

2012

An Approach for Intention-Driven, Dialogue-Based Web Search

Brian Small
University of Windsor

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Small, Brian, "An Approach for Intention-Driven, Dialogue-Based Web Search" (2012). *Electronic Theses and Dissertations*. Paper 5411.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

AN APPROACH FOR INTENTION-DRIVEN, DIALOGUE-BASED WEB SEARCH

by

Brian Small

A Thesis
Submitted to the Faculty of Graduate Studies
through Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2012

© 2012 Brian Small

AN APPROACH FOR INTENTION-DRIVEN, DIALOGUE-BASED WEB SEARCH

by

Brian Small

APPROVED BY:

Dr. G. Bhandari
Odette School of Business

Dr. S. Goodwin
School of Computer Science

Dr. X. Yuan, Advisor
School of Computer Science

Dr. Wu, Chair of Defense
School of Computer Science

14 September 2012

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

Web search engines facilitate the achievement of Web-mediated tasks, including information retrieval, Web page navigation, and online transactions. These tasks often involve goals that pertain to multiple topics, or domains. Current search engines are not suitable for satisfying complex, multi-domain needs due to their lack of interactivity and knowledge. This thesis presents a novel intention-driven, dialogue-based Web search approach that uncovers and combines users' multi-domain goals to provide helpful virtual assistance. The intention discovery procedure uses a hierarchy of Partially Observable Markov Decision Process-based dialogue managers and a backing knowledge base to systematically explore the dialogue's information space, probabilistically refining the perception of user goals. The search approach has been implemented in IDS, a search engine for online gift shopping. A usability study comparing IDS-based searching with Google-based searching found that the IDS-based approach takes significantly less time and effort, and results in higher user confidence in the retrieved results.

Dedication

To my family and friends.

Acknowledgements

I would like to express my sincere gratitude to everyone who has encouraged, supported, or inspired me. In particular, I would like to thank my advisor Dr. Yuan, for his guidance and patience; Dr. Goodwin, for his advice; and Dr. Bhandari, for his positive reinforcement. I am grateful to the Computer Science staff for their help, compassion, and kindness. Lastly, I would like to thank my family and friends for their unwavering support and generosity.

Table of Contents

Author's Declaration of Originality.....	iii
Abstract.....	iv
Dedication.....	v
Acknowledgements.....	vi
List of Tables.....	x
List of Figures.....	xi
Chapter 1: Introduction.....	1
1.1 Problem Statement.....	3
1.2 Contributions.....	4
1.3 Thesis Structure.....	4
Chapter 2: Search Engines.....	6
2.1 Objectives and Classification.....	6
2.2 Architecture.....	7
2.3 Web Search Goals.....	9
2.4 User Interaction.....	10
2.5 Problems with Web Search.....	13
Chapter 3: Spoken Dialogue Systems.....	14
3.1 Architecture.....	14
3.2 Dialogue Management.....	17
3.2.1 Finite State-Based Dialogue Management.....	17
3.2.2 Frame-Based Dialogue Management.....	19
3.2.3 Plan-Based Dialogue Management.....	20
3.2.4 Collaborative Agent-Based Dialogue Management.....	21
3.2.5 Information State-Based and Probabilistic Dialogue Management.....	22
3.2.6 Overview.....	23
3.3 Information State-Based Dialogue Management.....	23
3.3.1 Information Space Theory.....	23
3.3.2 Probabilistic Information State-Based Dialogue Management.....	25
3.3.3 MDP and POMDP.....	27
3.4 Summary.....	33
Chapter 4: Handling Multiple Domains.....	34
4.1 Multi-Domain Dialogue Systems.....	34

4.1.1	Objectives and Challenges	34
4.1.2	Architecture.....	35
4.1.3	Handling Multiple Domain Experts.....	37
4.1.4	Overview	38
4.2	Multi-Domain Search Systems.....	39
4.2.1	PowerAqua.....	39
4.2.2	Search Computing.....	40
4.3	Summary	41
Chapter 5:	Knowledge Representation and Reasoning.....	42
5.1	Handling Certain Knowledge.....	43
5.1.1	Plan and Goal Recognition	45
5.2	Handling Uncertain Knowledge.....	46
5.2.1	Probabilistic Representation using Bayesian Networks	47
5.2.2	Other Approaches	49
5.2.3	Plan and Goal Recognition	49
5.3	Overview	51
Chapter 6:	The Proposed Approach.....	52
6.1	Motivation	52
6.2	Method	53
6.3	Design.....	56
6.3.1	Probabilistic Knowledge.....	57
6.3.2	Other Knowledge	62
6.3.3	Input, Fusion, Output, and Fission.....	62
6.4	Algorithms.....	63
6.4.1	Descriptions	65
6.4.2	Time Complexity	69
Chapter 7:	Implementation	72
7.1	Data Layer	73
7.2	Logic and Date Access Layer.....	79
7.3	Presentation Layer.....	80
7.4	Implementation Tools	84
7.5	Notes on the Implementation	85
Chapter 8:	Usability Study.....	88
8.1	Description	88

8.2	Techniques and Measures	90
8.3	Approval.....	91
8.4	Results and Analysis	92
8.4.1	Descriptive Statistics.....	92
8.4.2	Comparing Time, Effort, and Confidence	98
8.4.3	Explaining the Time, Effort, and Confidence Differences	102
8.4.4	Measuring the Naturalness of the Intention-Driven Approach.....	108
8.5	Summary	110
Chapter 9: Conclusions and Future Work.....		112
Appendices.....		126
Appendix A: Consent Form		126
Appendix B: Recruitment Poster		129
Appendix C: Questionnaire.....		130
Appendix D: Scenarios		132
Guitar Scenarios.....		132
Home Improvement Scenarios.....		133
Appendix E: Experiment Data		135
Vita Auctoris		137

List of Tables

Table 1: Six modules in the general architecture.....	16
Table 2: Different probabilistic models.	30
Table 3: Knowledge representation techniques.	43
Table 4: Modular ontology languages.	45
Table 5: Non-probabilistic knowledge.....	62
Table 6: Results of the paired samples t-test for elapsed time.....	98
Table 7: Paired samples t-test for the effort scores.....	100
Table 8: Paired samples t-test for confidence differences.	101
Table 9: The interaction effects of the area and the scenario on the time differences....	104
Table 10: Interaction between area and scenario for confidence scores.....	105
Table 11: Distribution of the naturalness scores.....	109
Table 12: Summary of questionnaire responses.	111

List of Figures

Figure 1: Standalone search engine architecture [4].	8
Figure 2: User behaviour probabilities. Users tend to browse more than they query [189].	11
Figure 3: General architecture of a multimodal dialogue system [59].	15
Figure 4: Example interaction with a finite state-based dialogue system.	18
Figure 5: TRAINS planning architecture [60].	20
Figure 6: States, observations, and actions over time [73].	24
Figure 7: The probabilistic approach maintains a belief state accounting for different interpretations of user inputs and goals [74].	26
Figure 8: An example partitioning of the problem using HIS heuristic rules [167].	32
Figure 9: The centralized architecture approach [96].	36
Figure 10: PowerAqua workflow [104].	39
Figure 11: A two-slice DBN representation of a factored POMDP. Note that the arcs express conditional dependencies.	48
Figure 12: Bayesian network representing the possibility that going to the liquor store (ls2) is part of a liquor store shopping event (lss2) [123].	50
Figure 13: The information states for different domains are separable.	53
Figure 14: Domain-specific dialogue managers link together to interpret concepts from different points of view.	54
Figure 15: The many knowledge sources. Note that Controller encapsulates the system logic.	56
Figure 16: Hierarchical POMDP organization.	57
Figure 17: The top-POMDP delegates to its child POMDPs [189].	60
Figure 18: An H-POMDP's transitions [142].	61
Figure 19: Algorithm flowchart.	64
Figure 20: Main algorithm.	66
Figure 21: Observation creation.	67
Figure 22: Action selection.	68
Figure 23: Dialogue manager update.	68
Figure 24: Query construction.	69
Figure 25: System layout.	73
Figure 26: Database structure with foreign key constraints.	74
Figure 27: Budget1200 user action.	75
Figure 28: Information gathering template for acoustic guitars.	76
Figure 29: Value constraint for acoustic guitar frets.	76
Figure 30: Redacted fpomdp specification.	78
Figure 31: Hierarchy of dialogue managers.	80
Figure 32: Index page.	81
Figure 33: Index page after transition due to user input showing the system's response.	82
Figure 34: Constraint failure resulting in an error message.	83
Figure 35: Results page.	84
Figure 36: Proficiency distribution.	92
Figure 37: Distribution of the average number of searches per day.	92

Figure 38: Level of knowledge distribution.....	93
Figure 39: Google-based searching elapsed times.....	94
Figure 40: Google-based searching effort scores.....	94
Figure 41: Google-based searching confidence scores.....	95
Figure 42: Distribution of Web page views.....	95
Figure 43: Distribution of the number of sent queries.....	95
Figure 44: Distribution of the number of generated results pages.....	95
Figure 45: IDS-based searching elapsed times.....	96
Figure 46: IDS-based searching effort scores.....	96
Figure 47: IDS-based searching confidence scores.....	97
Figure 48: Distribution of the IDS-based searching naturalness scores.....	97
Figure 49: Distribution of the number of questions asked by the system.....	97
Figure 50: Distribution of the number of constraint failures.....	97
Figure 51: Distribution of the paired time differences.....	98
Figure 52: Distribution of the paired effort differences.....	100
Figure 53: Distribution of the paired confidence differences.....	101
Figure 54: Scatter plot of page views, results pages, and sent queries with respect to the time differences.....	106
Figure 55: Scatter plot of page views, results pages, and sent queries with respect to the effort differences.....	107
Figure 56: Page views vs. confidence differences.....	108

Chapter 1: Introduction

The World Wide Web is a core artifact of the Information Age. In its simplest form, the Web is merely a collection of interlinked documents accessed via a global network of computing devices (the Internet). From a more philosophical perspective, the Web is an embodiment of human knowledge whose borderless influence and decentralized control promote a diversity of cultures, beliefs, and attitudes [1].

Today's Web connects people to data and services [2]. A recent survey found that 91% of all American adults aged 18 or older that access the Internet use search engines to find information online [3]. In particular, 96% of adults aged 18-29, 91% of adults aged 30-49, 92% of adults between the ages of 50 and 64, and 80% of seniors aged 65 or older use search engines [3]. Over and above its informational capabilities, the Web is a global marketplace and a facilitator of real-world activities. For example, the Web allows users to research products online, buy, sell, and review them, and share their experiences using various media including text, sound, and video.

The Web's influence and popularity are in large part due to its search services. Search engines gather, process, and organize online information so that it can be synthesized and presented to users. Without search engines, users would have to painstakingly browse or share links to navigate through the Web [2]. Current search engines adapt traditional information retrieval techniques to deliver relevant results in the form of Web documents, such as Web pages or PDF files, or even answers to the user [4]. They infer relevancy using a variety of techniques, including textual analysis, hyperlink authoritativeness and popularity, and user behaviour analysis [5]. Search engines provide several basic interfaces to allow users to express their needs, including keyword-based, in which users enter several keywords to express their needs, and view-based, in which users incrementally choose categories to hone in on the areas of interest.

The difficulty in finding information is in part due to users' complex, multi-faceted needs that involve multiple implicitly or explicitly mentioned topics [6, 7]. For

example, searching for a home improvement product requires knowledge about the problem at hand, the function and attributes of different product solutions as well as the buyer's budget, experience, and training with these products. The aforementioned traditional methods of user input are not very effective at satisfying these types of queries. They do a poor job of capturing important contextual (background) information, severely limiting their capacity to disambiguate among the many possible goals and relationships the user is seeking information about [8]. This contextual information is important not only for interpreting the true requirements of the user, but also for orchestrating the effective presentation and ranking of the results [9]. For example, a user's query for information on his/her favourite musical artist should provide different results and advertisements if the intention is for album information as opposed to biographical content. Furthermore, existing systems typically require the user to possess expert-level knowledge about the relevant topics (including terminologies), and proficiency in generating high-quality query formulations, for keyword-based systems in particular [10].

In contrast to the basic Web user interfaces, dialogue-based systems provide a conversational interface to ease the knowledge burden on the user and to guide the user to communicate any relevant information. These systems operate over implicit or explicit knowledge, such as task and user models, to achieve reasonable, productive, goal-oriented interactions [11, 12]. They drive the dialogue forward in response to perceived user goals. However, dialogue systems are complicated by their need to address uncertainties in the dialogue process. A system cannot fully observe the intentions or mental state of the user—it has to refine its understanding of the user's needs through an interactive process [13]. This requires knowledge representations, specifically those that handle uncertainties, for example, using probabilities. The information state-based and probabilistic dialogue techniques are particularly important because they offer principled modelling of dialogues with the consideration of uncertainties such as the likelihood of user actions given an utterance or the probability of the user's goals given previous utterances.

Some efforts have been made to support multi-domain interactions and goals. Existing multi-domain dialogue systems are capable of conversing with the user by leveraging templates of information gathering requirements [14]. These systems typically have distributed designs and function by delegating the conversation to a selection of domain experts [15]. Despite recent work on representing and maintaining data about entities and querying this data to generate merged or integrated answers, there are no known dialogue-based multi-domain search engines on the Web.

1.1 Problem Statement

Unfortunately, existing mainstream search engines offer a time consuming search experience that is not conducive to the expression of multi-topic goals. Users are often forced to submit multiple search queries and sift through many different results pages. This problem is compounded by evidence that users are impatient, have difficulties forming high quality keyword-based queries, and are often unfamiliar or inexperienced with the areas they are seeking information about [16, 17, 18, 19]. Dialogue-based front ends attempt to elicit missing information from the user to satisfy the requirements of the scenario at hand. They can be made quite flexible and robust with the addition of probabilistic knowledge for recognizing user goals and speech utterances. However, probabilistic approaches have computational scalability issues—they become intractable as the size of the dialogue environment increases [20].

Existing multi-domain dialogue systems often do not account for uncertainties in the user goals and speech utterances and they typically do not support the processing of multiple domains at the same time. Meanwhile, Web-based search systems that accommodate multiple entities or topics require large curated knowledge bases with well-defined connections. In practice, these knowledge bases are too small (lack knowledge) and differ significantly in their quality and granularity [21]. In other words, the infrastructure is not mature enough to maintain and possess high-quality consistent facts about various entities and their relationships. The Web-based systems offer limited interactive support for non-expert users who may provide unreasonable information [22].

The problem, therefore, is how to outfit a multi-domain Web search application with robust, dialogue-based interaction to provide virtual assistance. The virtual assistance guides the user by generating appropriate suggestions and eliciting information in a natural way. Combining dialogue-based conversation with multi-domain search capabilities involves handling scalability concerns with dialogue management and accessing up-to-date Web resources with the consideration of their semantics.

1.2 Contributions

This thesis presents a novel intention-driven, dialogue-based Web search approach that uncovers and combines users' multi-domain goals to deliver helpful virtual assistance and highly relevant search results. The approach addresses users' difficulties in forming appropriate search engine queries, especially for topic areas that are unfamiliar, by providing expert advice throughout the interactive information gathering dialogue. Probabilistic information state-based dialogue management techniques are employed to enable the scalable consideration of users' complex goals via a hierarchical organization of multi-domain dialogue knowledge encoded in Partially Observable Markov Decision Processes. User errors and inconsistencies are easily detected and recovered from using constraints against a knowledge base, and Web results are fetched according to the recognized user goals. The intention-driven, dialogue-based approach has been implemented in a search engine for online gift shopping. A usability study found that the search approach takes significantly less time and effort, and achieves higher user confidence in the retrieved results, than the predominant Google-based Web search method.

1.3 Thesis Structure

This thesis begins with a reviews relevant background before presenting and examining the proposed method. Chapter 2 surveys existing search engine technologies

and challenges. Chapter 3 examines dialogue-based systems with a particular focus on their dialogue management approaches. The state-of-the-art in multi-domain systems is overviewed in Chapter 4, while Chapter 5 presents relevant knowledge representations and their applicability to the construction of dialogue-based systems. Chapter 6 presents this research work's method, design, and algorithms. Chapter 7 summarizes details of the implemented search engine. Chapter 8 describes and analyzes the usability study. Finally, Chapter 9 supplies potential future work and conclusions.

Chapter 2: Search Engines

Web search is a primary component of users' online activities [3]. Users participate in search sessions—sequences of one or more queries and the exploration of search results—to accomplish their goals. Armed with knowledge sources and processed Web documents, search engines attempt to leverage users' common-goal oriented behaviours to retrieve the applicable relevant results. This chapter overviews key aspects of search engine design, including objectives and architectures, the role of humans in the search process, and the limitations of existing systems.

2.1 Objectives and Classification

Traditionally, search engines process Web documents to return those that are relevant to users' needs [4]. In the Web context, these documents are typically Web pages or PDF files. Search engine performance is often measured in terms of relevancy using the evaluation metrics of traditional information retrieval systems (such as databases). The two key measures are recall, the total number of retrieved documents, and precision, the number of relevant documents [4].

The many relevancy factors that enable document ranking are based on content, link, and behavioural analysis [5]. Modern search engines have tens or hundreds of features that measure the textual relevance of a Web page [5]. These features include the frequency and position of occurrences of query terms, page structure, and graphical layout. Many approaches, such as Hyperlink-Induced Topic Search and Google's PageRank, take advantage of the link structure created by hyperlinking documents, where the links represent directed endorsements of pages whose contents are described by anchor text [23, 24]. Web query mining—the process of analyzing search engine logs to discover and investigate user search behaviours [25]—can be exploited to learn how to weigh the effects of result clicks and other query session data on relevancy [26, 27].

Recent efforts have focused on the importance of knowledge in searching [21]. Knowledge-based search systems, such as Wolfram-Alpha, gather and maintain large collections of assertions about real-world facts and translate user queries into requests over this knowledge to select the facts that are relevant to the query [6]. Likewise, question answering systems such as START take questions as input and return answers [28]. Essentially, these systems extend offline knowledge base techniques to deal with the challenges of scale on the Web.

Overall, search engines can be classified according to the scope of their data sets. General-purpose search engines attempt to cover a broad range of disparate domains [6]. Prototypical examples include Google, Yahoo, and Bing. Domain-specific, or vertical, search engines focus their expertise on specific fields of interest [6]. These search services take the form of weather forecasters, stock pricing monitors, and so on. Extensions of vertical search engines accommodate several related, highly coupled domains whose interconnections are well defined and common for typical search tasks [6]. The medical literature search engine PubMed is domain-specific, whereas PriceLine and Expedia are extensions as they integrate information about numerous topics, including airplane flights and hotel bookings. Finally, multi-domain systems combine partial results about numerous topics to generate integrated, global results [6]. There is very little research in this area as it is quite complex, requiring the maintenance and integration of numerous areas of knowledge. Multi-domain search systems are surveyed in Chapter 4.

2.2 Architecture

The standalone search engine architecture in Figure 1 contains Web crawlers, data processors, and indexers. Multiple distributed Web crawlers fetch online resources that are examined to create indexes or to populate knowledge bases. Indexes are used by document-based search engines to relate terms and features with the documents they are contained within [29]. When a query is submitted, the query terms are matched against the document indexes and the retrieved documents are ranked according to the frequency

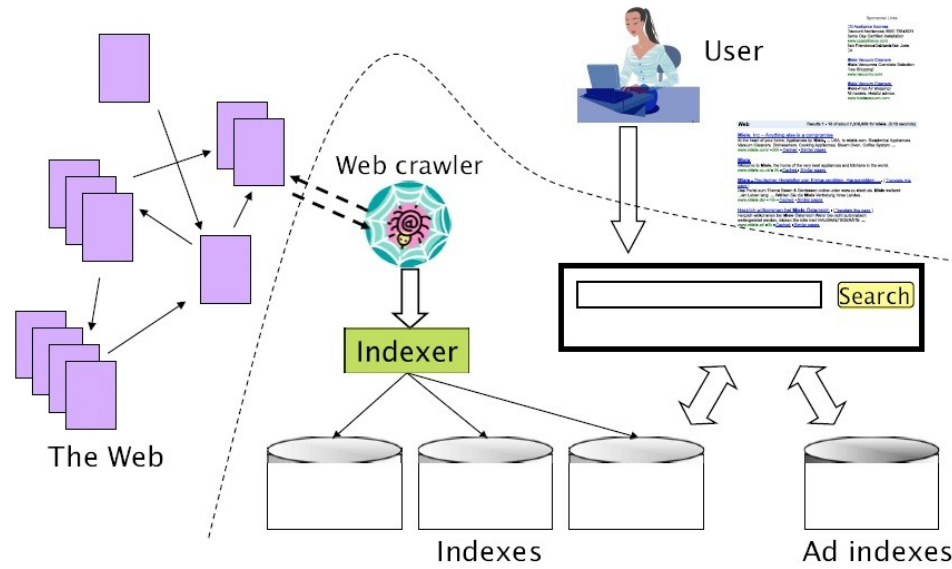


Figure 1: Standalone search engine architecture [4].

of occurrences of those terms along with other statistical scoring measures [4]. Knowledge bases, on the other hand, are created for use by knowledge-based search engines by processing the documents using sophisticated entity extraction techniques to obtain facts [30]. The population of knowledge bases has been aided by the growth of the Semantic Web, which involves the annotation of documents with vocabulary that permits semantic interpretations.

Another type of search engine called meta-search engines delegate the initial document collection and processing to other standalone or meta-search engines. A meta-search engine provides a single interface to multiple search engines and combines the results into an integrated results set [31]. The results are merged according to a fusion policy that takes into account the variability in the underlying search engines' ranking mechanisms.

The emerging trend is to augment traditional document-based search engines with knowledge sources [8]. For example, document-based search engines such as Google support limited knowledge-specific queries, such as checking the current local weather

forecast and browsing movie listings. In particular, knowledge sources are often used to augment and interpret user's queries [32]. General knowledge sources such as the WordNet thesaurus—an ontology that groups English nouns, verbs, adjectives, and adverbs into groups of cognitive synonyms [33]—are employed to add or remove terms from queries or to assess the meaning of the terms. Hypernyms (more general concepts) and synonyms are typically added to increase the number of relevant documents retrieved, while specific limiting terms are appended to both increase the focus of the search query and decrease the number of results [34]. Domain-specific knowledge sources are also used to try to recognize users' queries by identifying and interpreting the input in terms of domain vocabulary.

2.3 Web Search Goals

Users perform Web searching to accomplish specific goals. There are three broad classes of queries and corresponding goals: informational, navigational, and transactional [35]. Informational goals are satisfied by the delivery of static informational content. Navigational goals are achieved by reaching a specific destination Web page or online resource. Transactional goals are associated with the completion of Web-mediated tasks, such as online banking, shopping, or downloading files. This taxonomy has been modified in other works to differentiate between resource-driven searches, in which the user seeks access to online resources, and informational searches, in which the user has a need to access information about a specific topic [9].

Goals provide the motivation for the search but also the parameters for the computation and presentation of results [9]. For example, displaying relevant advertizing may be welcome in a shopping context, but unwelcome in a research context. The ranking and sorting of the results is also affected by the context established by the goal. For example, a search for advice on choosing a career may rate usage factors higher than term frequencies.

Users' goals often involve multiple implicitly or explicitly mentioned topics at the same time [6, 7]. This means that the searcher may start with one topic and then inspect additional topics throughout the search session as new information is acquired and learned. One recent study found that users make one to three topic changes per search session [36]. For example, finding an Italian restaurant close to a movie theatre that is playing a specific movie requires information about restaurants and movie theatres as well as general knowledge about geographic proximity. This type of goal illustrates the importance of considering multiple domains simultaneously. Identifying and discovering a checkout page for a home improvement device that will help with a real-world problem while taking into account the characteristics of the situation and the preferences of the user is another type of multi-domain goal. Such a search task may require consideration of the user's budget, brand biases, and installation expertise.

However, users' goals may be unclear. A user may not understand what he/she wants due to a lack of knowledge [37]. The user may make errors when forming the query, or specify an unachievable goal. In many cases, the user's query—the expression of the need—is ambiguous [38]. For example, a user seeking information about his/her favourite musical artist may wish to locate the artist's biography, official website, or download some of the artist's songs.

2.4 User Interaction

Due to its goal-oriented nature, searching is an interactive human-centric activity. As shown in Figure 2, human users employ a mixture of query and navigation (browsing) strategies to satisfy their goals [17, 9]. Starting with a known website or search engine, a user submits a query, quickly explores the results, and often reformulates the query using the same or a different search engine [12]. Especially for complex queries, users partake in information foraging, executing successive searches over time as the informational need becomes more concrete and the goal appears more achievable [39, 40]. Thus, the search process is multi-dimensional: It is a collaborative process wherein the user

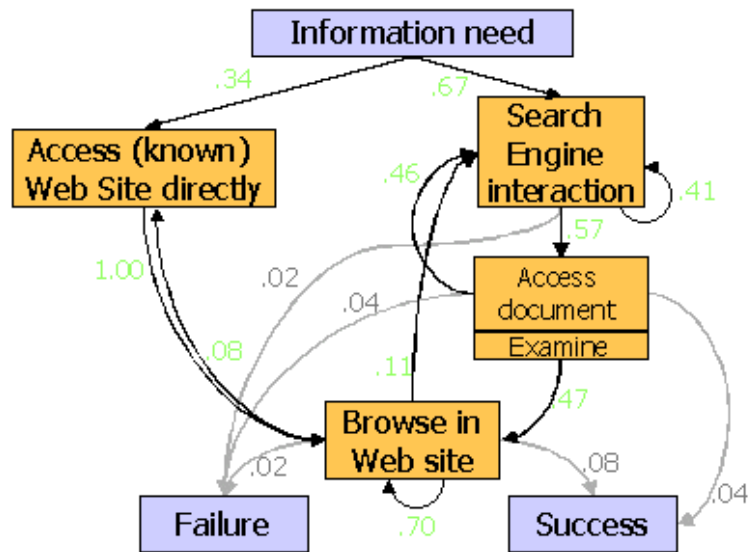


Figure 2: User behaviour probabilities. Users tend to browse more than they query [189].

discovers relevant materials by learning and adapting to the assistance provided by the search system.

The User in Figure 1 may interact with the search engine using various input mechanisms. The most prevalent approach is keyword-based, in which the user provides a short sequence of terms [4]. Keyword-based queries usually consist of about three words on average with a small number of frequently occurring terms and a much larger proportion of terms that rarely appear [36, 41]. A common technique is to combine subqueries into one query using Boolean connectives, such as AND, OR, and NOT [19]. A simple example: (fender OR gibson) AND acoustic. Unfortunately, users have a hard time forming appropriate queries. Analysis of searching behaviours indicates that users often form queries that are too specific or too general compared to the needs of their actual underlying goal [17]. In other words, many users experience difficulties constructing queries that represent the topic or subject they are looking for [18]. Users find it especially difficult to form queries about topics that are complicated or that they are unfamiliar with [19]. Furthermore, keyword-based queries lack expressivity—they cannot state relationships between words and they do not provide adequate context to disambiguate between different interpretations of the keywords [8].

Question answering systems accept natural language input commonly in the form of English language questions. These systems attempt to understand the type of answer the user is interested in by parsing the input to identify the appropriate question category from a hierarchy [42]. Some systems perform relatively simple manipulations of the input, including expanding the keywords and using synonyms and/or morphological variants [43, 44], while others use more sophisticated deep-level parsing to identify grammatical relations between entities in the text [45]. Research shows that knowledge-based systems are particularly well suited to natural language probing [46], but these systems are usually restricted to a specific vocabulary and a limited set of domains [8]. Grammars are difficult to create and are often domain dependent which makes natural language processing very costly in practice [47].

Multi-faceted or view-based searching allows the user to constrain the results by choosing restrictions from the terminological keywords provided by the search engine [48]. This means that the search interaction proceeds over a sequence of turns in which the user enforces or relaxes category constraints to explore the results. Usability studies show that the view-based approach is preferred when users do not know precisely what they want because it allows systematic exploration without the need to guess keywords [49, 50]. View-based searching constructs Boolean queries behind the scenes—adding a concept implicitly constrains the results with an AND while accounting for subconcepts of that concept with ORs [49]. Note that prominent keyword-based search engines like Google and Yahoo provide some support for category-based browsing, with basic support for choosing the type of answers (images, videos, Web).

Less prevalent input mechanisms include humming or singing interfaces that allow the user to perform a query based on content (a melody) rather than metadata (e.g. artist information). For example, Midomi searches for songs given singing or humming input. Obviously, this type of interface is not applicable for typical information seeking tasks.

2.5 Problems with Web Search

Unfortunately, the vast quantity of information available online makes it difficult to find useful information [51]. Although users are typically very confident in their searching abilities, they often feel overwhelmed by the search results or find that critically important information is missing from the results [3]. Users are often forced to sift through numerous documents and formulate multiple queries—a task that many find difficult to perform. Search engines and users alike must account for uncertainties in the authoritativeness of pages, considering the effects of diverse cultures, beliefs, and aims of the authors as well as deliberate search engine optimization schemes [52, 51]. Furthermore, search engines offer limited interactivity, which is a major obstacle to the information foraging activities involved in searching. Although keyword-based searching is simple, it does not allow the user to express relations between words and it lacks the contextual information needed to disambiguate between different interpretations [8].

The key to a better search experience is a deeper analysis of content with powerful support for reasoning about users' intentions [53]. Some researchers are calling for greater emphasis on user goals through intention-based searching, in which users' goals are established and assembled to retrieve results [53]. As discussed in Chapter 3, there is a strong basis for this type of approach in the form of dialogue-based systems that provide goal-based assistance to the user. Contextual information—background information relevant to the user's wishes—is essential for tuning the search to applicable content [12]. In a sense, domain-specific search engines try to reduce the possible context area by only maintaining and supporting certain (usually common) requests for information in a well defined area of knowledge [54]. Thus, the challenge for multi-domain searching is twofold: To simplify context maintenance using domain-specific techniques while providing the interactive capabilities of dialogue-based systems to accept and manipulate applicable contexts.

Chapter 3: Spoken Dialogue Systems

Spoken dialogue systems allow human users to interact with computer-based applications using verbal communication. They provide conversational interaction, which is very useful for eliciting user needs with the consideration of contextual information—an important characteristic for handling complex, multi-domain goals.

Dialogue systems use multiple sources of information, most notably some representation of the information to be derived throughout the dialogue (task model) and a model of user behaviours (user model). These models inform the dialogue manager component, which controls the flow of conversation, mediates between all system subcomponents, and selects system responses. This chapter centres its attention on dialogue management techniques, from the primitive finite state-based approach to more advanced, state-of-the-art probabilistic modelling. Due to its robust, principled handling of uncertainties, the information state-based dialogue management approach is covered in depth toward the end of the chapter.

3.1 Architecture

A spoken dialogue system performs several key operations. The system's main tasks are to accept and process user input, communicate with an external application, and deliver information back to the user. The processing of a single user utterance typically proceeds as follows:

1. The system converts the input speech utterance, consisting of a sequence of acoustic-phonetic parameters, into a string of words [11]. This string is analyzed to produce a meaning representation for the recognized utterance.
2. A dialogue management module orchestrates the updating of one or more dialogue components, including databases and dialogue agents, with the analyzed input utterance.

