs

The conversation in this appendix was produced using the computer-assisted dialogue program developed as part of the research described in this thesis.

	DIALOGUE ACT	Rank	Utterance
1	CONVENTIONAL-OPENING	1.4	hello customer
2	THANKING	2.78	thank you for contacting msn shopping
3	STATEMENT	1.41	i am mike and i will be assisting you today
4	OPEN-QUESTION	3.28	how are you doing today?
5	STATEMENT	C1.20	good
6	THANKING	C1.1	thanks
7	YES-NO-QUESTION	C0.0	are you a real person?
8	STATEMENT	2.6	i am a human
9	YES-NO-QUESTION	0.0	Please let me know how I can assist you on MSN Shopping today
10	STATEMENT	C1.79	i want to buy a gift for my brother
11	STATEMENT	C0.0	but i'm not sure what to get him
12	DOWNPLAYER	3.20	no problem
13	STATEMENT	1.100	i will help you in finding a great gift for your brother please be assured
14	THANKING	C1.1	thanks
15	DOWNPLAYER	1.2	you are welcome
16	YES-NO-QUESTION	0.0	Could you please let me know his interests and hobbies so that I can look for an appropriate gift for him?

17	YES-ANSWER	C0.0	he likes reading
18	THANKING	C0.0	and also rock climbing
19	YES-NO-QUESTION	0.0	Would you like me to search for a gift based on his hobbies?
20	YES-ANSWER	C3.23	yes
21	THANKING	1.9	thank you for the confirmation
22	YES-NO-QUESTION	0.0	May I know the price range please?
23	STATEMENT	C0.0	up to about 300
24	THANKING	2.84	thank you for the information
25	REQUEST	3.50	please give me a couple of minutes while i look into your request
26	RESPONSE-ACK	C1.1	ok
27	THANKING	1.2	thank you
28	YES-NO-QUESTION	C0.0	brb in 1 min
29	THANKING	3.26	thank you for waiting
30	STATEMENT	0.0	I have found a page that lists a wide variety of Rock Climbing Shoes and accessories
31	STATEMENT	1.46	i am sending you the page now the page opens in a new window on your screen now
32	STATEMENT	C0.0	back
33	DATA	0.0	http:...
34	STATEMENT	0.0	Just to inform, you need to scroll down the page and also click on the page numbers on top of the page to see the entire listing
35	REQUEST	2.14	please try and use the options on the left hand side of the page to narrow down your search
36	STATEMENT	C0.0	that's great, thanks
37	STATEMENT	C0.0	I have 1 more question
38	STATEMENT	C0.0	I just thought of
39	RESPONSE-ACK	2.77	sure
40	REQUEST	0.0	Please let me know your question
41	STATEMENT	C0.0	I live in New York and wanted to buy my cousin, who lives in France, a remote-control car
42	STATEMENT	C0.0	but I didn't want to buy one from here and ship it

43	YES-NO-QUESTION	C0.0	and I don't speak/read French so couldn't find a French toy store that could deliver it
44	YES-NO-QUESTION	C0.0	can you help me with that problem?
45	YES-NO-QUESTION	C0.0	or do you only do US deliveries?
46	STATEMENT	0.0	if you are looking for a store near you to buy the product, I suggest you to contact the merchant to know if he has a store in your area.
47	STATEMENT	C0.0	I was thinking it would be easier to buy it online from a French store with my credit card then have them deliver it
48	STATEMENT	0.0	Please click on the image of the product that you are interested in to contact the merchant.
49	YES-NO-QUESTION	0.0	That would be a great idea to buy online from a French store
50	STATEMENT	0.0	I am glad to inform you that we do have MSN International where you can buy online from many different countries
51	STATEMENT	C0.0	right, but I don't know of any and don't speak/read french
52	YES-NO-QUESTION	C0.0	Do you have access to toy stores in France?
53	STATEMENT	C0.0	ok, that's good news
54	YES-NO-QUESTION	C0.0	how do I use the international version?
55	STATEMENT	2.37	i am sending you the page on which they are listed
56	STATEMENT	1.66	i am sending you the page now the page opens in a new window on your screen now
57	DATA	0.0	http:...
58	STATEMENT	0.0	Please click on the country name that you wish to buy online from
59	YES-NO-QUESTION	2.6	did you receive the page customer?
60	YES-ANSWER	C1.1	yes
61	STATEMENT	C0.0	and it looks like it's has the right information. only problem is that it's in French - which I can't read :-(

62	YES-NO-QUESTION	C0.0	I also don't see an option to change the language
63	STATEMENT	0.0	I am sorry to hear that, Customer
64	STATEMENT	0.0	However, I suggest you to use our Web Search option available on our MSN Shopping home page to find the information of the product you are looking for on the world wide web
65	DOWNPLAYER	C3.20	ok
66	THANKING	C2.23	thanks for your help
67	STATEMENT	C0.0	the climbing shoes for my sister I think will be great gift
68	THANKING	2.82	thank you customer
69	CONVENTIONAL-CLOSING	C1.66	bye
70	STATEMENT	0.0	Customer, please type in few keywords related to your request in the 'Web Search' option available at the top right corner of the page and click on the 'Go' button to find the information on the world wide web
71	STATEMENT	C0.0	no, that doesn't work well
72	THANKING	C2.81	thanks
73	STATEMENT	0.0	I am sorry to hear that
74	STATEMENT	0.0	I am sorry that could not assist you effectively as my job expertise is limited to finding products on MSN Shopping
75	YES-NO-QUESTION	2.2	is there anything else i can assist you today with msn shopping?
76	NO-ANSWER	C2.8	no
77	THANKING	C2.28	thanks
78	STATEMENT	C0.0	you've been very helpful
79	THANKING	2.60	thank you
80	THANKING	3.31	thank you for contacting msn shopping
81	STATEMENT	0.0	Please feel free to contact us for any of your shopping needs. We are here 24 hours a day, 7 days a week to assist you
82	STATEMENT	1.21	have a great week
83	CONVENTIONAL-CLOSING	3.57	goodbye and take care

