Saarland University
Department of Computer Science
Master’s Program in Computer Science

Master Thesis

Contextual Media Retrieval
Using Natural Language Queries

Submitted by
Sreyasi Nag Chowdhury¹

Supervisors
Dr. Andreas Bulling¹ ² ³
Dr. Mario Fritz¹ ²

Adviser
M.Sc. Mateusz Malinowski¹ ²

Reviewers
Dr. Mario Fritz
Prof. Dr. Bernt Schiele¹ ²

January 16, 2015

¹Saarland University, Saarbrücken, Germany
²Max-Planck Institute for Informatics, Saarbrücken, Germany
³Cluster of Excellence on “Multimodal Computing and Interaction”, Saarbrücken, Germany
Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Declaration of Consent

I agree to make my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, January 16, 2015
To begin with, I would like to express my sincere indebtedness to my supervisors, Dr. Andreas Bulling and Dr. Mario Fritz for granting me the opportunity to work on this exciting project. With their constant guidance and critical comments they have helped make this thesis a challenging endeavor and a deep learning experience for me.

My completion of this project could not have been accomplished without the intriguing discussions I had with my co-supervisor, Mateusz Malinowski. I’m grateful to him for his constructive criticisms on the ideas developed and methods undertaken in the course of the thesis, and his continual counsel, support and encouragement.

I would like to thank my friend Paarijaat Aditya and distant colleagues Martín Ignacio Fallas Hidalgo and Mohamad Gad-elrab for helpful suggestions on Android development and client-server frameworks. I would also like to thank my colleagues Semih Korkmaz, Hosna Sattar, Julian Steil, Dominik Kirst, Xucong Zhang and Darya Dedik for participating in user studies at a short notice.

My dear friends – Deepan Basu, Prabhav Kalaghatgi, Juhi Kulshrestha, Mittul Singh and Yannic Haupenthal – had gone out of their way to help me improve this manuscript. I cannot thank them enough for their time and valuable suggestions.

Finally I extend my deepest gratitude to my loving family and closest friends for their unwavering moral and emotional support during the demanding times of my thesis.
Abstract

The 21st century has seen a rapid increase in the abundance of mobile devices with cameras. This, along with the evolution of digital photography and the internet, has presented mankind with a virtual mine of media content. The increasing number of images and videos rich with metadata (timestamps, GPS location, camera orientation etc.) has the potential to act as a collective memory dispersed in space and time. Put to good use, these can lead to an application which can be deemed as the closest approximation to spatio-temporal exploration of the world. However, such a virtual exploration is practically impossible without human interaction with the computer. In such a scenario, research on Ubiquitous Computing, supported by wearable technology, tries to make these interactions as friction-less as possible.

In our work we develop a query-retrieval system with which users can browse through the collective memory with natural language voice queries and enjoy the visual treat presented as egocentric images and videos. We have extended a state-of-the-art semantic parser to suit the dynamic egocentric environment that a spatio-temporal exploration calls for. Since the Google Glass has made human-computer interactions very easy (by allowing hands-free interactions through voice commands), we demonstrate our query-retrieval system on the Google Glass. The novelty of our work lies in the development of a visual collective memory, in the adaptation of a parser to handle spatial and temporal references in English language questions, in the addition of context or egocentrism to media retrieval and in the demonstration of a query-retrieval system in a dynamic environment.
Preface

This document is produced in partial fulfillment of the M.Sc. degree in Computer Science offered by the Faculty 6 - Natural Sciences and Technology I / Mathematics and Computer Science of Saarland University, Saarbrücken, Germany.

The entire document is divided into 3 broader parts - Introduction, Data Acquisition and Methods and Results and Conclusion. In Part I we discuss the motivation for undertaking this project and the past research works that had addressed similar problems. Part II describes in details our proposed Contextual Media Retrieval System, the prerequisites (data collection) and the user studies that were conducted for the evaluation and analysis of the complete architecture. We discuss the results of these evaluations in Part III along with a detailed overview of the challenges faced and the limitations of our system. We also point the reader to some possible future directions to this and related research works.

We hope to have presented our work in a compelling way invoking further interesting discussions and new research directions.
# Contents

## I Introduction

1 Motivation  
1.1 Overview of our system *Xplore-M-Ego*  

## II Related Work

2 Related Work  
2.1 Spatio-temporal Media Retrieval  
2.2 Natural Language Query Processing  
2.3 Media Retrieval Using Natural Language Queries  

## II Data Acquisition and Methods

3 Data Collection  
3.1 OpenStreetMap  
3.2 *Collective Memory*  
3.3 Training and Test data  
3.3.1 Synthetic Data  
3.3.2 Real Data  
3.4 Participants  

4 Contextual Media Retrieval System – *Xplore-M-Ego*  
4.1 Semantic Parser  
4.1.1 Dependency-based Compositional Semantics  
4.1.2 Induction of Logical Forms  
4.1.3 Answering from Denotations  
4.2 Media Retrieval from Denotations  
4.3 Adapting DCS to a Dynamic and Egocentric Environment  
4.4 Example Queries and DCS Trees  
4.5 Client-Server Communication  
4.5.1 Python Server  
4.5.2 Google Glass Client
5 User Study 51
  5.1 Relevance Feedback .............................................. 51
      5.1.1 Human evaluation of model trained with synthetic data
              (SynthModel) ............................................. 51
      5.1.2 The “human-in-the-loop” training .......................... 52
      5.1.3 Human evaluation of the model trained with real data
              (RealModel) .............................................. 52
      5.1.4 Human evaluation of temporal and contextual query-retrieval 53
  5.2 Prospective real-life questions ................................. 55

III Results and Conclusion 57

6 Evaluation 59
  6.1 SynthModel ....................................................... 61
  6.2 RealModel ......................................................... 62
  6.3 Temporal and Contextual Queries ............................. 64

7 Discussion 69
  7.1 Challenges ....................................................... 69
  7.2 Limitations of the Semantic Parser ........................... 71
  7.3 Accuracy of Performance ...................................... 72
  7.4 Future Work ...................................................... 74

8 Conclusion 77

A List of Abbreviations 79

B List of Figures 81

C List of Tables 83

Bibliography 85
Part I

Introduction
Chapter 1

Motivation

Thanks to digital photography and the large scale use of the internet in the last decade, the world seems to be visually well documented. With efficient search engines at our aid, viewing images and videos of unknown and distant places (mostly touristic) are just a few clicks away. But what about our local environments or contexts? What if we quickly want to know what is behind the building in front of us? What if we want to know what a particular cafe looks like to quickly locate it in a busy market area? What if we see remnants of celebratory decorations at a place and are just curious to know what had happened there the night before? What if we want to see what our new neighborhood looks like in the winter? For all these questions to be easily answered, a huge database of media content – a collective memory – has to be developed. This task only seems tractable if the process of media acquisition requires minimum human interaction. This can be achieved with the introduction of the paradigm of Ubiquitous and Wearable Computing.

After the rapid growth in the use of smart phones, it is presumably foreseen that the future generation of technologically equipped humans would welcome the advent and daily use of wearable devices. Such wearable devices, inevitably fitted with a camera, have already been developed such that they can be conveniently used for life-logging purposes. Devices like the Autographer (Figure 1.1a) and the Narrative Clip (Figure 1.1b) have clip-on cameras that continually take pictures on their own. In such a scenario, we conjecture that a huge number of images and videos will be generated on a daily basis. If these are harnessed in the right fashion, various applications can be developed to enrich man’s daily life.

Speaking of day-to-day life, the 21st century man has arguably grown extremely busy. Science has also adapted to his busyness by coming up with novel ideas and applications to shrink the time required for important daily activities; smart phones and other wearable devices are the latest innovations in this direction. For the user’s convenience, daily-aid applications have also increasingly become egocentric – for instance, route guides showing paths from the user’s current location to a destination display the direction in which the user is facing.
Recording memories or events in frames (images and videos) for later revival has been an important part of human life since the advent of suitable technology. However, a media retrieval application which caters to man’s growing dynamism and egocentrism has not yet been developed. Architectures that allow users to browse through a collection of media in a static environment exist. Past research had also attempted to improve human-computer interaction in such architectures by implementing media retrieval by typed-in natural language phrases. Gaining motivation from these existing frameworks, we develop a tool for spatio-temporal exploration of egocentric media and strive to make human interaction with the machine smoother by integrating the use of natural language voice queries. We have named our system \textit{Xplore-M-Ego} (read \textit{Explore Amigo}) – which stands for “Exploration(\textit{Xplore}) of Media(\textit{M}) Egocentrically(\textit{Ego})”.