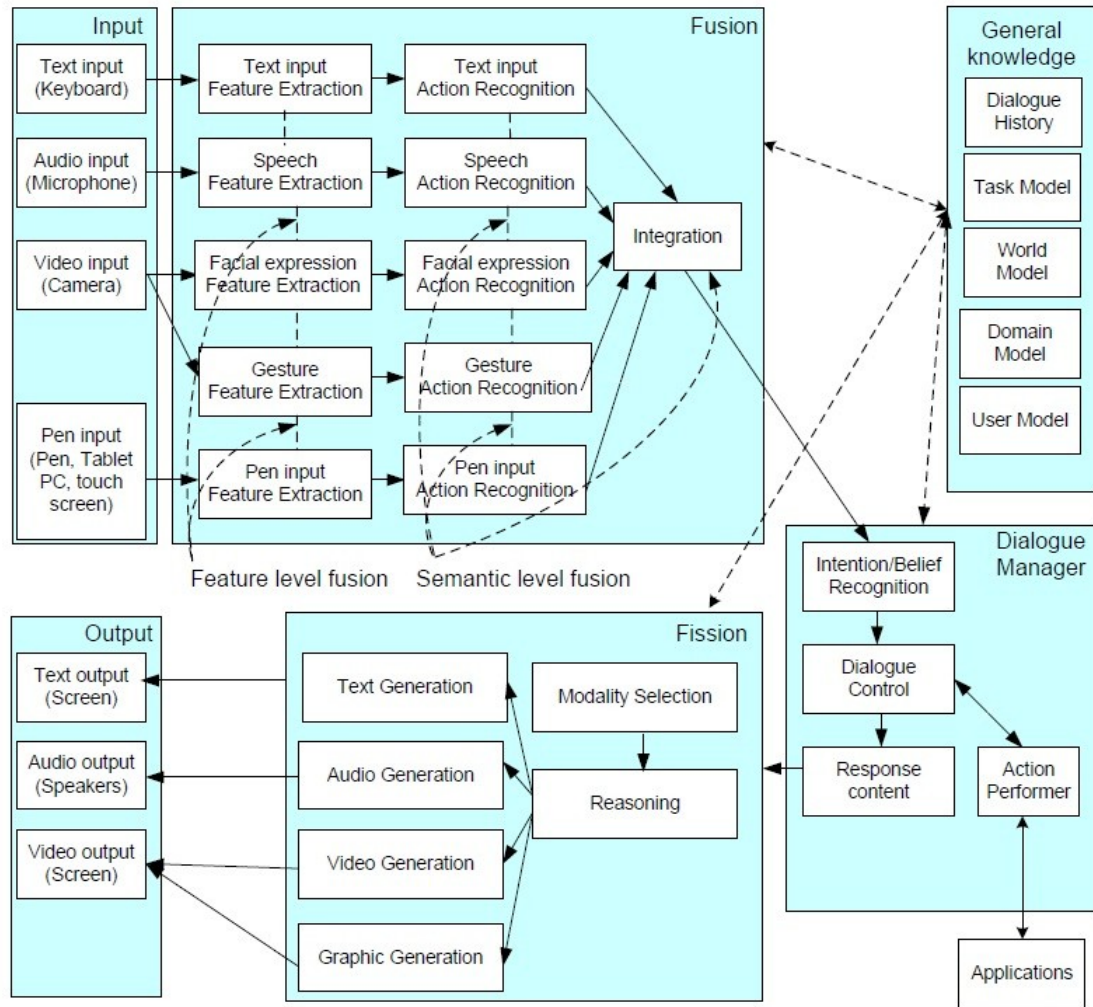


Figure 3: General architecture of a multimodal dialogue system [59].

3. The system creates a response message and outputs it using a text-to-speech synthesis operation.

Dialogue systems adopt a common architecture to implement the aforementioned workflow in a modular, decoupled, flexible way. This architecture accounts for various input modalities, including speech, physical gestures, and eye gaze, to deliver responses in multiple forms. The six modules of the general architecture are depicted in Figure 3 and described in Table 1.

Module	Description
Input	Captures user utterances and behaviours. Common input modalities include lip movements, speech and hand gestures, facial expressions, and eye gaze. The captured inputs are either active, intentionally performed by the user to convey a command to the computer, or passive, naturally occurring behaviour that does not indicate an explicit command to the computer [187].
Fusion	Extracts, recognizes, and integrates the features and actions captured by the input devices to produce a semantic representation that is sent to the dialogue manager.
Dialogue Manager	The controlling component that drives the dialogue process and coordinates interactions among system components.
General Knowledge	Consists of various pieces of knowledge that support the dialogue task.
Fission	Selects and arranges content to present to the user and coordinates the output over multiple output modalities.
Output	Presents the system's response to the user. Common output includes speech and text through the speakers and on the screen, and video output.

Table 1: Six modules in the general architecture.

The Dialogue Manager and the General Knowledge modules determine the flow of control and the rules that govern the conversation. The Dialogue Manager is the engine of the dialogue system. It is responsible for updating the dialogue context according to interpreted communications, providing context-dependent expectations for the interpretations of those communications, interfacing with and coordinating the dialogue system's components, and deciding which type of content to express and when to present it [55]. The Dialogue Manager relies on the dialogue model encoded in the General Knowledge module to provide knowledge that supports its operations. The dialogue model may include multiple knowledge sources [11]:

1. A dialogue history model records the history of interactions in the dialogue, including the mentioned propositions and entities.
2. A task record represents the information that the system must elicit from the user throughout the dialogue. This record is often a form, a template, or status graph.
3. A domain model contains specific information about the domain in question.
4. A world knowledge model encodes general information about the world that supports commonsense reasoning within the application domain.

5. A generic representation of conversational competence contains knowledge about the principles of conversational turn-taking and proper discourse behaviours.
6. A user model maintains information about the user that may be relevant to the dialogue, including the user's age, gender, beliefs, intentions, and preferences.

3.2 Dialogue Management

Using the information in the dialogue model, the Dialogue Manager executes a control strategy that dictates the flow of the conversation between the user and the system. Dialogue control may be user-led, system-led, or mixed-initiative [11]. In a user-led dialogue, the user controls the dialogue by asking questions to the system. By contrast, in a system-led dialogue, the system controls the dialogue flow by prompting the user for certain pieces of information. A mixed-initiative dialogue allows both the system and the user to take turns directing the conversation. The user can ask questions at any time, but the system can still demand required information or ask for clarification about unclear information.

Dialogue management techniques differ in how they represent and process dialogue tasks. The finite state-based approach encodes the possible pathways of interaction sequences necessary for satisfying a domain-specific need, whereas the frame-based technique encodes stereotypical situations or entities as templates to be filled. The plan- and collaborative agent-based methods rely on accurate representations of speech acts and their interrelations as well as planning algorithms to connect possible plans and goals with appropriate system responses or actions. Information state-based approaches store a summarized account of the dialogue itself and use it to plan and choose actions.

3.2.1 Finite State-Based Dialogue Management

In the finite state-based approach, the system elicits information from the user in a constant, well-defined sequence. The system maintains control of the dialogue and

System: What company would you like to pay?
User: Abbey National.
System: How much would you like to pay?
User: One hundred pounds next Monday.
System: What date would you like the payment to be made on?
User: Next Monday.

Figure 4: Example interaction with a finite state-based dialogue system.

produces prompts at each dialogue state. The user's input is recognized and parsed into specific words or phrases in response to the prompt, and the system then generates appropriate output messages.

For example, the Nuance automatic banking system allows users to conduct bank transactions over the telephone, such as paying a bill or obtaining an account balance. The user can enter relatively unrestricted speech input and he/she can specify certain combinations of values at once [11]. However, the system is not responsive to over-informative answers and cannot correct more than one error at a time [11]. Figure 4 shows an example interaction sequence from [11].

The finite state-based approach is simple to design and implement. State transition networks are easy to construct and they intuitively express the predetermined interaction sequence. The technique does not require complex natural language processing or speech recognizers because the accepted combinations of user inputs are predetermined [11]. It is particularly suitable for domains with highly structured tasks for which there are well known, widely accepted processes for information elicitation. For example, directory assistance and travel inquiries can be constrained to a series of system-led questions with well-defined responses.

Unfortunately, the technique is inflexible as it prescribes a specific sequence of system behaviours and expected inputs. It is not effective when the conversation does not follow a predictable order or when complex dependencies link the informational items [56]. Dependencies between items of information cause a combinatorial explosion

of states and transitions in the finite state graph [11]. Similarly, the graph grows unmanageably large if it allows users to change their answers.

3.2.2 Frame-Based Dialogue Management

A frame-based system asks questions to gather information to fill a predefined template of required information [57]. The dialogue approach guides the user to provide a value for each slot in the template. For example, the Philips Automatic Train Timetable Information System [58] delivers information over the telephone about train connections between selected German cities. By specifying the values for items such as the arrival time, destination, and departure time, the system helps the user to construct a database query that retrieves the desired timetable information.

Frame-based systems are flexible and efficient [59]. The dialogue flow is not predefined so questions are not asked in a predetermined order. Systems typically use a priority question ordering to choose which question to ask next [59]. The user can insert corrections to items that the system has misrecognized or misunderstood, and users' over-informative answers are parsed [11]. The system fills multiple slots to take into account all of the user-provided information. This saves time and reduces the number of questions the system asks.

Frame-based systems are not appropriate in all situations. They are not suitable for eliciting information about areas that are not well defined [11]. For this reason, frame-based systems cannot negotiate a task or collaboratively plan some activity. The system context that contributes to the determination of the next action is limited as it only considers the user's previous utterance and the filled-in slots [59]. Thus, this approach is not applicable for modelling a dynamic environment or world model. Although frames are simple to design, the application developer may have to do a significant amount of experimentation to ensure that rules fire appropriately in their particular contexts [11].

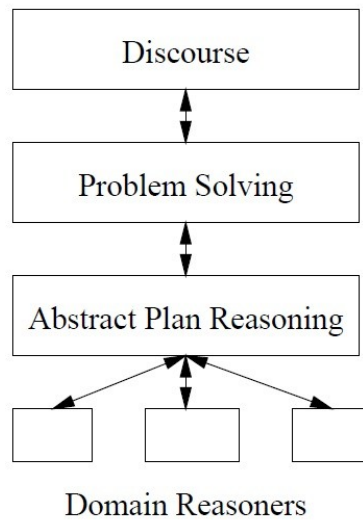


Figure 5: TRAINS planning architecture [60].

3.2.3 Plan-Based Dialogue Management

Plan-based dialogue managers view the dialogue as a sequence of interactions that form part of a plan that achieves an underlying goal [60]. A user’s utterance is typically perceived as a speech act—a function or action such as a request, promise, warning, or confirmation [61]. The dialogue manager tries to discover the user’s plan by reasoning about the observed speech acts. By recognizing the plan, the dialogue manager can effectively respond within the context-dependent dialogue. The idea is that by understanding the overall goal of the user, the system can direct the conversation in a natural way. For example, in response to the user’s question “Where are the steaks you advertized?”, the system may adeptly reply “How many do you want?” because it recognized the user’s plan to purchase steaks [62].

As an example, the TRAINS system supports collaborative problem solving using a plan-based approach [63]. As shown in Figure 5, the current plan is assessed by evaluating the input speech acts in the context of the discourse and finding causal and motivational connections between interpretations of those speech acts by problem solving and reasoning over possible compatible plans.

The plan-based approach is impractical in real-world applications. The process of plan recognition involves chaining from preconditions of plans to the system actions, which can be computationally intractable [11]. Furthermore, incorrect recognition and identification of the user's speech act could result in the incorrect assessment of the user's plan. Complex intention reassessment mechanisms are needed to work around this problem [15]. The plan-based approach is only applicable for restricted problem domains in which the reasoning is manageably small. Finally, plan-based approaches lack a sound theoretical basis for recognizing the plan [59].

3.2.4 Collaborative Agent-Based Dialogue Management

A collaborative agent-based dialogue manager models the communication as an interaction between two agents, the user and the system, each of which reasons about its own beliefs and actions (and perhaps those of the other participant) to achieve a common overall goal [59]. In contrast to other approaches, collaborative approaches attempt to capture the motivations behind the dialogue rather than just the structure of the dialogue itself [59]. There are many types of collaborative agent-based approaches, including theorem proving, distributed architectures, and conversational agents [11]. For example, TRIPS integrates the activities of a conversational agent and a problem solving agent to interpret user communications and create, rank, and adjust plans for system responses [64].

Collaborative agent-based approaches are very sophisticated and can handle complex dialogues that require problem solving and negotiation between the user and the system [59]. However, they demand many resources and processing capabilities [11]. Sophisticated natural language processing and deep semantic interpretation of the user's input are required to deal with open-ended, mixed-initiative dialogue. Existing systems are difficult to extend with support for additional domains. Since these systems often employ plan-based reasoners, their intention recognition functionality can be computationally intensive [11].

3.2.5 Information State-Based and Probabilistic Dialogue Management

The information state-based dialogue management approach focuses on maintaining a representation of the dialogue in terms of cumulative additions from previous actions to motivate future actions [55]. The approach models both the structure of the dialogue and user-centric notions such as beliefs, intentions, and plans to describe the dialogue in a way that enables a planning agent to choose effective actions.

The dialogue is described in a rich, flexible way containing multiple relevant pieces of knowledge, including [55]:

1. Descriptions and formal representations of informational components, including the participants, beliefs, obligations, commitments, and linguistic and intentional structures.
2. Dialogue moves that trigger updates to the information state.
3. Update rules that determine how the information state is altered.
4. An update strategy that decides which rules to apply and when to apply them.

Numerous toolkits apply information state-based dialogue management. Examples include TrindiKit and GoDiS [65, 66]. Specific dialogue systems include MATCH and Virtual Music Center [67, 68].

In recent years, probabilistic information state-based dialogue managers have emerged to account for uncertainties in the dialogue. Many systems model the dialogue as a Markov Decision Process (MDP), which enables the computation of dialogue strategies in a fully observable environment [69, 70]. Partially Observable Markov Decision Process (POMDP) modelling allows the dialogue state to be uncertain and is used in several dialogue systems [13, 71, 72].

3.2.6 Overview

The various dialogue management techniques offer different techniques for representing and reasoning about dialogues. The finite state-based and frame-based approaches seem to be the most prevalent due to their simplicity, but they do not provide the flexibility and robustness of the other techniques. The probabilistic information state-based technique has become particularly influential because it provides a principled, statistical method to capture and model the important parts of the dialogue and their effects on the system's planning decisions. Information state-based techniques in general promote the consideration of multiple pieces of knowledge, including user behaviours and specific domain factors or variables. Most importantly, the information state-based technique is a framework that naturally handles the inherent uncertainties in the dialogue, including the misrecognition of user input and the misidentification of possible user goals.

3.3 Information State-Based Dialogue Management

As previously mentioned, the information state-based dialogue management approach operates over an up-to-date representation of the dialogue. This dialogue representation encapsulates a history information state—a configuration of the dialogue in terms of summarized past interactions.

3.3.1 Information Space Theory

The information state-based approach to dialogue management is grounded in information space theory. Information space theory states that an agent acting in an uncertain environment can plan and act using its (noisy) perceptions of the world by maintaining a state representation in terms of its history of observations and actions [73]. The agent can use the information it knows to estimate the state, forming a plan and hoping that it works under reasonable estimation error [73]. Alternatively, the agent can

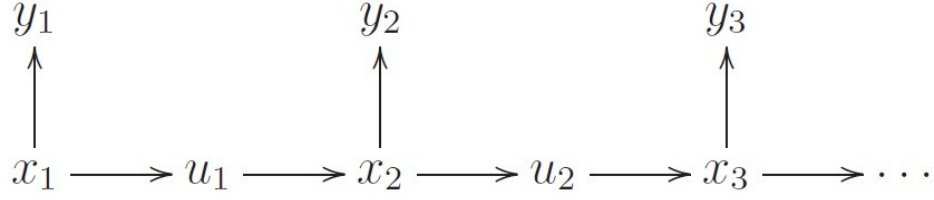


Figure 6: States, observations, and actions over time [73].

solve the task entirely in terms of the information space without ever actually knowing the exact state. This latter approach is simple and can be more computationally viable than the former technique [73].

As shown in Figure 6, the agent observes the state of the environment and uses this information along with its history to select and execute an action at each time step. In other words, the agent executes actions $u_k \in U$ in response to observations $y_k \in Y$ of the hidden states $x_k \in X$.

The agent's history at time step k is one particular configuration, or information state, within the history information space. Whereas the history information space defines every possible history, a history information state refers to one particular history. The history information space at time step k summarizes the initial (starting) conditions and the history of all actions and observations up to and including time step k :

$$I_k = I_0 \times \tilde{U}_{k-1} \times \tilde{Y}_k$$

where

I_0 denotes every possible set of initial conditions

\tilde{U}_{k-1} denotes the set of all action histories

\tilde{Y}_k denotes the set of all observation histories

Thus, the history information state at time step k is defined as:

$$\eta_k = (\eta_0, \tilde{u}_{k-1}, \tilde{y}_k)$$

where

η_0 denotes the initial conditions

\tilde{u}_{k-1} denotes all actions executed up to and including time step $k-1$

\tilde{y}_k denotes all observations up to and including time step k

If there are K stages, the history information space is:

$$I_{hist} = I_0 \cup I_1 \cup \dots \cup I_K$$

By casting the problem environment in terms of an agent that maintains and updates an information state, the planning task involves the construction of a plan over the history information space. The agent repeatedly interacts with the environment to learn a mapping from history information states to actions, $\pi : I_{hist} \rightarrow U$. Using this mapping, the agent attempts to minimize a cost function (or maximize a reward function) that is applied to each state-action history to find an optimal plan for the task. An optimal plan is thus one that incurs the lowest costs (or the highest cumulative rewards).

3.3.2 Probabilistic Information State-Based Dialogue Management

A popular extension of the information state-based approach is to model the uncertainties inherent in the dialogue process using a probabilistic information state. A probabilistic information state is a probability distribution over the possible true state configurations. An information states is called a belief state, as it represents the likelihoods of a specific configuration of information representing the dialogue. As shown in Figure 7, a probabilistic dialogue manager maintains a distribution across all

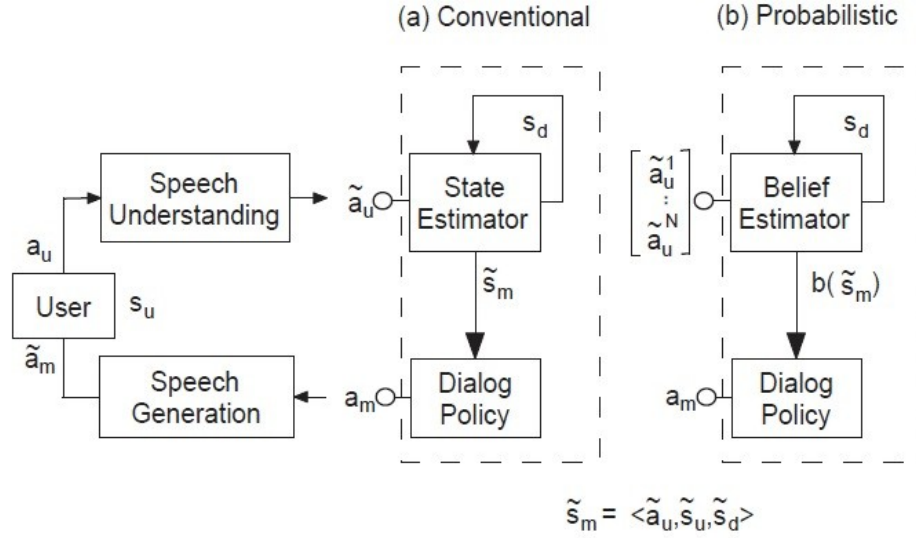


Figure 7: The probabilistic approach maintains a belief state accounting for different interpretations of user inputs and goals [74].

states rather than a point estimate of the most likely state. In other words, the dialogue manager tracks all possible dialogue paths rather than just the most likely one [74].

Beliefs are typically represented using Bayesian network-based formalisms. This type of approach, covered in more detail in Chapter 5, allows the specification of variables and their dependencies with respect to characteristics of the dialogue environment. For example, a bilingual hotline for real-time foreign exchange inquiries uses two goal-specific Bayesian networks and combines their decisions to identify the informational goal of the input query and to produce a system response to address missing information [75]. The system in [76] represents the dialogue as a hierarchy of Bayesian networks, choosing system actions that yield the highest information gain. Powerful probabilistic modelling tools such as Markov Decision Process (MDP) and Partially Observable Markov Decision Process (POMDP) can be represented as Bayesian networks and have recently been studied for dialogue modelling.

3.3.3 MDP and POMDP

An MDP encodes a fully observable problem environment. Early spoken dialogue systems, such as the one developed for the ARPA ATIS task, model the dialogue as an MDP using an additive expected dialogue cost function as an objective function to optimize [69]. These early dialogue systems are limited because they do not account for uncertainties in the speech recognition results and the goals of the user [13].

A POMDP extends an MDP by providing a complete and principled framework for modelling uncertainties [74]. It naturally considers the uncertainty in the estimate of the user's goal as well as the uncertainty in the speech recognition result [74]. Like an MDP, a POMDP follows Markovian dynamics: the last belief state and last executed action determine which action to perform next [77]. Formally, a POMDP model is a 7-tuple $\langle S_m, A_m, T, R, O, Z, \lambda, b_0 \rangle$:

1. A finite set of hidden environment states, S_m . The states are “hidden” because the agent cannot directly perceive them. The states typically represent the hidden goals of the user.
2. A set of actions that the machine may take, A_m .
3. A transition probability function, T , that specifies the likelihood of the next state given the current state and action, $P(s'_m | s_m, a_m)$
4. A reward function, R , that sets the positive or negative feedback the agent receives as a result of its interactions. Typically, the reward function is defined over each state-action pair, such that the expected immediate reward of executing action a_m in state s_m is given by $r(s_m, a_m)$.
5. The set of observations of user utterances, O .
6. An observation probability $P(o' | s'_m, a_m)$ defined by Z .
7. A discount factor, λ , where $0 \leq \lambda \leq 1$ that determines the relative influence of action rewards depending on when they occur. Future rewards usually have less influence than current rewards so the agent is encouraged to make the best move at each time step.

8. An initial belief state, b_0 . The initial belief state is a probability distribution over the states which describes the likelihood of starting in each state.

At each time step k , the machine is in some unobserved state $s_m \in S_m$. Due to uncertainties, the probability of being in each state s_m is given by the belief for that state, $b(s_m)$. Using the current belief state b , the machine selects and executes some action $a_m \in A_m$. The machine receives a reward for that action, as given by $r(s_m, a_m)$, and transitions to some new unobserved state s'_m . A user generates an utterance, which is recognized by the machine in the form of an observation $o' \in O$. Given this evidence of the unobserved state, the machine updates its belief distribution b using Bayes' probabilistic rule. For each state $s'_m \in S_m$ [78]:

$$\begin{aligned}
 b'(s'_m) &= P(s'_m | o', a_m, b) = \frac{P(o' | s'_m, a_m, b) P(s'_m | a_m, b)}{P(o' | a_m, b)} \\
 &= \frac{P(o' | s'_m, a_m) \sum_{s_m \in S_m} P(s'_m | a_m, b, s_m) P(s_m | a_m, b)}{P(o' | a_m, b)} \\
 &= \frac{1}{P(o' | a_m, b)} \times P(o' | s'_m, a_m) \sum_{s_m \in S_m} P(s'_m | a_m, s_m) b(s_m)
 \end{aligned}$$

The value of the generated plan is typically computed as the cumulative, infinite horizon, discounted reward given by [78]:

$$R = \sum_{t=0}^{\infty} \lambda^t r(b_t, a_{m,t}) = \sum_{t=0}^{\infty} \lambda^t \sum_{s_m \in S_m} b_t(s_m) r(s_m, a_{m,t})$$

Given multiple action choices at each state, reinforcement learning is used to systematically explore behaviours. Multiple simulations of the POMDP system are completed to compute the best plan for action selection based on rewards associated with each state transition [79]. An optimal plan, or policy, is always piecewise linear and convex in the belief space [80]. This means that it can be represented by a set of policy vectors, where each vector is associated with an action and the value for a specific state

on a vector yields the expected value of the optimal action in that state. In other words, a policy is a partitioning of belief space where each partition corresponds to an action [81].

As a concrete example of a simple POMDP-based dialogue system, researchers developed a nursing home robot assistant to allow users to find information about several domain-specific areas, including time, medications, and TV schedules [13]. In contrast to the typical modelling approach, the system models the state of the user rather than the system's state. The researchers found that as the speech accuracy degrades, the POMDP increasingly outperforms the non-probabilistic MDP approach.

3.3.3.1 Factored POMDP

A factored POMDP separates the definition of the state into multiple components. This makes it easier for POMDP designers to consider multiple factors in the transition function and it provides a richer state definition. The parameters that determine the transition probabilities can be made independent and, thus, estimated separately.

A factored POMDP architecture is used in a travel domain ticket purchasing dialogue system, a telephone-based question answering system, and a virtual tour guide [72, 68]. In this approach, the state variable $s_m \in S_m$ is separated into three components [72]:

1. The user's goal, $s_u \in S_u$. The goal corresponds to the user's need or motivation. For example, the user's goal may be to request information about a calendar or to choose a particular product configuration.
2. The user's actual action, $a_u \in A_u$. Examples include responding to a yes/no question or specifying a product's colour.
3. The state of the dialogue, $s_d \in S_d$, which indicates relevant dialogue state information from the user's perspective, such as which information is already specified. The dialogue state is important for providing dialogue context.

Model	Description	Formula
User goals	Indicates how the user's goal changes at each time step.	$P(s'_u s_u, a_m)$
User actions	Indicates which actions the user is likely to take at each time step.	$P(a'_u s'_u, a_m)$
Dialogue	Indicates how the user and the machine's actions affect the state of the conversation.	$P(s'_d s'_u, a'_u, s_d, a_m)$
Observation	Determines the most likely observations of user actions.	$P(o' a'_u)$

Table 2: Different probabilistic models.

Given the aforementioned factorization, the transition probabilities are decomposed into a user goal model, user action model, observation model, and a dialogue model (as described in Table 2). Making some independence assumptions, these models can be generated and designed separately, allowing the decoupling of significant areas of uncertainty modelling. For example, the observation model can be determined from a corpus or derived using a phonetic confusion matrix, language model, etc. [82].

The factoring also enables a richer reward function description. For example, the reward measures can incentivize or promote certain actions based on the user's goal or the dialogue state.

The belief state update equation for the factored POMDP is [78]:

$$\begin{aligned}
& b'(s'_u, a'_u, s'_d) \\
&= k \times P(o' | a'_u) P(a'_u | s'_u, a_m) \sum_{s_u} P(s'_u | s_u, a_m) \sum_{s_d} P(s'_d | s'_u, a'_u, s_d, a_m) b(s_u, s_d)
\end{aligned}$$

where

$$k = \frac{1}{P(o' | a_m, b)}$$

There are other state factorizations. The system presented in [71] splits the POMDP state into a user intention component as well as a hidden system state component. This allows low-level information obtained from the multi-modal inputs to

be considered in the assessment of the observation as well as the state. In addition, the actions are divided into two classes to simplify the transition function. Only actions that gather more information from the user are assumed to cause state transitions. The approach in [83] incorporates the user's emotional or affective state into the dialogue model by factoring the state into four components (the user's goal, the user's affective state, the user's action, and the user's grounding state) and by including the affective state in the observation definition.

3.3.3.2 Scalability

POMDP solution procedures do not scale well. The basic exact solution algorithm, value iteration, involves the repeated computation of the policy vectors for all possible action-observation pairs [74]. As the number of iterations increases, the estimated value function converges to the actual (optimal) value function from which the policy is derived. However, even with pruning of some generated policy vectors, this approach is computationally intractable. The size of the policy space grows exponentially with the size of the observation set and doubly exponentially with the number of time steps from the horizon [84].

Approximate solution algorithms use heuristics to get a near-optimal solution. Some approaches, such as MDP approximation, assume that the state is fully observable, thus ignoring the uncertainties or relying on a reasonable error estimate. Grid-based approximation involves considering only a few belief states. Different strategies are used to select these belief states, including random selection and picking those that define the extremities of the state space [71]. Examples include point-based value iteration [85] and value directed compression with bounded policy iteration [86].

There are two main approaches to achieve a practical and tractable POMDP-based dialogue system [78]. The state can be factored into simple discrete components each of which has an associated probability distribution. This technique is used in slot filling applications, where the purpose of the dialogue is to provide values for all of the slots, or

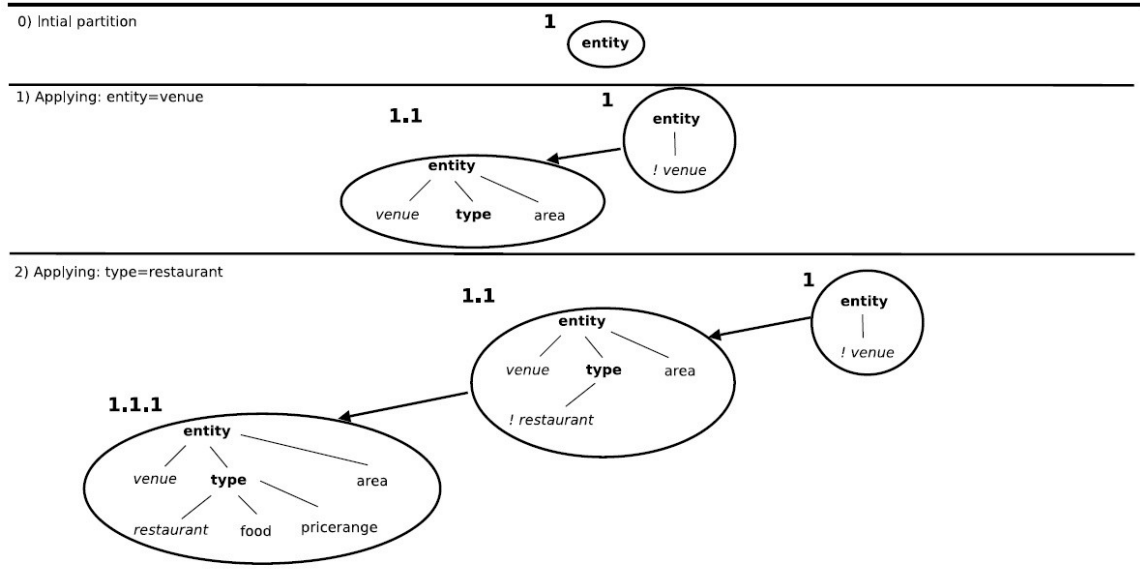


Figure 8: An example partitioning of the problem using HIS heuristic rules [167].

properties. For example, the Bayesian Update of Dialogue State (BUDS) framework represents the state of a POMDP with a set of slots [87]. Conditional independence assumptions are made so that the belief update acts on a per slot basis. A slot's associated beliefs are only updated if the slot is activated. This technique may introduce a summary space to simplify belief maintenance [81]. A summary space consists of the top N user goal states from a master space and a simplified encoding of the user actions, observations, and dialogue history. At each dialogue turn, the belief state is updated in master space and mapped to a belief state in summary space. Then, an optimized (simpler) dialogue policy is applied in summary space to select a new machine action. This machine action is mapped back into master space and then executed.

Another method is to retain a full, rich state representation but only maintain probability estimates over the most likely states. Essentially, this approach maintains probabilities across a set of conceptual dialogue managers [78]. At each dialogue turn, the probability of each dialogue manager representing the true state of the dialogue is computed and the system response is based on the probability distribution over all the dialogue managers. For example, in the HIS system, similar belief states are grouped

into partitions and a single belief state is maintained for each partition [88]. The space of all user goals is defined by some domain-specific ontological rules. When a user performs a speech act, it is matched against each partition's goal. If there is no exact match, the partition is refined (i.e. partitioned) according to the rules [88]. For example, the probability mass is redistributed in Figure 8 to give a higher likelihood to restaurant venues. The HIS system also makes use of a summary space: the master belief state is mapped into summary belief state and the nearest policy belief point is found and used to identify a machine action which is mapped to a master space machine action. This approach can become unwieldy as the dialogue progresses over time and more partitions are created [87].

3.4 Summary

Spoken dialogue systems are useful tools for extracting necessary information from the user. They typically provide support for a single domain area by capturing expected inputs and associating them with goals or plans. Robust dialogue management approaches that deal with uncertainties, such as information state-based methods, do not scale very easily to new domains or alternative constraints and values. However, many techniques have been presented to simplify the belief state update procedure to increase computability.

Chapter 4: Handling Multiple Domains

Although most dialogue systems only support a single domain area, some try to accommodate multiple areas. These multi-domain dialogue systems incorporate a scalable distributed architecture with frame-based domain experts. In the Web context, current multi-domain search systems that systematically integrate results by domain area do not allow verbal communication.

4.1 Multi-Domain Dialogue Systems

Existing spoken dialogue systems usually support a single domain or area of expertise [11]. For example, Jupiter is a telephone dialogue system for obtaining weather information, and TOSBURG-II is a fast food ordering system [89, 90]. Other restricted-domain spoken dialogue systems have been developed for flight reservations [69, 91], train travel [92], bus information [15, 93], and in-car navigation [94].

Limiting the conversation to one or a limited set of domains is problematic. Users must be aware of the limitations of the system to ensure that their utterances are understood [15]. As users' tasks often require information from multiple domains, systems must be able to maintain knowledge and support dialogue about them. For example, a driver support system should support various task domains, such as the air conditioner, car radio, navigation system, and vehicle information system [95].

4.1.1 Objectives and Challenges

The main challenges for designing multi-domain dialogue systems are scalability/extensibility and robustness/consistency [15]. The key functional requirements are summarized in these conditions:

1. The dialogue system should work reasonably well even as new domain support is added [15, 95].
2. The system should handle many different user utterances consistently across domain areas by identifying the correct domains and switching among them as needed [96].
3. Speech recognition errors should be managed and recovered from in an appropriate manner [15].

However, performance degradation is inevitable as new domains are added due to the expansion of the vocabulary size and grammar rules and the addition of language models and domain knowledge [96]. The speech recognition performance degrades as the vocabulary size increases. Furthermore, it is very difficult to tune one domain without affecting another [96].

4.1.2 Architecture

Existing multi-domain spoken dialogue systems typically use a distributed agent architecture of domain experts and system modules to achieve the scalability and robustness objectives. In this architecture, the system is composed of two types of components: those that are designed independently of all other domains, and those that consider domain relations [15]. Systems attempt to minimize the impact of the latter type of components to create more extensible and modifiable implementations.