84	THANKING	C1.8	thanks
85	CONVENTIONAL-CLOSING	C1.1	bye
86	CONVENTIONAL-CLOSING	1.11	bye
1	CONVENTIONAL-OPENING	1.1	hello customer welcome to msn shopping
2	STATEMENT	0.0	I am Sally and I am here to assist you for your shopping
3	STATEMENT	0.0	I understand that you are looking for information for crashed browser
4	YES-NO-QUESTION	C0.0	is this a computer?
5	STATEMENT	2.5	i am a human
6	STATEMENT	2.63	i wish to inform you that msn shopping is a platform where the merchants advertise and sell their products
7	THANKING	0.0	Thank you for approaching us
8	STATEMENT	0.0	I would surely try to help you today
9	STATEMENT	0.0	My expertise lies in finding the products on MSN Shopping
10	STATEMENT	C0.0	then how could you understand that i'm looking for information on crashed browsers?
11	STATEMENT	C0.0	i was talking to mike and my browser crashed
12	YES-NO-QUESTION	C0.0	can you transfer me to him again?
13	STATEMENT	C0.0	he found a gift i wanted
14	STATEMENT	0.0	Customer I will try my best to help you find the gift and please let me know the request
15	STATEMENT	0.0	Mike is not available at this point of time
16	STATEMENT	C0.0	but mike already found it
17	YES-NO-QUESTION	C0.0	isn't he there?
18	STATEMENT	C0.0	it was a remote control car
19	STATEMENT	0.0	Mike is not available right now
20	STATEMENT	0.0	I am here to assist you with MSN Shopping
21	RESPONSE-ACK	3.20	sure
22	STATEMENT	0.0	I will search for the remote control car
23	REQUEST	2.7	i would be thankful to you if you give me a moment while i get back to you with the information

24	YES-NO-QUESTION	C0.0	how many people are there where you're working?
25	YES-NO-QUESTION	0.0	There are many people working working with MSN Shopping
26	THANKING	3.12	thank you for waiting
27	YES-NO-QUESTION	C0.0	over 100?
28	STATEMENT	2.94	i am sending you the page right across
29	REQUEST	2.90	please have a look at it
30	DATA	0.0	http:...
31	STATEMENT	0.0	Please try and use the options on the left hand side of the page to narrow down your search
32	YES-NO-QUESTION	2.25	did you receive the page customer?
33	YES-ANSWER	C1.1	yes
34	REQUEST	2.57	please have a look at the pages and let me know if the sent link was helpful to you
35	STATEMENT	0.0	I am sorry for the typing mistake
36	STATEMENT	0.0	I am waiting for a response from you
37	YES-NO-QUESTION	C0.0	was looking at the page
38	STATEMENT	C0.0	looks good
39	THANKING	2.31	thank you customer
40	YES-NO-QUESTION	1.9	is there anything else i can assist you today with msn shopping?
41	NO-ANSWER	C1.10	no
42	THANKING	C2.33	thanks
43	YES-NO-QUESTION	C0.0	but can you tell me which city you're in?
44	THANKING	0.0	thank you for the interest
45	STATEMENT	0.0	we are not authorized to give any personal information
46	EXPRESSIVE	C3.65	lol
47	STATEMENT	C0.0	a phone number is personal, not a city in india. but that's ok. you've been very helpful
48	RESPONSE-ACK	C0.0	:)
49	THANKING	3.15	thank you customer
50	STATEMENT	0.0	Same here
51	THANKING	2.14	thank you for your interest
52	OPEN-QUESTION	3.37	how may i help you with msn shopping today?

53	STATEMENT	C0.0	ooh
54	YES-NO-QUESTION	C0.0	this dialogue is at least computer assisted, right?
55	NO-ANSWER	3.11	no customer
56	STATEMENT	0.0	I am saying it and really we take immense pleasure in having you a sour esteemed customer
57	THANKING	3.20	thank you for your time and patience
58	STATEMENT	C0.0	you've been fun to talk to
59	YES-NO-QUESTION	C0.0	do you have a lot of other poeple waiting for your assistance?
60	STATEMENT	0.0	Why Customer, am I not talking good
61	EXPRESSIVE	C3.62	haha
62	NO-ANSWER	C3.43	no
63	STATEMENT	C0.0	that wasn't what i was implying. you've been talking fine. i'm just curious about this service
64	YES-NO-QUESTION	C0.0	MS must hire so many people to help everyone do their shopping online
65	STATEMENT	0.0	THis is the only service which will take care of the customers in a very special way
66	STATEMENT	0.0	Please feel to contact MSN Shopping for any of your shopping needs. We are available 24 hours a day and 7 days a week
67	THANKING	C2.75	thanks
68	STATEMENT	C1.92	have a great day!
69	THANKING	C0.0	:-)
70	DOWNPLAYER	1.5	you are most welcome
71	OPEN-QUESTION	0.0	:)
72	STATEMENT	C0.0	hehe
73	STATEMENT	0.0	you are great and appreciating me a lot
74	THANKING	2.94	thank you
75	CONVENTIONAL-CLOSING	0.0	Bye Customer and have a wonderful time
76	CONVENTIONAL-CLOSING	C0.0	k
77	THANKING	C0.0	cya
78	YES-ANSWER	2.72	sure
79	CONVENTIONAL-CLOSING	2.80	bye