![Life-logging devices](image)

(a) The Autographer  
(b) The Narrative Clip

\textbf{Figure 1.1: Life-logging devices}

\section{Overview of our system \textit{Xplore-M-Ego}}

The most convenient medium for humans to interact is via speech [1, 2]. Literature from as far back as the ’80s show that it has been a nurtured topic in computer science research. For instance, Lea [3] states:

“...you will want to use speech whenever possible because it is the human’s most natural communication modality.”

while Viglioni [2] writes:

“With speech recognition and speech response systems, man can communicate with machines using natural language human terminology.”

More recently, in an era where human-computer symbiosis is gradually becoming indispensable, making the machine understand spoken languages is of utmost
interest to the research community [4, 5, 6]. Questions as to how efficiently speech can be used for human-computer interactions have also been raised in the past [7]. Till date, considerable progress has been made in this regard although such an attempt is far from perfect. Since human interaction with computer is an integral part of our project, we make it smoother by integrating the use of natural language voice queries for media retrieval.

Excited by the idea of a prospective contextual/egocentric spatio-temporal exploration of the world by natural language voice queries, the next module to think about was the geographic scope of our system and the nature of questions that it would entertain. For our study, we picked the Saarland University campus as our geographic scope. Further, we realized that in a real-life scenario, given an application that answers questions about the locality of the user, users might be interested in asking questions from a variety of domains. Since that was deemed to be a difficult problem because of problems related to information acquisition, we decided to restrict the nature of questions to only spatio-temporal references. Thus evolved a blue-print of our system which would retrieve images/videos with natural language queries of the following types:

- “What is there in front of the university cafeteria?”
- “How does the computer science building look?”
- “What is the view of the bio-informatics building?”
- “How does the university look in December?”
- “What happened here (user’s geographic location) 5 days ago?”
- “What is on the right of the Max Planck Institute’s parking lot?”
- “Show me images of the campus center.”
- “Which building is behind the mathematics department?”

An existing semantic parser [8] was used for the task of natural language understanding. It will be discussed in greater detail in Sections 2.2 and 4.1. The Google Glass (Figure 1.2), being the most convenient wearable device developed till now, was used to demonstrate our query retrieval system. (An info-graphic by Martin Missfeldt demonstrates the working of the Google Glass and can be found in supplementary Figure A.1.)
An overview of the basic function of our query-retrieval system is shown in Figure 1.3.\footnote{Free vector graphics templates from www.freepik.com have been used to create the Infographics in this manuscript.} The \textit{collective memory} is constantly updated with the images and videos that users capture on a daily basis using a wearable device, for example, the Google Glass (by assumption). The user query, along with metadata such as the GPS coordinates of the user, his/her heading direction and the timestamp at the time of query are sent to a pre-processing step where the temporal reference in the query and the user metadata are written to a dynamic database and the query is modified according to the user-centric reference frame (calculated from the direction the user faces). The semantic parser then predicts answers (names of media files) to the query. The files are then extracted from the \textit{collective memory} and sent to the user for viewing.
Before describing and analyzing our system in greater detail, we would like to discuss some of the prominent research works encompassing the areas of media retrieval (with and without the use of natural language questions) and machine understanding of natural language.
Chapter 2

Related Work

The abundance of digital media in the last century has not gone unnoticed in the computer science research fraternity. Researches have jumped into the arena to develop novel applications for browsing and exploring the expanding collection of images and videos [9, 10, 11, 12].

With paramount importance being attached to human-computer-symbiosis, quite a lot of progress has also been made in the area of machine understanding of the spoken language [13, 14, 15, 16, 17]. A new paradigm in this area has evolved with proposed methods which are trained towards natural language question-answering [8, 18, 19, 20, 21]. For the natural language understanding module of our project we extended one of these methods by Liang et al. [8].

Inspired by the advancement in both fields, computer scientists have also proposed their amalgamations resulting in the development of media retrieval applications which use natural language queries [22, 23, 24, 25, 26, 27, 28, 29, 30].

In this chapter we will discuss some of these related literature that motivated us to undertake this project and point towards our contributions in contrast to existing architectures. A brief overview of this discussion can be found at the end of the chapter in Table 2.1.

2.1 Spatio-temporal Media Retrieval

Spatio-temporal media retrieval is the browsing of media content captured in different geographic locations at various times in the past. In this section we would describe the expanse of media retrieval applications that have been developed till date along with brief descriptions of their features and functionality.

Snavely et al. [9] presented a novel system for browsing and organizing large unorganized photo collections. Their system computes the location, orientation and field of view of the cameras from the images themselves and constructs a sparse 3D geometric representation of the images and the underlying scene. The system performs the following major functions:
1. Enables the user to explore popular world sites in 3D by morphing between photos.

2. Offers a search functionality where the user can look for all images containing a particular object (which can be specified by drawing a bounding box around the object).

3. Finds out the location of the photographer.

4. Provides information about objects in images by transferring annotations from similar images.

The estimated camera pose (location, orientation, field of view) can be used to place each picture into a 3D coordinate system. The user can then move in 3D space from one picture to another using the photo explorer that the system offers. Morphing techniques are used for smooth transition between photos. The broader scene of a particular region is presented as a backdrop by displaying reconstructed features as a 3D point cloud and by using non-photorealistic rendering techniques that provide a better sense of the scene appearance (Figure 2.1).

Figure 2.1: Example of 3D reconstructed view of the location of images and the underlying scene (left) and the user interface (right) presented by Snavely et al. (figure taken from [9])

The system also allows users to add content to a scene by registering new photographs at run-time and by adding text annotations to a selected region of an image. The annotations are stored and can be linked to other resources like maps, guidebooks, video clips etc. Scalability is a limitation for this system since it becomes slow as the number of registered photographs grow.

To address challenges in the construction management industry, Wu et al. [10] designed PhotoScope. It is an interactive tool which visualizes spatio-temporal coverage of photos in a photo collection which users can browse with space, time and standardized content specifications. It also emphasizes on the spatial coverage of the photos rather than camera positions (as in [9]). Figure 2.2 shows the overview of the system. Users can browse through the spatial...
2.1. SPATIO-TEMPORAL MEDIA RETRIEVAL

scope by clicking on the floor map and can slide through the timeline to view images from a particular month.

Tompkin et al. [12] developed a system called Vidicontexts that embed videos in a panoramic frame of reference (referred to as context) and enables simultaneous visualization of videos in different foci. This allows the exploration of spatial and temporal relationships between videos even when there is no direct visual match between them. Vidicontexts takes as input a panoramic image (from online repositories, Google Street View, DSLR stitches etc.) and a collection of videos (taken with smart-phones) with time stamps, GPS data and orientation sensor data. A spatio-temporal index of where and when each video intersects the context is constructed. This data is shown as heat maps with which the user can identify regions in the context which was most covered in videos and navigate to them (Figure 2.3).

The Vidicontexts interface (Figure 2.4) allows the user to pan, zoom and smoothly switch between perspectives. Each video has a local timeline. A global timeline is also shown. The user can slide and adjust the times to see the spatial effect.
of the videos on the context. Users can also displace videos spatially (click and drag a frame on the context) to see their temporal changes.

Figure 2.4: User interface of the Vidicontexts system (figure taken from [12])

Vidicontexts was found to provide better accuracy than other similar systems. However, the main limitation of this system is that sensor orientation data is required to align the videos within the context. This information is not always available for the large collection of existing videos and panoramas. Also, Vidicontexts would fail if large changes occur in the environment between the panoramas and the videos.

Another system proposed by the same authors, VideoScapes [11] facilitates exploration of unstructured but related video collections by navigating between video clips spatially and/or temporally. Videoscapes is a graph structure with videos as edges and portals (automatically identified transition opportunities) as vertices. When temporal context is relevant videos are temporally aligned to offer correctly ordered transitions (Figure 2.5).