The most common approach is the master-slave architecture in which a master module coordinates the selection of slaves (domain experts) which, in turn, determine how the user's utterance is processed and how a response is generated [15]. [95] provides a compositional architecture of hierarchical modules based on the notion of passing fragments between system modules. Different domain managers control work modules that know how to converse about specific domains. A master module decides the relevancy of each input fragment (recognition of a user utterance) for each work module, distributes the fragment to all of the work modules for processing, and then

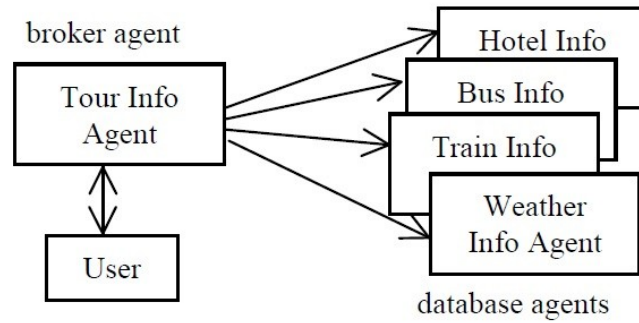


Figure 9: The centralized architecture approach [96].

integrates the responses to generate a system response. In [15], the system employs a central module that performs speech recognition and selects the expert that will contribute the next system response. Each domain expert processes the user's utterance, but only the expert that is selected for the next dialogue turn retains its updated dialogue state. [96] proposes a three component architecture, consisting of a user interface agent, one or more spoken dialogue agents (one for each domain), and a shared data store containing state-dependent data. A facilitator component switches control between agents by loading the dialogue state and history persisted in the data store into the newly selected agent. The facilitator decides when to perform domain switching by transforming the input utterance into a rich semantic structure (such as a phone lattice) and choosing the expert that is most compatible with it.

The centralized approach has one component that manages the entire dialogue state as well as the domain knowledge. As shown in Figure 9, a broker agent accepts and understands the user's requests and sends formatted queries to domain experts. This approach is not practical because the broker agent must be extremely complicated and it must possess a lot of knowledge to allow the dialogue to switch smoothly across different domains [96]. This approach is difficult to manage and scale with additional knowledge.

Another option is the blackboard technique. Communication between agents is mediated by a blackboard module that notifies specific agents when relevant changes are made [97]. There is no central agent responsible for planning or coordination. For

example, the SesaME dialogue manager's Interaction Manager implements a blackboard that stores and modifies the dialogue information in response to events including dialogue moves, internal events, and changes in the user's external context [97].

4.1.3 Handling Multiple Domain Experts

The systems assign different responsibilities and labels to the experts. Some systems require dialogue agents to perform very simple tasks, while others demand extensive discourse knowledge and behaviours [14]. In [98], simple generic error-handling agents ask the user to repeat misunderstood input. Most systems, however, employ agents that implement a skill set for a substantial dialogue or subdialogue for a specific transactional area. [99] distinguishes between experts that cause verbal actions and experts that cause physical (robot) actions. In addition, these experts are classified as user-initiated or system-initiated experts. SesaME has task-specific agents as well as decision agents, which evaluate results produced by the task-specific agents [97]. [100] differentiates between service agents, that encapsulate behaviours typical of a particular business domain, and support agents, that provide cross-domain functionalities. [15] treats the experts as independent dialogue managers with their own language understanding modules and dialogue updating procedures.

In order to accommodate the many domain experts, the dialogue systems employ various domain selection procedures. A domain selection procedure chooses one or more experts to process the user's input and generate a system response. Many conventional methods perform domain selection by estimating the most likely domains based on the speech recognition results [15]. SesaME extracts topic vectors and keywords from the domain descriptions and the user input to identify the domains of interest [97]. Many systems consider the history of domain selections. [96] gives preference to the previously selected domain expert by adding a score when comparing the N-best candidates of the speech recognition for each domain. [14] imposes subtask completion behaviour: the system does not change its domain until the current subtask is completed. [15] considers multiple factors in the domain selection procedure, including the previous

domain, the domain whose speech recognition results have the highest recognition score, and the possibility that the current speech recognition interpretations are incorrect. The system presented in [101] identifies the target domain by examining the input's keywords (with well-known links to domains) as well as linguistic and semantic features.

The domain experts employ various dialogue management approaches. STAR uses a frame- and collaborative agent-based architecture based on the TRIPS framework [102]. The Task Manager is frame-based, representing each domain as a separate template. The Queen's Communicator is also collaborative with frame-based, distributed agents. The agents collect and manipulate frames of information containing types, values, levels of confirmation, and rules for detecting database-determined constraints and for determining the agent's reaction to the information combinations [14]. [15] and [96] represent each expert as a frame-based system with common slots shared between experts.

4.1.4 Overview

The overarching theme in multi-domain dialogue system design is the pursuit of scalability. Systems employ scalable infrastructures consisting of decoupled dialogue control and domain knowledge, distributed agents, and advanced agent selection techniques. This infrastructure provides support for the addition of new domains, or the ability for the system to handle more and more user goals.

However, current systems are error-prone as they rely on speech recognition and various domain switching algorithms to drive the dialogue process [15]. It is easy for a system to misidentify the intended task domain and follow-up with inappropriate questions [103]. Existing approaches do not leverage the rich knowledge available via the dialogue's information state to handle the uncertainties in domain selection or utterance interpretations. Most systems employ frame-based dialogue subsystems, which are not well suited to modelling an environment where information is coming into the system and causes effects on multiple domain areas at the same time.

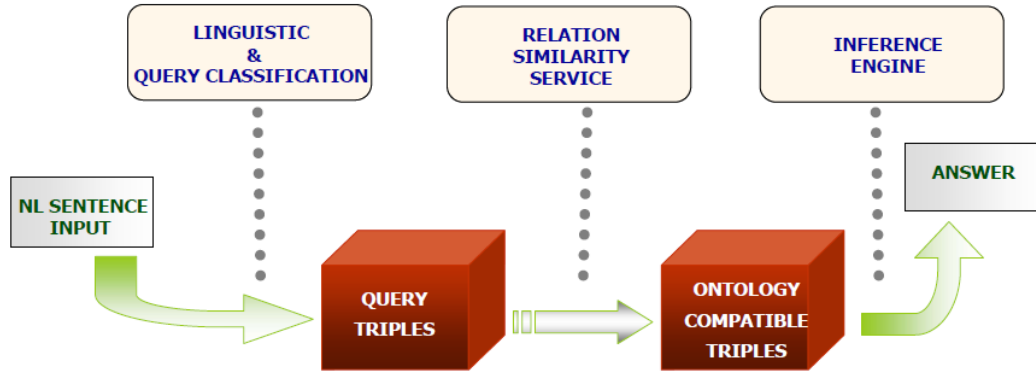


Figure 10: PowerAqua workflow [104].

4.2 Multi-Domain Search Systems

A multi-domain system has the capacity to accommodate many different areas of knowledge as it seeks to understand the user's intention and communicate effectively. As described above, several dialogue systems achieve multi-domain support using a distributed architecture of domain experts. In the Web context, there are two frameworks in particular that dynamically select and reason with domain experts.

4.2.1 PowerAqua

PowerAqua offers a natural language query interface to publicly accessible, heterogeneous knowledge sources published on the Web [104]. The user's input is converted into a series of statements that are matched against the knowledge sources using similarity measures and heuristics. The selected knowledge bases are queried and their partial results are merged and ranked. As shown in Figure 10, a PowerAqua interaction involves the linguistic analysis and statement identification of the input and subsequent mapping of the input statements to facts in the knowledge bases.

PowerAqua generates global results by merging and ranking partial results obtained from dynamically selected knowledge sources. Unfortunately, the system has severe limitations. It cannot answer questions that contain negations, comparatives, or superlatives (e.g. “the most”, “the best”) [105]. The system is very slow even for queries constructed by users that are aware of the available knowledge (about 15.39 seconds on average per query) [106]. The poor performance is in part due to a lack of query context that results in imprecise matching between the query and the knowledge sources. The knowledge bases are often sparse and heterogeneous in terms of their granularity (level of detail) and quality which produces poor retrieval of concrete answers.

4.2.2 Search Computing

The Search Computing framework uses registered knowledge sources with well-defined semantics and linkages. These knowledge sources are abstractions over one or more concrete data sources that store information about specific entities [107]. The results from each knowledge source are composed to generate the global results as specified by predefined connection patterns. These connection patterns are merely handcrafted queries that expose attributes of the knowledge sources. The system provides a “liquid query interface” with which the user can select connection patterns to incrementally build the query [108]. The framework includes advanced optimization and execution techniques for load balancing the subqueries over data sources, and for producing and consuming chunks of results at a time for efficiency [109].

Current state-of-the-art multi-domain search systems have adopted the Search Computing framework. CrowdSearcher enables the querying of domain-specific service marts and the combination of their results with the consideration of opinions derived from social media [110]. For example, the user may search for job offers weighing home rental results with the added advice of selected friends in their social network. The biomedical-molecular search system can be used to explore various biomedical domains, ultimately creating globally-ranked results from these multi-domain interactions [111].

The framework is limited mainly due to its reliance on predefined connection patterns and registered, curated knowledge sources. These knowledge sources must be developed, published, and managed by experts who know how to construct appropriate connection patterns. The types of satisfiable queries are determined by the quality and availability of knowledge sources and corresponding connection patterns. Although the system provides interactivity in the form of data warehouse-like drill-down and roll-up operations, the user must be completely familiar with the domains of interest in order to hone in on the results that meet his/her needs.

4.3 Summary

Multi-domain dialogue systems are typically hierarchically-constructed: They are composed of domain experts and mechanisms for choosing them dynamically. Unfortunately, selecting domain experts and choosing most likely contexts necessarily forces a loss of interpretability. If the incorrect domain expert is consulted, important interpretations of user input may be lost and the conversation may proceed in an unnatural manner. Existing systems do not leverage principled statistical techniques, such as POMDPs, to handle uncertainties in the dialogue.

Web-accessible data-driven systems that attempt to support multi-domain queries are not dialogue-based and are inherently limited by the presence of well-maintained knowledge sources with defined interrelations. These systems illustrate some of the challenges of data integration on the Web that make it difficult to support robust multi-domain functionality.

Chapter 5: Knowledge Representation and Reasoning

Spoken dialogue systems represent and reason about dialogue phenomena and goals by encoding applicable knowledge in its General Knowledge module (see Figure 3). For example, a finite state-based dialogue system implicitly encodes the relationships between goals and user actions as a connected network of transitions. Richer techniques, such as those that use information states, explicitly model various aspects of the dialogue, often accommodating the expression of uncertainties.

Throughout the course of the dialogue, the system must keep track of user-provided statements that indicate preferences, needs, or desires, and it must monitor and detect incompatibilities. The statements associated with users' speech acts are instantiated, maintained, and used to draw inferences to make sense of any underlying plans or goals.

Various knowledge representation technologies can be used to describe these statements and their related contexts. These knowledge representations differ in terms of their modelling viewpoints, expressivity, and the performance of their inferencing procedures [112]. The modelling stance imposed by the representation language determines the point of view and methodology for describing the concepts of interest. Expressivity refers to the richness of the descriptions. A highly expressive language is very descriptive and allows the knowledge designer to give a great deal of detail, including cardinality restrictions, disjointness, and individual correspondences or equivalencies. However, the performance of the inferencing procedures generally decrease as the expressivity increases [112].

Name	Strengths	Weaknesses
Frame System	Natural, cognitive-based theory of representation. Simple, object-oriented approach where objects (frame instances) inherit from parent frames. Offers efficient means for decidable reasoning [112].	Lacks expressiveness: Property constraints permit modelling cardinality restrictions on slot values and inverse and disjoint relations, but only subsumption (subclass-superclass) relationships are allowed between classes.
Description Logic	Polynomial-time subsumption testing (in practice) with exponential worst-case time complexity [108]. Fairly expressive.	Combining multiple description logics usually requires the alignment of their terminologies.
Production Rule System	Easy to understand. Clearly observable side effects. Often used to increase the expressive power of description logics.	Not appropriate for modelling non-procedural knowledge, such as objects and their properties.

Table 3: Knowledge representation techniques.

5.1 Handling Certain Knowledge

Many knowledge representations allow the formal, explicit description of a domain in terms of certain rules, identities, roles, and relations that are either true, false, or unknown [113]. The most commonly studied representation and logic framework, first-order logic, is built around objects and defining relations between them [112]. The knowledge designer gives a set of axioms that make assertions about a domain. The axioms and logical consequences derived from them comprise a theory that is interpreted by assigning constants, predicates, and functions to the terminology [114]. First-order logic is a very powerful framework but it has undecidable reasoning, meaning that there is no way to derive truths for every possible question [113].

Several knowledge representation and reasoning formalisms address this computability concern by supporting less expressive constructs and by suggesting specific modelling methodologies. The frame system approach represents stereotypical situations, like being in a certain kind of environment, by capturing relevant properties and attaching default

values and/or procedures to them [115]. The structure and organization of the frames are inspired by human cognitive activities for knowledge management [116]. Description logics combine the object-centred approach of frames with the logic-based constructs of first-order logic. Knowledge is encoded using concepts, roles (properties and relations), and individuals using a generative approach: like first-order logic, statements are built on top of other statements. Production rule systems implement a subset of first-order logic to represent procedural knowledge using if-then Horn clause rules. Satisfying the if-conditions results in the addition, removal, or modification of statements in a way that is governed by rule selection strategies [112]. Table 1 outlines the strengths and weaknesses of these approaches.

Recent efforts have focused on creating and combining modular knowledge encoded in ontologies. An ontology is a formal, explicit representation of a system of concepts and their relations from a particular point of view [117]. The growth of the Semantic Web has led to the publication of an abundance of ontologies that are expressed using description logic-based languages, such as OWL. The ϵ -connections framework allows the combination of separate decidable logics (modules) through link properties [118]. Within this framework, an ϵ -connection is a set of connected modules that capture a specific subset of knowledge. The Distributed Description Logic (DDL) approach uses directed semantic mappings (bridge rules) to connect concepts and individuals across modules [119]. The Package-based Description Logic (P-DLs) method is quite different in terms of its semantics. A P-DL encapsulates individuals, concepts, and roles from different modules (packages) by importing those terms defined in foreign modules [120]. Yet another approach, Integrated Distributed Description Logics (IDDL), formalizes mappings as semantic relations between items of different modules stated from a global, external perspective [121]. Modules are connected via bidirectional semantic mappings, or ontology alignments, that assert relations between concepts, roles, or individuals.

There are clear benefits to the modularization of ontologies, including ease of maintenance, faster inferencing (over a subset of the ontologies), and easier debugging

Name	Strengths	Weaknesses
ϵ -Connections	Natural way to infer knowledge in one module based on knowledge in another.	Can only be used to combine ontologies that contain disjoint terminologies. A concept cannot be declared as a subclass of a concept described in another ontology [117]. An instance in one ontology cannot be an instance of another ontology [118].
Distributed Description Logics	Can support the propagation of the role hierarchy between modules and mappings across a chain of ontologies [119]. Modules do not need disjoint terminologies. Since the bridge rules are independent from the modules, different mappings can be used to connect the same modules to generate different views [120].	Lacks expressivity: New constructs cannot be created across modules [117].
Package-based Description Logics	Provides a structured, organized package hierarchy. Scope modifiers can be used to control importing operations [118].	Does not allow role inclusions nor using foreign roles to construct local concepts [117]. Currently no known implementation [118].
Integrated Distributed Description Logics	Very good for reasoning about the mappings.	Cannot be used to combine ontologies in a hierarchical way [139]. Does not provide importing constructs.

Table 4: Modular ontology languages.

[118, 122]. The strengths and weaknesses of the aforementioned modelling techniques are covered in Table 4.

5.1.1 Plan and Goal Recognition

Dialogue systems can use these types of knowledge representation technologies to assess users' intentions. Many intention recognizers follow a plan-based model of dialogue and attempt to use logical methods to exclude goals and plans based on learned

information. These logic-based approaches typically employ propositional chaining. Given a single observed action, the system in [60] uses heuristics with backward chaining (working from the goals to the actions) to figure out which pieces of information to provide to the user as system responses. [123] extends this work to cover multi-utterance dialogues in a two-phase manner. After identifying any immediate goals implied by the utterance, a goal is selected based on global analysis that fits one of the immediate goals into the context of previous utterances. A powerful approach proposed by Kautz represents the set of possible plans in an event hierarchy, representing goals and actions as complex schemas with parameter values [124]. This approach minimizes the number of top-level plans to reduce the plan recognition problem to that of nonmonotonic deduction (statements can be added or removed) [125]. In other words, the process is simplified to identifying the plans that are consistent with the observed actions.

Another approach is to assess goals by examining goal graphs. The approach in [126] constructs a goal graph to represent observed actions, state information, and achieved goals as well as connections between them at consecutive steps. This graph is analyzed at each step to recognize goals that are consistent with the actions that have been achieved so far. Other systems try to reduce the set of possible goals by pruning away those that are inconsistent with observed actions, under the assumption that a user constructs plans without any irrelevant actions [127].

5.2 Handling Uncertain Knowledge

Typical knowledge representations do not account for ambiguities or uncertainties. This is problematic because, as previously noted, the dialogue environment is filled with uncertainties in goals and observations. The aforementioned logic- and graph-based techniques are unable to handle multiple consistent hypotheses [128]. They assume expert-level user behaviour: They cannot understand a non-expert user's requirements, in particular someone who has cognitive impairments and who may execute actions erroneously or in confusion [129]. Dialogue systems attempt to resolve ambiguities either by asking the user for clarifications or by specifically representing the

uncertainties to allow the system to reason with the likelihood of any given interpretation [130]. The former technique is undesirable as the number of clarification questions can easily make the dialogue unnatural and cumbersome.

Thus, dialogue systems often explicitly model probabilities and uncertainties using appropriate knowledge representation languages. One approach is to augment first-order logic with constructs that allow the expression of proportions or degrees of belief for statements about objects [114]. Bacchus's logic enables the specification that a given proportion of objects in a domain possesses a certain property [131]. Halpern's logic can express both proportion expressions and degrees of belief in those proportions [132]. However, neither of these logical systems provide a mechanism to express theories in a modular, composable manner [114].

5.2.1 Probabilistic Representation using Bayesian Networks

Graphical probability models emerged to represent probabilistic knowledge in a logically coherent way, providing efficient algorithms for inference, search, optimization, and learning [114]. A graphical probability model encodes dependencies between hypotheses as a graph and “local” probability information for each hypothesis as probability distributions.

A Bayesian network (BN), or belief network, is a commonly used directed acyclic graphical probability model. A node in the belief network represents a random variable and an arc between nodes conveys conditional dependence—that the probability distribution of the target variable depends on the value of the source variable. Together, the graph structure and the probability distributions define a joint distribution that allows the computation of the probability of any set of hypotheses given any set of observations [133].

Dynamic Bayesian Networks (DBNs) convey dynamic systems, where interactions occur over a sequence of time slices. Each time slice in a DBN is a BN that

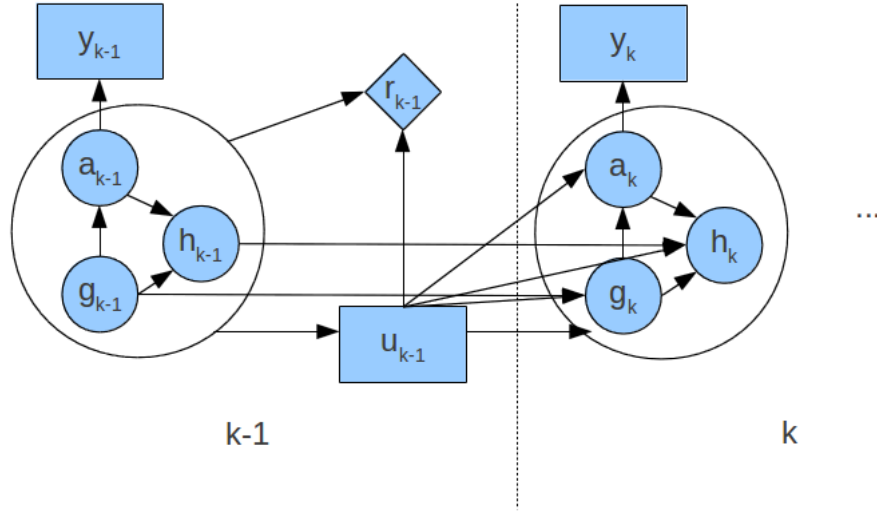


Figure 11: A two-slice DBN representation of a factored POMDP. Note that the arcs express conditional dependencies.

is connected to the next time slice according to a transition model. A DBN represents a restricted “window” of the random variables by using a compact belief state to summarize the past observations [134]. Note that a POMDP can be concisely expressed as a DBN using two slices (since, following the Markov assumption, only the last slice and the current slice matter), as shown in Figure 11.

Extensions to the Bayesian network formalism enable modular construction with repeated substructures [114]. The Probabilistic Relational Modelling approach represents objects with attributes that are affected by the attributes of other objects [135]. Given a partial specification of the state of the world, a probabilistic relational model gives a probability distribution over the possible joint assignments of values to the random variables (attributes of objects). The framework also supports modelling the probability that certain relationships hold between objects (existence and reference uncertainties). Similarly, the Object-Oriented Bayesian Network method models complex domains using a collection of inter-related objects [136]. Each object has stochastic functions associated with its attributes. These functions define probability distributions over the values of the attributes. An object or a class is thus represented by a Bayesian Network composed of connected attributes with associated probability distributions. In the Multi-Entity

Bayesian Network approach [114], areas of knowledge are represented as probability distributions over related hypotheses. These hypotheses are combined and specified in a MEBN Theory, which represents a joint probability distribution for the area of interest.

5.2.2 Other Approaches

Dempster-Shafer Theory (DST) and Certainty Factors (CF) provide alternative ways of managing uncertainty. DST is a generalization of the Bayesian theory of subjective probability that uses interval-valued degrees of belief to represent the probability that evidence supports a proposition [113]. In DST, the degree of belief for one question can be obtained from subjective probabilities for a related question [137]. Dempster's rule enables the combination of degrees of belief based on independent items of evidence. Certain Factors (CFs) were used in the MYCIN medical diagnosis and treatment expert system [138]. A CF is the expected change in belief in a hypothesis given some evidence. For example, a CF between -1 and 0 indicates a decrease in belief, whereas a CF between 0 and 1 indicates an increase in belief. In the CF model, uncertainties in if-then rules are expressed using CFs. The rules along with their attached CFs are chained together to compute the change in belief in any hypothesis in the network. The CF model often leads to errors in reasoning due to changes in belief due to ignorance of context. Furthermore, rules in the CF model were shown to be unnatural to design, leading to errors [133].

5.2.3 Plan and Goal Recognition

Approaches that deal with uncertainty in plans and goals are mainly based on Bayesian networks and Markov models [128]. As depicted in Figure 12, Charniak and Goldman's system dynamically constructs Bayesian networks by introducing new nodes for hypotheses accounting for the new evidence (previous utterances, plan roles of items in the current utterance, etc.) [125]. The new network yields a joint probability distribution for the plan hypotheses represented by the root nodes. The approach in [139]

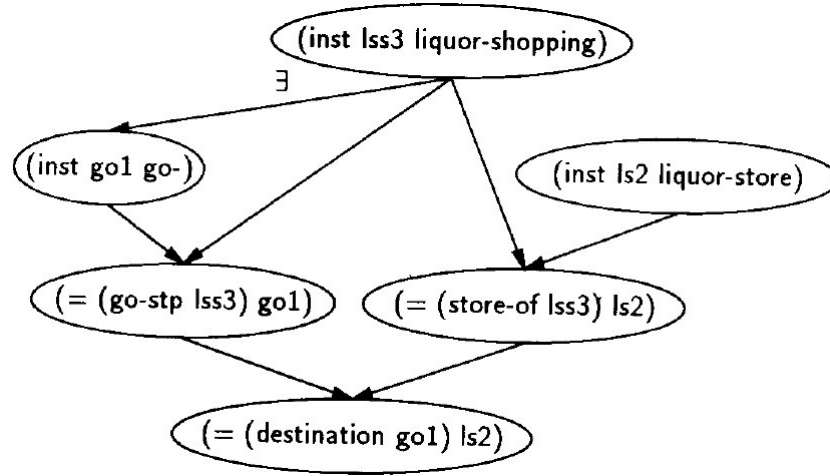


Figure 12: Bayesian network representing the possibility that going to the liquor store (ls2) is part of a liquor store shopping event (lss3) [123].

uses a Dynamic Bayesian Network to predict goals using probabilities derived from user logs.

Some approaches decompose the goal or attempt to recognize it at various levels of detail. A 3-layer DBN can be used to recognize user goals at various levels of abstraction [76]. The top-level goal is abstract, whereas lower-level goals are more concrete. Progressively more detailed levels are passed all of the linguistic and non-linguistic evidence observed by the system. Another approach, Probabilistic State-Dependent Grammars (PSDGs), represents the probabilities of having specific plans using production rules [140]. Subgoals are modelled as non-terminals in a grammar. The recognition procedure keeps track of the current plans and state variables as a DBN, choosing the most likely string of plans as the current goal structure.

In [141], Bauer employed Dempster-Shafer Theory to represent and combine the probability of goals given observed actions. The system explicitly accounted for ignorances due to incomplete information about the situation and/or a lack of knowledge about the agent's typical behaviours. The basic procedure involves the reallocation of probability mass to goals that become more likely as indicated by the a priori goal probabilities and the probabilities of goals given observations.

5.3 Overview

Knowledge representations are essential for creating knowledge-intensive systems, like dialogue systems. Since dialogue systems typically adopt the speech acts theory of dialogue, viewing the users' utterances as important indicators of intentional behaviours, they require formalisms to represent and reason with domain and user knowledge to identify goals. In particular, research efforts have focused on assessing users' underlying plans to effectively establish an appropriate context for conversation. Probabilistic approaches enable the consideration of multiple possible hypotheses and uncertainties about dialogue phenomena. Among these approaches, graphical probabilistic models have emerged as the dominant, most popular technique for encoding uncertain knowledge. Bayesian Networks are easy to use and are based on probabilistic theory that supports the intuitive construction and interpretation of conditional dependencies. Dynamic Bayesian Networks are especially popular as they enable temporal modelling. This type of approach is effectively implemented in probabilistic information state-based dialogue management in the form of POMDP-based dialogue managers, discussed in Chapter 3. Recent works attempt to create and process modular knowledge representations both with and without the consideration of uncertainty. However, it is often difficult to model knowledge in a modular way as modularization approaches inherently enforce certain modelling rules [142]. Several goal recognition systems induce layers of abstraction, or subgoals, to provide more fine-grained inferencing as well as smaller and faster inferencing procedures. This goal decomposition approach can allow the system to guide the conversation along a natural path of convergence toward shared understanding, resulting in fewer misunderstandings and uncertainties [76].

Chapter 6: The Proposed Approach

This chapter presents a method for the design and construction of a modular intention-driven, multi-domain, dialogue-based Web search engine. The motivations and methodology are overviewed followed by a description of the design components and the algorithmic procedures.

6.1 Motivation

Many search interactions are part of a complex process of expressing goals and achieving tasks that involve numerous domains [6]. One particular example is e-commerce in which users weigh criteria related to multiple topics to choose products to purchase. The problem is that existing Web search engines provide inadequate support for complex, multi-domain queries. Users are often forced to submit many ambiguous keyword-based queries and sift through numerous results pages, finding results that are irrelevant to their goals [143].

Dialogue-based systems provide an opportunity to assess the context of a user's search and to render a natural, interactive, helpful conversational experience. However, existing systems are typically geared toward single-domain support [11]. Systems that accommodate multiple domains do not take advantage of principled, probabilistic methods for handling the dialogue's uncertainties in goals and speech recognition. Web-based multi-domain systems that integrate information from various semantic-rich sources provide limited interactivity and are restricted by their lack of factual knowledge.

Knowledge representations that explicitly encode and manipulate uncertainties are needed to deliver robust, goal-driven dialogue management. A common approach, especially for information state-based dialogue management, is to model the dialogue environment as a Bayesian network. However, Bayesian networks become computationally intractable to reason with as they grow in size [20].

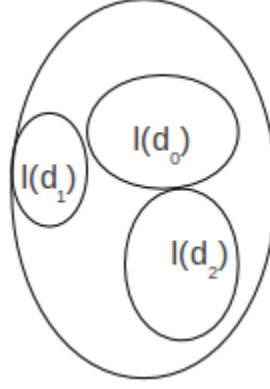


Figure 13: The information states for different domains are separable.

6.2 Method

This thesis presents a novel dialogue-based method for the incremental evaluation and integration of users' multi-domain goals to deliver a natural and helpful Web search experience. The approach accommodates users' potential lack of topic area knowledge by providing dialogue-based assistance. Multi-domain support is produced by partitioning the global information space into separate domain-specific regions that are modelled and managed using well-known tractable procedures. Overall, the method's dialogue process is driven by the need to assess and determine users' goals. These goals are combined to form a high-quality query that fetches precise, highly relevant multi-domain Web search results.

The partitioning of the information space enables the scalable construction of multiple probabilistic dialogue managers to handle uncertainties in the recognition of user goals and actions. Intuitively, each domain or topic is associated with a subset of knowledge and possible interaction histories defined by a segment of the information space, as shown in Figure 13. By decomposing the information space into domain-specific regions, the global dialogue process is defined in terms of smaller, easier to generate domain-specific action policies. These domain-specific regions are connected using probabilistic transitions via higher-level action policies. Intuitively, this connotes

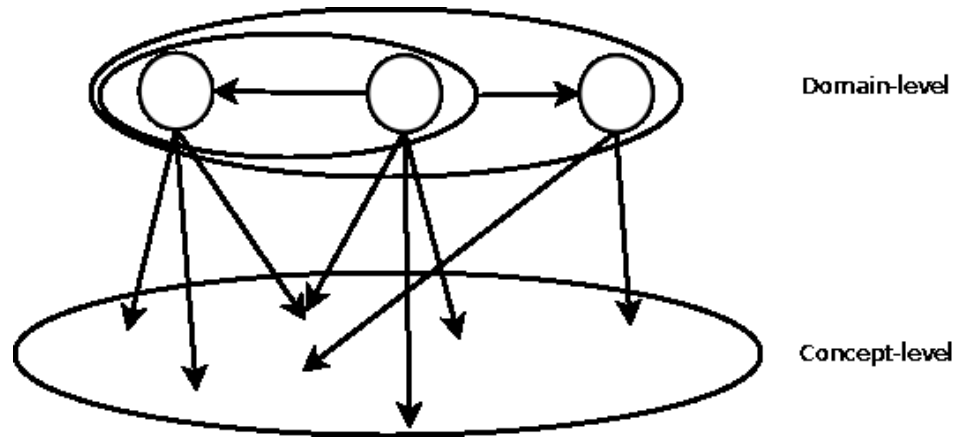


Figure 14: Domain-specific dialogue managers link together to interpret concepts from different points of view.

two-levels of abstraction: concept-level, grouping concepts into domains, and domain-level, grouping domains into meta-domains. This dichotomy is depicted in Figure 14. For example, the GuitarPurchasing meta-domain controls the GuitarAcoustic and GuitarElectric domains (among others), where GuitarAcoustic interprets concepts such as the type of top wood and the number of frets, and GuitarElectric interprets concepts like the types of pickups and vibrato mechanisms. Thus, the global dialogue is modelled using a collection of domain-specific dialogue managers that are linked together by higher-level dialogue managers.

Within a dialogue, meta- and domain-specific dialogue managers chain together to render one integrated dialogue and overall goal. For example, the high-level goal of purchasing a guitar can be recognized as a problem in the GuitarPurchasing meta-domain. The GuitarPurchasing meta-domain may include several subdomains that cover information pertaining to the user's budget, qualities of the guitar (colour, condition, dimensions, etc.), the user's musical preferences, and so on. This allows conversational support to span over numerous domains to ease the knowledge burden on users. Users do not have to be experts in the domains of interest because the system seamlessly guides the user through likely relevant topics. Since meta-level domains may overlap in their knowledge requirements, they can share subdomain processes. In this case, the modular design facilitates knowledge reuse so that common, shared properties do not need to be

redefined for each high-level domain. In other words, the same concept can be interpreted in different ways according to the context.

Since user-provided information may have consequences for multiple domain areas, each dialogue manager handles all the utterance observations that affect its beliefs. For example, the user may state that he/she is interested in an “electric device”. The system must interpret this utterance as it pertains to various domains. For example, “electric” may refer to a vibrant colour, an electric-powered device, or a particular music style. This multi-domain interpretation capability is particularly important for the processing and consideration of information provided in response to questions about one domain that cause changes in the system’s beliefs in other domains. From the user’s point of view, this enables the system to handle diverse, over-informative input. For example, a user’s wish to sound like Stevie Ray Vaughan should increase the system’s confidence in the types of instruments, manufacturers, and music styles the user is interested in.