1	CONVENTIONAL-OPENING	1.1	hello customer welcome to msn shopping
2	STATEMENT	0.0	I am Mike and I will be assisting you today
3	OPEN-QUESTION	2.57	how are you doing today?
4	STATEMENT	C1.21	good
5	OPEN-QUESTION	C1.23	and you?
6	STATEMENT	0.0	That's nice to hear from you
7	STATEMENT	1.50	i am well
8	THANKING	1.73	thank you
9	STATEMENT	C0.0	my friend said that you're a computer
10	YES-NO-QUESTION	C2.1	is that correct?
11	NO-ANSWER	1.11	no customer
12	STATEMENT	0.0	I am a human
13	EXPRESSIVE	C3.51	haha
14	EXPRESSIVE	C0.0	cool
15	STATEMENT	0.0	I am sorry if I sound too robotic
16	EXPRESSIVE	C3.19	lol
17	OPEN-QUESTION	3.37	how may i help you today with msn shopping site?
18	OPEN-QUESTION	C0.0	so how many people do you help at once?
19	STATEMENT	0.0	Well, you are the only one whom I am assisting
20	RESPONSE-ACK	C3.11	ok
21	YES-NO-QUESTION	C0.0	can you help me find a gift for my cousin who lives in france?
22	YES-ANSWER	3.12	sure
23	STATEMENT	0.0	With pleasure
24	STATEMENT	C0.0	that I can buy online from a store local in france so that I don't have to pay for international shipping?
25	STATEMENT	0.0	I wish to inform you that MSN Shopping is a platform where the merchants advertise and sell their products
26	STATEMENT	0.0	Therefore, the products that are available are sold and shipped by these merchants independently of MSN Shopping
27	STATEMENT	0.0	For better assistance with shipping and payment details you need to contact the merchant directly

28	STATEMENT	0.0	However, I will be glad to help you find a Gift for your Cousin
29	STATEMENT	C1.1	fine
30	YES-NO-QUESTION	C0.0	can we find a merchant in france?
31	STATEMENT	0.0	all the merchants listed on our site generally ships to US only
32	STATEMENT	0.0	However, I once again suggest you to contact them as they might help you in shipping the product to France
33	YES-NO-QUESTION	0.0	is the Gift for a Man or Woman?
34	STATEMENT	C0.0	a boy - 11
35	THANKING	3.32	thank you for the confirmation
36	YES-NO-QUESTION	C0.0	where are you based by the way?
37	STATEMENT	0.0	We are in India
38	STATEMENT	0.0	May I know your cousin's hobbies and interests and also if you have any price range for the gift please?
39	STATEMENT	C0.0	that's cool. my friend here is from bangalore
40	STATEMENT	C0.0	he wants a remote control car
41	STATEMENT	C0.0	price isn't too important
42	THANKING	2.75	thank you for the information
43	YES-NO-QUESTION	C0.0	so do people in india still take these calls at all hours or is the service taken over by some other country in a different timezone when it's night there?
44	YES-ANSWER	3.1	yes
45	STATEMENT	0.0	Its 1pm here
46	THANKING	2.85	thank you for the interest
47	YES-NO-QUESTION	C0.0	so people work there even at 3am?
48	YES-ANSWER	1.77	yes
49	STATEMENT	0.0	We are here round the clock for your help
50	YES-NO-QUESTION	C0.0	wow
51	STATEMENT	C0.0	that's amazing
52	THANKING	2.17	thank you
53	STATEMENT	2.16	i am sending you the page right across
54	REQUEST	3.33	please have a look at it

55	RESPONSE-ACK	C1.1	ok
56	STATEMENT	0.0	This page will pop up in a new window on your screen
57	REQUEST	2.12	please go through it
58	STATEMENT	0.0	I request you to disable the pop up blocker if you have any, as it might block this page.
59	STATEMENT	C0.0	hm
60	RESPONSE-ACK	C1.1	ok
61	THANKING	2.2	thank you
62	YES-NO-QUESTION	0.0	So, may I send you the page now?
63	YES-ANSWER	C3.11	yes
64	THANKING	1.1	thanks
65	STATEMENT	0.0	Here comes the page.
66	STATEMENT	1.99	it has four pages and you may narrow down the search by clicking on the options on the left side of the page
67	YES-NO-QUESTION	1.19	did you receive the page?
68	YES-ANSWER	C1.1	yes
69	STATEMENT	1.1	fine
70	STATEMENT	0.0	Just to inform you, there are 4 pages in the above sent link.
71	REQUEST	2.69	please have a look at the page and let me if the page was helpful
72	STATEMENT	0.0	It seems that we were disconnected.
73	STATEMENT	0.0	However, I hope the information provided is helpful to you.
74	THANKING	2.78	thank you for contacting msn shopping
75	STATEMENT	0.0	If you need us anytime, please feel free to contact us. We are available round the clock for your assistance.
1	CONVENTIONAL-OPENING	1.1	hello customer welcome to msn shopping
2	STATEMENT	1.2	my name is krishna and i am your online shopping assistant today
3	OPEN-QUESTION	1.84	how are you doing today?
4	STATEMENT	C1.21	fine
5	OPEN-QUESTION	1.29	how may i help you with msn shopping today?