Figure 2.5: Overview of Videoscapes computation; portals are shown in green (figure taken from [11])
2.1. SPATIO-TEMPORAL MEDIA RETRIEVAL

An interactive explorer application (Figure 2.6) allows the user to navigate through Videoscapes. While viewing a video, when a portal arises, the user is given a choice to switch between videos. If (s)he chooses to do so, a thumbnail panel with the possible transitions appears. A clock icon signifies that a video is time synchronous with the one presently being viewed. Common road sign icons signify dead ends or repeats. A toggle-able mini map view shows the view frustum and direction. This mini-map is expandable to full-screen where eye icons show possible portals. A user can view a video tour by clicking on these eye icons in succession. Another map view shows real-world traveled paths as lines. A user can select a path and view corresponding videos. Users can also search for videos with images or labels.

![Image of Videoscapes](figure.png)

Figure 2.6: The interactive exploration mode of Videoscapes (figure taken from [11])

The authors further prove with user studies that their proposed system has more benefits when compared with existing systems in terms of spatial awareness, video summarization and video browsing.

Our approach to spatio-temporal media retrieval stands out in comparison to these state-of-the-art methods because we take into account the context of the user (user’s geographic location and the direction in which (s)he faces). No other media retrieval architecture in the literature employ egocentrism as its intrinsic property. Also, all these systems function with (somewhat) complicated touch/click-based Graphical User Interfaces (GUI). We aim to offer a very simple GUI by which exploration of images and videos are possible merely through voice commands. This is further described in Section 4.5.2.
2.2 Natural Language Query Processing

The understanding and successful answering of natural language queries by machines involves mapping the queries into logical forms (semantic parsing). Previously developed approaches to semantic parsing can be categorized into the following prominent methods by the level of supervision:

- By manual annotation of the logical forms [31, 32, 33, 14, 34, 35, 15, 36].
- By weak supervision [37].
- By modeling the logical forms as latent variables and training using only question-answer pairs [38, 8].

Manual annotation of logical forms (where a mapping from textual sentences to logical forms is learnt explicitly) has been found to be expensive while weak supervision cannot handle complex sentence structures. The question-answer based approach doesn’t need expensive annotations of the logical forms (since they are treated as latent variables) and hence are much more scalable (as it is easier to obtain data). This makes the third approach most appealing and convenient. Liang et al. [8] present a parser of this group which gives a very competitive performance to other heavily-supervised methods. Malinowski et al. [21] use this parser to answer questions based on real-world images. Since we also deal with a question-answering task, the work of Liang et al. [8] and Malinowski et al. [21] appeared to be good starting points.

Liang et al. [8] introduce the idea of inducing logical forms as latent variables from natural language questions instead of manually annotating them for training the semantic parser. These induced logical forms are then used to predict answers to the questions based on a world \( w \) of static facts. They define this world \( w \) in the form of a database written in Datalog\(^1\). The also formulate a set of rules and lexical triggers to induce these latent logical forms which are basically constraint satisfaction problems. The solutions to these CSPs, called the denotations, are then used against the world \( w \) to predict textual answers to the natural language questions. An EM-like algorithm, which optimizes the parameters with every iteration, is used to train the model. Testing their system on two standard datasets Geo and Jobs, they achieved superior performance to other semantic parsers trained solely from question-answer pairs and comparable performance to other approaches that required more supervision. Chapter 4 holds a detailed discussion about the semantic parser designed by Liang et al. [8]. Further details can be found in their paper [8].

Drawing inspiration from the class of semantic parsers which are trained using only question-answer pairs, Malinowski et al. [21] proposed an architecture for question-answering tasks based on real-world indoor images. They integrated scene analysis with natural language understanding for this task. They also extended the single world approach of Percy Liang [8] to a multi-world approach.

\(^1\)The syntax of Datalog is similar to that of Prolog; Datalog is more associated with databases.
2.2. NATURAL LANGUAGE QUERY PROCESSING

which takes into account different interpretations of scenes and natural language questions. A general overview of their proposed system is shown in Figure 2.7. Their system answers questions involving count (e.g. “how many chairs are in image 59?”), count and color (e.g. “how many blue balloons are in image 45?”), category (e.g. “what type of room is depicted in image 87?”), comparison (e.g. “what is the largest object in image 2?”) and logical statements (e.g. “which images do not have sofa?”).

In contrast to Liang et al. [8], the authors also deal with subjective interpretations of the scenes rather than depending on hard facts. Hence additional challenges arose due to the inconsistencies in human utterance of natural language and the ambiguities with respect to reference frames. Moreover, they studied how humans differ in their reference to the same objects in a scene. In another article, Malinowski et al. [39] pointed out some of the challenges faced in machine interpretation of natural scenes and languages and formulation of question-answering tasks. Here they also argue that answers, otherwise impossible to obtain by scene segmentation algorithms, can be guided by common sense knowledge of humans (for example, “Which object on the table is used for cutting?” narrows down human search to possible answers(Figure 2.8).

The difference between our work and the two previous papers discussed in this section is that we target a dynamic egocentric environment as the world based on which answers are evaluated by the parser. Whereas, Liang et al. [8] and Malinowski et al. [21] work with a static database of geographic/job/image data. However, the similarity with Malinowski et al. lies in the challenges faced due to the use of spatial relations in natural language queries and the human inconsistencies thereof.
2.3 Media Retrieval Using Natural Language Queries

Previous research on media retrieval using natural language queries varies vastly in the methods used to process the natural language utterances. Lum et al. [22] proposed a method which matches semantic net representations (Figure 2.9) of queries with those of natural language descriptions of media data (by human annotators). They first match the subject noun and action verb in the sentence/semantic net representation with that of the media-file descriptions in the database and then proceed to match the descriptor nodes like color, count etc. They retrieve approximate matches (when subject and action matches while the minor descriptions do not) since they hypothesize that approximate matches are enough for retrievals. However, this is different from a question-answering task where perhaps more accuracy is desired.

Kucuktunc et al. [24] proposed a Part-of-Speech(POS)-tag based pattern matching approach where relations described in Prolog are triggered by matching the POS ordering of the natural language queries with existing query patterns. They apply their procedure for querying a video database from which salient objects and relationships between objects in video frames have been pre-extracted and stored as facts.

There have also been previous works on ontology based image or video retrieval. They represent the media metadata (scene objects and their relationships) as RDF/OWL encoding and query them either by matching nouns and verbs from user provided natural language queries with RDF-triples (Subject-Predicate-Object) [23] or by translating the queries into a SPARQL query (Fig-
2.3. MEDIA RETRIEVAL USING NATURAL LANGUAGE QUERIES

Figure 2.9: Semantic net representation of the sentence “Stephen drives a blue car” (figure taken from [22])

Contrary to the existing research, our work does not involve any computer vision implementations or human annotations for extracting descriptions of object/entities from images and videos. We aim to extract media files simply based on their geographic location (GPS coordinates).

There is prior research which uses geometric features in videos [27] or object configurations in images [29] to answer natural language questions containing spatial relationships. Tellex et al. [27] explored spatial relations in surveillance videos by a classification task which handles two prepositions, “across” and “along”. Resolving spatial relations involve considering an underlying convention of coordinate axes. However, which convention to chose and whether regular users follow the same convention is most often ambiguous. Addressing this ambiguity, they define their own coordinate axes according to the motion detected in the video (Figure 2.11). The line connecting the beginning and end points of the motion trajectory in the video frame is considered as the primary axis. The perpendicular bisector of this line is the secondary axis.

Lan et al. [29] used structured object queries to query for images with spatial relationships between objects (Figure 2.12). Improving over traditional keyword-based classification problems in image retrieval, they introduced a new
CHAPTER 2. RELATED WORK

(a) Video frames from surveillance videos showing motion of a person “across the kitchen”.
(b) Axes that the motion in the video frames above impose on the ground

Figure 2.11: Definition of coordinate axes with respect to motion in video frames (figure taken from [27])

parameter – the configuration of objects in images to reflect the spatial relationships.