The system of dialogue managers accesses a shared knowledge base that encodes information about the covered domains, their interrelations, and their combinations for goal formation. This knowledge base consists of concepts, roles, individuals, and constraints that make up the world that the dialogue can operate over. Conceptually, the constraints describe integrity and consistency checks to ensure that the system detects conflicts in the statements provided by the user. Specifically, the constraints ensure that the user does not provide multiple (conflicting) values for the same property, inappropriate values, or incompatible values across related attributes. Over the course of the dialogue, the active dialogue managers request statements to be added to the knowledge base in response to recognized user inputs.

A search engine query is constructed from the statements contained within the knowledge base once the dialogue is complete by interpreting the statements with respect to the identified high-level search context. This query is sent to an external system-selected search service that allows some level of structured keyword-based querying.

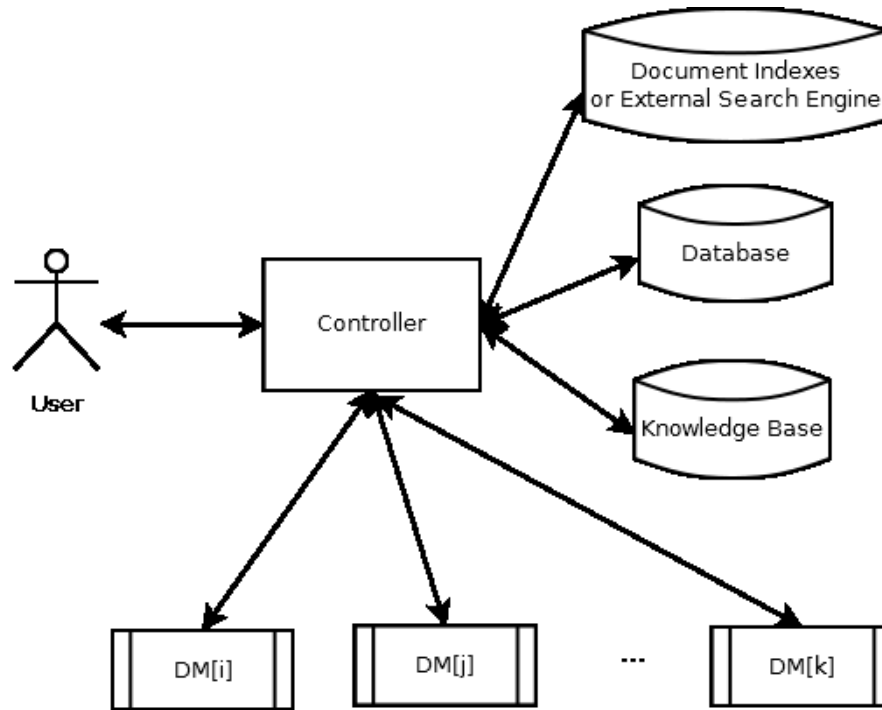


Figure 15: The many knowledge sources. Note that Controller encapsulates the system logic.

Thus, the approach attempts to achieve a reasonably wide coverage of relevant online knowledge and documents without having to assemble and maintain rich, entity-centric knowledge bases or indexes.

6.3 Design

The design follows the general spoken dialogue system architecture covered in Figure 3. For simplicity, the knowledge modules are separated into two groups: probabilistic knowledge, which encapsulates the dialogue history model, task record, conversational competence model, and user models; and other knowledge, which covers both domain and world knowledge as well as information used for generating human readable responses. Probabilistic dialogue knowledge is encoded using POMDPs while other knowledge is contained within relational databases or description logic-based ontologies. The knowledge layout is depicted in Figure 15.

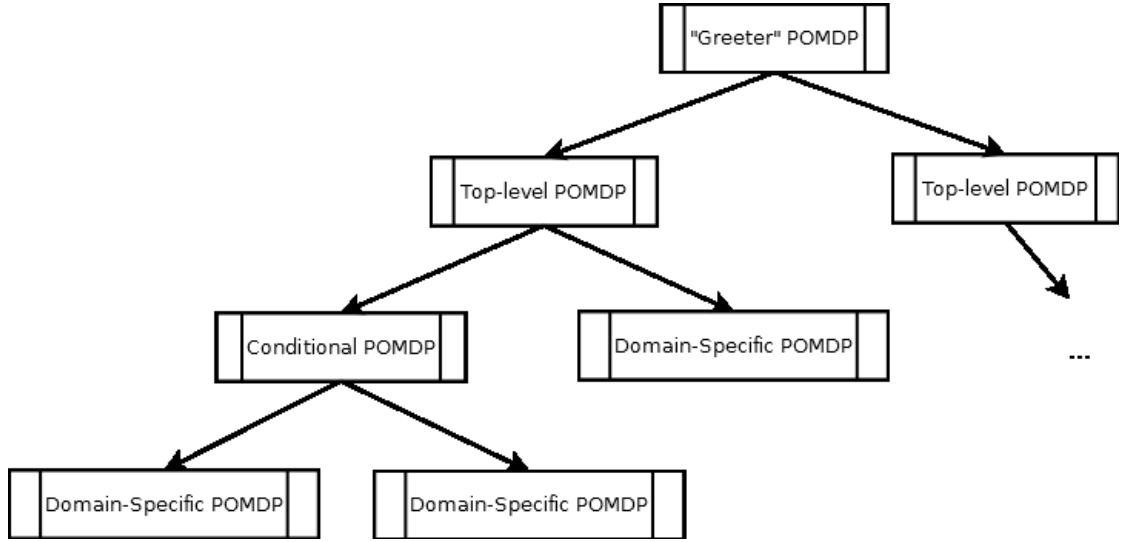


Figure 16: Hierarchical POMDP organization.

6.3.1 Probabilistic Knowledge

The dialogue management task of supporting different scenarios across multiple domains is achieved by the effective segmentation of the global dialogue management task into hierarchical, related subtasks. Robust, principled, probabilistic POMDPs represent these subtasks, or subdomains, at multiple levels of abstraction. The design involves three types of POMDPs: top-level, which provide the highest level of context abstraction; conditional, which select (and reject) some domain areas to drive the conversation; and domain-specific, which maintain information about specific sets of knowledge.

The POMDPs are organized in a hierarchical network, as illustrated in Figure 16. The top-level POMDPs are situated at the top of the hierarchy. Conditional POMDPs and domain-specific POMDPs form the rest of the network's structure, with the conditional POMDPs acting as hubs for descending the hierarchy. The "Greeter" POMDP is a conditional POMDP that maintains beliefs about the user's top-level search context. Its job is to figure out which high-level search context the user has in mind so subsequent multi-domain interactions can proceed. For example, the high-level context

could be shopping for a musical instrument. Domain-specific POMDPs are “terminal” network nodes.

The top-level POMDPs do not represent any specific goals, which allows them to have simplified search spaces. They maintain an approximation of the level of specification of the POMDPs they control. A top-level controller is formally defined by:

1. The user goals, S_u , which only contains the “null” goal.
2. The user actions, A_u , the names of the POMDPs whose defined subtasks were just completed. A user action conveys which previously under-specified POMDPs have been fully defined either explicitly by the user or implicitly through inferences. The user action is “null” if no subtask was just completed.
3. The dialogue states, S_d , which are the possible levels of specification of the controlled POMDPs. For example, $s_d = fff$ means that all three of the controlled POMDPs are deemed to be fully specified.
4. The system actions, A_m , are the names of the controlled POMDPs. A control-level POMDP selects a dominant POMDP that will contribute the next system action according to the context.
5. The transition function, reward function, and discount rate.

Conditional POMDPs reason about the system’s knowledge to direct the conversation over certain domains in response to user-provided information. For example, HIMeta only directs the conversation to the SumpPump domain if it determines that the user is interested in sump pumps, and consequently ignores the Cupola domain entirely. Such a POMDP is described as follows:

1. The user goals, S_u , are the names of the POMDPs that the conditional control-level POMDP makes decisions about. For example, SumpPump and Cupola are HIMeta’s user goals. HIMeta decides which POMDP to allow into the conversation.

2. The user actions, A_u , are the names of the POMDPs whose defined subtasks influence the conditional POMDP's assessment of the goal. Certain domain-specific utterances may indicate the goal. For example, if a user says he/she wants a sump pump, the system is obviously more apt to want the SumpPump domain in the future. The user's action is "null" if no subtask was just completed.
3. The dialogue states, S_d , refer to the possible levels of specification of the controlled POMDPs. For example, $s_d = fff$ means that all three of the controlled POMDPs are deemed to be fully specified.
4. The system actions, A_m , are the names of the controlled POMDPs. The POMDP selects a dominant POMDP that will contribute the next system action according to the context.
5. The transition function, reward function, and discount rate.

Finally, the domain-specific POMDPs enable the interpretations of user utterances in well-defined low-level contexts. Conceptually, they provide slots or properties that the user fills in throughout the dialogue. A domain-specific POMDP consists of:

1. User goals that define all possible combinations of input slot values.
2. User actions and associated observations covering all recognized user utterances.
3. Dialogue states that indicate the level of specification of knowledge for the domain area covered by the POMDP.
4. System actions: The responses the system emits. These are typically questions, confirmations, or suggestions. For example, the system may ask about the user's favourite musical artist or suggest an artist based on the history of the dialogue.
5. The transition function, reward function, and discount rate.

The global dialogue management task is thus separated into multiple subtasks that are implemented by separate POMDP-based dialogue managers. These dialogue managers operate independently, maintaining their own states and beliefs. However, they are aware of the positions of their knowledge areas (POMDPs) in the context of the global hierarchy. In addition to its beliefs, each dialogue manager keeps track of which

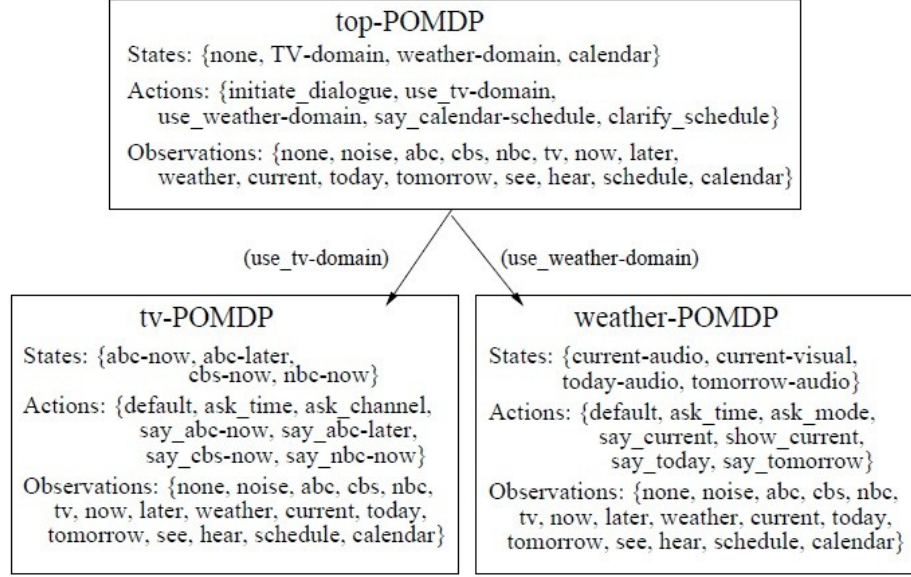


Figure 17: The top-POMDP delegates to its child POMDPs [189].

subtasks are hierarchically related to it. Hierarchically-related dialogue managers communicate using peer-to-peer messaging, allowing update requests to propagate through the network of dialogue managers. This peer-to-peer style allows the dialogue managers to be extended with additional knowledge without affecting other unrelated dialogue managers.

6.3.1.1 Justification

The design's hierarchical partitioning is easily extendable to support new domains and leverages the factored POMDP approach to generate expressive representation. By contrast, hierarchical approaches to POMDP decomposition, such as the HPOMDP and H-POMDP methods, are limited in their scalability.

In the HPOMDP approach, the problem environment is divided into independent subtasks by partitioning the action set [144, 145]. These subtasks are glued together by a POMDP whose actions are abstract, indicating the need to query an underlying subtask's policy. Most notably, this decomposition of the POMDP environment is not suitable for

the multi-domain searching case because each POMDP has the same (global) observation set, which makes the approach inefficient and non-scalable as the number of subtasks increases. Furthermore, the POMDPs themselves do not take advantage of the computational and representational benefits of factorization. For example, in Figure 17 the top-POMDP delegates to either the tv-POMDP or the weather-POMDP to handle the user request (and the ensuing system response).

The H-POMDP approach constructs one POMDP consisting of both vertical and horizontal transition probabilities [146, 147]. Figure 18 depicts a hierarchical POMDP with two primitive actions, $a1$ and $a2$, which cause transitions to various states $s1, \dots, s7$, and transitions to different vertical levels or subtasks via emission states, $e1, \dots, e3$. This representation allows subtasks to be modelled at different levels in a hierarchy. By imposing restrictions on the structure of the POMDP (e.g. the state transitions), a simple, unified representation is created. However, this approach is not modular—to add support for new domains and/or actions, the entire POMDP has to be altered and re-evaluated.

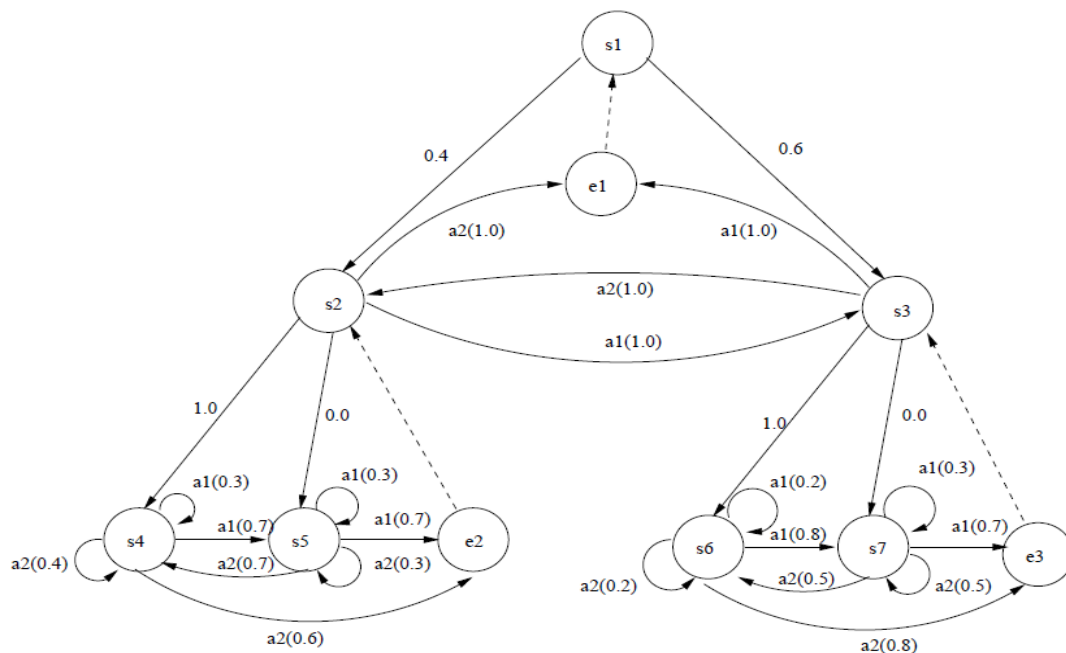


Figure 18: An H-POMDP's transitions [142].

Type	Description
Core Domain Knowledge	Encodes terminology and rules that describe the domains. In particular, this knowledge includes relations between concepts in different domains. The dialogue managers are provided with a shared understanding of their underlying domains through this unified domain model. Constraint and integrity checking are performed against this model throughout the dialogue.
Goal Knowledge	Acts as an interface to the domain knowledge and formalizes users' multi-domain goals as queries against the domain knowledge.
User Input Knowledge	Stores patterns that determine which parts of the captured inputs are potentially relevant to specific dialogue managers. Basically, these patterns serve as a rudimentary mechanism to forward captured inputs to dialogue managers that will perform action recognition. This module also contains knowledge to provide human readable output from the system's action token strings.
System Output Knowledge	Contains mappings from system action token strings to human readable output including relevant help text.

Table 5: Non-probabilistic knowledge.

6.3.2 Other Knowledge

The core domain knowledge and goal knowledge components are represented using description logic. This allows them to be defined separately (if needed) and integrated despite their potential distribution over multiple physical locations, or files. The description logic-based approach provides computational guarantees and tractable reasoning, which are essential for the real-time usage of this knowledge in a dialogue system [113]. The user input and system output knowledge are stored in a relational database for efficient access. The other knowledge sources are summarized in Table 5.

6.3.3 Input, Fusion, Output, and Fission

Although it was created to support spoken dialogue conversations, the current design only accommodates text-based input and output. This simplification enables a focused investigation of the methodology of intention-driven searching without complications pertaining to the usability and performance of modern-day speech recognition and generation components.

6.4 Algorithms

A search session with the system consists of a dialogue phase, in which the user and the system participate in a collaborative, turn-based conversation to identify the needs of the user, and a query phase, in which the system generates a query and retrieves product results. In a typical dialogue turn, the input recognition component parses the input and sends parts of it to specific dialogue managers for context-specific processing. The system then updates its dialogue knowledge and checks for inconsistencies or constraint failures by leveraging domain knowledge. Lastly, the system selects a global response and presents it to the user. When the system has gained sufficient information to identify the underlying goals of the user, the system constructs a query, sends it to an external search service, and presents the retrieved results. The overall algorithm is summarized in Figure 19.

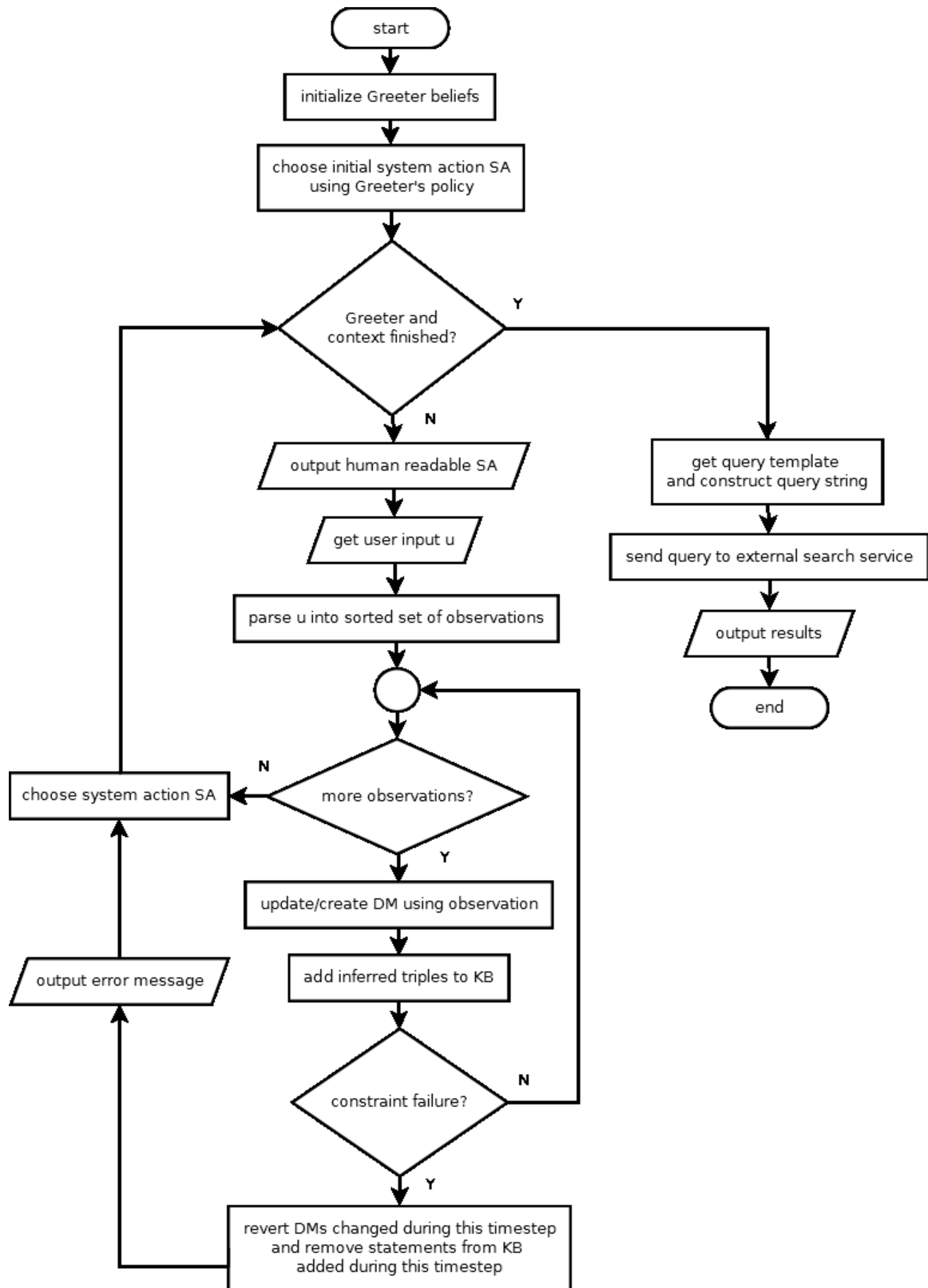


Figure 19: Algorithm flowchart.

6.4.1 Descriptions

The system maintains probabilistic beliefs throughout the course of the dialogue by updating POMDPs and by storing collected information in the knowledge base. Several key operations occur including the translation of user input into observations, the processing of the observations by POMDPs to gain intentional knowledge and to assess the user's actions, and the selection of the next global system action.

The main system logic in Figure 20 outlines the system's initialization and dialogue operations. The Greeter DM (dialogue manager) has a unique role as the uppermost conditional process that prunes away most of the information space by directing the conversation to a certain subtree, or context. First, the Greeter DM is initialized and the starting system action is chosen from the Greeter's policy (lines 1-4). The user provides some input and the system converts this input into a set of observations (line 6). The observations that affect the Greeter are processed first because the Greeter is the highest-level managing DM (lines 7-17). Each observation is dealt with independently by allowing the corresponding DM to assess the user actions, adding any implied triples to the knowledge base, and reverting all previous updates for the dialogue turn (lines 8-15). Note that if the dominant context is newly established, the Greeter sets the high-level search context (lines 16-17). Lines 18-26 process observations addressed to DMs if the context is known. This involves propagating the update throughout the dynamically-created DM hierarchy (line 20), adding the appropriate (inferred) triples into the knowledge base (line 23), and reverting if necessary (lines 25-27). After DM processing, the global system action is selected according to the status of the Greeter and the current context (lines 28-34). Finally, the knowledge base is queried to fetch information to create a query and the results are fetched (lines 35-36).

```

1.  construct and initialize Greeter DM
2.  choose global system action from Greeter's policy
3.  query database for human readable output associated with the system
    action
4.  output the system's response

5.  while dialogue is not finished:
6.    get user utterance as observations
7.    for each observation addressed to the Greeter:
8.      update Greeter DM
9.      get user action as determined by the DM
10.     if user action is not null
11.       add triples to knowledge base
12.       if constraint failure
13.         revert Greeter DM to its previous state
14.         remove statements added during this outer
            iteration
15.       exit
16.     else if Greeter recognizes context
17.       set and instantiate context
18.   if context is not null
19.     for each other observation:
20.       propagate update through context tree (instantiate
          DMs whenever needed)
21.       get user action
22.       if user action is not null
23.         add triples to knowledge base
24.         if constraint failure
25.           revert all DMs affected during this
              outer iteration to their previous
              states
26.           remove statements added during this
              outer iteration
27.       exit
28.   if Greeter is not finished
29.     choose system action prescribed by Greeter
30.   else if context is not null
31.     if context is finished
32.       finished <- true
33.   else
34.     choose system action prescribed by context
35.   query database for human readable output associated with the
    system action
36.   output the system's results

```

Figure 20: Main algorithm.

The user input is parsed into observations (Figure 20, line 6) using regular expression-based matching to associate parts of the recognized text with dialogue managers that may be applicable for processing, as shown in Figure 21. Since the input for the system is text-based, this involves some manipulation of the text first. (Note that observation creation from text takes the place of observation creation from speech input in the implemented system.) First, the current system action is associated with the pre-processed user input as a way to easily convey the current question under discussion (lines 1-2). Then, the text is matched against templates that recognize concepts and their relationships to DMs in the system (line 3-5).

The dialogue update procedure (Figure 20, lines 8, 20) is recursive and hierarchical, spanning multiple levels of knowledge abstractions for context updates. Each dialogue manager stores a copy of its state before it is updated to enable later reversion. In Figure 23, “next system action” refers to the specific dialogue manager instance’s next system action and not the globally selected system action. First, the dialogue manager stores its current configuration in case it needs to be reverted (lines 1-2). If the observation was addressed to the DM (and not just “passing through” the DM on its way down the hierarchy), the DM’s beliefs are updated (line 4), its next prescribed action is chosen (line 4), and it returns the user action it used for the update (line 7). Otherwise, the observation is addressed to another DM. The update request is forwarded to the appropriate child DM (lines 9-11). On return from the recursive calls, the tree is

```

1. concatenate the currently executed system action to the front of
   the
       text input
2. trim surrounding whitespace, convert to lowercase, remove all
   punctuation except `:', convert all whitespace to one space
   character
3. for each regular expression that identifies an observation:
4.     if it matches the text
5.         get the topic/domain and recognition component (the
           concept) associated with the match and store it as
           an observation

```

Figure 21: Observation creation.


```

1.  if the dialogue manager's state is not backed up
2.    store the current state

3.  if the observation is addressed to this DM
4.    update beliefs and get the next system action
5.    if next system action is bye
6.      finished <- true
7.    return user action
8.  else
9.    consult local forwarding table to get the name of the child
        DM to sent the observation to
10.   construct this child DM if it does not exist
11.   call update on the child      // Recursive call

12.  if child is newly finished    // Upon return from recursion
13.    get finished response from child and set it as the next
        system action
14.  else
15.    set next system action to null
16.  if next system action is bye
17.    finished <- true
18.  return the user action

```

Figure 23: Dialogue manager update.

updated to signal changes in the lower-levels and to affect the upper levels (lines 12-17). A finished child causes a change in the parent DM (line 13), otherwise no changes (line 15).

Action selection (Figure 20, lines 29, 34) is the recursive process of finding the next system action for the dialogue. Basically, the procedure in Figure 22 consists of descending the hierarchy of linked dialogue managers starting from the root of the context until the current subject domain-specific dialogue manager is found and its prescribed system action is returned (line 3).

```

1.  function chooseAction():
2.    if children == null
3.      return next system action
4.    else
5.      if next system action DM is inactive
6.        create it
7.      return (next system action DM).chooseAction()

```

Figure 22: Action selection.

1. get the most specific type of context/product
2. execute SELECT templates associated with that type to get specific information
3. piece together the information to achieve a formatted query string

Figure 24: Query construction.

The search engine query is constructed using information collected from the dialogue (Algorithm 1, line 35). Each context is associated with a predefined template of relevant property values. This template encodes a way of combining and accessing the information before it is put into an implementation-specific query format (e.g. a Google Search API for Shopping request). As shown in Figure 24, query construction involves getting the most specific type of the context (e.g. GuitarAcoustic is more specific than Guitar) and retrieving and formatting data obtained from the knowledge base (lines 2-3).

6.4.2 Time Complexity

An algorithm's time complexity is a measure of the amount of work that it performs in terms of key (computational) operations. The following analysis examines the operations in Figure 20. Line 1 takes $O(N)$ time, as it simply involves the assignment of initial, pre-computed values to the Greeter's N states. The selection of the global system action from the Greeter (line 2) involves the identification of the highest-valued alpha vector for the current belief state. This means solving each alpha vector using the current belief values, and choosing the vector with the maximum value—a procedure analogous to solving N linear equations, where N is the number of hyperplanes or alpha vectors. Each hyperplane is a linear function, thus requiring $O(|S|)$ operations to solve (state size is $|S|$), so the action selection procedure takes $O(N \cdot |S|)$ to find the maximum-valued vector and its associated action. Line 3's database query is implementation-specific, taking at worst $O(N)$ for a largest table size N used in the lookup. Identifying observations from the input utterance (line 6) requires $O(N)$ operations, matching the input text with N regular expressions. The exact belief update procedure in line 8 takes

$O(|S|^2)$ operations, where $|S|$ is the number of states. The updating procedure computes an $|S| \times |S|$ matrix, where each row i 's entries represent the probabilities of reaching the i^{th} state from any other state (given the current system action, observation, and state) [Bui]. Line 9's user action query is a constant-time lookup operation that simply parses the system's interpretation of the user action. Adding inferred triples is a simple process of creating a model with N statements and adding this model to the existing model. Similarly, removing N statements requires N operations. Thus, adding and removing statements (lines 11, 14, 23, 26) takes $O(N)$ operations, where N is the number of statements/triples. Constraint checking requires the assessment of C constraints by submitting queries against the knowledge. Each query takes polynomial time (against a description logic-based knowledge base [148]), so the complexity is $O(C \cdot N^K) = O(N^K)$ for some constant K and number of facts N . The belief state reversion in Line 13 requires two constant-time assignment operations: setting the belief vector to its previous value and setting the previous action. The propagation of the belief update through the hierarchy of DMs (line 20) involves performing the belief update to a maximum depth D , processing one DM at each depth. Thus, the complexity of the belief updating down the tree is $O(D \cdot |S|^2)$, for depth D and maximum state size $|S|$. Action selection from lines 20 to 34 proceeds by identifying the proper context (if there is one) and executing action lookups down the context-rooted subtree until a domain-specific terminal node is reached. In the absence of precomputed cached actions, this requires $O(D \cdot N \cdot |S|)$ operations, extending to a depth D where each step requires $O(N \cdot |S|)$ operations to assess the optimal action. As the description logic used to encode the knowledge base requires simple restrictions and axioms, such as enumerations and domain and range restrictions, querying the knowledge base to get information to build the query string (line 35) should take polynomial time with respect to the number of facts, e.g. $O(N^K)$ for some constant K [148].

Overall, the algorithm's complexity is determined by the most expensive operations. The size of the knowledge base, the number of constraints, and the number of regular expressions/concepts may, in practice, be the dominating characteristics (as opposed to the size of the belief states, which is limited due to the production of many

dialogue managers). Weighing all of the operations, the time complexity is given by the largest growth function which is polynomial, $O(N^K)$ for some constant K and the dominant factor's size N (e.g. number of facts).

Note that the computational cost of generating the POMDP alpha vectors is not a runtime consideration for the algorithm. The POMDPs are solved in advance using exact or approximate techniques. Solving the POMDPs using an exact algorithm, such as value iteration, is exponential with respect to the planning horizon [149]. This is referred to as the curse of history problem. For example, using standard value iteration, as the number of time steps to consider increases, the number of generated alpha vectors increases exponentially: there are $|A||\Gamma_{N-1}|^{|Z|}$ vectors produced at time step N where $|A|$ is the number of system actions, $|\Gamma_{N-1}|$ is the number of alpha vectors generated at the $N - 1^{th}$ time step, and $|Z|$ is the number of observations [149].

Chapter 7: Implementation

This thesis's Web searching method has been implemented in IDS, a search engine for online gift shopping. Two types of shopping contexts are supported: purchasing a guitar and buying a home improvement product. Although these areas are not exhaustive of every possible topic, they bring about different considerations: the home improvement case requires some diagnostic assessment, whereas guitar purchasing is more focused on the recipient's preferences. The hierarchy of dialogue managers and, thus, the organization of topics or domains is depicted in Figure 31.

The search engine is a stateful Web browser-based Java application. The user makes requests to an HTTP servlet (controller) which delegates to the appropriate application code and creates responses. Throughout the dialogue, a collection of Java-based dialogue managers handles uncertainties in utterances, goals, and domain ordering, a Jena knowledge base augmented with SPIN constraints keeps track of user-provided information, and a MySQL database provides data used for input processing and output generation. When the dialogue is complete, a query string is created specifically for processing by Google's Search API for Shopping. Product results are returned in JSON format and examined, formatted, and presented using the jQuery JavaScript library.

The system's architecture is multi-tiered in its construction and design to ensure a robust, decoupled, and flexible implementation. This allows the data, logic and data access, and presentation layers to be modified independently while maintaining the integrity of the system. Figure 25 shows the layout of the system components from the presentation layer all the way down to the data layer. Note that the figure shows one DialogueManager instance and its relationships to other components. In reality, there are many DialogueManager instances.