6	YES-NO-QUESTION	C0.0	what exactly does this service provide?
7	STATEMENT	0.0	MSN Shopping is a service designed to help you find products available through our participating merchants and my expertise lies with finding the products on MSN shopping site.
8	STATEMENT	C0.0	i see
9	YES-NO-QUESTION	C0.0	what would you recommend for the person who has everything?
10	YES-NO-QUESTION	0.0	I would like to confirm that if you are looking for a gift?
11	YES-ANSWER	C0.0	yep
12	THANKING	1.12	thank you for confirming that
13	YES-NO-QUESTION	0.0	Is there a special occasion for this gift?
14	YES-ANSWER	C0.0	christmas
15	STATEMENT	0.0	I am trying to get a sense of what gift would be suitable
16	YES-NO-QUESTION	0.0	Who is the gift item for?
17	YES-ANSWER	C0.0	a successful business woman
18	THANKING	3.28	thank you
19	YES-NO-QUESTION	0.0	Do you have a price range in mind for this?
20	YES-ANSWER	C0.0	let say less than \$150
21	YES-NO-QUESTION	2.60	may i know few of his interests?
22	YES-ANSWER	C0.0	travel, fine dining, active sports like running, yoga
23	THANKING	2.35	thank you for the information
24	YES-NO-QUESTION	1.54	would you be interested in having a look at 'personalized gifts' which is a unique gift idea where you can also add their initials to make that gift of their own!
25	YES-ANSWER	C0.0	nah
26	RESPONSE-ACK	2.43	ok
27	STATEMENT	0.0	I understand that you are looking for a special gift within the price range of \$150
28	YES-NO-QUESTION	2.1	is that correct?
29	YES-ANSWER	C1.1	yes
30	THANKING	2.12	thank you for confirming that

31	REQUEST	1.80	please give me a couple of minutes while i look into your request
32	RESPONSE-ACK	C1.4	sure
33	STATEMENT	0.0	Sorry for the delay as I am still finding a suitable item for you
34	STATEMENT	0.0	I will be back momentarily
35	THANKING	0.0	thanks for your patience
36	STATEMENT	2.10	i will search for the link
37	STATEMENT	2.51	i have found a page which matches your request
38	REQUEST	2.96	please have a look at this page
39	YES-NO-QUESTION	1.42	did you receive the page?
40	YES-NO-QUESTION	0.0	Could you please reply back so that I can further assist you?
41	STATEMENT	2.14	i am waiting for a response from you
42	YES-NO-QUESTION	2.62	are you still there?
43	STATEMENT	0.0	I regret that we could not continue our chat session as it has been disconnected. Hope the page sent would be helpful to you. However, if you need further assistance or information, please feel free to get back to us anytime
44	THANKING	2.80	thank you for contacting msn shopping
45	STATEMENT	0.0	We are available 24 hours a day, 7 days a week for your assistance
46	CONVENTIONAL-CLOSING	2.84	goodbye and take care
47	YES-NO-QUESTION	C0.0	hello?
48	THANKING	C3.55	thanks
49	STATEMENT	0.0	I am glad to know that you are back
50	YES-NO-QUESTION	2.5	did you receive the page?
51	YES-ANSWER	C1.1	yes
52	THANKING	C2.2	thank you
53	YES-NO-QUESTION	1.8	was it helpful?
54	YES-ANSWER	C1.1	yes
55	THANKING	C2.2	thank you
56	DOWNPLAYER	1.58	you are welcome
57	YES-NO-QUESTION	2.57	is there anything else i can assist you today with msn shopping?

58	STATEMENT	0.0	I appreciate it if you reply back so that I can assist you further
59	YES-NO-QUESTION	2.4	are you still there?
60	THANKING	C3.58	thanks for your help
61	CONVENTIONAL-CLOSING	C0.0	good night
1	CONVENTIONAL-OPENING	1.3	hello customer
2	STATEMENT	1.3	my name is krishna and i am your online shopping assistant today
3	CONVENTIONAL-OPENING	C1.7	hi
4	OPEN-QUESTION	2.1	how are you doing today?
5	STATEMENT	C1.1	fine
6	STATEMENT	C1.2	good
7	THANKING	C2.1	thanks
8	DOWNPLAYER	1.69	you are welcome
9	OPEN-QUESTION	1.8	how may i help you with msn shopping to-day?
10	STATEMENT	C0.0	I want to buy a gift for my brother
11	THANKING	3.16	thank you for the information
12	STATEMENT	2.51	i understand that you are searching for gift
13	YES-NO-QUESTION	2.33	is that correct?
14	YES-ANSWER	C1.1	yes
15	STATEMENT	C0.0	that's what I said
16	YES-NO-QUESTION	C0.0	are you chatting with many people?
17	STATEMENT	0.0	I am here to assist you and will certainly put in the best of my efforts to find a suitable gift item.
18	STATEMENT	0.0	I am with you in chat
19	YES-NO-QUESTION	2.52	would you be interested in having a look at 'personalized gifts' which is a unique gift idea where you can also add her initials to make that gift of her own!
20	NO-ANSWER	C2.10	no
21	YES-NO-QUESTION	0.0	Do you have any specific gift idea in your mind?
22	STATEMENT	C0.0	not really
23	OPEN-QUESTION	C3.35	are you a person or computer?
24	THANKING	2.9	thank you for confirming that