Figure 2.12: An example of image retrieval with structured object queries (figure taken from [29])

It is important to note that the nature of queries entertained in these architectures are short phrases of the same structure such as “across the kitchen”, “table beside chair” etc. and thus do not fully qualify as complete natural language queries. In contrast, our media retrieval architecture aims to use rich natural language sentences as user queries. Moreover, it can answer questions containing many other spatial relations like “in front of”, “behind”, “near”, “left of”, “beside”, “opposite to”, “right of”, “ahead of” etc. and is not limited to the identification of spatial configurations of entities in images and videos.

Yet another method adopted to parse natural language queries is Recursive Neural Networks (RNN) [28]. Recent RNN based approaches ([28], [30]) have been more and more successful in representing the meaning of a sentence. To represent the meaning of a whole sentence, the semantic representation of its words is merged according to some rules that resemble building a syntactic tree (Figure 2.13).
Socher et al. [30] use such an RNN based approach for finding images. They map natural language sentences and images into a common embedding space in order to retrieve one from the other (Figure 2.14).

However, our procedure differs from the prior work in that we do not retrieve media based on their contents, but rather based on their spatial localization on a map (by matching GPS coordinates). Also, we aim to develop a question-answering task which these works do not address.

It is important to note that although these approaches to media retrieval deal with some kind of representation of semantics of natural language sentences, the vocabulary that they handle is most often restricted by certain structures (for instance, object-relation-object [29]). We address this limitation by increasing the richness of natural language queries that our architecture manages. Also, to the best of our knowledge, none of the previous works venture into contextual media retrieval by taking into account the user’s current location and viewing direction as necessary criteria. Also, accepting voice queries as user inputs hasn’t been encountered in the literature. We think that the introduction of egocentrism and voice queries in architectures developed for browsing large media collections can have many practical applications. Not only does it open another unexplored dimension for media retrieval (vis-à-vis, egocentrism), but also aids in human interaction with the computer.
With a robust knowledge and understanding of the related research works that inspired us to undertake this project, we would now delve deeper into development of our contextual media retrieval system, the challenges faced, a thorough discussion about the limitations, and a few proposed future research directions.

Table 2.1: Related Work in a nutshell

<table>
<thead>
<tr>
<th>Category</th>
<th>References</th>
<th>Features Offered</th>
<th>Our contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatio-temporal Media Retrieval</td>
<td>[9], [10], [12], [11]</td>
<td>Browsing media collections in a static allocentric setting; Click-based GUI</td>
<td>Browsing media collections in a dynamic egocentric setting; hands-free voice-command based GUI</td>
</tr>
<tr>
<td>Natural Language Query Processing</td>
<td>[8], [21]</td>
<td>Answering natural language questions with respect to a static world; Returning textual information as answers</td>
<td>Answering natural language questions with respect to a dynamic world; Returning media files as answers</td>
</tr>
<tr>
<td>Media Retrieval Using Natural Language Queries</td>
<td>[22], [24], [23], [26], [27], [29], [29], [28], [30]</td>
<td>Retrieving media based on scene contents; Using short structured phrases as queries; Does not take into account user’s context</td>
<td>Retrieving media based on geographic location; Using rich complete natural language sentences as queries; Takes into account user’s context</td>
</tr>
</tbody>
</table>
Part II

Data Acquisition and Methods
Chapter 3

Data Collection

The idea behind this project was to enable the exploration of a certain geographic area (at a certain time) without being physically present there. For this we inherently required the following:

- Some kind of map information which recorded physical features on the ground along with their types (e.g. building, cafe, highway, etc.), names and GPS locations.

- A collection of media files (images and videos) covering a vast spatial and temporal extent which would form our proposed collective memory.

OpenStreetMap was found to be a freely-available and well-documented collection of not only geographic information, but also information about the name and type of the entity, the address (in case of buildings, cafes, sports centers etc.) and sometimes also the kind of facilities it provides (for example, whether a particular cafe is wheelchair accessible, whether a sports center has a badminton court etc.). Moreover, the OpenStreetMap data of a particular geographic area can be exported in the form of an XML file, making it easy to extract relevant information from it. This fulfilled our first requirement.

To fulfill our second requirement, that is, the development of a collective memory, we collected media files from human participants for about a month. Also, to train the semantic parser in order to understand spatio-temporal natural language queries we needed a collection of train and test question-answer pairs. We also collected these natural language questions from human users to make the system as human-centric as possible.

In the subsequent sections we will briefly describe the OpenStreetMap, its prominent features and the information that was extracted from it. Further, we will talk about the development of the collective memory and collection of natural language questions from human users.
CHAPTER 3. DATA COLLECTION

3.1 OpenStreetMap

OpenStreetMap is an open-source map of the world being created by volunteers with knowledge about his/her locality. Its basic data primitives (nodes, ways and relations) are called Elements. They represent entities in the physical world such as building, cafe, highway, lake, river, forest etc. The tags describe specific geographic attributes of the entity – for example, ‘name’, ‘address’, ‘lat’, ‘lon’, ‘amenity’ etc. The basic definitions of the elements are as follows:

- A node represents a standalone point feature in space defined by an unique id, its latitude and longitude. It is accompanied by a tag to define it’s purpose. For example, a restaurant may be tagged with amenity=restaurant.

Example:

```xml
<node id="344240596" visible="true" version="6" changeset="9208001"
     timestamp="2011-09-04T11:43:28Z"
     user="arnhar"
     uid="495739"
     lat="49.2562752"
     lon="7.0436771">
  <tag k="amenity" v="bus_stop"/>
  <tag k="name" v="Universität Mensa"/>
</node>
```

The above entry in the map describes a bus-stop named Universität Mensa. nodeid is an unique id assigned to it, user and uid refer to the voluntary user who updated this information in the OpenStreetMap, lat and lon denote the GPS latitude and GPS longitude of the physical entity.

- A way is an ordered list of nodes used to represent linear features like roads or rivers, or closed boundaries like buildings or forests. In the case of closed polygonal areas, the first and the last node in the way are the same (referred to by unique node ids – nd ref).

Example:

```xml
<way id="42822009" visible="true" version="5" changeset="19178027"
     timestamp="2013-11-29T13:01Z"
     user="sekhan"
     uid="1822774">
  <nd ref="535606984"/>
  <nd ref="535606985"/>
  <nd ref="535606988"/>
  <nd ref="535606989"/>
  <nd ref="535606984"/>
  <tag k="addr: housenumber" v="E1.6"/>
  <tag k="building" v="yes"/>
  <tag k="fixme" v="yes"/>
  <tag k="height" v="15m"/>
</way>
```
• A relation is an ordered list of nodes, ways or other relations representing routes (for example, bus routes, cycle routes), turn directions or multi-polygons (areas with an outer and an inner boundary).

Example:

\[
\begin{align*}
&\langle \text{relationid} = "1363074" \text{ visible} = "true" \text{ version} = "3" \text{ changeset} = "17839254" \text{ timestamp} = "2013-09-14T19:52:30Z" \text{ user} = "arnhar" \text{ uid} = "495739" \rangle \\
&\langle \text{member type} = "way" \text{ ref} = "37474735" \text{ role} = "to" \rangle \\
&\langle \text{member type} = "node" \text{ ref} = "156965353" \text{ role} = "via" \rangle \\
&\langle \text{member type} = "way" \text{ ref} = "237987413" \text{ role} = "from" \rangle \\
&\langle \text{tag k} = "fixme" \text{ v} = "yes" \rangle \\
&\langle \text{tag k} = "fixme:note" \text{ v} = "Abbiege einschr{"a}\n\text{nkung vor Ort pr{"a}ufen!}" \rangle \\
&\langle \text{tag k} = "restriction" \text{ v} = "only\_straight\_on" \rangle \\
&\langle \text{tag k} = "type" \text{ v} = "restriction" \rangle \\
\end{align*}
\]

The above example member type define the type of the constituent members (nodes, ways, relations) and role defines their part in the route.

The geographic or spatial scope of this project was restricted to the Saarland University campus (Figure 3.1) since we want to deal with a limited number of facts (Section 7.1 contains a detailed explanation).

Figure 3.1: Saarland University campus from OpenStreetMap
CHAPTER 3. DATA COLLECTION

The Document Object Model Parser (DOMParser) was used to extract information (nodes and ways) from the exported XML file of the OpenStreetMap. Since our proposed method did not require any direction (routes or turns) information, relations were not included. From this map data we then created a database of static facts which would be used by the semantic parser. This database included predicates which define the type of the physical entity. Each predicate contained information such as the name of the entity and its latitude and longitude.