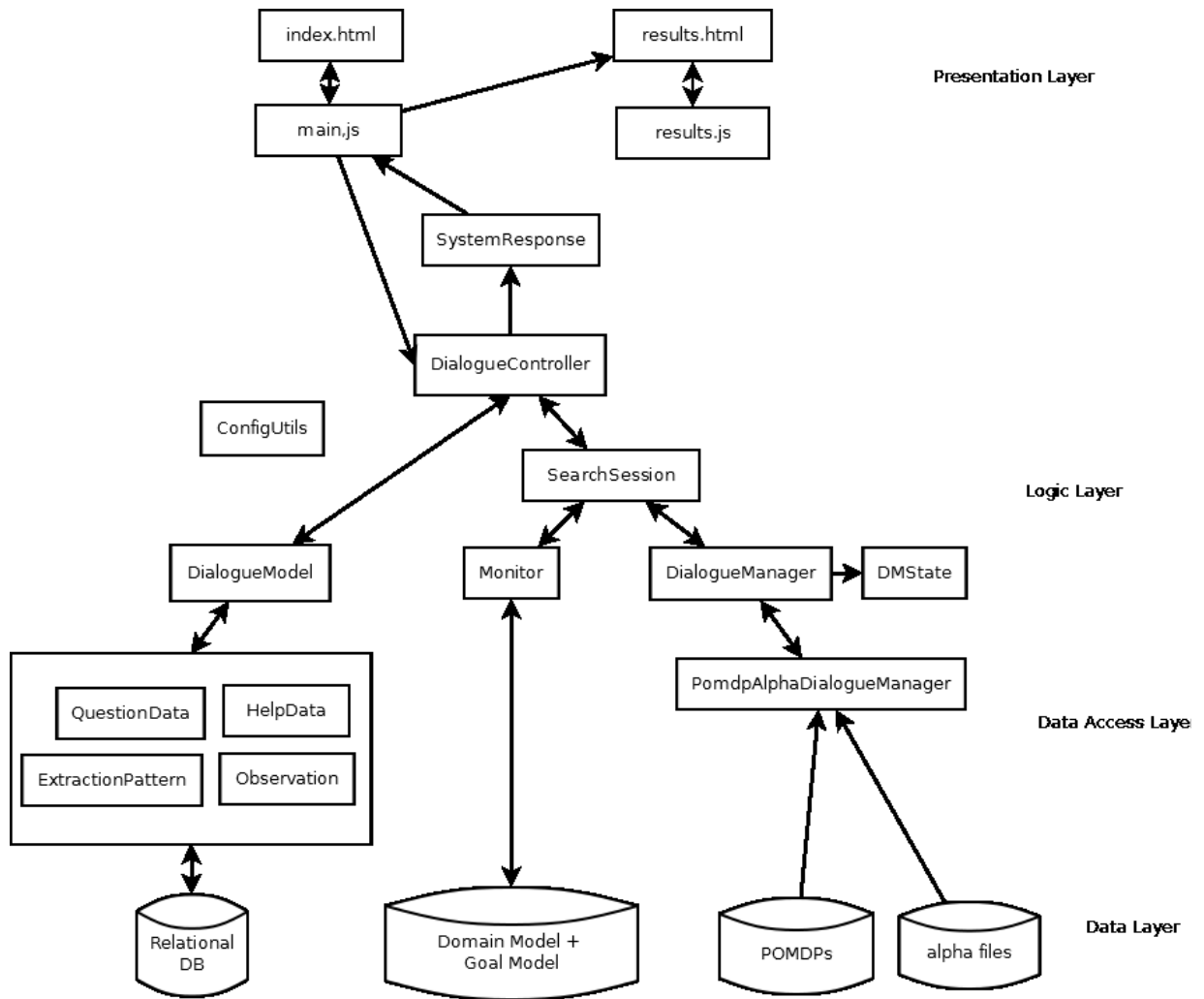


Figure 25: System layout.

7.1 Data Layer

The data layer stores and maintains the application's data. This data includes knowledge used for dialogue processing, maintenance, and response.

A MySQL relational database maintains information used to recognize utterances and generate rich help content for system responses. MySQL is fast, robust, and scalable allowing changes in the data to occur easily and seamlessly [150]. The logical design of the relational tables is shown in Figure 26. (Note that the foreign key constraints are omitted in the actual implementation merely for convenience.) The regexes table contains the regular expressions used to match input text with concepts. These concept-regular expression patterns are related to topics (stored in the topics table) via the regexes_topics join table. System question information, help text, and product classes are stored in the questions, help, and products tables, respectively. These tables are conceptually

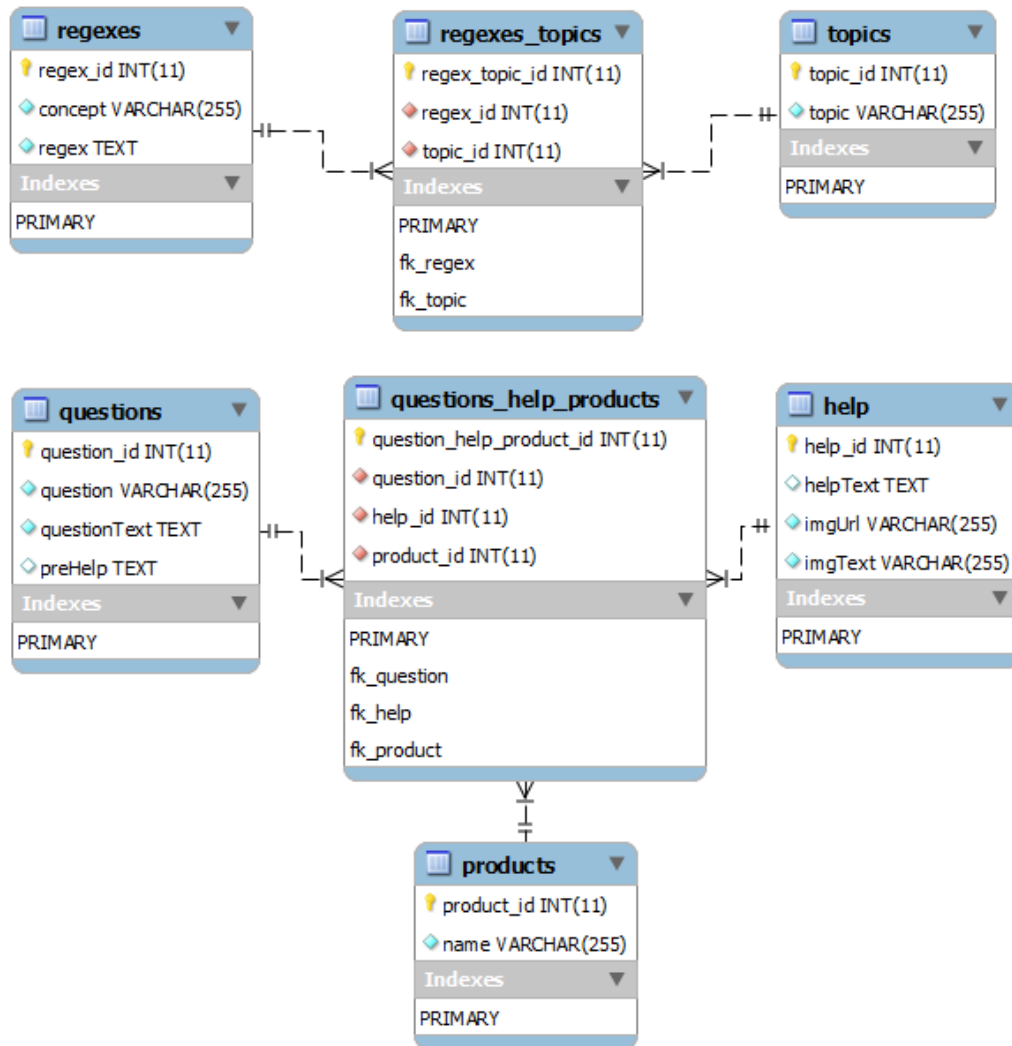


Figure 26: Database structure with foreign key constraints.

```

CONSTRUCT {
    dk:BUDGET dk:hasUpperBound 1200 .
}
WHERE {
}

```

Figure 27: Budget1200 user action.

combined via the `questions_help_products` join table to fetch human readable output. The data used to populate the tables are stored in comma-separated value (CSV) files which are accessed by an SQL script to perform database creation and loading on demand.

Domain and goal knowledge are encoded in separate OWL ontologies, or models. The Web Ontology Language (OWL) is a popular description logic-based knowledge representation language whose three dialects, OWL Lite, OWL DL, and OWL Full, offer increasing levels of expressivity (with the associated increased cost of inference procedures) [151]. It is based on the Resource Description Framework—a model for data representation that encodes knowledge as subject-predicate-value statements, or triples [152]. OWL enables rich class descriptions using disjointness relations and property restrictions (e.g. defining a class of objects according to its properties). Furthermore, properties can relate objects to data values or other objects and can have special semantics, including transitivity, symmetricity, functionality, and inversibility.

SPARQL Inferencing Notation (SPIN) is used to represent queries that add or remove statements from the knowledge base or express constraints within the ontologies [153]. These queries are expressed using SPARQL, a popular well-established RDF query language [154]. Statements can be easily added or removed by representing user actions as SPARQL queries that generate statements implied by the user actions. For example, the Budget1200 user action represented by the SPARQL CONSTRUCT query in Figure 27 creates a model that states that the budget has an upper bound of 1200. This model can be added or subtracted from the pre-existing statements in the knowledge base.


```

CONSTRUCT {
  _:b0 a spin:ConstraintViolation .
  _:b0 rdfs:label "An acoustic guitar can have 19 or 20 frets." .
  _:b0 spin:violationRoot ?this .
  _:b0 spin:violationPath dk:hasFrets .
}
WHERE {
  ?this dk:hasFrets ?frets .
  FILTER ((?frets != 19) && (?frets != 20)) .
}

```

Figure 29: Value constraint for acoustic guitar frets.

Constraint checking is performed by directly associating constraints (as SPARQL queries) with domain concepts and assessing the semantics of those queries with respect to the constraints. For example, the CONSTRUCT query in Figure 29 generates a ConstraintViolation object if the condition in the WHERE clause, that the number of frets associated with the AcousticGuitar instance is not 19 or 20, are satisfied. Several types of constraints are implemented in the system, including unsatisfactory value constraints (e.g. “the colour must be white or blue”) and conflicting value constraints (e.g. “the colour cannot be both white and blue”).

SPIN is also used to encode goal templates that encapsulate the informational

```

SELECT ?topWoodText ?cutawayTPLT ?cutaway ?colourText ?handedText
WHERE {
  dk:PRODUCT dk:hasProductTop ?top .
  ?top dk:hasMaterial ?topWood .
  ?topWood rdfs:label ?topWoodText .
  dk:isCutaway rdfs:label ?cutawayTPLT .
  dk:PRODUCT dk:isCutaway ?cutaway .
  dk:PRODUCT dk:hasColour ?colour .
  OPTIONAL {
    ?colour rdfs:label ?colourText .
  } .
  dk:PRODUCT dk:hasHandedness ?handedness .
  OPTIONAL {
    ?handedness rdfs:label ?handedText .
  } .
}

```

Figure 28: Information gathering template for acoustic guitars.

requirements of specific goals. General-purpose templates fetch information that is needed regardless of the context, such as the budget or price range, the desired product condition, and favoured brands. Other context-specific templates retrieve information that is relevant to the context. For example, the template for retrieving information about acoustic guitars in Figure 28 acquires the type of top wood, whether the guitar is a cutaway or not, the colour, and the orientation.

Probabilistic knowledge about the dialogue is encoded using factored POMDPs that are organized hierarchically at different levels of abstraction. The POMDPs are represented in regular expression-based formats (`dlgpomdp` or `fpomdp`) and are converted into standard, canonical form (`pomdp`) using the POMDP Toolkit’s dialogue specification parser [155]. The canonical POMDP specifications are solved using the ZMDP Solver to find acceptable belief state-action policies. The ZMDP Solver implements several heuristic search algorithms to solve MDPs and POMDPs, including RTDP, LRTDP, HDP, and HSVI2 [156]. The `fpomdp` notation is especially helpful for minimizing the size of the POMDP specification because it allows the declaration of variables whose values are substituted at pre-processing time. For example, Figure 30 shows a heavily redacted snippet of an `fpomdp`-based POMDP specification.

```

user-goals: {brand}{yearsActive}{madeIn}
user-actions: BrandFender, BrandGibson, ..., null
dialogue-states: nnn, nnf, ..., fff, snn, snf, sfn, sff
system-actions: askBrand, ..., bye
brand: BrandFender, BrandGibson, ..., BrandAny
yearsActive: YearsActive5, YearsActiveAny
madeIn: MadeInUS, MadeInAny

DLGPOMDP:
discount: 0.95
legal-states:
  (BrandFender|...|BrandAny).* : $1 : f..
  ...
start:
  .* : null : nnn

SU: (.* ) : .* : $1 1.0
SU: (.* ) : .* : .* 0.0

AU: f.. : .* : .* : .* 1.0/15
AU: f.. : .* : .* : .* 0.0
AU: .* : askBrand :
  (BrandFender|...|YearsActiveAny) (MadeInUS|MadeInAny) : $1 0.5
  ...
AU: .* : askBrand :
  (BrandFender) (YearsActive5|YearsActiveAny) (MadeInUS|MadeInAny) : $3
  0.1
  ...
AU: .* : bye : .* : null 1.0
AU: .* : bye : .* : .* 0.0

SD: fff : .* : .* : .* : fff 1.0
SD: fff : .* : .* : .* : .* 0.0

SD: .(..) : .* : .* :
  BrandFender|BrandGibson|BrandMartin|BrandTaylor|BrandGoodDirections|Br
  andWhitehall|BrandFlotec|BrandZoeller|BrandAny : f$1 1.0
SD: .(..) : .* : .* :
  BrandFender|BrandGibson|BrandMartin|BrandTaylor|BrandGoodDirections|Br
  andWhitehall|BrandFlotec|BrandZoeller|BrandAny : .* 0.0
  ...
transition-to-end:

R: .* : .* : f.. : askBrand -50
R: .* : .* : .* : askBrand -1
  ...
R: BrandFender.* : .* : f.. : confirmBrandFender -50
R: BrandFender.* : .* : s.. : confirmBrandFender 1000
R: .* : .* : .* : confirmBrandFender -100
  ...
R: .* : .* : fff : bye 1000
R: .* : .* : .* : bye -1000

O: (.* ) : $1 1.0
O: (.* ) : .* 0.0

```

Figure 30: Redacted pomdp specification.

7.2 Logic and Data Access Layer

The logic and data access layer acts as an intermediary between the presentation and data layers. In addition to the core business processes that drive the application, the layer coordinates the application by facilitating client-server communication and access to the data layer.

As shown in Figure 25, the `DialogueController` is a Java servlet that mediates between the business logic and the presentation layer. It is responsible for capturing HTTP GET and HTTP POST requests and guiding system control to appropriate business logic before returning a result to the user. The `DialogueController` maintains a connection to the relational database through its `DialogueModel` object and manages users via session-specific `SearchSession` objects. A `DialogueModel` uses objects, such as `QuestionData` and `ExtractionPattern`, to access the relational database. For performance reasons, the `DialogueModel` stores regular expression extraction patterns for recognizing user utterances in memory.

A `SearchSession` maintains the dialogue state for a user's search session, which requires interfacing with other business logic components. Its `process()` method takes a sorted set of observations and, in turn, delegates to the appropriate `DialogueManager` objects so they can update themselves with respect to their local contexts. The `SearchSession` also interacts with a `Monitor` instance that oversees additions to the knowledge base, checking for constraint failures and responding to them appropriately.

Each `SearchSession` has its own set of `DialogueManagers` that are dynamically instantiated on demand. The full network of connected dialogue managers is shown in Figure 31. The `DialogueManager`'s `update()` method operates over the subtree rooted at the `DialogueManager` instance. The low-level belief state update operation is delegated to the `PomdpAlphaDialogueManager` instance associated with the `DialogueManager`. As shown in Figure 25, a `PomdpAlphaDialogueManager` accesses a POMDP specification

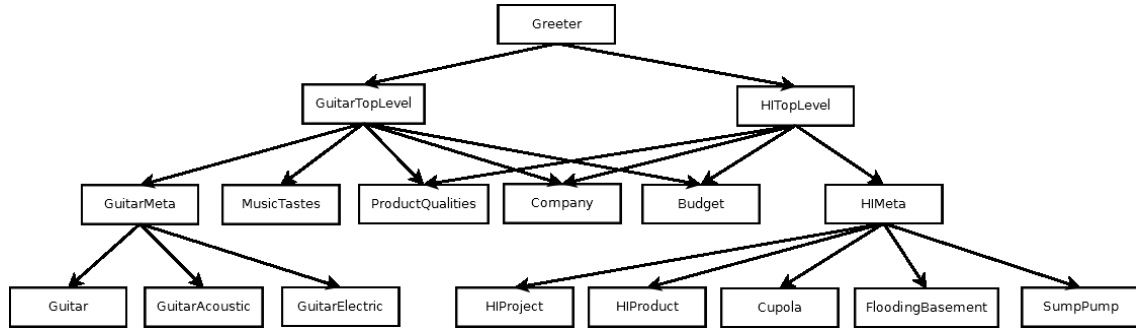


Figure 31: Hierarchy of dialogue managers.

and the corresponding alpha file containing policy vectors to update beliefs and to choose system actions.

The domain and goal models are programmatically accessed and manipulated by the Monitor using the Jena framework. Jena provides a collection of tools and Java libraries that facilitate reading, writing, processing, and querying RDF data, including OWL [157]. In particular, the Ontology API enables dynamic access and management of ontology concepts and inferencing mechanisms, and Jena’s ARQ query engine allows SQL-like retrieval of statements.

Data is passed between the presentation and logic layers via JSON messages. JavaScript Object Notation (JSON) is a data interchange format that represents objects as sets of key-value pairs. Its syntax is very simple, making it easy for humans and machines to write and understand [158]. Google’s Gson is a Java library that is used to convert data (in the form of Java objects) into JSON representation so they can be processed in the presentation layer [159].

7.3 Presentation Layer

The presentation layer interacts directly with the user and is responsible for generating rich output. The client relies on JavaScript functionality to communicate with the business logic implemented on the server.

The dialogue phase is performed using the index.html page whose content is managed by main.js. main.js uses the jQuery and jQuery UI JavaScript libraries to achieve a responsive user interface that is updated automatically in response to user input [160, 161]. The jQuery library provides convenient access to the index page's HTML elements and advanced functionality for error handling, animations, and AJAX interactivity that facilitates asynchronous communication between the client and the server. AJAX requests update the state and the index page's content after receiving JSON data from the server. The slimScroll jQuery plugin is also used to create a scroll bar for the help content if the height of the help text exceeds its boundaries on the page. slimScroll furnishes an attractive scroll bar that can be customized and hidden on demand [162].

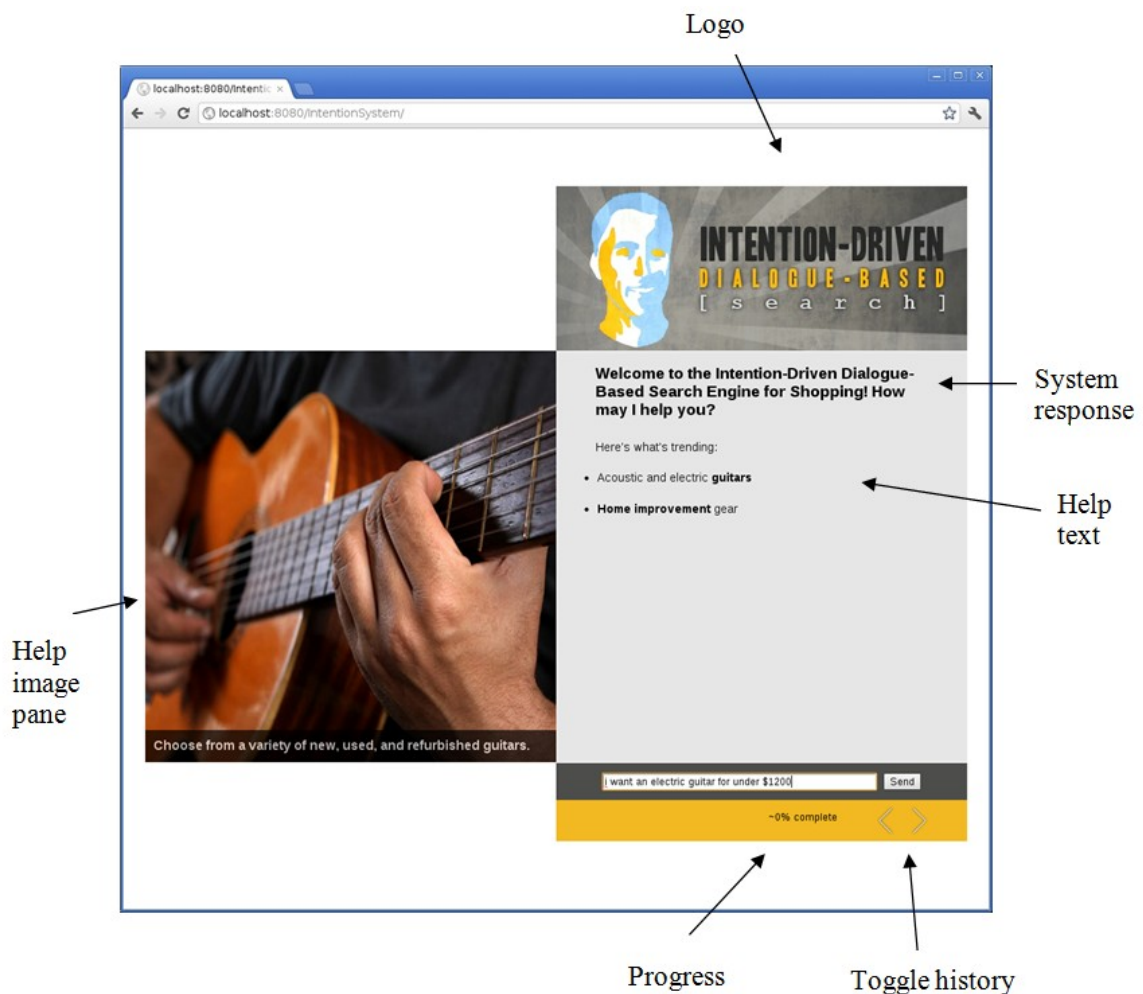


Figure 32: Index page.

The index page features a slide-based page transition scheme where system responses are signalled by new content sliding in from the right of the screen and the user can view his/her previous answers by navigating to past slides. The index page's interface is illustrated in Figure 32. The system response text is prominently large and is followed by help text. Moving the mouse over the help text causes the associated image to appear in the help image pane. The progress percentage is updated throughout the conversation as new information is gained by the system about the wishes of the user. For example, after stating a need for an electric guitar for less than \$1200, the progress percentage jumps from 0% to 37% (see Figure 32 and Figure 33). An error message overlays the help image pane to alert the user in case of any inconsistencies in the information they provided with respect to all current and past information given (see Figure 34).

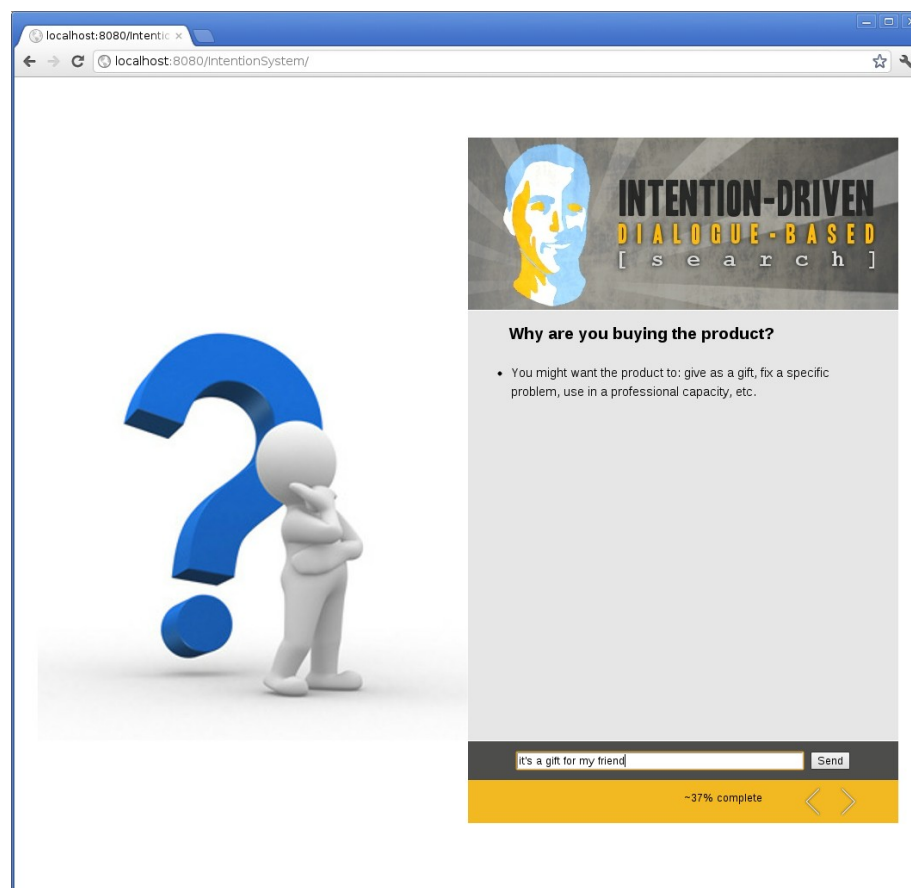


Figure 33: Index page after transition due to user input showing the system's response.

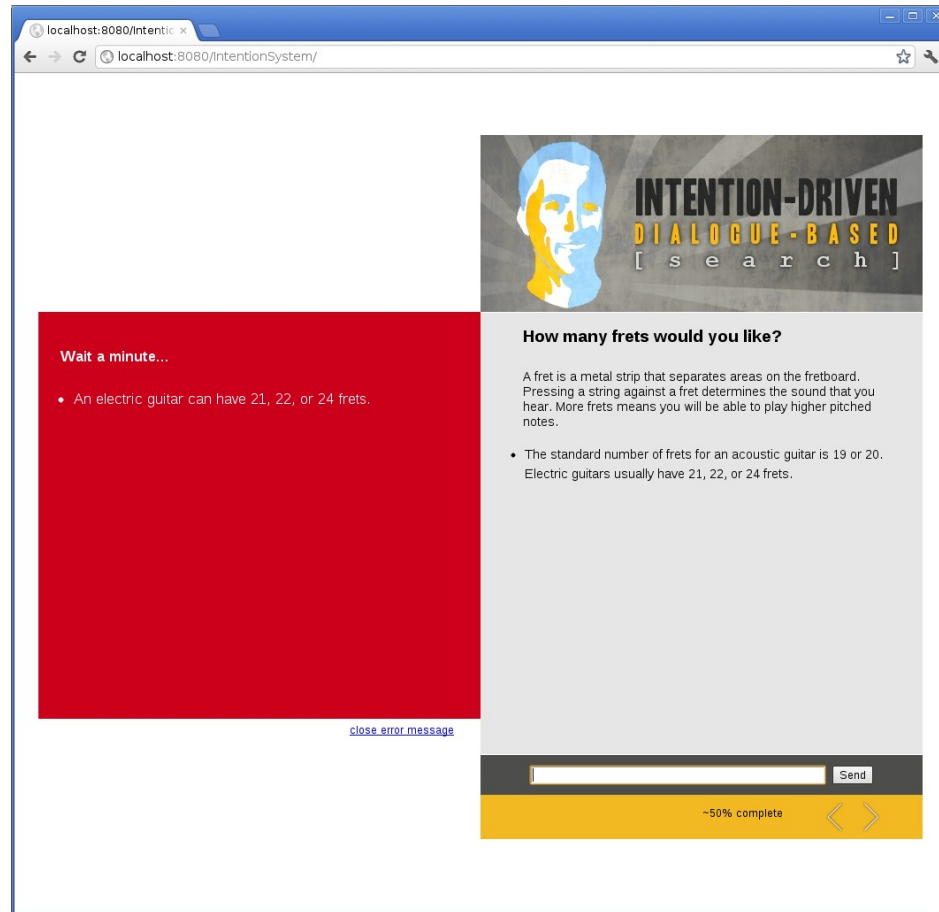


Figure 34: Constraint failure resulting in an error message.

The query results are presented in the results.html page whose content is managed by results.js. results.js takes the system-generated query, sends it to Google's Search API for Shopping using an asynchronous JSONP request, and processes the retrieved JSON product data in real-time. Google's Search API for Shopping takes HTTP GET requests to probe data that has been uploaded to Google's Merchant Center [163]. A variety of product attributes are retrieved in the response including product descriptions, images, prices, brands, and conditions. JSONP allows data from Google (a different domain) to be returned as a parameter to results.js so it can be inserted into the script and executed. The code implemented in results.js essentially loops through the product objects returned by Google, analyzes them using regular expressions, and generates an array of HTML list content to add to results.html. Note that new data is fetched, processed, and appended to

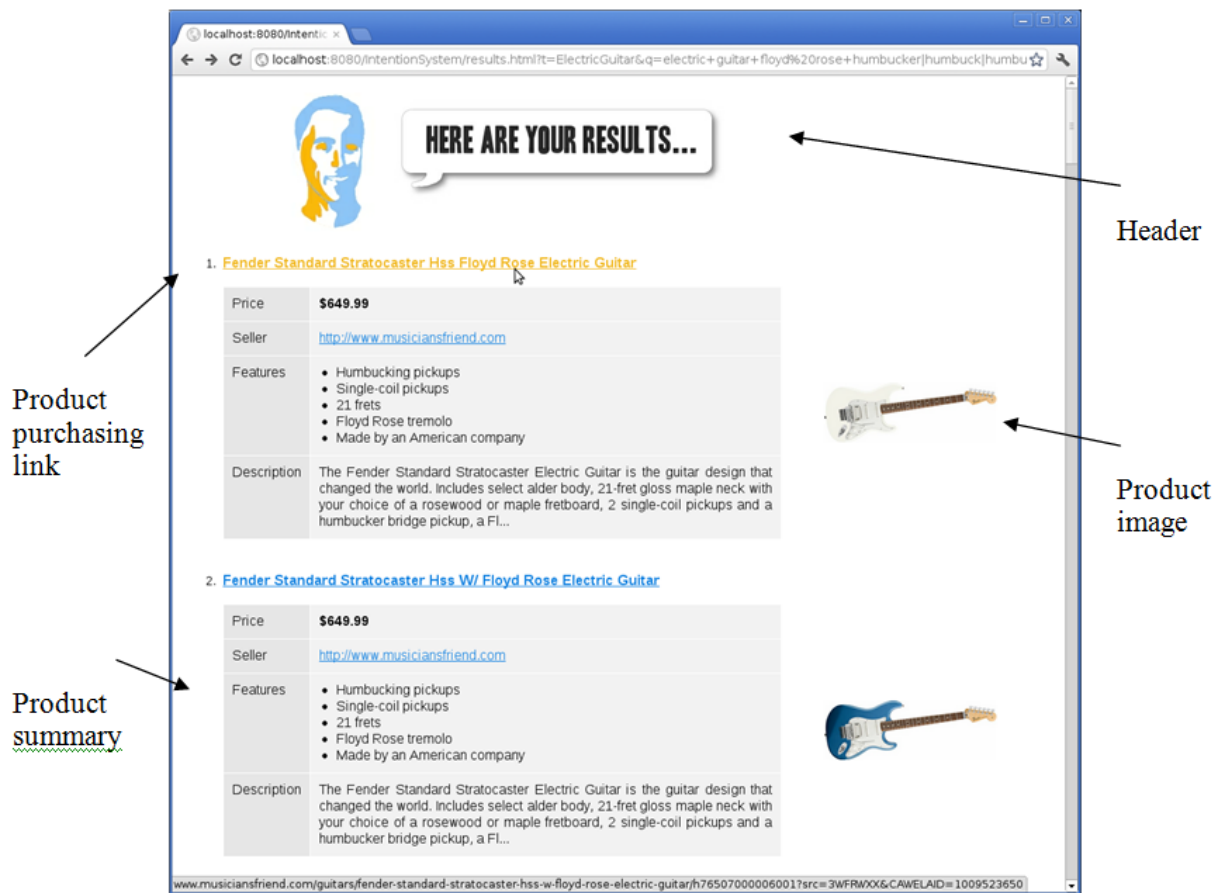


Figure 35: Results page.

the results page whenever the user scrolls to the bottom of the page. For example, Figure 35 shows the results of a search for an electric guitar. Clicking on the product purchasing link will take the user to the product offer's Web page. Notice the structured representation of relevant information pertinent to the user's goals in the product summary tables.

7.4 Implementation Tools

The search engine was developed using a variety of software tools including the Eclipse Java Enterprise Edition integrated development environment [164], TopBraid Composer (Free Edition) [165], Google Chrome [166], and the GNU Image Manipulation

Program (GIMP) [167]. Eclipse was used due to its popularity and extensibility through various plugins. The Eclipse installation contained the Eclipse Web Tools Platform and Eclipse Data Tools Platform plugins to support database integration and Web development. The Web application was executed within an Apache Tomcat 7.0 servlet container managed by Eclipse [168]. TopBraid Composer (Free Edition) was used to create the ontologies. Built on the Eclipse platform, it offers a familiar interface and comprehensive support for developing, managing, and testing configurations of knowledge models and their instance knowledge bases [165]. The Free Edition natively supports OWL ontologies, SPARQL querying, and SPIN constraints and rules. The Google Chrome Web browser was employed for testing and debugging the Web interface. In particular, its Developer Tools came in handy for manipulating HTML elements and styles and for debugging JavaScript functionality. The GIMP was used to create and/or edit images. The GIMP offers many powerful graphics editing capabilities. Admittedly, the GIMP provides more than enough functionality for relatively basic graphics editing.