25	YES-NO-QUESTION	1.32	is there a price range you would like to stay within?
26	STATEMENT	1.47	i am a human
27	THANKING	C0.0	under 1000
28	THANKING	1.82	thank you
29	REQUEST	2.82	please give me a couple of minutes while i look into your request
30	RESPONSE-ACK	C1.1	ok
31	STATEMENT	1.48	i am sending you the page now the page opens in a new window on your screen now
32	YES-NO-QUESTION	1.24	did you receive the page?
33	YES-ANSWER	C2.1	yes
34	YES-NO-QUESTION	3.10	was the page helpful to you customer?
35	STATEMENT	C0.0	it's very general
36	YES-NO-QUESTION	C0.0	is there some way we can find something that suits him better?
37	YES-ANSWER	2.74	sure
38	STATEMENT	0.0	I will help you in finding a great gift for your brother. Please be assured.
39	YES-NO-QUESTION	2.60	may i know few of her interests?
40	YES-ANSWER	C0.0	he plays video games and is into bike riding mostly
41	THANKING	1.27	thank you for the information
42	YES-NO-QUESTION	0.0	Is there any specific video games that he would be interested in?
43	STATEMENT	C0.0	not that i know of
44	REQUEST	C0.0	he plays age of empires and chess
45	RESPONSE-ACK	C0.0	so something along those lines
46	STATEMENT	0.0	I have a wide range of video games in the page I am sending to you now.
47	REQUEST	2.1	please have a look at it
48	YES-NO-QUESTION	1.21	did you receive the page?
49	YES-ANSWER	C1.1	yes
50	THANKING	C2.1	thanks
51	YES-NO-QUESTION	1.6	was it helpful?
52	YES-NO-QUESTION	C0.0	do you know what game is similar to age of empires?

53	STATEMENT	C0.0	I think he'd like that.
54	REQUEST	2.83	please give me a moment while i get back to you with the information
55	RESPONSE-ACK	C1.1	ok
56	STATEMENT	0.0	Sorry for the delay as I am still finding a suitable item for you
57	STATEMENT	1.83	i will be back momentarily
58	RESPONSE-ACK	C1.24	ok
59	THANKING	3.17	thank you for your time and patience
60	STATEMENT	0.0	I have found a page which matches your request
61	REQUEST	2.7	please check it
62	REQUEST	0.0	It has four pages and you may narrow down the search by clicking on the options on the left side of the page
63	YES-NO-QUESTION	2.24	did you receive the page?
64	YES-ANSWER	C1.1	yes
65	RESPONSE-ACK	C2.53	looks good
66	STATEMENT	0.0	I am glad to know that
67	YES-NO-QUESTION	2.2	is there anything else i can assist you today with msn shopping?
68	NO-ANSWER	C2.10	no
69	STATEMENT	C1.73	i think i can find something suitable in that link
70	THANKING	C1.5	thanks for your help
71	DOWNPLAYER	1.69	you are welcome
72	STATEMENT	3.50	it was my pleasure assisting you today
73	THANKING	C1.1	thanks
74	YES-NO-QUESTION	C0.0	can I have a copy of this sent to me via email?
75	STATEMENT	0.0	This chat transcript will be mailed to your e mail address that you have given before you signed in to this chat service
76	THANKING	2.77	thank you for contacting msn shopping
77	STATEMENT	2.67	if you need us anytime please feel free to contact us we are available round the clock for your assistance

78	CONVENTIONAL-CLOSING	3.20	goodbye and take care
79	THANKING	C1.8	thanks
80	CONVENTIONAL-CLOSING	C1.1	bye
1	CONVENTIONAL-OPENING	1.1	hello customer welcome to msn shopping
2	STATEMENT	0.0	My name is Krishna and I am your online Shopping assistant today
3	CONVENTIONAL-OPENING	C3.27	hello
4	OPEN-QUESTION	1.1	how are you doing today?
5	STATEMENT	C0.0	I am well
6	OPEN-QUESTION	C2.95	you?
7	STATEMENT	1.23	fine
8	STATEMENT	0.0	I understand that you are searching for gift
9	YES-NO-QUESTION	2.2	is that correct?
10	YES-ANSWER	C1.1	yes
11	STATEMENT	C0.0	I am looking for a gift for my 12 yr old niece
12	THANKING	3.8	thank you for the information
13	DOWNPLAYER	C1.2	no problem
14	YES-NO-QUESTION	0.0	What are the desired features/characteristics you are looking for in the gift?
15	STATEMENT	C0.0	Something typical of a new teenage girl. Either something trendy that she would be wearing or something new and different that will make here stand out a bit.
16	YES-NO-QUESTION	0.0	Do you want to go for dress?
17	YES-ANSWER	C3.37	sure
18	THANKING	1.8	thank you for the information
19	REQUEST	1.97	please give me a couple of minutes while i look into your request
20	RESPONSE-ACK	C1.1	ok
21	STATEMENT	0.0	I am sending you a page that lists dress which might interest you. The page will open in a new window.
22	REQUEST	2.1	please have a look at it
23	RESPONSE-ACK	C1.1	ok
24	STATEMENT	C0.0	looking at now
25	REQUEST	1.70	please click on the image of the product that you are interested in to contact the merchant

26	YES-NO-QUESTION	1.21	is there anything else i can assist you today with msn shopping?
27	RESPONSE-ACK	C3.15	ok
28	STATEMENT	C0.0	I am sorry, I was thinking of something more everyday which doesn't have to be a dress
29	YES-NO-QUESTION	C0.0	Are there any site geared toward teen girls, or is it in the list?
30	STATEMENT	C0.0	This seems more on a women's level more so a teen.
31	RESPONSE-ACK	3.4	sure
32	STATEMENT	0.0	I will search for the link
33	REQUEST	2.2	please give me a couple of minutes while i look into your request
34	RESPONSE-ACK	C1.1	ok
35	THANKING	2.9	thank you for your time and patience
36	DOWNPLAYER	C1.63	no problem
37	REQUEST	3.48	please have a look at it
38	RESPONSE-ACK	C1.1	ok
39	YES-NO-QUESTION	2.85	was it helpful?
40	STATEMENT	C0.0	They seem like halloween costumes.
41	REQUEST	0.0	Please have a look at this page.
42	RESPONSE-ACK	C3.20	ok
43	YES-NO-QUESTION	0.0	Was I able to provide all the information you needed?
44	RESPONSE-ACK	C0.0	I really don't see a section (catagory) specifically geared toward teen girls.
45	RESPONSE-ACK	C1.1	ok
46	STATEMENT	C0.0	i see one now
47	THANKING	3.42	thank you
48	YES-NO-QUESTION	1.10	is there anything else i can assist you today with msn shopping?
49	NO-ANSWER	C1.10	no
50	STATEMENT	C0.0	I will search the page
51	THANKING	C3.5	thanks for your help
52	STATEMENT	0.0	I hope the page I sent is helpful. If you need further assistance, please log in again. We are available 24 hours a day, 7 days a week.