A few examples of the entries in this database is shown below:

atm(‘postbank’,49.2547788,7.0415639,49.2548,7.0416).
highway(‘universitaet_campus’,49.2548058,7.0405196,49.2548,7.0405).
parking(‘parkhaus_ost’,49.25741,7.048056,49.2574,7.0481).

3.2 Collective Memory

Our collective memory is a pool of images and videos (along with the metadata such as timestamp and GPS coordinates) taken by a lot of people in diverse locations at various points in the past. Such a collection can be developed when humans capture media with smart phones or wearable devices on a daily basis. A hands-free wearable device is more convenient and thus better suited for this kind of media acquisition aimed at producing a huge collection of media files. Keeping the Google Glass in mind as the potential wearable device, we developed an Android app (DataCollectionApp) for the creation of the collective memory. With this app users can capture images or record videos. This takes minimum interaction with the device (after the initiation of the app) since images are captured continuously with a 20 second interval between consecutive captures. However, the video capture could not be made continuous due to heavy consumption of the battery. Users can switch to the video mode manually. The media files thus captured are automatically saved in a common server along with their metadata such as the GPS location of the device and the timestamp. The GUI of the application is shown in Figure 3.2.

![Figure 3.2: GUI of DataCollectionApp](image)
Due to the absence of a large number of wearable devices which could be used for media capture, we mimicked the situation with the help of smartphones. Participants interested in data collection were asked to walk around the Saarland University campus and capture media (images/videos) at various locations. In total the collective memory consists of 1025 images and 175 videos and has a temporal span of about a month. This process was coupled with the collection of natural language questions in that the users first thought of a question and captured the media that they would expect as an answer to that particular question. Some examples of these natural language questions will be discussed in Section 3.3.2.

The collective memory is finally represented as a database of facts written in Datalog logic which will then be used by the semantic parser for predicting answers to natural language spatio-temporal questions. The Datalog representation is as follows:

\[
\begin{align*}
\text{image}(\text{id}, \text{timestamp}, \text{latitude}, \text{longitude}, \text{month}) & : \\
\text{image}(\text{img}_{20141111}, 20141111, 49.2566, 7.0442, \text{november}). \\
\text{image}(\text{img}_{20141112}, 20141112, 49.2554, 7.0396, \text{november}). \\
\text{image}(\text{img}_{20141226}, 20141226, 49.2569, 7.0441, \text{december}). \\
\text{video}(\text{id}, \text{timestamp}, \text{latitude}, \text{longitude}, \text{month}) & : \\
\text{video}(\text{vid}_{20141121}, 20141121, 49.2569, 7.0456, \text{november}). \\
\text{video}(\text{vid}_{20141123}, 20141123, 49.2530, 7.0338, \text{november}). \\
\text{video}(\text{vid}_{20141226}, 20141226, 49.2518, 7.0405, \text{december}).
\end{align*}
\]

3.3 Training and Test data

In order for the semantic parser to understand spatio-temporal natural language queries and predict answers efficiently, it was necessary to train the model with question-answer pairs.

Through the data collection procedure we observed that the human perception of “frame of reference” is rife with inconsistencies and ambiguities. For example, the spatial relation “left of” instigated imagery of different physical locations for different users – for some users it meant “north of” while some others interpreted it as “west of”. Most often it depended on which direction the front entrance of a particular entity (building, cafe, sports center etc.) faced. Since such information was not available from the OpenStreetMap data, and also because these inconsistencies would lead to a poorly trained model, we use a canonical reference frame for training the semantic parser. This called for the creation of a synthetic data set in which the questions followed fixed templates and the answers were generated by consistent rules. Later on the performance of the model trained by this synthetic data set (SynthModel) would be tested on questions collected from human users (the real data set). The real data set would also be used to train a different model of the semantic parser (RealModel) through the “human-in-the-loop” training (Section 5.1.2).
3.3.1 Synthetic Data

We use a template-based approach to generate the synthetic data set similar to Malinowski et al. [21]. The names of all the entities in and around the Saarland University campus were extracted from the OpenStreetMap. For each entity the synthetic data set consisted of questions adhering to the following templates:

- “What is there in front of {name of entity}?"
- “What is there behind {name of entity}?"
- “What is there on the left of {name of entity}?"
- “What is there on the right of {name of entity}?"

Example:

```
“what is there in front of mpi inf?”
“what is there behind mpi inf?”
“what is there on the right of mpi inf?”
“what is there on the left of mpi inf?”
```

These questions were accompanied by answers in Datalog syntax which referred to predicates defined in the semantic parser. The question-answer pairs in the synthetic data set were written as Datalog statements and had the following format:

```
parse([what, is, there, in, front, of, mpi_inf, ?],
answer(A,(const(B,buildingid('mpi_inf')),frontOf(B,A)))).
parse([what, is, there, behind, mpi_inf, ?],
answer(A,(const(B,buildingid('mpi_inf')),behind(B,A)))).
parse([what, is, there, on, the, right, of, mpi_inf, ?],
answer(A,(const(B,buildingid('mpi_inf')),rightOf(B,A)))).
parse([what, is, there, on, the, left, of, mpi_inf, ?],
answer(A,(const(B,buildingid('mpi_inf')),leftOf(B,A)))).
```

Here, `const`, `building`, `leftOf`, `rightOf`, `behind` and `frontOf` are the predicates (rules written in Datalog).

3.3.2 Real Data

We collected the real data from human participants who were familiar with the Saarland University campus. They were instructed to use the following spatial relations in their questions: “front of”, “behind”, “left of” and “right of”. However they weren’t asked to use exact templates, resulting in several different forms of the same question such as:

- “What is there on the left of MPI-INF?”
- “What is on the left of MPI-INF?”
3.4. PARTICIPANTS

- “What is to the left of MPI-INF?”
- “What is on the left side of MPI-INF?”
- “What building is left of MPI-INF?”
- “What is the view in front of MPI-INF?”
- “How does the building on the left of MPI-INF look?” etc.

The participants also naturally used additional spatial relations such as “ahead of”, “opposite to”, “near”, “beside” etc.

A total of 1000 questions were collected. Out of this, 500 questions were used for the “human-in-the-loop” training of the semantic parser (Section 5.1.2). The rest of the 500 questions were held aside for testing the various different models of the semantic parser that would eventually be trained.

As pointed out in Section 3.2, the process of collecting questions and generating the database of media files were juxtaposed. Table 3.1 shows a few examples of the questions and associated images that the participants had collected.

3.4 Participants

To introduce sufficient amount of variations in natural language we chose participants belonging to different cultures and speaking different mother tongues (detailed information can be found in Table 3.2). There were 15 participants – 8 male and 7 female – of the age group 22-28. They were asked to use the DataCollectionApp that we developed for the Google Glass (but was customized for use with Android powered smart phones) for capturing images and videos. As mentioned in the previous section, the participants also paired each of these images and videos with a natural language question that they would like to ask a media retrieval system.