7.5 Notes on the Implementation

The implementation required around 20 different technologies and tools for its development and execution. During the course of the development phase, several other technologies were also considered but ultimately rejected due to malfunctioning behaviours, incompatibilities, or personal developer preferences. This section describes some of these technologies.

A significant amount of development work was carried out for ontology building. The Pellet reasoner [169] and the SWOOP ontology editor [170] were intended to be used for the creation of a modular, distributed collection of ε -connected ontologies to form the knowledge base. However, the SWOOP ontology editor did not properly allow the creation of ε -connection link properties and the Pellet reasoner for ε -connected ontologies was no longer supported by the developers. The Protege ontology editor [171]

was experimented with, but TopBraid Composer (Free Edition) was chosen instead due to its more familiar interface and built-in support for the SPIN API.

Java Enterprise Edition (J2EE) 6 technologies were closely examined and applied to develop the system. In particular, JavaServer Faces (JSF) Technology with Enterprise JavaBeans (EJBs) were considered for the construction and management of the search application. JSF is a relatively new standard for building server-side user interfaces [172]. An EJB encapsulates business logic and makes use of the J2EE container for transaction and scalability management [173]. Initial development work stored business logic in EJBs and connected the EJBs with the JSF presentation layer using JSF managed beans and/or Contexts and Dependency Injection (CDI) services. CDI simplifies application development by allowing J2EE components to be bound to lifecycle contexts, acquire references to other components through dependency injection, and respond to observed events in a decoupled way [174]. In addition, JavaServer Pages (JSP) technology was considered for the generation of presented content [175]. Instead of JSF (or JSP), EJBs, and CDI, the implementation uses the simplest approach wherein a servlet mediates between the client and the server, and the server uses Plain Old Java Objects (POJOs) with synchronized method execution to keep track of stateful, session-specific information. The user interface is dynamically generated on the client side using AJAX.

In addition, the Semantic Web Rule Language (SWRL) was investigated to represent the integrated multi-domain goals using Horn-like rules. SWRL combines OWL DL and OWL Lite with Unary/Binary Datalog RuleML sublanguages to extend the set of OWL axioms to include Horn-like rules [176]. In other words, it allows the expressive procedural declaration of ontological axioms. For example, the *BrightSoundingGuitar* concept can be defined by chaining together different desired properties. This approach is intuitive, but it requires the definition of many different goal classes. A simpler approach has been used instead, where the goals are captured by context-specific SPARQL queries.

Even the integrated development environment (IDE) choice changed during the development phase. Initially, the NetBeans IDE [177] was configured and used. However, after a few months of programming, the Eclipse IDE was selected due to its powerful extensibility and popularity at the University.

Chapter 8: Usability Study

A usability study was performed to investigate users' search interactions using Google-based searching and intention-driven searching (as provided by this thesis's IDS search engine). The usability study addressed four hypotheses:

1. The intention-driven, dialogue-based search method will take less time than traditional keyword-based searching (e.g. Google-based) when the user is unfamiliar with the topics that comprise their search goal.
2. The intention-driven, dialogue-based search method will take less effort than traditional keyword-based searching (e.g. Google-based) when the user is unfamiliar with the topics that comprise their search goal.
3. The user will be more confident in the results he/she achieves with the intention-driven approach.
4. IDS's dialogue-based interaction will be natural and helpful.

8.1 Description

A usability study assesses the extent with which a specific set of users can achieve goals effectively, efficiently, and with satisfaction in a given context of use [178]. Usability is typically measured in terms of task completion rates (for a measure of effectiveness), mean task completion times (for a measure of efficiency), and mean participant satisfaction ratings (for a measure of satisfaction) [179]. Other possible measurements include the number of tasks completed within a specified time limit, the number of wrong menu choices, and the number of user errors [180].

A usability study was chosen to investigate users' searching behaviours because it allows the collection of quantitative and qualitative data from real human users. Although many dialogue-based systems have been examined using computer simulations of user behaviours (e.g. [83, 181, 82]), a usability study avoids issues pertaining to the

viability or appropriateness of any simulated human behaviours in a multi-domain search environment.

In this study, participants engaged in search sessions to find products that met the expectations outlined in fictional scenarios. Two fictional scenarios were selected by the researcher from an area of knowledge that the participant chose as one he/she was least familiar with. The participant conducted a typical Google-based search—one in which the user starts with a Google search and continues onwards through the browsing and searching phases unrestricted in which sites they can access (as in [182])—to satisfy the needs of one scenario, and a search using IDS to meet the needs outlined in the other scenario. After the completion of each scenario's search session, the participant graded his/her experience with the employed search method. Note that the order of search system usage and the assignment of scenarios were both alternated to mitigate biases in participants' judgments induced by ordering.

University of Windsor students of all ages, genders, ethnicities, and majors were recruited to participate in the study via posters affixed to campus bulletin boards. The compensation for participation was entry into a random draw with the chance to win one of 10 cash prizes, each valued at \$25. Although this population is representative of a relatively well-educated group of individuals that are experienced with computer technologies, the lack of restrictions in student recruitment theoretically enables the assessment of individuals with diverse backgrounds, proficiencies, and interests.

The study took place in a noise- and distraction-free computer lab. The test computer system was a Debian operating system-based desktop computer with a quad-core 2.40 GHz CPU and 3 GB of DDR2 SDRAM. The sequence of procedures is summarized as follows:

1. The participant is briefed on users' search techniques and behaviours, the goal-oriented nature of searching, and the role of multiple topics on goals.

2. After providing some information about his/her experience level with search engines, the participant chooses an area he/she is least comfortable with (in the context of gift buying).
3. The researcher toggles the order of system usage (Google-based or IDS) and the order of the given scenarios.
 - a. The participant executes a Google-based search session to find a product that meets the needs outlined in the first scenario. Then, the participant grades his/her experiences.
 - b. The participant performs a search session using this research work's system to find a product that satisfies the requirements of the second scenario. The participant then grades his/her experiences.
4. The participant finishes the study by evaluating his/her perceptions of the strengths and weaknesses of the two search methods as well as his/her subjective preferences concerning those methods.

8.2 Techniques and Measures

The usability study employed two experiment techniques: participant observation and a questionnaire/survey. The questionnaire assessed background information concerning the experiences of the user with searching and the chosen gift purchasing area, as well as the user's impressions or subjective judgments of the search methods (see Appendix C: Questionnaire).

The observation part of the study was designed to acquire quantitative data. Participant observation involved the following measures:

1. The elapsed time for the search session, starting from the first submitted query to the selection of the final answer.
2. For Google-based searching:
 - a. The number of queries explicitly submitted on any search site.

- b. The number of Web resources accessed, such as Web pages and PDFs, excluding search results pages.
 - c. The number of search results pages viewed.
 - d. The quality of the final answer as determined by a comparison of the attributes of the user's answer with those of an intended product.
- 3. For intention-driven searching:
 - a. The number of questions answered by the user.
 - b. The number of error messages produced by the system.

Note that the number of submitted queries consists of keyword-based queries as well as any explicit requests by the user through online form-based mechanisms. For example, the common act of filtering the results by specifying category or property restrictions (view-based searching) counted as a submitted query.

8.3 Approval

As the study involved human participants from the University, the aforementioned experiment techniques and procedures were outlined and submitted to the University's Research Ethics Board (REB) for approval. As a prerequisite, the Tri-Council Policy Statement (TCPS2) Course on Research Ethics (CORE) training was completed to ensure adequate knowledge of ethical experiment practices with human participants. The human testing application process required the clear identification of the study's purpose and objectives, the methods for meeting those stated objectives, and a description of all aspects pertaining to the recruitment and treatment of participants and the collected data. All materials used to obtain participants and formalize the experiment, such as the consent form and recruitment poster (see Appendix A: Consent Form and Appendix B: Recruitment Poster) were included in the REB application. Low risks were indicated for the participants, including psychological, physical, social, and data security factors. Note that the approval process took several weeks to complete (with one requested set of revisions). The experiment culminated with the submission of a final report to the REB.

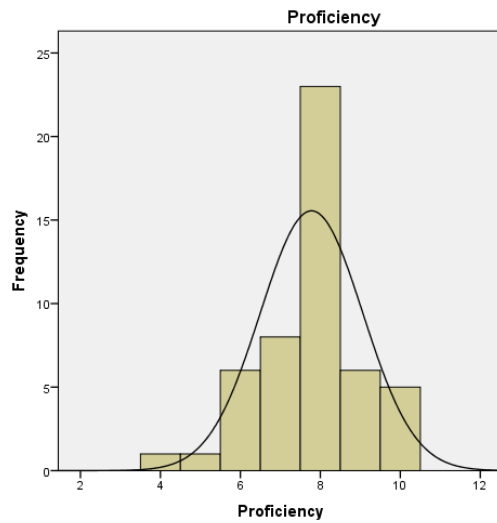


Figure 36: Proficiency distribution.

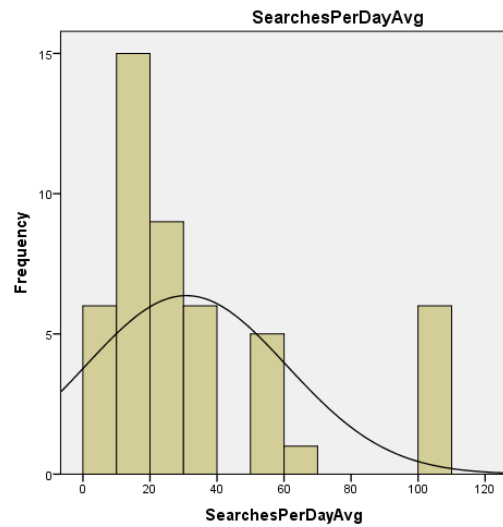


Figure 37: Distribution of the average number of searches per day.

8.4 Results and Analysis

The Descriptive Statistics section details the data collected during the study. After a review of the statistics, statistically significant differences in performance for the two approaches are discussed in Comparing Time, Effort, and Confidence. Explaining the Differences investigates these differences (or lack thereof) using analysis of variance tests and linear regression. The naturalness of the IDS-based searching technique is examined and, lastly, the overall results are summarized and analyzed.

8.4.1 Descriptive Statistics

The sample population consisted of 50 students, with 64% (32 of 50) majoring in Computer Science and 36% (18 of 50) having other majors. Overall, participants were quite proficient using Google-based searching, with a mean of 7.78 out of 10 (SD=1.282, N=50) proficiency for the reasonably Normal proficiency distribution (skewness -0.477 and kurtosis 0.747), shown in Figure 36. The participants were often experienced, performing between 2 and 100 searches per day with a median of 20. These statistics

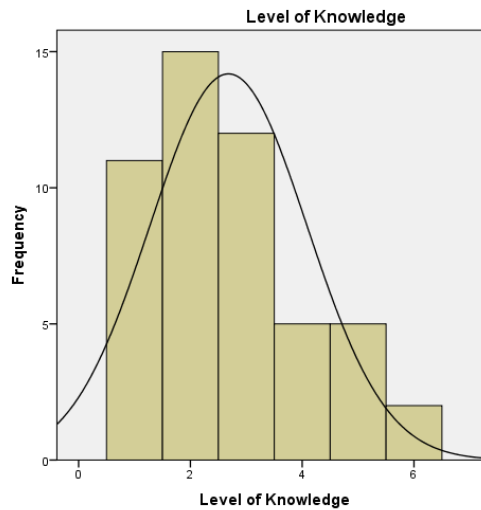


Figure 38: Level of knowledge distribution.

were computed with two missing values—two participants entered “too many [searches] to count”. This demonstrates a nonnormal distribution that is skewed by the participants who perform a very high number of queries, as shown in Figure 37.

Given the experiment setup procedures, participants were responsible for selecting the area they felt least familiar with. The Guitar shopping area was chosen 76% of the time (38 of 50), while the Home Improvement shopping area was chosen 24% of the time (12 of 50). Overall, participants expressed a very low level of knowledge about their chosen area, with 72% reporting a level of knowledge of 3 or lower (out of 10). The reasonably Normal distribution of the subjective level of knowledge (as evidenced by skewness 0.693 and kurtosis -0.278) are shown in Figure 38.

The order of system usage and the order of scenario selections were randomly assigned to participants, controlling for each participant’s major and chosen shopping area. 50% (25 of 50) of the participants used Google before IDS, with the remaining 50% (25 of 50) using IDS before Google. Similarly, 50% (25 of 50) of the participants were given scenario A and then B, while the other 50% were given scenario B and then

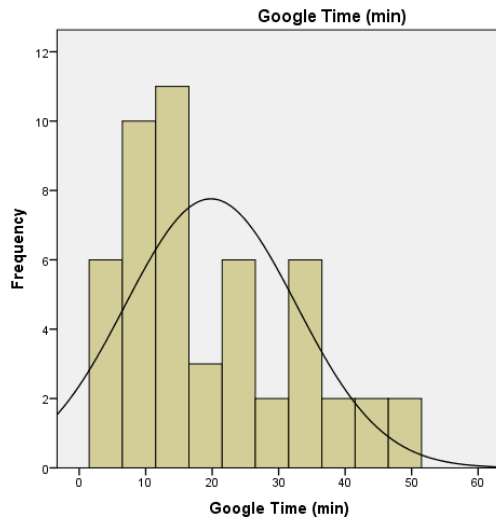


Figure 39: Google-based searching elapsed times.

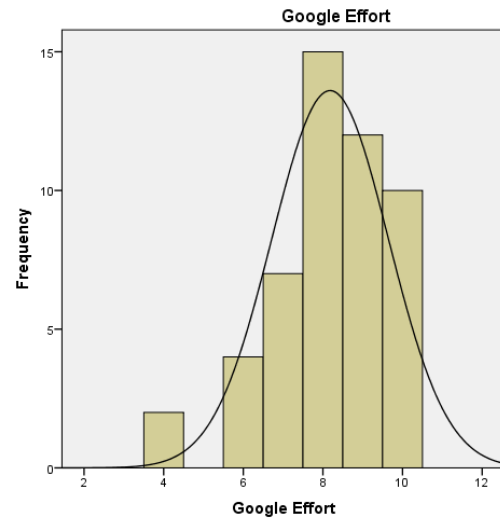


Figure 40: Google-based searching effort scores.

A. Overall, the scenario and order selections were proportional for all combinations of majors and shopping areas (see Appendix E: Experiment Data for frequencies).

Google-based searching was characterized by a high amount of time and effort and a moderate level of confidence in the results. The elapsed time (in minutes) for Google-based searching was reasonably Normally distributed (skewness 0.702, kurtosis - 0.624), with a mean of 19.8 (SD=12.854, N=50). Although it was relatively Normally distributed, the elapsed time observations had a wide range, with a minimum of 4 and a maximum of 49. Using Google-based searching, participants sent between 3 and 55 queries, with a median of 16. The number of page views was nonnormal (skewness 2.879, kurtosis 11.883), ranging from 2 to 104. The five-number summary of the page views is (2, 10, 19, 27, 104), indicating a median of 19 and a large variance. The number of generated results pages was nonnormal (skewness 1.05, kurtosis 0.23), ranging from 3 to 57 with a median of 16. Given these behaviours, the amount of effort, measured on a scale from 1 to 10, was reasonably Normal (skewness -0.891, kurtosis 0.945), with a mean of 8.18 (SD=1.466, N=50). Similarly, the amount of confidence in the achieved results were also reasonably Normal (skewness -0.553, kurtosis -0.369), with a mean of

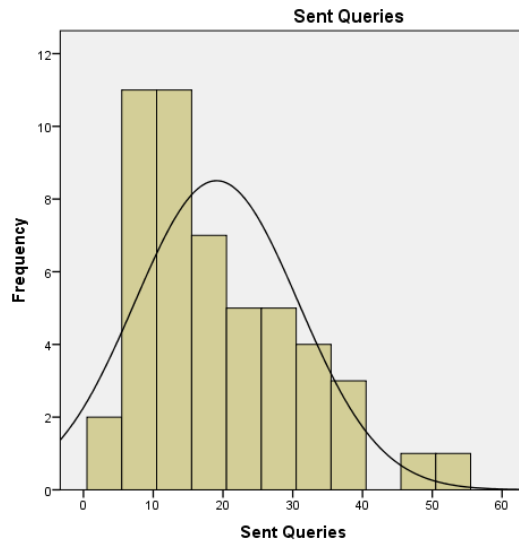


Figure 43: Distribution of the number of sent queries.

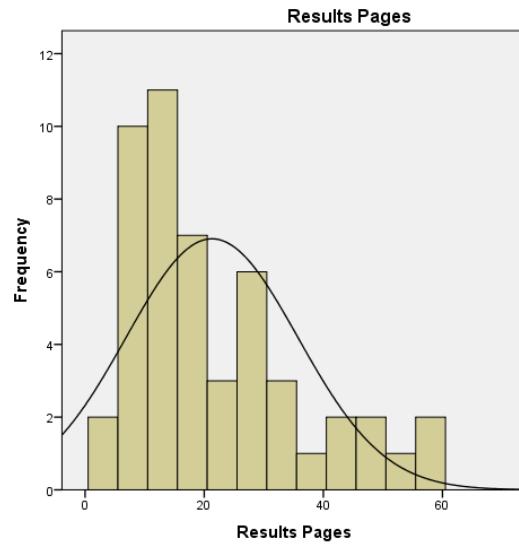


Figure 44: Distribution of the number of generated results pages.

6.04 (SD=2.285, N=50). The distributions (with Normal curves) of these Google-based search measures are shown from Figure 39 to Figure 42.

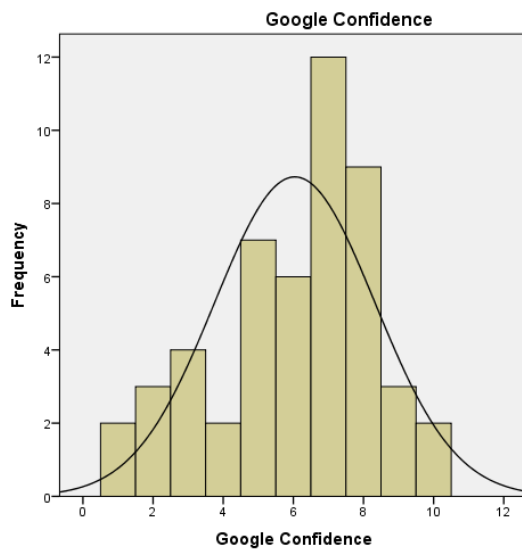


Figure 41: Google-based searching confidence scores.

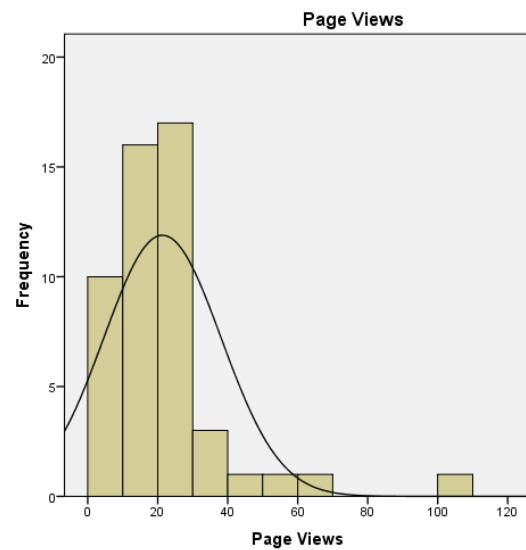


Figure 42: Distribution of Web page views.

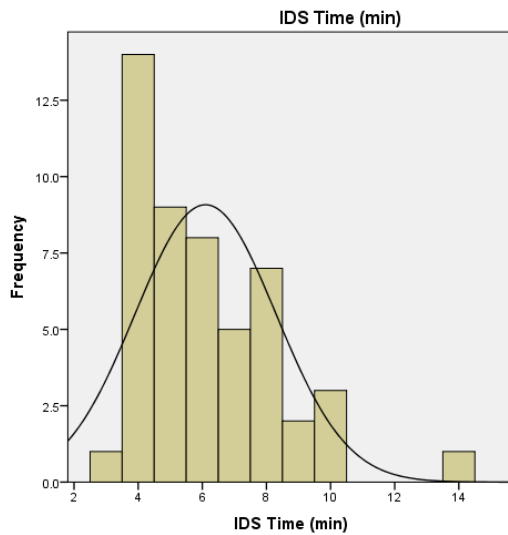


Figure 45: IDS-based searching elapsed times.

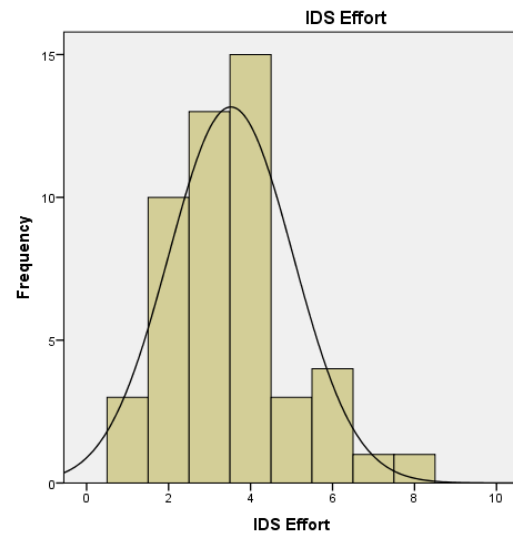


Figure 46: IDS-based searching effort scores.

IDS-based searching, on the other hand, had low time and effort demands and produced confident results. The elapsed time (in minutes) was nonnormal (skewness 1.203, kurtosis 1.959), ranging from 3 to 14, with a median of 6. The time values tended toward the lower values, with a high peak at 4. The number of questions followed a reasonably Normal distribution (skewness -0.33, kurtosis -0.453), having a mean of 12.3 (SD=2.435, N=50). The number of error messages generated by the system (indicated by constraint failures) was nonnormal (skewness 0.855, kurtosis 1.009), ranging from 1 to 5 where 62% of the failures were less than or equal to 1. Given these interaction measurements, the amount of effort, as determined by participants on a scale from 1 to 10, was reasonably Normal (skewness 0.73, kurtosis 0.709) with a mean of 3.52 (SD=1.515, N=50). The confidence in IDS-generated results was highly skewed and peaked (skewness -1.304, kurtosis 1.795), with a minimum of 6 and a maximum of 10. A full 42% (21 of 50) of the participants chose 10 out of 10 as the level of confidence in the results. Finally, the naturalness of the conversation was nonnormal (skewness -1.304, kurtosis 0.78), ranging from 4 to 10 with a median of 8. The histograms (with Normal curves) for the IDS-based searching measures are shown from Figure 45 to Figure 50.

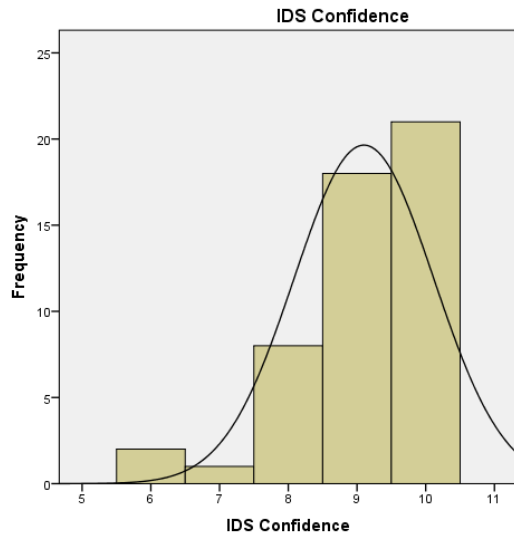


Figure 47: IDS-based searching confidence scores.

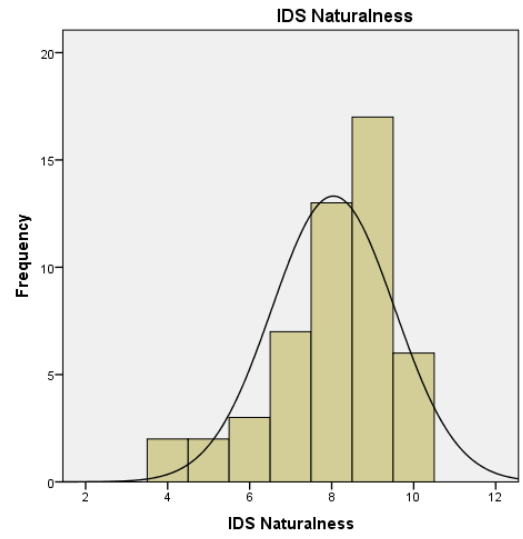


Figure 48: Distribution of the IDS-based searching naturalness scores.

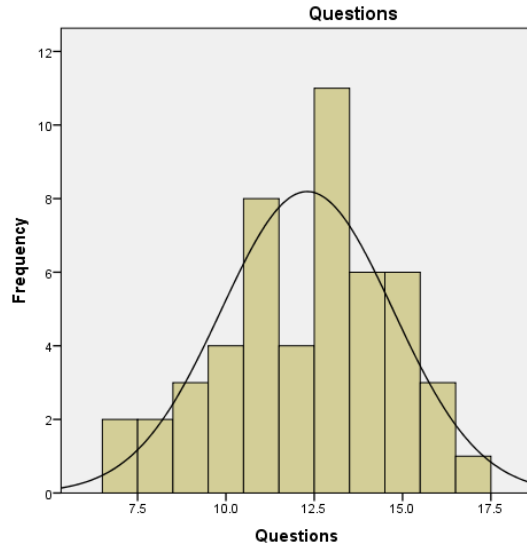


Figure 49: Distribution of the number of questions asked by the system.

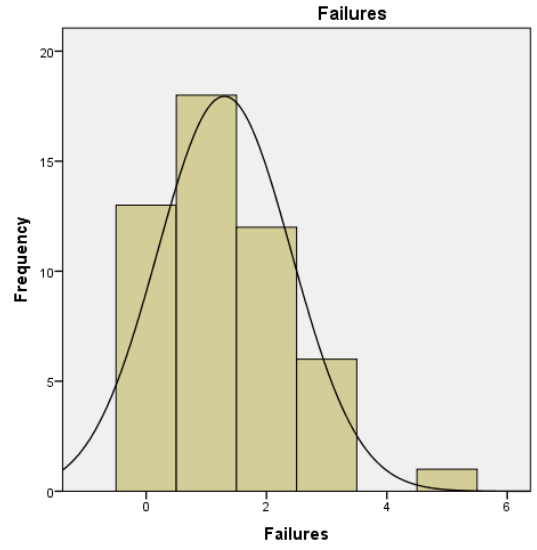


Figure 50: Distribution of the number of constraint failures.

8.4.2 Comparing Time, Effort, and Confidence

Hypothesis 1: *The intention-driven, dialogue-based search method will take less time than traditional keyword-based searching (e.g. Google-based) when the user is unfamiliar with the topics that comprise their search goal. Equivalently: Traditional keyword-based searching (e.g. Google-based) will take more time than the intention-driven, dialogue-based search method when the user is unfamiliar with the topics that comprise the search goal.*

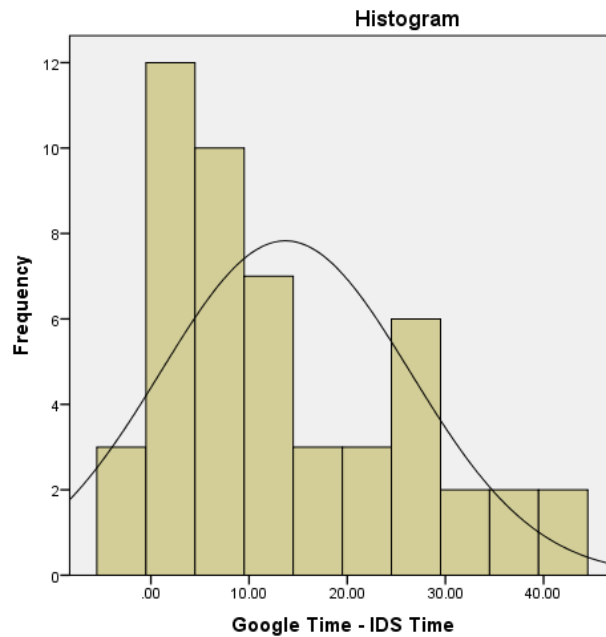


Figure 51: Distribution of the paired time differences.

Paired Samples Test									
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	Google Time (min) - IDS Time (min)	13.700	12.740	1.802	10.079	17.321	7.604	49	.000

Table 6: Results of the paired samples t-test for elapsed time.

A paired t-test was performed to determine whether Google-based searching took more time than IDS-based searching. The distribution of the differences is reasonably Normal, as evidenced by the skewness and kurtosis values (skewness 0.756, kurtosis -0.444). The mean time difference ($M=13.700$, $SD=12.740$, $N=50$) was significantly greater than zero, $t(49) = 7.604$, one-tailed $p < 0.001$, indicating that Google-based searching required significantly more time than IDS-based searching. The distribution of the time differences are shown in Figure 51. Since the mean difference was quite large (13.7 minutes, shown in Table 6), these time differences are very significant.

Hypothesis 2: *The intention-driven, dialogue-based search method will take less effort than traditional keyword-based searching (e.g. Google-based) when the user is unfamiliar with the topics that comprise their search goal. Equivalently: Traditional keyword-based searching (e.g. Google-based) will take more effort than the intention-driven, dialogue-based search method when the user is unfamiliar with the topics that comprise the search goal.*

A paired t-test was generated to ascertain whether the Google-based searching technique required more effort than IDS-based searching. The differences in effort do not appear to follow a Normal distribution, as indicated by the large absolute values of the skewness and kurtosis (skewness -1.317, kurtosis 3.256) and as shown in Figure 52. This is likely due to the noticeable outliers located outside of the Normal curve (see Figure 52). Since the number of samples is “large” (greater than 40), it is reasonable to invoke the paired t-test (due to the Central Limit Theorem). The mean effort difference ($M=4.660$, $SD=2.228$, $N=50$) was significantly greater than zero, $t(49) = 14.790$, one-tailed $p < 0.001$, which implies that Google-based searching takes more effort than IDS-based searching. The mean difference in effort was substantial (4.660), as shown in Table 7.

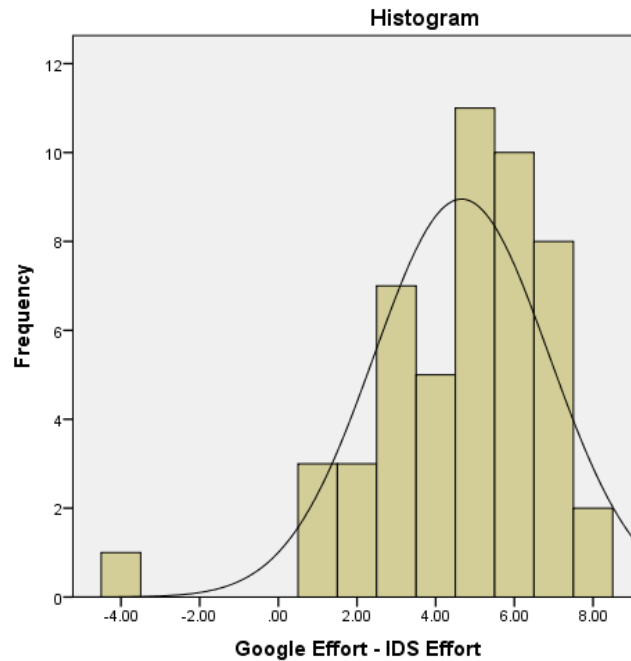


Figure 52: Distribution of the paired effort differences.

Paired Samples Test									
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	Google Effort - IDS Effort	4.660	2.228	.315	4.027	5.293	14.790	49	.000

Table 7: Paired samples t-test for the effort scores.

Hypothesis 3: *The intention-driven, dialogue-based search method will produce more trustworthy results than traditional keyword-based searching (e.g. Google-based) when the user is unfamiliar with the topics that comprise their search goal. Equivalently: The user will be less confident in the results he/she achieved using Google-based searching.*

A paired t-test was performed to determine whether users were less confident in the results they achieved using Google-based searching. The assumption of normality for

the paired t-test is satisfied as indicated by the skewness and kurtosis (skewness -0.483 and kurtosis -0.059). The distribution of the differences is depicted in Figure 53. Since the mean confidence difference (M=-3.060, SD=2.502, N=50) is negative, with $t(49) = -8.647$ and one-tailed $p < 0.001$, there is sufficient evidence that confidence in results derived from Google-based searching is less than confidence in results obtained through IDS-based searching. Overall, the difference is meaningfully lower, with a mean difference of -3.06, as shown in Table 8.