53	THANKING	3.45	thank you for contacting msn shopping
54	CONVENTIONAL-CLOSING	1.77	have a nice day! good bye and take care
55	THANKING	C1.12	thank you
1	CONVENTIONAL-OPENING	1.1	hello customer welcome to msn shopping
2	STATEMENT	0.0	My name is Krishna and I am your online Shopping assistant today
3	CONVENTIONAL-OPENING	C0.0	Hello
4	STATEMENT	C0.0	I'm trying to find a sports watch
5	YES-NO-QUESTION	C2.34	are you still in the chat?
6	STATEMENT	0.0	I understand that you are looking for sports watch
7	YES-NO-QUESTION	2.2	is that correct?
8	YES-ANSWER	C1.1	yes
9	OPEN-QUESTION	C2.70	are you a person or computer?
10	OPEN-QUESTION	0.0	Man
11	REQUEST	2.46	please give me a couple of minutes while i look into your request
12	RESPONSE-ACK	C1.1	ok
13	THANKING	2.10	thank you for your time and patience
14	DOWNPLAYER	C1.63	ok
15	STATEMENT	2.43	i am sending you the page right across
16	YES-NO-QUESTION	1.53	are you able to view the page now?
17	RESPONSE-ACK	C2.13	ok
18	STATEMENT	C0.0	the Polar Heart Rate Monitor is closest to what I want, but I would like one that monitors the heart rate from the watch itself, not a strap around the chest
19	YES-NO-QUESTION	C0.0	do you have anything like that?
20	STATEMENT	C0.0	I think there's something wrong with this connection
21	YES-NO-QUESTION	C2.13	are you still in the chat?
22	YES-ANSWER	1.1	yes
23	REQUEST	0.0	please visit this page
24	YES-NO-QUESTION	2.10	are you able to view the page now?
25	YES-ANSWER	C1.1	yes
26	THANKING	C2.1	thanks
27	YES-NO-QUESTION	1.3	is there anything else i can assist you today with msn shopping?

28	YES-ANSWER	C1.1	yes
29	STATEMENT	C0.0	that seems to be the same as the other one & it has a device you strap around your chest to measure the heart rate. i think there are some watches that read it from your pulse on your wrist, which is what i'm looking for
30	YES-NO-QUESTION	2.25	customer is there anything else i may assist you with?
31	NO-ANSWER	C1.8	no
32	STATEMENT	C0.0	that seems to be the right item
33	THANKING	C2.100	thank you
34	THANKING	3.53	thank you for your time and patience
35	DOWNPLAYER	C2.61	no problem
36	CONVENTIONAL-CLOSING	C0.0	good bye
37	THANKING	2.28	thank you
38	STATEMENT	0.0	I hope you found our session today helpful
39	STATEMENT	0.0	If you have any additional questions or you need additional information, please log in again to chat with us. We are available 24 hours a day, 7 days a
40	THANKING	3.6	thanks for your help
41	THANKING	3.46	thank you for contacting msn shopping
42	CONVENTIONAL-CLOSING	2.75	have a nice day! good bye and take care
1	CONVENTIONAL-OPENING	1.1	hello customer welcome to msn shopping
2	STATEMENT	1.3	my name is krishna and i am your online shopping assistant today
3	CONVENTIONAL-OPENING	C1.6	hi
4	STATEMENT	0.0	I understand that you are looking for new headphones
5	YES-NO-QUESTION	2.23	is that correct?
6	YES-ANSWER	C1.1	yes
7	THANKING	2.9	thank you for the confirmation
8	STATEMENT	C0.0	they're for my friend for christmas
9	STATEMENT	C0.0	but he works in a noisy environment
10	STATEMENT	1.1	fine
11	YES-NO-QUESTION	0.0	Is there any specific brand you are looking for?

12	NO-ANSWER	C3.31	no
13	STATEMENT	C0.0	brand is not important
14	STATEMENT	C0.0	but i've heard that you can get some that block out noise
15	YES-NO-QUESTION	2.30	is there a price range you would like to stay within?
16	STATEMENT	C0.0	around \$200 or so i think
17	THANKING	3.37	thank you customer
18	YES-NO-QUESTION	0.0	Are you looking for headphones with any specific features?
19	YES-NO-QUESTION	C0.0	you mean other than blocking out noise?
20	NO-ANSWER	C2.19	no
21	STATEMENT	C0.0	nothing apart from that
22	THANKING	0.0	Okay
23	REQUEST	3.11	please give me a moment while i get back to you with the information
24	RESPONSE-ACK	C1.1	ok
25	STATEMENT	0.0	I have found a page that lists the latest headphone available on MSN Shopping. I am sending you the page that will pop up in a new window on your screen.
26	REQUEST	2.2	please have a look at it
27	STATEMENT	0.0	However, I will also send you the page for various headphones on MSN Shopping
28	STATEMENT	C0.0	those JVC headphones don't seem to block out the noise
29	YES-NO-QUESTION	C0.0	do they?
30	STATEMENT	0.0	I have found some noise-canceling head phones
31	STATEMENT	1.48	i am sending you the page right across
32	REQUEST	3.22	please have a look at it
33	YES-NO-QUESTION	1.44	did you receive the page customer?
34	YES-ANSWER	C1.1	yes
35	STATEMENT	C0.0	but then one of the links made it disappear again
36	YES-NO-QUESTION	C0.0	can you send it again please?
37	DOWNPLAYER	3.26	no problem
38	STATEMENT	2.41	i will search the page