Having gathered the prerequisites for building our media retrieval system we proceeded with the development of the actual architecture. The following chapters explain our system in details.
<table>
<thead>
<tr>
<th>Question</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>“What is there beside MPI-INF?”</td>
<td><img src="image1.jpg" alt="Image" /></td>
</tr>
<tr>
<td>“What is on the left of E 1.3?”</td>
<td><img src="image2.jpg" alt="Image" /></td>
</tr>
<tr>
<td>“How does the campus bus stop look?”</td>
<td><img src="image3.jpg" alt="Image" /></td>
</tr>
<tr>
<td>“What is on the right side of the university campus?”</td>
<td><img src="image4.jpg" alt="Image" /></td>
</tr>
<tr>
<td>“What is behind the main entrance?”</td>
<td><img src="image5.jpg" alt="Image" /></td>
</tr>
<tr>
<td>“What is in front of the campus center?”</td>
<td><img src="image6.jpg" alt="Image" /></td>
</tr>
<tr>
<td>“How does the university main entrance look?”</td>
<td><img src="image7.jpg" alt="Image" /></td>
</tr>
<tr>
<td>“What is in front of the university bus terminal?”</td>
<td><img src="image8.jpg" alt="Image" /></td>
</tr>
</tbody>
</table>
### 3.4. PARTICIPANTS

<table>
<thead>
<tr>
<th>User</th>
<th>Age</th>
<th>Gender</th>
<th>Mother Tongue</th>
<th>Most fluent language</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>26</td>
<td>M</td>
<td>Malayalam</td>
<td>Malayalam</td>
</tr>
<tr>
<td>User 2</td>
<td>24</td>
<td>F</td>
<td>Uzbek</td>
<td>Russian</td>
</tr>
<tr>
<td>User 3</td>
<td>24</td>
<td>F</td>
<td>Telugu</td>
<td>Telugu</td>
</tr>
<tr>
<td>User 4</td>
<td>24</td>
<td>M</td>
<td>Telugu</td>
<td>Telugu</td>
</tr>
<tr>
<td>User 5</td>
<td>25</td>
<td>M</td>
<td>Tamil</td>
<td>Tamil</td>
</tr>
<tr>
<td>User 6</td>
<td>28</td>
<td>M</td>
<td>Arabic</td>
<td>Arabic</td>
</tr>
<tr>
<td>User 7</td>
<td>24</td>
<td>M</td>
<td>Telugu</td>
<td>Telugu</td>
</tr>
<tr>
<td>User 8</td>
<td>25</td>
<td>M</td>
<td>Urdu</td>
<td>Urdu</td>
</tr>
<tr>
<td>User 9</td>
<td>27</td>
<td>F</td>
<td>Bengali</td>
<td>English, Bengali</td>
</tr>
<tr>
<td>User 10</td>
<td>24</td>
<td>M</td>
<td>Hindi</td>
<td>Bhojpuri, Hindi</td>
</tr>
<tr>
<td>User 11</td>
<td>25</td>
<td>F</td>
<td>Tamil</td>
<td>Tamil</td>
</tr>
<tr>
<td>User 12</td>
<td>25</td>
<td>F</td>
<td>Kannada</td>
<td>English</td>
</tr>
<tr>
<td>User 13</td>
<td>25</td>
<td>F</td>
<td>Bengali</td>
<td>Bengali</td>
</tr>
<tr>
<td>User 14</td>
<td>25</td>
<td>F</td>
<td>Telugu</td>
<td>Hindi, Telugu</td>
</tr>
<tr>
<td>User 15</td>
<td>22</td>
<td>M</td>
<td>Bengali</td>
<td>Bengali</td>
</tr>
</tbody>
</table>
Chapter 4

Contextual Media Retrieval System – *Xplore-M-Ego*

There are two layers to our contextual media retrieval system - the front end (client side) with which users ask questions and view returned results - and the back end (server side) where the main task takes place. Overall, the complete architecture has 4 main modules (it can also be viewed in the black-box representation of *Xplore-M-Ego* in Figure 1.3):

- Module 1: reception of query from the user, recording his metadata and sending these to the server
- Module 2: conversion of a dynamic and egocentric world to a static world
- Module 3: semantic parsing of the query and generating denotations to predict answers
- Module 4: extraction of the actual media files from the collective memory and sending them to the user

Module 1 constitutes the client side and Modules 2-4 constitute the server side. It is important to take into account the distinction between the prediction of answers by Module 3 and the actual extraction of media from the collective memory. In the former, the predictions, from induced DCS trees, are just textual names of the suitable media files registered in our media database. Once these names are known, they are trivially fetched from the collective memory – a collection all media files.

The bulk of our application depends on the semantic parsing of natural language questions and therein lies most of our challenges and limitations. Hence we would first like to have a detailed discussion about the state-of-the-art semantic parser [8] that we extend in our project. We would then describe our extensions to it to meet our requirements of working with a dynamic and egocentric environment. We would finally conclude the chapter with a description of the complete client-server architecture that *Xplore-M-Ego* embodies.
4.1 Semantic Parser

Semantic parsing deals with the challenging task of enabling a machine to understand the meaning of a natural language sentence. It refers to building a semantic representation of a natural language utterance from predicates representing meanings of single words. Such representations are referred to as logical forms. Traditional approaches to semantic parsing used manual annotations of logical forms as training examples for a heavily supervised learning method [31, 32, 33, 14, 34, 35, 15, 36]. Modern semantic parsing approaches implement weakly supervised learning algorithms to train the parser simply with natural language question-answer pairs [38, 8]. The logical forms in this approach are induced as latent variable and hence conveniently gets rid of costly manual annotations.

Liang et al. [8] used the question-answer based training approach to develop a semantic parser which answers natural language questions based on a static set of facts. There are three most important components of their semantic parser – a set of rules that can produce multiple derivations or representations of a single sentence, a machine learning based approach to selectively reject incorrect derivations, and the principle of compositionality, where the meaning of a whole sentence is derived from its sub-parts. Then, with the help of a lexicon, words from the text are mapped to a set of candidate predicates. In Liang et al. [8], the POS tags of words are used to accomplish this mapping. Identification of POS tags, however, is generally a difficult problem since the same words may have different parts of speech depending on the context of the sentences they appear in. In spite of recent advances in this area, understanding the intended meaning of a human natural language utterance seems to be too challenging a ask for a machine to accomplish.

We had briefly discussed the semantic parser of Liang et al. [8] in Section 2.2. In the following section we elaborate on the kind of data they operate on, how they encode their constraint satisfaction problem, their lexical triggers and the types of natural language questions which are answered by their system.

4.1.1 Dependency-based Compositional Semantics

Dependency-based Compositional Semantics (DCS) formalism is introduced by Liang et al. [8] for building logical forms. These can be represented by trees with nodes labeled with predicates and edges labeled with relations. Such a DCS tree encodes a Constraint Satisfaction Problem (CSP) and is computationally efficient. An example of a DCS tree (both in mathematical and graphical notation) can be found in Figure 4.1. The logical form $z$ (from the figure) encodes a DCS tree. city, major, loc and CA are predicates with edges corresponding to the relations between them (for example, the edge between loc and CA mean that the 2nd argument of predicate loc should unify with the 1st argument of the predicate CA). Such encoding can be equivalently encoded in a lambda

\[ \lambda_{\text{CA}}(\text{city}, \text{major}, \text{loc}) \]

The way in which Prolog matches two terms is called unification. (http://www.dai.ed.ac.uk/groups/ssp/bookpages/quickprolog/node12.html)
4.1. SEMANTIC PARSER

Liang et al. [8] have argued that DCS is a more tractable representation than lambda calculus.

Example: major city in California

\[
z = \langle \text{city}, \text{major}, \text{loc}, \text{CA} \rangle
\]

\[
\lambda c \exists m \exists \ell \exists s. \\
\text{city}(c) \land \text{major}(m) \land \\
\text{loc}(\ell) \land \text{CA}(s) \land \\
c_1 = m_1 \land c_1 = \ell_1 \land \ell_2 = s_1
\]

(a) DCS tree

(b) Lambda calculus formula

(c) Denotation: \([z]_w = \{\text{SF, LA, ...}\}\)

Figure 4.1: Example of a DCS tree; denotations refer to a set of feasible solutions of a particular CSP instance (figure taken from [8])

The logical forms thus induced as DCS trees, by the virtue of encoding a CSP, generate a list of feasible solutions called denotations which actually answer the original natural language question. In Figure 4.1, the denotations of the DCS tree are SF(San Francisco), LA(Los Angeles) etc. which are the major cities in California, thus answering the main question. These two separate but consecutive processes – inducing DCS and predicting answers from denotations – form Module 3 of our contextual media retrieval system.

4.1.2 Induction of Logical Forms

Figure 4.2 shows the probabilistic model of Liang et al. [8]. For a particular question \(x\), the latent logical form \(z\) is induced according to a probability distribution parametrized with \(p_\theta(z|x) \propto e^{\phi(x,z)^T \theta}\). In the distribution, \(\phi(x,z)\) are feature vectors. The answer \(y\) is then evaluated from this logical form based on a database of facts (world \(w\)).