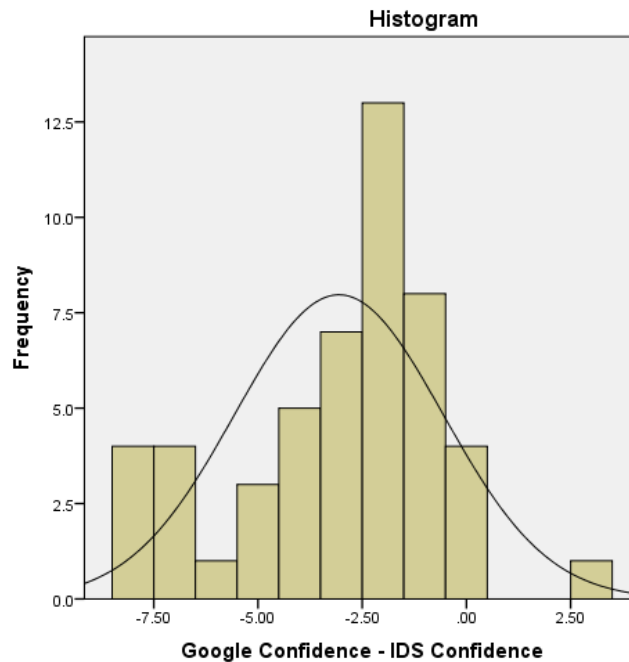


Figure 53: Distribution of the paired confidence differences.

Paired Samples Test									
		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
					Lower	Upper			
Pair 1	Google Confidence - IDS Confidence	-3.060	2.502	.354	-3.771	-2.349	-8.647	49	.000

Table 8: Paired samples t-test for confidence differences.

8.4.3 Explaining the Time, Effort, and Confidence Differences

This investigation also attempts to examine possible reasons for the difference scores in the hopes of uncovering relationships that suggest areas for future work. First, the relationship between the pre-trial (background) variables and the time, effort, and confidence differences were evaluated using one-way analysis of variance tests. The independent variables were considered as fixed factors to investigate their effects on the dependent difference variables. Note that the proficiency score was transformed into categories to generate populated subgroups. The level of knowledge was not examined because they were quite invariant due to the specification that participants must select the area they are least familiar with. Second, multiple linear regression equations were derived from in-trial participant behaviours to determine any linear relationships between those behaviours and the observed differences. Specifically, the Google-based observations (number of sent queries, number of pages viewed, number of results pages) and the IDS-based observations (number of questions, number of errors/failures) were examined.

The results are summarized as follows:

1. The participants' majors were shown to have insignificant effects on the time and confidence differences. There was insufficient/incompatible data to assess the effort differences.
2. Using the proficiency, area, and scenario as fixed factors, the scenario was influential to the time differences. The A scenarios took longer using Google than the B scenarios took using IDS and, in general, A scenarios were harder. This also affected the confidence values, as the larger confidence differences occurred for Home Improvement with Google applied to scenario A.
3. The number of sent queries and page views were the best predictors of the time differences, which suggests that Google-based behaviours were more variable and dominated the difference calculation.

4. The number of sent queries and the number of questions were the best predictors for the effort differences. This indicates that user effort for Google-based searching was best approximated by the number of sent queries, while user effort in IDS-based searching was best approximated by the number of questions answered.
5. The number of Google-based page views was the best predictor for the confidence differences, showing that as the number of page views increases, the confidence decreases. This makes sense: Less confident users will look at more pages.

8.4.3.1 Effects of Major

First, one-way analysis of variance tests were performed to investigate the influence of the participants' majors on the time, effort, and confidence differences. The assumption of Normality was already established for the time differences. Levene's test of equal variances produced an F-value of 0.038, $p = 0.846$, confirming the assumption of equal variances. Applying Cohen's criteria for effect size, the major was shown to have a small, insignificant effect on the time difference ($F=1.260$, $p=0.267$, partial eta squared=0.026).

The distribution of effort differences was shown to be nonnormal and Levene's test produced $F=4.301$, $p=0.043$, rejecting the assumption of equal variances. ANOVA is robust to violations of the equal variance assumption provided that the largest group variance is less than or equal to three times the smallest group variance [183]. However, the computed variance for the Computer Science group of 6.66 is more than the 1.99 variance for the Other group. Therefore, the ANOVA is an inappropriate test for the effort differences.

The distribution of confidence differences was shown to be Normal and Levene's test generated $F=0.123$, $p = 0.728$, allowing the assumption of equal variances. The

major is insignificant with respect to the confidence differences ($F < 0.001$, $p=0.993$, partial eta squared < 0.001).

8.4.3.2 Effects of Proficiency, Area, and Scenario

One-way ANOVA tests using a General Linear Univariate Model and Bonferroni's significance adjustment were performed to ascertain whether the proficiency, area, and scenario assignments were influential with respect to the time differences. The time differences were Normal and Levene's test of group variance homogeneity produced $F=1.477$, $p=0.177$, maintaining the assumption of homogeneity. The test shows that the scenario selections ($F=8.937$, $p=0.005$, partial eta squared=0.186) as well as the interaction between the area and the scenario selections ($F=6.789$, $p=0.013$, partial eta squared 0.148) were significant. As shown in Table 9, the time difference was the largest when users searched with Google having scenario A. In particular, the Home Improvement scenario A took a lot longer to solve using Google than scenario B using IDS for Home Improvement (32.14 mean time difference). The large positive mean difference in time depending on the Home Improvement scenario may suggest that scenario A is very difficult to solve using the Google-based approach, but it is much more manageable when tackled using IDS-based searching.

5. Area * Google Scenario

Dependent Variable: Google Time - IDS Time

Area	Google Scenario	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
Guitars	A	17.356 ^a	3.156	10.973	23.740
	B	15.067 ^a	2.970	9.060	21.074
HI	A	32.140	4.437	23.166	41.114
	B	8.227 ^a	4.530	-.936	17.391

a. Based on modified population marginal mean.

Table 9: The interaction effects of the area and the scenario on the time differences.

One-way ANOVA tests using a General Linear Univariate Model and Bonferroni's significance adjustment were performed to determine whether the proficiency, area, and scenario assignments affected the effort differences. The distribution was nonnormal, but Levene's test produced $F=1.338$, $p=0.24$, enabling the assumption of equal variances. None of the factors were shown to be significantly influential.

Finally, one-way ANOVA tests using a General Linear Univariate Model and Bonferroni's significance adjustment were performed to determine whether the proficiency, area, and scenario assignments affected the confidence differences. The distribution was reasonably Normal and Levene's test generated $F=1.542$, $p=0.153$, allowing the assumption of equal variances. The area ($F=9.105$, $p = 0.004$, partial eta squared= 0.189) and the interaction between the area and the scenario ($F=6.275$, $p=0.017$, partial eta squared= 0.139) were significantly influential. The interaction effects are depicted in Table 10. Google-based searching for Home Improvement scenario A (and using IDS for scenario B) resulted in a large average confidence difference (-5.485). In general, the Home Improvement area exhibited a larger confidence difference (average -4.535). This may suggest that there was a larger gap in the difficulties of the Home Improvement scenarios than the Guitar scenarios.

5. Area * Google Scenario

Dependent Variable: Google Confidence - IDS Confidence

Area	Google Scenario	Mean	Std. Error	95% Confidence Interval	
				Lower Bound	Upper Bound
Guitars	A	-2.021 ^a	.679	-3.395	-.647
	B	-2.642 ^a	.639	-3.935	-1.350
HI	A	-5.485	.955	-7.416	-3.554
	B	-3.268 ^a	.975	-5.240	-1.296

a. Based on modified population marginal mean.

Table 10: Interaction between area and scenario for confidence scores.

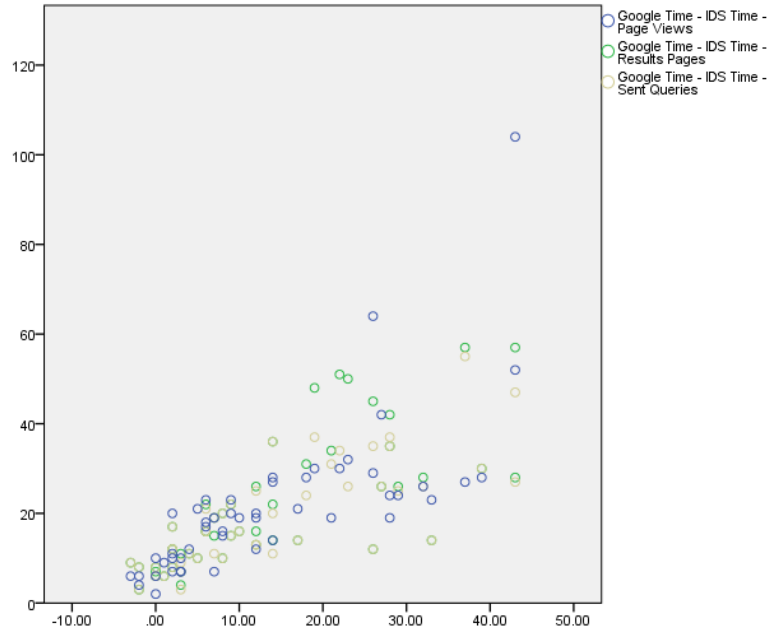


Figure 54: Scatter plot of page views, results pages, and sent queries with respect to the time differences.

8.4.3.3 Analyzing the Behaviours

After performing multiple linear regression using the backwards variable selection technique and a probability of removal given by $F \geq 0.1$, the number of sent queries and the number of page views were chosen as significant predictors for the time differences ($t = 4.748$, $p < 0.001$ for Sent Queries; $t = 3.648$, $p = 0.001$ for Page Views). The regression equation was as follows:

$$\text{Predicted Google Time - IDS Time} = -3.356 + 0.558 * \text{Sent Queries} + 0.3 * \text{Page Views}$$

Pearson's correlation statistics showed strong significant positive linear relationships between the number of sent queries, number of results pages, and number of page views and the time difference ($r = 0.774$, $p < 0.001$; $r = 0.740$, $p < 0.001$; $r = 0.733$, $p < 0.001$). This relationship is shown in Figure 54. Coupled with the generated regression equation, this indicates that the predicted time difference was most influenced by the Google-based interactions, which supports the idea that Google-based searching dominated the difference calculation due to its fluctuations in variability.

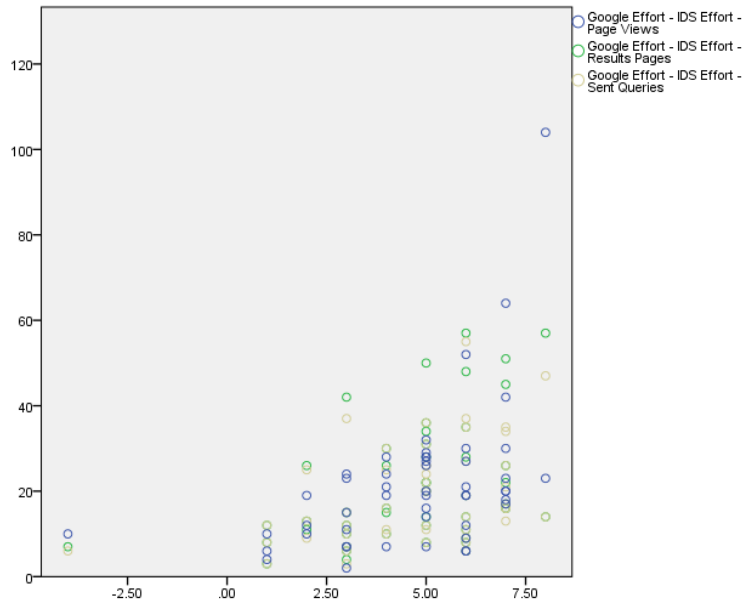


Figure 55: Scatter plot of page views, results pages, and sent queries with respect to the effort differences.

Using backwards variable selection with $F \geq 0.1$ as the removal condition, a regression equation was created to predict the effort differences. The number of sent queries and the number of questions were significant predictors ($t = 3.497$, $p = 0.001$; $t = 2.237$, $p = 0.03$), yielding the regression equation:

$$\text{Predicted Google - IDS effort} = -0.164 + 0.085 * \text{Sent Queries} + 0.261 * \text{Questions}$$

Pearson's correlation statistics showed that the number of sent queries, number of page views, and the number of results pages were moderately correlated with the difference ($r=0.46$, $p=0.002$; $r=0.402$, $p<0.001$; $r=0.402$, $p = 0.002$), as shown in Figure 55. These correlations make sense: As the indicators of Google effort increase, so too do the effort differences but it is tempered by the effort for IDS, which is dominated by the number of questions.

The multiple regression equation derived to predict the confidence differences using backwards variable selection and $F \geq 0.1$ as the removal condition was based on

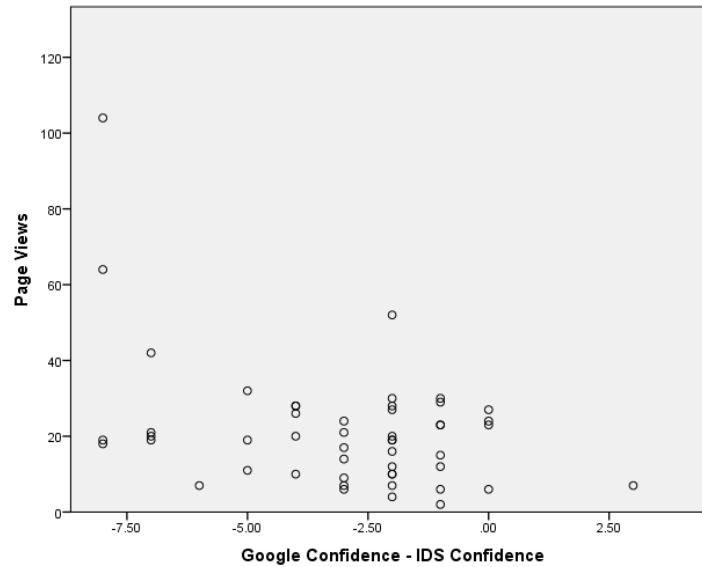


Figure 56: Page views vs. confidence differences.

one significant factor, the number of page views ($t=-2.968$, $p=0.005$). The regression equation was:

$$\text{Predicted Google - IDS confidence} = -1.807 - 0.59 * \text{Page Views}$$

Pearson's correlation statistics reveal that the number of page views was significantly negatively correlated with the confidence differences ($r=-0.394$, $p=0.002$), as shown in Figure 56. In other words, as the number of page views increases, the confidence generally decreases. This makes sense because users with less confidence will presumably continue to visit pages until they give up or reach a reasonable result.

8.4.4 Measuring the Naturalness of the Intention-Driven Approach

Hypothesis 3: *IDS's generated dialogue-based interaction is natural and helpful.*

As mentioned in the Descriptive Statistics, the naturalness of the conversation was nonnormal, ranging from 4 to 10 with a median of 8 (as shown in Figure 48). 72% (36 of 50) of the participants chose a naturalness value from 7 to 9. The distribution is

IDS Naturalness				
	Frequency	Percent	Valid Percent	Cumulative Percent
4	2	4.0	4.0	4.0
5	2	4.0	4.0	8.0
6	3	6.0	6.0	14.0
7	7	14.0	14.0	28.0
Valid 8	13	26.0	26.0	54.0
9	17	34.0	34.0	88.0
10	6	12.0	12.0	100.0
Total	50	100.0	100.0	

Table 11: Distribution of the naturalness scores.

described in detail in Table 11. In other words, users felt that IDS was quite natural to use. From questionnaire responses: “IDS ... worked with me to find what I was looking for ... it helped me through the process by guiding my search, giving me suggestions and doing most of the work for me.” Google “requires me to already be knowledgeable or do research on the topic”, whereas IDS “directs one step at a time, similar to a salesperson”. IDS provided an “iterative search rather than research and query” as opposed to one that “spews out a lot more search results that [aren’t] defined or do not suggest a specific path”.

The questionnaire responses provided several possible explanations for low values in the distribution:

1. The system has limited accepted user inputs at this time.
2. A spoken dialogue instead of a text-based interaction would be more natural.
3. The help text that the system provided was sometimes too wordy or unclear.

8.5 Summary

The hypotheses were confirmed. The intention-driven system provided a natural experience that took less time and effort than the Google-based approach and users were more confident in the results they achieved using IDS. Many statistical tests were done to explain the findings. These tests showed that the scenarios and the areas were influential with respect to the time and confidence differences, but the proficiency and major of the user were insignificant to the differences. Overall, Google-based search behaviours were better predictors of the differences, suggesting that interactions with IDS were quite stable, while interactions with Google were more variable and had larger magnitude effects.

The statistical conclusions of the analyses are tempered somewhat by the relatively small sample size consisting of users who were quite familiar with Web searching. Even though participants were recruited from all faculties, it is likely that those with more interest in search engines were willing to take part in the study. Since these experienced searchers found Google-based searching to be quite difficult, inexperienced searchers will presumably struggle even more.

98% (49 out of 50) of the participants preferred the intention-driven search method. Table 12 summarizes the strengths and weaknesses of both search methods as perceived by the participants. Users concluded that IDS was better than Google in case of limited product knowledge or if the target product is not known in advance. The one participant who preferred Google-based searching was on a self-imposed strict time limit so he wanted fewer questions and access to immediate dynamic results throughout the conversation. In other words, he was eager to see immediate results. Users expressed that the familiar Google-based searching would be better for more exploratory searching or when the exact target product is fully identified before the search. The intention-driven approach was preferred for shopping-like situations in which users need help making selections.

	Strengths	Weaknesses
Google-based	Instant results.	Time consuming for complicated needs.
	No limitations on the topics or accepted inputs.	No interactive support to help the user.
	Autocorrect and dynamic suggestions in the search box.	Provides many results which are irrelevant to your global search goal.
	Familiar.	Too easy to get sidetracked during the search.
	Can search images and videos in addition to Web pages.	Difficult to mentally combine partial results throughout the search session.
		Can be difficult to find the correct “unambiguous” search terms to express your needs, especially for searches that involve a lot of constraints or conditions.
		It is easy to overlook important pages.
		Many of the results are redundant.
IDS-based	Easy to use.	No instant results: It takes time to have a conversation.
	Conversational, interactive, and “personal”.	The help text can be long, wordy, or unclear.
	Has a “nice” interface.	At times, it did not ask questions that were expected.
	Makes relevant suggestions and asks relevant questions to guide the search.	Asks too many questions.
	Provides a completion percentage so you know how much time and effort are still needed.	Limited to a few topic areas.
	Very precise, specific results.	Too directed, pushing you to make decisions.
	Gives direct product information summaries in addition to links.	
	Does not require you to be knowledgeable about the subject areas.	

Table 12: Summary of questionnaire responses.

Chapter 9: Conclusions and Future Work

The interactive, goal-driven search mechanism presented in this thesis outlines a new way of searching for information on the Web that overcomes the limitations of current approaches to produce an efficient, natural, and in-depth search experience. The approach tackles users' lack of expert-level knowledge about the topics of interest and query formulations by providing rich dialogue-based assistance. The dialogue itself is generated through a systematic examination of information space regions, where a set of hierarchical dialogue managers operates over specific areas of the information space, probabilistically responding to user inputs and refining the perception of user goals. This expert-based design enables dialogue managers to focus on specific subsets of dialogue knowledge, which allows them to derive and function with smaller action policies that are easier to generate. Furthermore, this enables system knowledge extensibility by adding or removing dialogue experts. The approach exploits external search services to avoid having to gather, process, and maintain its own collections of Web data.

The search approach has been implemented in IDS, a search engine for online gift shopping. IDS employs many state-of-the-art technologies to provide a highly responsive user interface, and robust and efficient application management. This thesis's usability study found that the intention-driven, dialogue-based approach provided by IDS significantly improves the quality of the searching experience. The intention-driven approach was faster to use and took less effort than the currently dominant Google-based search method. In addition, users were significantly more confident in the results they achieved using the intention-driven approach. 98% of the study participants preferred the intention-driven approach for finding products that meet the requirements of specific scenarios. IDS's helpful guidance allowed users to focus on their needs rather than the challenging task of browsing through a plethora of Web pages, executing numerous queries along the way.

In light of the usability study's findings, there are several areas for future experimental work. Since the participants' prior proficiencies were quite high and insignificant with respect to the differences, a wider range of technically non-proficient participants should be studied. Future studies should assess a larger, more diverse population of individuals, including students, seniors, and youths, using disparate topics and scenarios with varying difficulties. The correlation analyses in Chapter 8 were handicapped by a lack of samples in relation to the number of independent variables under consideration. Furthermore, future studies should investigate the needs and behaviours of expert users that are familiar with the topics of interest.

There are many opportunities for improving the implemented search engine. The system can be easily extended by adding support for other domains and contexts. Although the system was designed to handle dialogue-based input, it only processes textual utterances. A speech interface can be constructed to handle spoken dialogue. This would require the introduction of system clarification questions and rules for inserting statements into the knowledge base only when the speech recognition is reasonably high. Inspired by the suggestions of a few users, the system could enable real-time results presentation throughout the dialogue, dynamically showing changes to a results set as the user partakes in the conversation. An even more exciting addition would be to allow the conversation to carry on after the presentation of the results, allowing the user to ask for help in interpreting the results with respect to the context of the preceding dialogue. The POMDPs could be altered to enable better support for mixed-initiative and to allow the user to change his/her goals during the dialogue. Finally, the system could send queries to other search services to provide better coverage of online content.

The intention-driven, dialogue-based approach is an important development for information retrieval and task completion on the Web. As the Web continues to grow in size and importance, so too must our capacity to consume it in new, easier, and more intuitive ways. This type of human-centric searching—of having the system adjust to the

user rather than the other way around—is a clear sign of progress. The searcher, armed with a need, may confidently move forward in a Web of uncertainty.

References

- [1] "About The World Wide Web," 24 01 2001. [Online]. Available: <http://www.w3.org/WWW/>. [Accessed 4 09 2012].
- [2] R. A. Baeza-Yates, P. Milka and H. Zaragoza, "Search, Web 2.0, and the Semantic Web," *IEEE Intelligent Systems*, vol. 23, pp. 80-82, 2008.
- [3] K. P. Purcell, J. Brenner and L. Rainie, "Search Engine Use 2012," 9 3 2012. [Online]. Available: http://pewinternet.org/~media/Files/Reports/2012/PIP_Search_Engine_Use_2012.pdf. [Accessed 17 9 2012].
- [4] C. D. Manning, P. Raghavan and H. Schutze, Introduction to information retrieval, Cambridge University Press, 2008, pp. I-XXI, 1-482.
- [5] H. Zaragoza and M. Najork, "Web Search Relevance Ranking," in *Encyclopedia of Database Systems*, 2009, pp. 3497-3501.
- [6] S. Ceri, "Search Computing," in *SeCO Workshop*, 2009.
- [7] A. Bozzon, D. Braga, M. Brambilla, S. Ceri, F. Corcoglioniti, P. Fraternali and S. Vadacca, "Search computing: multi-domain search on ranked data," in *SIGMOD Conference*, 2011.
- [8] V. Lopez, V. S. Uren, M. Sabou and E. Motta, "Is Question Answering fit for the Semantic Web?: A survey," *Semantic Web*, vol. 2, no. 2, pp. 125-155, 2011.
- [9] D. E. Rose and D. Levinson, "Understanding user goals in web search," in *WWW*, 2004.
- [10] D. Barbieri, A. Bozzon, D. Braga, M. Brambilla, A. Campi, S. Ceri, E. D. Valle, P. Fraternali, M. Grossniklaus, D. Martinenghi, S. Ronchi and M. Tagliasacchi, "Data-driven optimization of search service composition for multidomain queries," in *Workshop on Using Search Engine Technology for Information Management (USETIM) at VLDB 2009*, 2009.
- [11] M. F. McTear, "Spoken dialogue technology: enabling the conversational user interface," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 90-169, 2002.
- [12] J. F. Allen, D. K. Byron, M. Dzikovska, G. Ferguson, L. Galescu and A. Stent, *Towards Conversational Human-Computer Interaction*, 2001.
- [13] N. Roy, J. Pineau and S. Thrun, "Spoken Dialogue Management Using Probabilistic Reasoning," in *ACL*, 2000.
- [14] I. M. O'Neill, P. Hanna, X. Liu and M. F. McTear, "The Queen's Agents: Using Collaborating Object-Based Dialogue Agents in the Queen's Communicator," in *COLING*, 2004.
- [15] K. Komatani, K. Naoyuki, M. Nakano, K. Nakadai, H. Tsujino, T. Ogata and H. Okuno, "Multi-Domain Spoken Dialogue System with Extensibility and Robustness against Speech Recognition Errors," in *7th SIGdial Workshop on Discourse and Dialogue*, Sydney, Australia, 2006.
- [16] B. J. Jansen, A. Spink and T. Saracevic, "Real life, real users, and real needs: a

- study and analysis of user queries on the web," *Inf. Process. Manage.*, vol. 36, no. 2, pp. 207-227, 2000.
- [17] D. Downey, S. Dumais, D. Liebling and E. Horvitz, "Understanding the Relationship between Searchers' Queries and Information Goals," in *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining*, New York, NY, USA, 2008.
 - [18] F. M. Yamin and T. Ramayah, "User web search behavior on query formulation," in *Proc. Int Semantic Technology and Information Retrieval (STAIR) Conf*, 2011.
 - [19] E. Hyvönen, S. Saarela and K. Viljanen, *Ontogator: combining view- and ontology-based search with semantic browsing*, 2003.
 - [20] X. Boyen and D. Koller, "Tractable Inference for Complex Stochastic Processes," in *UAI*, 1998.
 - [21] G. Weikum, "Search for Knowledge," in *SeCO Workshop*, 2009.
 - [22] R. W. a. D. Morris, "Investigating the Querying and Browsing Behavior of Advanced Search Engine Users," in *SIGIR*, Amsterdam, 2007.
 - [23] J. M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," *J. ACM*, vol. 46, no. 5, pp. 604-632, 1999.
 - [24] L. Page, S. Brin, R. Motwani and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, 1999.
 - [25] R. A. Baeza-Yates, L. Calderón-Benavides and C. N. González-Caro, "The Intention Behind Web Queries," in *SPIRE*, 2006.
 - [26] U. Lee, Z. Liu and J. Cho, "Automatic identification of user goals in Web search," in *WWW*, 2005.
 - [27] T. Joachims and F. Radlinski, "Search Engines that Learn from Implicit Feedback," *IEEE Computer*, vol. 40, no. 8, pp. 34-40, 2007.
 - [28] B. Katz, "Annotating the World Wide Web using Natural Language," in *RIAO*, 1997.
 - [29] A. Harth, A. Hogan, R. Delbru, J. Umbrich, S. O'Riain and S. Decker, "SWSE: Answers Before Links!," in *Semantic Web Challenge*, 2007.
 - [30] K. Kaiser and S. Miksch, "Information Extraction: A Survey," 2005.
 - [31] O. E. Erik Selberg, *The MetaCrawler Architecture for Resource Aggregation on the Web*, 1997.
 - [32] R. Navigli, "Word sense disambiguation: A survey," *ACM Comput. Surv.*, vol. 41, no. 2, 2009.
 - [33] G. A. Miller, *WordNet: A Lexical Database for English*, 1995.
 - [34] C. Mangold, "A survey and classification of semantic search approaches," *IJMSO*, vol. 2, no. 1, pp. 23-34, 2007.
 - [35] A. Z. Broder, "A taxonomy of web search," *SIGIR Forum*, vol. 36, no. 2, pp. 3-10, 2002.
 - [36] A. Spink, B. J. Jansen, D. Wolfram and T. Saracevic, "From E-Sex to E-Commerce: Web Search Changes," *IEEE Computer*, vol. 35, no. 3, pp. 107-109, 2002.

- [37] R. W. White, S. T. Dumais and J. Teevan, "Characterizing the influence of domain expertise on web search behavior," in *WSDM*, 2009.
- [38] N. Stojanovic, "On Analysing Query Ambiguity for Query Refinement: The Librarian Agent Approach," in *ER*, 2003.
- [39] A. Spink, J. Bateman and B. J. Jansen, "Searching heterogeneous collections on the Web: behaviour of Excite users," *Inf. Res.*, vol. 4, no. 2, 1998.
- [40] S. C. Peter Pirolli, "Information Foraging," Cognitive and Neural Science and Technology Division, Office of Naval Research, 1999.
- [41] R. Lempel and S. Moran, "Predictive caching and prefetching of query results in search engines," in *WWW*, 2003.
- [42] L. Hirschman and R. J. Gaizauskas, "Natural language question answering: the view from here," *Natural Language Engineering*, vol. 7, no. 4, pp. 275-300, 2001.
- [43] R. K. Srihari and W. Li, "Information Extraction Supported Question Answering," in *TREC*, 1999.
- [44] A. Ittycheriah, M. Franz, W. jing Zhu, A. Ratnaparkhi and R. J. Mammone, *IBM's Statistical Question Answering System*, 2000.
- [45] S. Harabagiu, D. Moldovan, M. Pasca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Girju, V. Rus and P. Morarescu, *FALCON: Boosting Knowledge for Answer Engines*, 2000.
- [46] A. B. Esther Kaufmann, "How Useful are Natural Language Interfaces to the Semantic Web for Casual End-users?," in *6th International Semantic Web Conference*, Busan, Korea, 2007.
- [47] A. Copestake and K. S. Jones, "Natural Language Interfaces to Databases," in *Knowledge Engineering Review, Special Issue on the Applications of Natural Language Processing Techniques*, 1989.
- [48] A. S. Pollitt, *The key role of classification and indexing in view-based searching*, 1998.
- [49] J. English, M. Hearst, R. Sinha, K. Swearingen and K.-P. Yee, *Flexible Search and Navigation Using Faceted Metadata*, 2002.
- [50] K.-P. Yee, K. Swearingen, K. Li and M. Hearst, *Faceted Metadata for Image Search and Browsing*, 2003.
- [51] E. J. Glover, S. Lawrence, M. D. Gordon, W. P. Birmingham and C. L. Giles, "Web Search - Your Way," *Commun. ACM*, vol. 44, no. 12, pp. 97-102, 2001.
- [52] L. P. Sergey Brin, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*, 1998.
- [53] R. A. Baeza-Yates and P. Raghavan, "Next Generation Web Search," in *SeCO Workshop*, 2009.
- [54] S. Lawrence, "Context in Web Search," *IEEE Date Engineering Bulletin*, vol. 23, no. 3, pp. 25-32, 2000.
- [55] D. R. Traum and S. Larsson, "The Information State Approach to Dialogue Management," [Online]. Available: <http://people.ict.usc.edu/~traum/Papers/traumlarsson.pdf>. [Accessed 3 7 2012].