39	THANKING	C1.1	thanks
40	DOWNPLAYER	2.56	you are welcome
41	OPEN-QUESTION	C0.0	hmm
42	STATEMENT	C0.0	every time i click on the Sony ones from Walmart it messes up the page. i'll try it in a new browser window
43	RESPONSE-ACK	2.98	sure customer
44	STATEMENT	C0.0	nope - i've lost the page
45	YES-NO-QUESTION	C0.0	could you send it again please?
46	YES-ANSWER	0.0	Okay, Customer
47	STATEMENT	0.0	I will send you the link of the page
48	STATEMENT	0.0	Please copy the link into a new browser window to view it
49	YES-NO-QUESTION	2.28	were you able to view the page i sent you?
50	YES-ANSWER	C1.1	yes
51	THANKING	C2.1	thanks
52	YES-NO-QUESTION	1.50	was the page i have sent to you helpful?
53	STATEMENT	C0.0	i think so
54	STATEMENT	C0.0	still looking
55	STATEMENT	0.0	Please take your time and let me know if you need further assistance
56	DOWNPLAYER	C3.29	ok
57	YES-NO-QUESTION	3.22	is there anything that i would help you with?
58	NO-ANSWER	C2.9	no
59	STATEMENT	C1.74	i think i can find an appropriate gift in there
60	THANKING	C1.6	thanks for your help
61	DOWNPLAYER	1.56	you are welcome
62	STATEMENT	0.0	It was my pleasure assisting you today
63	THANKING	2.97	thank you for contacting msn shopping
64	STATEMENT	0.0	We will be available 24 Hours a day and 7 Days a week for your help
65	THANKING	C2.75	thanks
66	CONVENTIONAL-CLOSING	C1.60	bye
67	STATEMENT	1.93	have a great day!

Appendix C

Dialogue Act Bigram transition probabilities

Initial State	Transition State	P
<i>None</i>	CONVENTIONAL-OPENING	1.0
	DOWNPLAYER	0.333
	STATEMENT	0.333
	CONVENTIONAL-CLOSING	0.111
	REQUEST	0.111
OPEN-QUESTION	YES-No-QUESTION	0.111
	STATEMENT	0.8
	OPEN-QUESTION	0.12
	THANKING	0.04
	YES-No-QUESTION	0.04
YES-ANSWER	STATEMENT	0.375
	THANKING	0.333
	REQUEST	0.125
	RESPONSE-ACK	0.083
	OPEN-QUESTION	0.042
YES-No-QUESTION	YES-No-QUESTION	0.042
	STATEMENT	0.357
	YES-ANSWER	0.343
	No-ANSWER	0.157
	YES-No-QUESTION	0.057
	RESPONSE-ACK	0.043
	THANKING	0.029
	EXPRESSIVE	0.014

CONVENTIONAL-OPENING	STATEMENT	0.636
	OPEN-QUESTION	0.182
	THANKING	0.182
REQUEST	RESPONSE-ACK	0.464
	STATEMENT	0.179
	YES-NO-QUESTION	0.179
	REQUEST	0.071
	DOWNPLAYER	0.036
	EXPRESSIVE	0.036
	OPEN-QUESTION	0.036
THANKING	STATEMENT	0.369
	YES-NO-QUESTION	0.262
	DOWNPLAYER	0.123
	CONVENTIONAL-CLOSING	0.108
	REQUEST	0.062
	OPEN-QUESTION	0.031
	THANKING	0.031
RESPONSE-ACK	RESPONSE-ACK	0.015
	STATEMENT	0.471
	THANKING	0.235
	YES-NO-QUESTION	0.118
	CONVENTIONAL-CLOSING	0.059
	OPEN-QUESTION	0.059
	REQUEST	0.059
STATEMENT	STATEMENT	0.294
	THANKING	0.212
	YES-NO-QUESTION	0.206
	REQUEST	0.088
	RESPONSE-ACK	0.065
	OPEN-QUESTION	0.053
	EXPRESSIVE	0.047
CONVENTIONAL-CLOSING	CONVENTIONAL-CLOSING	0.018
	CONVENTIONAL-OPENING	0.018
	THANKING	0.444
	RESPONSE-ACK	0.222
	CONVENTIONAL-CLOSING	0.111
	CONVENTIONAL-OPENING	0.111
	REQUEST	0.111

NO-ANSWER	STATEMENT	0.583
	THANKING	0.167
	OPEN-QUESTION	0.083
	RESPONSE-ACK	0.083
	YES-NO-QUESTION	0.083
EXPRESSIVE	STATEMENT	0.455
	EXPRESSIVE	0.091
	NO-ANSWER	0.091
	OPEN-QUESTION	0.091
	RESPONSE-ACK	0.091
	THANKING	0.091
	YES-NO-QUESTION	0.091

Table C.1: Bigram dialogue act transition probabilities.

Appendix D

Utterance Classifications with Dialogue Acts

This appendix lists the manual clusterings that were performed on utterances within dialogue acts. Clusterings were based on pragmatic equivalence.