For the induction of these logical forms or DCS trees [8] the semantic parser relies on a set of rules or predicates written in Datalog. These rules define the constraints that need to be fulfilled in establishing relations between different nodes. Some of the predicates used by Liang et al. [8] are shown below:

\[\text{Predicate Example:\quad} \text{city}(c) \land \text{major}(m) \land \text{loc}(\ell) \land \text{CA}(s) \land c_1 = m_1 \land c_1 = \ell_1 \land \ell_2 = s_1\]

\[\text{Predicate Definition:\quad} p_\theta(z|x) \propto e^{\phi(x,z)^T \theta}\]

\[\text{Feature Vector:\quad} \phi(x,z)\]

\[\text{Database of Facts:\quad} \text{world } w\]

\[\text{Evaluation of Answer:\quad} y = \text{eval}(z, w)\]

\[\text{Contextual Media Retrieval System:\quad} \text{Module 3}\]

\[\text{Semantic Parser:\quad} \text{DCS Trees}\]

\[\text{Induction of Logical Forms:\quad} \text{Probability Distribution}\]

\[\text{Rule Definition:\quad} \text{Datalog}\]

\[\text{Predicate Usage:\quad} \text{Semantic Parser Output}\]

\[\text{Feature Engineering:\quad} \text{Feature Vectors}\]

\[\text{Evaluation Framework:\quad} \text{Answer Evaluation}\]

\[\text{System Architecture:\quad} \text{Module 3 Architecture}\]

\[\text{Discussion about lambda calculus is being omitted here since it is not directly relevant for understanding this work.}\]
These are compositional rules that are required by the parser which combine simpler rules into more complex rules. The variables on the left hand side of the ‘:-’ are unified with those on the right hand side. ‘_’ represent ‘don’t-care-variables’ which can have any value. Also, the predicates suffixed ‘id’ are used to refer to proper nouns like names of states, cities, rivers etc.

The semantic parser also requires a set of lexical triggers. These are hand-designed set of mappings between syntactic categories of words and a set of candidate predicates. The authors define a basic set of lexical triggers \( L \) in the form of pairs \((x, p)\) where \(x\) is a sequence of words or Parts of Speech (POS) tags in \{JJ,NN,NNS\} \(^4\) and \(p\) is a predicate. This set is shown below:

---

\(^3\)We would like to direct the reader to The Prolog Dictionary for a greater insight into the terms related to Prolog/Datalog that we use in this manuscript – http://www.cse.unsw.edu.au/~billw/prologdict.html.

\(^4\)JJ – Adjective, NN – Noun, NNS – Noun, plural
4.1. SEMANTIC PARSER

\[
((\text{function words, predicate}))
\]
\[
(\text{most, size}).
\]
\[
(\text{total, sum}).
\]
\[
(\text{called, nameObj}).
\]

\[
((\text{POS tags}, [\text{predicates}]))
\]
\[
(WRB, \text{loc}).
\]
\[
([-NN,NNS],[\text{city, state, country, lake, mountain, river, place}]).
\]
\[
([-NN,NNS],[\text{person, capital, population}]).
\]
\[
([-NN,NNS,JJ],[\text{len, negLen, size, negSize, elevation}]).
\]
\[
([-NN,NNS,JJ],[\text{negElevation, density, negDensity, area, negArea}]).
\]
\[
(JJ, \text{major}).
\]

The POS tags follow the Penn Treebank Project [40]. For the sake of convenience, the set of POS tags referred to above are being stated below:

\[
WRB : \text{Wh-adverb}
\]
\[
NN : \text{Noun, singular or mass}
\]
\[
NNS : \text{Noun, plural}
\]
\[
JJ : \text{Adjective}
\]

For an improved performance, the authors also define an augmented lexicon \(L^+\) containing prototype words for all the predicates which were mapped to a POS tag in \(L\). This is a hard-coding which cancels the effect of predicates triggered by the words' POS tag. For example, the POS tags \(NN, NNS, JJ\) would trigger the predicates \(\text{len, size, negSize, elevation}\) when using \(L\). However when using \(L^+\) the words \(\text{long, large, small, high}\) (from the list of prototype words given below), irrespective of their POS tags, would trigger the predicates \(\text{len, size, negSize, elevation}\) respectively. Since \(L^+\) has more hard-codings, it gives a better performance than \(L\). However, creating \(L^+\) or an even more fine-grained list of triggers requires manually finding all synonyms that a predicate would ideally represent. Since this is a complex procedure which doesn’t guarantee completeness and also because it tends to over-train the model to a certain vocabulary, it is best to loosely define a mapping of POS tags into a set of predicates (as in \(L\)). Some of the prototype words defined in \(L^+\) are shown below:

\[
(\text{long, len}).
\]
\[
(\text{large, size}).
\]
\[
(\text{small, negSize}).
\]
\[
(\text{high, elevation}).
\]

The number of DCS trees generated per question was restricted by beam search which is a special form of best first search involved in truncating the search space.
4.1.3 Answering from Denotations

The next step after the generation of the logical forms is to actually answer the natural language question in textual form. For this, a world \( (w) \) is required (Figure 4.2) which is implemented as a database. Liang et al. [8] use two such benchmark databases – GEO and JOB. We will use the GEO dataset to exemplify their system. It contains data on the names of States, constituent cites, population, names and length of rivers flowing through them, highest/lowest points, names of mountains, roads and lakes. Below is a small excerpt from their database:

```
state('alabama','al','montgomery',3894.0e+3,51.7e+3,22,'birmingham','mobile', 'montgomery','huntsville').
city('alabama','al','birmingham',284413).
city('alabama','al','mobile',200452).
river('arkansas',2333,['colorado','kansas','oklahoma','arkansas']).
border('alabama','al',['tennessee','georgia','florida','mississippi']).
highlow('arizona',az,'humphreys peak',3851,'colorado river',21).
mountain('alaska','ak','mckinley',6194).
road(86,['massachusetts','connecticut']).
lake('superior',82362,['michigan','wisconsin','minnesota']).
country('usa',307890000,9826675).
```

The lines in the database can be interpreted as follows: the State Alabama (in predicate `state`), also referred to by acronym `al`, has Montgomery as its capital city and Birmingham, Mobile and Huntsville as its major cities; the river Arkansas (in predicate `river`) is 2333 km long and flows through the cities Colorado, Kansas, Oklahoma and Arkansas and so on. Here, it is important to note that this database of facts is static and raises no possibility of disagreement between human users while referring to an entity or while evaluating the correctness of the retrieved results.

The concrete implementation of the questions-answering architecture by Liang et al. [8] had an accuracy of 88.6% with \( L \) and 91.1% with \( L+ \) on the GEO dataset [8]. Let us now look at some of the example queries that the semantic parser can answer.

```
"What is the highest point in Florida?"
"Which State has the shortest river?"
"What is the tallest mountain in the United States?"
"What is the capital of Maine?"
"What are the populations of states through which the Mississippi river run?"
"Name all the lakes of US?"
"Which state is the city Denver located in?"
```

As pointed out previously, the questions contain no ambiguity of reference frames. Moreover, reference resolution (automatically mapping acronyms to elaborate proper nouns in the database of facts) is not required. Also, chances
of human inconsistency in question-answering (which is most often considered as ground truth or baseline) is nil.

The semantic parser developed by Liang et al. [8], although being the state-of-the-art, is not completely rid of this problem. Since our methodology extends this parser, our proposed system is also similarly limited.

4.2 Media Retrieval from Denotations

One of the major differences of our contextual media retrieval application and Percy Liang’s DCS was that we desired to retrieve media as answers to natural language questions instead of textual information. Naturally, the denotations of the DCS trees that our system would generate had to be the names of media files. For example, denotations for the question “What is there on the right of MPI?” would have to be something like ['image234', 'image12', 'image58',...], where ‘image234’, ‘image12’, ‘image58’ and so on would show geographic entities on the right of MPI. To implement such a requirement, we generated a database containing the names and metadata of all the media files present in the collective memory. Such a database was written in Datalog and had entries of the following form:

<table>
<thead>
<tr>
<th>Image denotation</th>
<th>Timestamp</th>
<th>GPS Coordinates</th>
<th>Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>image('img_20141111_165828',20141111,49.2566,7.0442,'november').</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>image('img_20141112_092045',20141112,49.2554,7.0396,'november').</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>video('vid_20141121_120149',20141121,49.2569,7.0456,'november').</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>video('vid_20141123_165241',20141123,49.2530,7.0338,'november').</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The predicates image and video define the names of images and videos respectively along with their timestamps, GPS coordinates and the name of the month in which they were captured. Taking this world as a basis the names of the media files could now be predicted as denotations of the DCS trees.