- [56] C. Kamm, A. Kamm, M. Walker and D. Litman, *Evaluating Spoken Language Systems*.
- [57] M. F. McTear, "Modelling spoken dialogues with state transition diagrams: experiences with the CSLU toolkit," in *ICSLP*, 1998.
- [58] H. Aust, M. Oerder, F. Seide and V. Steinbiss, "The Philips automatic train timetable information system," *Speech Communication*, vol. 17, no. 3-4, pp. 249-262, 1995.
- [59] T. H. Bui, "Multimodal Dialogue Management - State of the art," Centre for Telematics and Information Technology, University of Twente, Enschede, Netherlands, 2006.
- [60] J. Allen, "Computational Models of Discourse," M. Brady and R. Berwick, Eds., MIT Press, 1983, pp. 107-166.
- [61] J. R. Searle, *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press, 1970.
- [62] P. Cohen, "Survey of the State of the Art in Human Language Technology," H. Uzokoreit, Ed., Cambridge University Press, 1997, pp. 204-209.
- [63] G. Ferguson, J. Allen and B. Miller, *TRAINS-95: Towards a mixed-initiative planning assistant*, 1996.
- [64] J. F. A. George Ferguson, *TRIPS: An Integrated Intelligent Problem-Solving Assistant*, 1998.
- [65] "TRINDI," [Online]. Available: <http://www.ling.gu.se/projekt/trindi/>. [Accessed 01 09 2012].
- [66] S. Larsson, P. Ljunglöf, R. Cooper, R. Cooper, E. Engdahl, E. Engdahl and S. Ericsson, *GoDiS - An Accommodating Dialogue System*, 2000.
- [67] M. Johnston, S. Bangalore, G. Vasireddy, A. Stent, P. Ehlen, M. A. Walker, S. Whittaker and P. Maloor, "MATCH: An Architecture for Multimodal Dialogue Systems," in *ACL*, 2002.
- [68] T. H. Bui, B. van Schooten and D. Hofs, *Practical Dialogue Manager Development using POMDPs*, 2007.
- [69] R. P. Esther Levin, *A Stochastic Model of Computer-Human Interaction for Learning Dialogue Strategies*, 1997.
- [70] S. P. Singh, D. J. Litman, M. J. Kearns and M. A. Walker, "Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System," *J. Artif. Intell. Res. (JAIR)*, vol. 16, pp. 105-133, 2002.
- [71] B. Zhang, Q. Cai, J. Mao and B. Guo, "Planning and Acting under Uncertainty: A New Model for Spoken Dialogue System," in *UAI*, 2001.
- [72] J. D. Williams, P. Poupart and S. Young, *Factored partially observable Markov decision processes for dialogue management*, 2005.
- [73] S. M. LaValle, *Planning Algorithms*, Cambridge University Press, 2006.
- [74] S. Young, "Using POMDPs for dialog management," in *IEEE/ACL Workshop on Spoken Language Technology*, Aruba, 2006.
- [75] C. Wai, H. M. Meng and R. Pieraccini, "Scalability and Portability of a Belief

- Network-based Dialog Model for Different Application Domains," in *Human Language Technology Conference*, Toulouse, France, 2001.
- [76] T. P. Eric Horvitz, "A Computational Architecture for Conversation," in *User Modeling*, Banff, Canada, 1999.
 - [77] T. Cassandra, *Background on Solving POMDPs*, 2009.
 - [78] S. Young, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson and K. Yu, *The Hidden Information State model: A practical framework for*, 2009.
 - [79] D. J. Litman, M. S. Kearns, S. Singh and M. A. Walker, *Automatic Optimization of Dialogue Management*, 2000.
 - [80] E. Sondik, "The Optimal Control of Partially Observable Markov Decision Processes," 1971.
 - [81] J. D. Williams and S. Young, "Scaling up POMDPs for Dialog Management: The ``Summary POMDP" Method," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, 2005.
 - [82] S. Keizer, M. Gasic, F. Mairesse, B. Thomson, K. Yu and S. Young, "Modelling user behaviour in the HIS-POMDP dialogue manager," in *SLT*, 2008.
 - [83] T. H. Bui, J. Zwiers, M. Poel and A. Nijholt, "Affective Dialogue Management Using Factored POMDPs," in *Interactive Collaborative Information Systems*, 2010, pp. 207-236.
 - [84] L. P. Kaelbling, M. L. Littman and A. R. Cassandra, "Planning and Acting in Partially Observable Stochastic Domains," *Artif. Intell.*, vol. 101, no. 1-2, pp. 99-134, 1998.
 - [85] J. Pineau, G. J. Gordon and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *IJCAI*, 2003.
 - [86] P. Poupart and C. Boutilier, "VDCBPI: an Approximate Scalable Algorithm for Large POMDPs," in *NIPS*, 2004.
 - [87] B. Thomson and S. Young, "Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems," *Computer Speech & Language*, vol. 24, no. 4, pp. 562-588, 2010.
 - [88] S. Young, J. Schatzmann, K. Weilhammer and H. Ye, "The hidden information state approach to dialogue management," in *ICASSP 2007*, Honolulu, Hawaii, 2007.
 - [89] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. J. Hazen and L. Hetherington, "Jupiter: A Telephone-Based Conversational Interface for Weather Information," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 85-96, 2000.
 - [90] Y. Takebayashi, H. Tsubo, Y. Sadamoto, H. Hashimoto and H. Shinichi, "A real-time speech dialogue system using spontaneous speech understanding," in *ICSLP*, 1992.
 - [91] R. San-Segundo, B. Pellom, W. Ward and J. M. Pardo, "Confidence measures for dialogue management in the CU Communicator system," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing ICASSP '00*, 2000.
 - [92] L. Lamel, S. Rosset, J. L. Gauvain, S. Bennacef, M. Garnier-Rizet and B. Prouts, "The LIMSI ARISE System," *Speech Communication*, vol. 31, no. 4, pp. 339-354,

2000.

- [93] A. Raux and M. Eskenazi, "Non-Native Users in the Let's Go!! Spoken Dialogue System: Dealing with Linguistic Mismatch," in *HLT-NAACL*, Boston, Massachusetts, USA, 2004.
- [94] W. Minker, U. Haiber, P. Heisterkamp and S. Scheible, "The SENECA spoken language dialogue system," *Speech Communication*, vol. 43, no. 1-2, pp. 89-102, 2004.
- [95] N. Kawaguchi, S. Matsubara, K. Toyama and Y. Inagaki, *An Open Architecture for Multi-Domain Multi-Modal Conversational Systems*, 1999.
- [96] B. shen Lin, H. min Wang and L.-S. Lee, "A Distributed Architecture For Cooperative Spoken Dialogue Agents With Coherent Dialogue State And History," in *ASRU*, Colorado, USA, 1999.
- [97] B. Pakucs, *Towards Dynamic Multi-Domain Dialogue Processing*, 2003.
- [98] M. Turunen and J. Hakulinen, "Agent-Based Adaptive Interaction and Dialogue Management Architecture for Speech Applications," in *TSD*, 2001.
- [99] M. Nakano, Y. Hasegawa, K. Funakoshi, J. Takeuchi, T. Torii, K. Nakadai, N. Kanda, K. Komatani, H. G. Okuno and H. Tsujino, "A multi-expert model for dialogue and behavior control of conversational robots and agents," *Knowl.-Based Syst.*, vol. 24, no. 2, pp. 248-256, 2011.
- [100] I. M. O'Neill, P. Hanna, X. Liu and M. F. McTear, "The queen's communicator: an object-oriented dialogue manager," in *INTERSPEECH*, 2003.
- [101] C. Lee, S. Jung, S. Kim and G. G. Lee, "Example-based dialog modeling for practical multi-domain dialog system," *Speech Communication*, vol. 51, no. 5, pp. 466-484, 2009.
- [102] P. Madeira, M. Moura and N. J. Mamede, "Star - A Multiple Domain Dialog Manager," in *ICEIS (2)*, 2003.
- [103] I. R. Lane and T. Kawahara, "Incorporating Dialogue Context and Topic Clustering in Out-of-Domain Detection," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '05)*, 2005.
- [104] V. Lopez, E. Motta and V. Uren, "Poweraqua: Fishing the semantic web," in *ESWC*, Budva, Montenegro, 2006.
- [105] "PowerAqua," Knowledge Media Institute, [Online]. Available: <http://technologies.kmi.open.ac.uk/poweraqua/demo.html>. [Accessed 3 1 2012].
- [106] V. Lopez, M. Fernández, E. Motta and N. Stieler, "PowerAqua: supporting users in querying and exploring the Semantic Web content," 2010.
- [107] A. Campi, S. Ceri, G. Gottlob, A. Maesani and S. Ronchi, "Service Marts," in *SeCO Workshop*, 2009.
- [108] A. Bozzon, M. Brambilla, S. Ceri, P. Fraternali and I. Manolescu, "Liquid Queries and Liquid Results in Search Computing," in *SeCO Workshop*, 2009.
- [109] D. Braga, S. Ceri, F. Corcoglioniti and M. Grossniklaus, "Panta Rhei: Flexible Execution Engine for Search Computing Queries," in *SeCO Workshop*, 2009.
- [110] A. Bozzon, M. Brambilla and S. Ceri, "Answering search queries with

- CrowdSearcher," in *WWW*, 2012.
- [111] M. Masseroli, G. Ghisalberti and S. Ceri, "Bio-Search Computing: Integration and global ranking of bioinformatics search results," *J. Integrative Bioinformatics*, vol. 8, no. 2, 2011.
 - [112] K. Trentelman, "Survey of Knowledge Representation and Reasoning Systems," Defence Science and Technology Organisation, Edinburgh, Australia, 2009.
 - [113] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2 ed., Pearson Education, 2003.
 - [114] K. B. Laskey, "MEBN: A Language for First-Order Bayesian Knowledge Bases," *Artificial Intelligence*, vol. 172, no. 2-3, pp. 140-178, 2008.
 - [115] M. Minsky, "The Psychology of Computer Vision," McGraw-Hill, 1975, pp. 211-277.
 - [116] R. Davis, H. Shrobe and P. Szolowits, "What is a Knowledge Representation?," *AI Magazine*, vol. 14, no. 1, pp. 17-33, 1993.
 - [117] V. Devedzic, "Understanding ontological engineering," *Commun. ACM*, vol. 45, no. 4, pp. 136-144, 2002.
 - [118] B. C. Grau and O. Kautz, "Modular Ontology Languages Revisited," in *SWeCKa 2007: Proc. of the IJCAI-2007 Workshop on Semantic Web for Collaborative Knowledge Acquisition*, Hyderabad, India, January 7, 2007, 2007.
 - [119] A. Borgida and L. Serafini, "Distributed Description Logics: Directed Domain Correspondences in Federated Information Sources," in *CoopIS/DOA/ODBASE*, 2002.
 - [120] J. Bao, D. Caragea and V. G. Honavar, "Towards collaborative environments for ontology construction and sharing," in *International Symposium on Collaborative Technologies and Systems*, Las Vegas, USA, 2006.
 - [121] A. Zimmermann, *Integrated Distributed Description Logics (extended abstract)*, 2007.
 - [122] C. Parent and S. Spaccapietra, "An Overview of Modularity," in *Modular Ontologies*, 2009, pp. 5-23.
 - [123] S. Carberry, *Plan Recognition in Natural Language Dialogue*, MIT Press, 1990.
 - [124] H. A. Kautz, H. A. Kautz, R. N. Pelavin, J. D. Tenenbergs and M. Kaufmann, *A formal theory of plan recognition and its implementation*, 1991.
 - [125] E. Charniak and R. P. Goldman, "A Bayesian Model of Plan Recognition," *Artif. Intell.*, vol. 64, no. 1, pp. 53-79, 1993.
 - [126] J. Hong, "Goal Recognition through Goal Graph Analysis," *JAIR*, vol. 15, pp. 1-30, 2001.
 - [127] N. Lesh and O. Etzioni, "A Sound and Fast Goal Recognizer," in *IJCAI*, Barcelona, Spain, 1995.
 - [128] T. A. Han and L. M. Pereira, *State-of-the-Art of Intention Recognition and its use in Decision Making*, 2002.
 - [129] P. Roy, B. Bouchard, A. Bouzouane and S. Giroux, "A Hybrid Plan Recognition Model for Alzheimer's Patients: Interleaved-Erroneous Dilemma," in *Proc.*

- IEEE/WIC/ACM Int. Conf. Intelligent Agent Technology IAT '07*, 2007.
- [130] N. Blaylock, "Retroactive Recognition of Interleaved Plans for Natural Language Dialogue," University of Rochester, Rochester, NY, USA, 2001.
 - [131] F. Bacchus, A. J. Grove, J. Y. Halpern and D. Koller, "From Statistical Knowledge Bases to Degrees of Belief," *Artif. Intell.*, vol. 87, no. 1-2, pp. 75-143, 1996.
 - [132] J. Y. Halpern, "An Analysis of First-Order Logics of Probability," *Artificial Intelligence*, vol. 46, no. 3, pp. 311-350, 1990.
 - [133] D. Heckerman and E. H. Shortliffe, "From certainty factors to belief networks," *Artificial Intelligence in Medicine*, vol. 4, no. 1, pp. 35-52, 1992.
 - [134] U. Kjærulff, "A Computational Scheme For Reasoning In Dynamic Probabilistic Networks," in *UAI*, Stanford, CA, USA, 1992.
 - [135] L. Getoor, N. Friedman, D. Koller, A. Pfeffer and B. Taskar, *5 Probabilistic Relational Models*.
 - [136] D. Koller and A. Pfeffer, "Object-Oriented Bayesian Networks," in *UAI*, 1997.
 - [137] G. Shafer, "Dempster-Shafer Theory," [Online]. Available: <http://www.glennshafer.com/assets/downloads/articles/article48.pdf>. [Accessed 10 08 2012].
 - [138] E. Shortliffe, *Computer Based Medical Consultation: MYCIN*, Elsevier Science Publishers, 1976.
 - [139] D. W. Albrecht, I. Zukerman and A. E. Nicholson, "Bayesian Models for Keyhole Plan Recognition in an Adventure Game," *User Modeling and User-Adapted Interaction*, vol. 8, no. 1-2, pp. 5-47, 1998.
 - [140] D. V. Pynadath and M. P. Wellman, "Probabilistic State-Dependent Grammars for Plan Recognition," in *UAI*, 2000.
 - [141] M. Bauer, *A Dempster-Shafer Approach to Modeling Agent Preferences for Plan Recognition*, 1995.
 - [142] L. Serafini and M. Homola, "Modular Knowledge Representation and Reasoning in the Semantic Web," in *Semantic Web Information Management*, 2009, pp. 147-181.
 - [143] C. Barry, *The Identification of User Criteria of Relevance and Document Characteristics: Beyond the Topical*, PhD thesis, Syracuse University, 1993.
 - [144] J. Pineau and S. Thrun, *Hierarchical POMDP Decomposition for A Conversational Robot*, 2001.
 - [145] J. Pineau, N. Roy and S. Thrun, *A Hierarchical Approach to POMDP Planning and Execution*, 2001.
 - [146] G. Theodorou, K. Rohanimanesh and S. Mahadevan, "Learning Hierarchical Partially Observable Markov Decision Process Models for Robot Navigation," in *ICRA*, 2001.
 - [147] G. Theodorou, K. P. Murphy and L. P. Kaelbling, "Representing Hierarchical POMDPs as DBNs for Multi-scale Robot Localization," in *ICRA*, 2004.
 - [148] "OWL 2 Web Ontology Language Profiles," 27 10 2009. [Online]. Available: <http://www.w3.org/TR/owl2-profiles/>. [Accessed 1 9 2012].

- [149] T. H. Bui, "Toward Affective Dialogue Management using Partially Observable Markov Decision Processes," University of Twente, Enschede, Netherlands, 2008.
- [150] "Why MySQL," 2012. [Online]. Available: <http://www.mysql.com/why-mysql/>. [Accessed 1 9 2012].
- [151] G. Antoniou and F. van Harmelen, A semantic web primer, MIT Press, 2004, pp. I-XX, 1-238.
- [152] W3C, "Resource Description Framework (RDF)," 1 4 2004. [Online]. Available: <http://www.w3.org/RDF/>. [Accessed 1 9 2012].
- [153] "SPIN SPARQL Inferencing Notation," [Online]. Available: <http://spinrdf.org/>. [Accessed 4 1 2012].
- [154] "SPARQL Query Language for RDF," 15 1 2008. [Online]. Available: <http://www.w3.org/TR/rdf-sparql-query/>. [Accessed 4 1 2012].
- [155] B. v. Schooten, T. H. Bui and D. Hofs, "POMDP Toolkit," [Online]. Available: <http://wwwhome.cs.utwente.nl/~hofs/pomdp/>. [Accessed 4 1 2012].
- [156] T. Smith, "ZMDP Software for POMDP and MDP Planning," [Online]. Available: <http://www.cs.cmu.edu/~trey/zmdp/>. [Accessed 4 1 2012].
- [157] "Apache Jena," 2012. [Online]. Available: <http://jena.apache.org/>. [Accessed 4 1 2012].
- [158] "Introducing JSON," [Online]. Available: <http://www.json.org/>. [Accessed 4 1 2012].
- [159] "google-gson," Google, [Online]. Available: <http://code.google.com/p/google-gson/>. [Accessed 5 4 2012].
- [160] "jQuery," The jQuery Foundation, 2012. [Online]. Available: <http://jquery.com/>. [Accessed 3 3 2012].
- [161] "jQuery user interface," The jQuery Foundation, 2012. [Online]. Available: <http://jqueryui.com/>. [Accessed 3 3 2012].
- [162] P. Rochala, "jQuery slimScroll," 16 10 2011. [Online]. Available: <http://rocha.la/jquery-slimScroll>. [Accessed 1 5 2012].
- [163] "Search API for Shopping," Google, 23 3 2012. [Online]. Available: <https://developers.google.com/shopping-search/>. [Accessed 3 3 2012].
- [164] "Eclipse IDE for Java Developers," The Eclipse Foundation, 2012. [Online]. Available: <http://www.eclipse.org/downloads/moreinfo/java.php>. [Accessed 1 9 2012].
- [165] "TopBraid Composer," TopQuadrant, 2012. [Online]. Available: http://www.topquadrant.com/products/TB_Composer.html. [Accessed 2 1 2012].
- [166] "Google Chrome," Google, 2012. [Online]. Available: https://www.google.com/intl/en_uk/chrome/browser/. [Accessed 1 9 2012].
- [167] "gimp 2.8," The GIMP Team, 2012. [Online]. Available: <http://www.gimp.org/>. [Accessed 1 9 2012].
- [168] "Apache Tomcat," The Apache Software Foundation, 6 9 2012. [Online]. Available: <http://tomcat.apache.org/>. [Accessed 1 9 2012].
- [169] "Pellet: OWL 2 Reasoner for Java," Clark & Parsia, 2011. [Online]. Available:

- <http://clarkparsia.com/pellet/>. [Accessed 3 1 2012].
- [170] "swoop - Semantic Web Ontology Editor," 2007. [Online]. Available: <http://code.google.com/p/swoop/>. [Accessed 2 1 2012].
 - [171] "welcome to protege," Stanford Center for Biomedical Informatics Research, 2012. [Online]. Available: <http://protege.stanford.edu/>. [Accessed 2 1 2012].
 - [172] "JavaServer Faces Technology," Oracle Corporation, 2012. [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html>. [Accessed 1 4 2012].
 - [173] "Enterprise JavaBeans Technology," Oracle Corporation, 2012. [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>. [Accessed 1 4 2012].
 - [174] E. Ort, "Introducing the Java EE 6 Platform: Part 1," Oracle Corporation, 2009. [Online]. Available: <http://www.oracle.com/technetwork/articles/javaee/javaee6overview-141808.html>. [Accessed 1 9 2012].
 - [175] "JavaServer Pages Technology," Oracle Corporation, 2012. [Online]. Available: <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>. [Accessed 1 9 2012].
 - [176] I. Horrocks, P.-S. Peter, H. Boley, S. Tabet, B. Grosz and M. Dean, "SWRL: A Semantic Web Rule Language," W3C, 21 5 2004. [Online]. Available: <http://www.w3.org/Submission/SWRL/>. [Accessed 1 2 2012].
 - [177] "NetBeans," Oracle Corporation, 2012. [Online]. Available: <http://netbeans.org/>. [Accessed 2 1 2012].
 - [178] *Common industry format for usability test reports (ANSI-NCITS 354-2001)*, 2001.
 - [179] J. R. Lewis, "Usability Testing," in *Handbook of Human Factors and Ergonomics, Third Edition*, John Wiley & Sons, 2006, pp. 1275-1316.
 - [180] J. S. Dumas and J. C. Redish, *A Practical Guide to Usability Testing*, Intellect Ltd., 1999.
 - [181] J. Schatzmann, K. Weilhammer, M. N. Stuttle and S. Young, "A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies," *Knowledge Eng. Review*, vol. 21, no. 2, pp. 97-126, 2006.
 - [182] A. Aula, N. Jhaveri and M. K{\a}ki, "Information search and re-access strategies of experienced web users," in *WWW*, 2005.
 - [183] A. J. Schwab, "Solving One-way ANOVA Problems as a General Linear Model," 2007. [Online]. Available: http://www.utexas.edu/courses/schwab/sw388r7_spring_2007/SolvingProblemsInSPSS/SolvingOne-wayANOVAProblemsasGeneralLinearModel.pdf. [Accessed 1 09 2012].
 - [184] S. Young, *Effective Handling of Dialogue State in the Hidden Information State POMDP-based Dialogue Manager*.
 - [185] H. Stuckenschmidt, "Implementing Modular Ontologies with Distributed Description Logics," in *WoMO*, 2006.
 - [186] C. Ghidini, L. Serafini and S. Tessaris, "Complexity of Reasoning With Expressive

- Ontology Mappings," in *FOIS*, 2008.
- [187] J. Bao, D. Caragea and V. Honavar, "On the Semantics of Linking and Importing in Modular Ontologies," in *International Semantic Web Conference*, 2006.
- [188] S. Oviatt, "Multimodal Interfaces," in *The Human-Computer Interaction Handbook*, J. Jacko and A. Sears, Eds., New York, Lawrence Erlbaum Associates, 2002, pp. 413-432.
- [189] C. Holscher and G. Strube, "Web search behavior of Internet experts and newbies," *Computer Networks*, vol. 33, no. 1-6, pp. 337-346, 2000.

Appendices

Appendix A: Consent Form



CONSENT TO PARTICIPATE IN RESEARCH

Title of Study: Intention-Driven, Dialogue-Based Search Engine for Online Gift Shopping

You are asked to participate in a research study conducted by Brian Small from the Department of Computer Science at the University of Windsor that will contribute to his master's thesis research.

If you have any questions or concerns about the research, please feel free to contact Brian Small (phone: 519-253-3000 ext. 4407, email: smalld@uwindsor.ca) or Dr. Xiaobu Yuan (phone: 519-253-3000 ext. 3790, email: xyuan@uwindsor.ca).

PURPOSE OF THE STUDY

The study will evaluate the usability of both the dominant keyword-based search engine (Google) and this research work's intention-driven, dialogue-based search system for users with varying levels of familiarity with the topics involved in their searches. The study will investigate the practicality and applicability of the intention-driven method for searches that involve multiple topic considerations.

PROCEDURES

If you volunteer to participate in this study, we would ask you to do the following things:

1. Listen to a brief introduction on users' Web search techniques and on the role of multiple topics on users' search engine goals. The search task often involves terminology from many different areas of knowledge. Users are often unfamiliar with the knowledge needed to complete their search tasks and search engines have a hard time incorporating user-provided information about multiple topics. (Duration: 5 minutes)
2. Use Google to search for a product that satisfies the needs outlined in a fictional scenario. (Duration: Estimated 10-20 minutes)
3. Use the system generated by the researcher to find a product that satisfies the needs outlined in another fictional scenario. (Duration: Estimated 5-10 minutes)
4. Answer questions about the usability of the two systems. (Duration: Estimated 5 minutes)

In total, your participation should take about 30-40 minutes. Each participant will perform the task one at a time in a noise-free computer lab located on the third floor of Erie Hall.

POTENTIAL RISKS AND DISCOMFORTS

This study involves minimal risks to the participant. You will be given two impersonal, fictional scenarios that describe a need for a particular product. Using a computer workstation, you will conduct two search sessions under observation from the researcher. The researcher will gather data on the usability of the systems—not on you or your ability to perform the search tasks. Please feel free to adjust your ergonomic setup to ensure physical comfort throughout the tasks.

POTENTIAL BENEFITS TO SUBJECTS AND/OR TO SOCIETY

You will benefit by gaining an understanding of the multi-topic nature of online information retrieval tasks and an appreciation for the limitations of existing systems. This research work will validate a new approach to searching that involves the interactive, knowledge-intensive identification and integration of user goals.

PAYMENT FOR PARTICIPATION

Ten \$25 cash prizes will be awarded at random to participants.

CONFIDENTIALITY

Any information that is obtained in connection with this study and that can be identified with you will remain confidential and will be disclosed only with your permission.

You will be required to provide your name and email address to enter into the prize draw. This information will be written on a prize ballot and will be stored separately from the collected experiment data. The data collected during the course of the experiment will be linked to you only through a unique numerical identifier. All data will be kept in secure locations for at most two months after the date of your participation.

PARTICIPATION AND WITHDRAWAL

You can choose whether to be in this study or not. If you volunteer to be in this study, you may withdraw at any time during or within three days of your participation without consequences of any kind. Notification of your withdrawal must be through email request to the principal investigator. The investigator may withdraw you from this research if circumstances arise which warrant doing so.

You will be asked to withdraw involuntarily if you do not complete the questionnaire. The data collected from any participant who withdraws will be permanently deleted.

FEEDBACK OF THE RESULTS OF THIS STUDY TO THE SUBJECTS

The study's findings will be made available online at the specified Web address. Participants are also welcome to attend the researcher's thesis defence whose date will be specified on the Web page.

Web address: <http://cs.uwindsor.ca/~smalld>

Date when results are available: September 18, 2012

SUBSEQUENT USE OF DATA

This data may be used in subsequent studies.

RIGHTS OF RESEARCH SUBJECTS

You may withdraw your consent at any time during or within three days of your participation and discontinue participation without penalty. If you have questions regarding your rights as a research subject, contact: Research Ethics Coordinator, University of Windsor, Windsor, Ontario N9B 3P4; Telephone: 519-253-3000, ext. 3948; e-mail: ethics@uwindsor.ca

SIGNATURE OF RESEARCH SUBJECT/LEGAL REPRESENTATIVE

I understand the information provided for the study “Intention-Driven, Dialogue-Based Search Engine for Online Gift Shopping” as described herein. My questions have been answered to my satisfaction, and I agree to participate in this study. I have been given a copy of this form.

Name of Subject

Signature of Subject

Date

SIGNATURE OF INVESTIGATOR

These are the terms under which I will conduct research.

Signature of Investigator

Date

Appendix B: Recruitment Poster



INTENTION-DRIVEN, DIALOGUE-BASED SEARCH ENGINE FOR ONLINE GIFT SHOPPING

STUDY FOCUS

We are interested in testing a new interactive, goal-driven approach to online information retrieval.

REQUIREMENTS

In this study you will be asked to use a popular search engine (Google) as well as a system generated by the researcher to find products that meet the needs outlined in fictional scenarios. You will complete a questionnaire to evaluate the systems. Your participation will require about 30-40 minutes.

COMPENSATION

Ten \$25 cash prizes by random draw.

TO PARTICIPATE

Please contact the researcher to set up a meeting date and time.

QUESTIONS

Please feel free to contact the researcher.

CONTACT INFO

Researcher (Primary Investigator): Brian Small; M.Sc. Computer Science candidate; smallb@uwindsor.ca

This research has received clearance from the University of Windsor Research Ethics Board.

Appendix C: Questionnaire



QUESTIONNAIRE

Participant ID: _____

Date: _____

EXPERIENCE

1. How many times per day do you use a search engine? _____
2. On a scale from 1 to 10 (1 = low, 10 = high), rate your level of proficiency using Google or other similar search engines (circle one).
1 2 3 4 5 6 7 8 9 10

CASE STUDY

1. Select the area that you are least familiar with (circle one).
a) Home improvement
b) Guitars
2. Rate your level of knowledge about the subject area you chose (1 = low, 10 = high).
1 2 3 4 5 6 7 8 9 10

Complete Part ____ BEFORE Part ____ (chosen by the researcher).

Part 1: Google Search

Read the researcher-provided scenario that corresponds with the subject area you chose above. Using as many Google searches as you want, find a product that meets the scenario's requirements. Then, answer the following questions:

1. On a scale from 1 to 10 (1 = low, 10 = high), how much effort did the task require from you?
1 2 3 4 5 6 7 8 9 10
2. On a scale from 1 to 10 (1 = low, 10 = high), how confident are you that the product fully meets the requirements outlined in the scenario?
1 2 3 4 5 6 7 8 9 10

Part 2: Intention-Driven Search

Read the researcher-provided scenario that corresponds with the subject area you chose above. Engage with the intention-driven system to find a product that meets the requirements outlined in the scenario. Then, answer the following questions:

1. On a scale from 1 to 10 (1 = low, 10 = high), how much effort did the task require from you?
1 2 3 4 5 6 7 8 9 10
2. On a scale from 1 to 10 (1 = low, 10 = high), how natural was the conversation?
1 2 3 4 5 6 7 8 9 10
3. On a scale from 1 to 10 (1 = low, 10 = high), how confident are you that the product fully meets the requirements outlined in the scenario?
1 2 3 4 5 6 7 8 9 10

WRAP-UP

1. Which system was easier to use and why?

2. Identify the strengths and weaknesses of both systems. Which system did you prefer to use and why?

Appendix D: Scenarios

Guitar Scenarios

Scenario A

Don is an advanced electric guitar player. He wants a sunburst-coloured electric guitar that provides a “chunky” or “fat” tone. The guitar must have 21 frets and a vibrato mechanism that permits extreme pitch variations while keeping the guitar in tune. Don was not impressed with his last guitar, a Japanese Ibanez, so he would prefer an American-made guitar. Heavily influenced by the blues, one of Don’s favourite players is Stevie Ray Vaughan. You want to buy Don a new right-handed guitar made by the company known for its close relationship with Stevie Ray Vaughan. You have \$800 to spend.

Summary:

- Electric guitar
- Right-handed
- New
- Sunburst-coloured
- Has a “fat” sound
- 21 frets
- Made by an American company
- Made by the company strongly linked with Stevie Ray Vaughan
- For blues-style playing
- Needs to have a vibrato/tremolo system that doesn’t detune the guitar and that permits extreme pitch changes
- Up to \$800

Scenario B

Inspired by her love of Classical music, Alice decides that she wants to learn to play the acoustic guitar. She is looking for a solid top acoustic guitar that provides a warm, dark tone suitable for fingerstyle playing. Since Alice is left-handed, she needs a left-handed guitar. As she wants flexibility in her note choices, she wants as much access to the upper frets as possible. You want to buy a new guitar for Alice for under \$1200.

Summary:

- Acoustic guitar
- Left-handed
- New
- For classical (fingerstyle) music

- Has a “dark” tone
- As much access to the upper frets of the guitar as possible
- Can spend up to \$1200

Home Improvement Scenarios

Scenario A

Don would like to use his detached garage as a workspace for tinkering with home improvement projects. Unfortunately, his garage is unbearably hot and stifling in the late summer months. To solve this problem, Don would like to install a product that adds onto his garage roof to allow hot air to escape and fresh air to enter. The product’s dimensions should be 18” by 22” and it should be made from low maintenance materials. To match the rest of the garage, the base of the product should be white. You want to buy Don a product made by a company that has been in business for more than five years. The product must be new and no more than \$1200.

Summary:

- Roof addition for garage
- New
- For ventilation only
- 18” by 22” measurements
- Made from low maintenance materials
- Made by an experienced company (5 years or more)
- White-coloured
- \$1200 to spend

Scenario B

Alice’s basement floods frequently. Her basement’s walls are properly waterproofed, her eavestroughs are correctly set up, and her house is properly graded to push water away from it. The basement is, however, below the water table level. The product should come from an established manufacturer (more than five years of experience). Since frequent power outages occur in her neighbourhood, Alice would like a water pressure-powered device. You want to help Alice by purchasing a new product that will fix her problem for no more than \$300.

Summary:

- Basement flooding
- Diagnostic information:
 - The house is properly graded
 - The eavestroughs are set up properly and are not clogged

- The basement walls are waterproofed
 - The basement is below the water table level
- Made by a company with more than 5 years of experience
- Powered by water pressure
- New
- Costs up to \$300

Appendix E: Experiment Data

	Descriptive Statistics								
	N	Minimum	Maximum	Mean	Std. Deviation	Skewness		Kurtosis	
	Statistic	Statistic	Statistic	Statistic	Statistic	Statistic	Std. Error	Statistic	Std. Error
Answer Quality	50	.00	1.00	.6620	.28536	-1.317	.337	.879	.662
Google Confidence - IDS Confidence	50	-8.00	3.00	-3.0600	2.50233	-.483	.337	-.059	.662
Google Effort - IDS Effort	50	-4.00	8.00	4.6600	2.22793	-1.317	.337	3.256	.662
Google Time - IDS Time	50	-3.00	43.00	13.7000	12.73954	.756	.337	-.444	.662
Failures	50	0	5	1.30	1.111	.855	.337	1.009	.662
Google Confidence	50	1	10	6.04	2.285	-.553	.337	-.369	.662
Google Effort	50	4	10	8.18	1.466	-.891	.337	.945	.662
Google Time (min)	50	4	49	19.80	12.854	.702	.337	-.624	.662
IDS Confidence	50	6	10	9.10	1.015	-1.304	.337	1.795	.662
IDS Effort	50	1	8	3.52	1.515	.730	.337	.709	.662
IDS Naturalness	50	4	10	8.04	1.498	-1.020	.337	.780	.662
IDS Time (min)	50	3	14	6.10	2.197	1.203	.337	1.959	.662
Level of Knowledge	50	1	6	2.68	1.406	.693	.337	-.278	.662
Page Views	50	2	104	21.34	16.777	2.879	.337	11.883	.662
Proficiency	50	4	10	7.78	1.282	-.477	.337	.747	.662
Questions	50	7	17	12.30	2.435	-.330	.337	-.453	.662
Results Pages	50	3	57	21.30	14.440	1.050	.337	.230	.662
Searches Per Day	48	2	100	30.82	30.079	1.473	.343	1.060	.674
Sent Queries	50	3	55	19.08	11.726	.962	.337	.631	.662
Valid N (listwise)	48								

Order (1 = Google, 2 = IDS)	Area	Google Scenario	N
(1, 2)	Guitars	A	10
		B	9
		Total	19
	HI	A	3
		B	3
		Total	6
	Total	A	13
		B	12
		Total	25
(2, 1)	Guitars	A	9
		B	10
		Total	19
	HI	A	3
		B	3
		Total	6
	Total	A	12
		B	13
		Total	25
Total	Guitars	A	19
		B	19
		Total	38
	HI	A	6
		B	6
		Total	12
	Total	A	25
		B	25
		Total	50

Number of participants with several factors.

Vita Auctoris

Brian Small was born in 1988 in Windsor, Ontario. He graduated from Sandwich Secondary High School in 2006 and achieved a Bachelor of Computer Science (Honours) degree from the University of Windsor in 2010. He is currently a candidate for the degree of Master of Science at the University of Windsor and hopes to graduate in Fall 2012.