Utterance	Class
DOWNPLAYER	
you are most welcome	1
ok	2
that's ok	2
you are welcome anytime	1
no problem	2
you are welcome	1
you are welcome customer	1
EXPRESSIVE	
:-)	1
:)	1
wow	2
haha	1
ooh	3
lol	4
hmm	5
hm	5
hehe	1
CONVENTIONAL-CLOSING	
bye customer and have a wonderful time	1
have a great day bye customer	1

good night	2
good bye	2
have a nice day! good bye and take care	2
goodbye and take care	2
cya	2
bye	2
NO-ANSWER	
no customer	1
nah	1
not that i know of	1
not really	1
no	1
YES-ANSWER	
yes actually	1
sure	1
okay customer	1
yes that is correct	1
yep	1
yes	1
yes it worked	1
RESPONSE-ACK	
that's fine	1
great	1
sure	1
ok	1
okay customer	1
i see	2
k	1
okay	1
sure customer	1
looks good	1
THANKING	
thank you!	1
thanks for your patience	2
thank you for the interest	3
thank you for your time and patience	2
thank you for the information customer	4
thank you for approaching us	5

thank you for your appreciation	6
thank you for confirming that	4
thank you for your valuable time	2
thank you for your interest	3
thank you very much	1
thank you	1
thank you for the information	4
thank you for waiting	2
thanks	1
thank you for allowing us to assist you regarding wrist watch	7
thank you for the confirmation	4
thanks for your help	8
thank you customer	1
thank you for contacting msn shopping	5
CONVENTIONAL-OPENING	
hello customer welcome to msn shopping	1
hi	2
hello	2
hello customer	1
hello customer welcome to msn shopping online support	1
hello krishna	3
hello?	4

Appendix E

Online Chat Support Services

This chapter includes a list of companies that provide online chat support and companies that provide software and services for online chat support. The information contained here is correct as of 28 April 2006.

E.1 Online Chat Support

This section lists a selection of companies that provide online chat support to customers.

Company Name	URL
Telstra BigPond	http://my.bigpond.com/internetplans/
Lands' End	http://www.landsend.com/
Dell Inc.	http://support.dell.com/support/topics/global.aspx/ support/en/chat
Hewlett-Packard	http://welcome.hp.com/country/us/en/contact/chat_1. html

E.2 Online Chat Support Software and Services

This section contains a selection of companies that offer software and services for online chat support. Typical features include: canned responses to common questions; scripts that can be selectively run; concurrent sessions, allowing multiple chats at the same time; co-browsing, allowing the agent to view the same web page as the customer; page pushes, where the agent can send a URL to the customer which typically opens in a new browser window; chat history; chat invites, where an agent can initiate or offer a chat with the customer while the customer is browsing the web site.

Company Name	Product name	URL
Bravestorm, LLC	Boldchat	http://www.boldchat.com
LivePerson, Inc.	Timpani Chat	http://www.liveperson.com
Provide Support LLC	Live Chat	http://www.providesupport.com
HumanClick Ltd	Live Chat	http://www.humanclick.com
Lotus/IBM	Lotus Sametime	http://www-142.ibm.com/software/sw-lotus/products/product3.nsf/wdocs/homepage
KUIconnect!	KUIconnect	http://www.kuiconnect.com
Omnistar Interaction	OmnistarLive	http://www.omnistarlive.com
iSupporter	iSupporter Online Assistant	http://isupporter.com
Parker Software	WhosOn	http://www.whoson.com
Lanset America Corporation	Hostik	http://livechat.hostik.com

Appendix F

Penn Treebank Part of Speech tag set

This appendix lists the Penn Treebank tag set from Marcus *et al.* (1993). It contains 36 POS tags and 12 other tags used for punctuation and currency symbols.

1	CC	Coordinating conjunction
2	CD	Cardinal number
3	DT	Determiner
4	EX	Existential <i>there</i>
5	FW	Foreign word
6	IN	Preposition/subordinating conjunction
7	JJ	Adjective
8	JJR	Adjective, comparative
9	JJS	Adjective, superlative
10	LS	List item marker
11	MD	Modal
12	NN	Noun, singular or mass
13	NNS	Noun, plural
14	NNP	Proper noun, singular
15	NNPS	Proper noun, plural
16	PDT	Predeterminer
17	POS	Possessive ending
18	PRP	Personal pronoun
19	PP	Possessive pronoun
20	RB	Adverb
21	RBR	Adverb, comparative
22	RBS	Adverb, superlative
23	RP	Particle

24	SYM	Symbol (mathematical or scientific)
25	TO	<i>to</i>
26	UH	Interjection
27	VB	Verb, base form
28	VBD	Verb, past tense
29	VBG	Verb, gerund/present participle
30	VBN	Verb, past participle
31	VBP	Verb, non-3rd ps. sing. present
32	VBZ	Verb, 3rd ps. sing. present
33	WDT	<i>wh</i> -determiner
34	WP	<i>wh</i> -pronoun
35	WP\$	Possessive <i>wh</i> -pronoun
36	WRB	<i>wh</i> -adverb
37	#	Pound sign
38	\$	Dollar sign
39	.	Sentence-final punctuation
40	,	Comma
41	:	Colon, semi-colon
42	(Left bracket character
43)	Right bracket character
44	"	Straight double quote
45	'	Left open single quote
46	"	Left open double quote
47	'	Right close single quote
48	"	Right close double quote



Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:

Ivanovic, Edward

Title:

Automatic instant messaging dialogue using statistical models and dialogue acts

Date:

2008

Citation:

Ivanovic, E. (2008). Automatic instant messaging dialogue using statistical models and dialogue acts. Masters Research thesis, Faculty of Engineering, Computer Science and Software Engineering, University of Melbourne.

Publication Status:

Unpublished

Persistent Link:

<http://hdl.handle.net/11343/39499>

File Description:

Automatic Instant Messaging Dialogue using Statistical Models and Dialogue Acts

Terms and Conditions:

Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.