After the generation of the denotations the actual media files were extracted from the collective memory – physically, a file system storing all captured media. These extracted media files were then returned to the user as answers to the natural language question asked.

However, since we considered a different domain of questions (spatio-temporal questions) in contrast to Percy Liang’s DCS as the target natural language questions to be answered, underlying our system was a completely different set of predicates and lexical triggers. It was also necessary for us to implement a dynamic and egocentric environment which our system embodies within the existing static setup presented in DCS [8]. In the following sections we would like to discuss these extensions in details and show some examples of DCS trees generated in response to the type of questions our system handles.
4.3 Adapting DCS to a Dynamic and Egocentric Environment

In the light of the discussion on Percy Liang’s DCS, we now proceed to point out our extensions of the semantic parser.

We extend the DCS parser [8] to deal with a dynamic and egocentric environment and hence implement the idea of Contextual Media Retrieval. Revising our requirements once again, our set up is dynamic due to the following reasons:

- The source of the query is an user who is interacting with a wearable device and dislocating in space and time.
- By assumption, the collective memory, and hence the set of media files in our database is always increasing in size.

We also utilize egocentrism by:

- Taking into account the user’s current location for contextual temporal queries.
- Retrieving answers in accordance with the user’s heading direction.

We also return media files as answers to the natural language queries whereas the original DCS parser returned textual information.

To incorporate dynamism and egocentrism into the pre-existing framework of DCS, we conceptualized a partition in the world \( w \) that the parser uses to answer natural language questions. Our world consists of a static part \( w_s \) and a dynamic part \( w_d \). \( w_s \) is a database of geographic facts (Section 3.1). \( w_d \) (\( w_d = w_{du} + w_{dm} \)) constitutes a dynamic database of user metadata (\( w_{du} \)) and the collective memory (Section 3.2) \( (w_{dm}) \). The DCS parser [8] was originally designed to work with the static world \( w_s \). Our approach to modeling dynamism is to set \( w_d \) anew for each query before feeding it to the parser. \( w_{dm} \) was also updated as and when new media content was available. Hence, although our world \( (w = w_s + w_d) \) is continuously changing, to the DCS parser it always appears static.

Below we show a few example entries in our static and dynamic databases:

**Static Database of Geographic Facts \( w_s \)**

<table>
<thead>
<tr>
<th>Name</th>
<th>Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>atm('postbank_atm')</td>
<td>49.2573855,7.0430358,49.2574,7.0430</td>
</tr>
<tr>
<td>bank('bank1saar')</td>
<td>49.2545957,7.0401859,49.2546,7.0402</td>
</tr>
<tr>
<td>bar('canossa')</td>
<td>49.2572934,7.0429204,49.2573,7.0429</td>
</tr>
<tr>
<td>building('department_of_culture')</td>
<td>49.25343,7.0414877,49.2534,7.0415</td>
</tr>
<tr>
<td>cafe('icoffee')</td>
<td>49.2574952,7.0453556,49.2575,7.0454</td>
</tr>
<tr>
<td>highway('campus')</td>
<td>49.25573,7.0389795,49.2557,7.0390</td>
</tr>
<tr>
<td>library('state_library')</td>
<td>49.2533533,7.038327,49.2534,7.0383</td>
</tr>
<tr>
<td>parking('uni_nord')</td>
<td>49.25751,7.041421,49.2575,7.0414</td>
</tr>
<tr>
<td>research_institution('dfki')</td>
<td>49.25717,7.041499,49.2572,7.0415</td>
</tr>
</tbody>
</table>
4.3. ADAPTING DCS TO A DYNAMIC AND EGOCENTRIC ENVIRONMENT

The static database of geographic facts serves the purpose of the map data of our spatial scope. Proper nouns in the natural language utterances are mapped to the entity names in the database. For example, the query “what is there in front of postbank atm?” triggers the predicate atm from ws stated above.

Dynamic Database of Media Content \( w_{dm} \)

<table>
<thead>
<tr>
<th>Image</th>
<th>Timestamp</th>
<th>Coordinates</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>img_20141111_165828</code></td>
<td>20141111, 49.2566, 7.0442</td>
<td>November</td>
<td></td>
</tr>
<tr>
<td><code>img_20141112_092045</code></td>
<td>20141112, 49.2554, 7.0396</td>
<td>November</td>
<td></td>
</tr>
<tr>
<td><code>vid_20141121_120149</code></td>
<td>20141121, 49.2569, 7.0456</td>
<td>November</td>
<td></td>
</tr>
<tr>
<td><code>vid_20141123_165241</code></td>
<td>20141123, 49.2530, 7.0338</td>
<td>November</td>
<td></td>
</tr>
</tbody>
</table>

This database is appended when new media files are added to the collective memory by the assumption that people with wearable devices generate a large number of everyday images and videos.

Dynamic Database of User Metadata \( w_{du} \)

<table>
<thead>
<tr>
<th>Person</th>
<th>Coordinates</th>
<th>View Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>person(49.2578, 7.0454, 'n')</code></td>
<td>(x, y) in the user’s current GPS location, 'n' refers to 'north'</td>
<td></td>
</tr>
<tr>
<td><code>day(20141104)</code></td>
<td>The date of the user’s query</td>
<td></td>
</tr>
</tbody>
</table>

Here, in `person(x, y, 'n')`, ‘x’, ‘y’ refer to the current GPS location of the user and ‘n’ refers to ‘north’ – an example of user’s viewing direction. The viewing direction is calculated from the azimuth angle (the angle the user’s viewing makes with the geomagnetic north) of the magnetometer reading of the wearable device. In `day(k)`, ‘k’ refers to the temporal reference in the user query – for example, for the query “what happened here 5 days ago?”, ‘k’ is the timestamp of 5 days prior to the current date. The user metadata, along with the natural language query (transcribed to textual form the voice query) are sent from the client to the server at the time of the query. This information is then extracted and written to the dynamic database \( w_{du} \) in a pre-processing step at the server side.

Having given a notion about our world \( w \) we now need to define the predicates that the semantic parser would use to build the DCS trees. Our predicates pertain to the spatial and temporal references that we aim at handling in English language questions. We resolve the spatial relations “front of”, “behind”, “left of” and “right of” by defining a convention based on the geomagnetic reference frame. We calculate these relations by comparing the values of the GPS coordinates of the entities. The temporal references are resolved by comparing the timestamps in the predicate `day` in database \( w_{du} \) (after modifying it in query-time) with the timestamps of the media files in our collective memory. The spatio-temporal references (for example, “what happened here 2 days ago?”) are resolved by comparing both GPS coordinates and timestamps of entities in the databases. Table 4.1 shows the logical definitions of these predicates along with example queries.
### Table 4.1: Definitions of predicates in our DCS

<table>
<thead>
<tr>
<th>Predicates</th>
<th>Definitions</th>
<th>Example Query</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spatial</td>
<td></td>
<td></td>
</tr>
<tr>
<td>frontOf(A,B)</td>
<td>lat(B)&gt;lat(A), lon(A)=lon(B)</td>
<td>“what is in front of A?”</td>
</tr>
<tr>
<td>behind(A,B)</td>
<td>lat(B)&lt;lat(A), lon(A)=lon(B)</td>
<td>“what is behind A?”</td>
</tr>
<tr>
<td>rightOf(A,B)</td>
<td>lon(B)&gt;lon(A), lat(A)=lat(B)</td>
<td>“what is on the right of A?”</td>
</tr>
<tr>
<td>leftOf(A,B)</td>
<td>lon(B)&lt;lon(A), lat(A)=lat(B)</td>
<td>“what is on the left of A?”</td>
</tr>
<tr>
<td>Temporal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>view2(M,B)</td>
<td>month(B)=M, lat(B)=user’s lat, lon(B)=user’s lon</td>
<td>“how did this place look in M?”</td>
</tr>
<tr>
<td>view1(B)</td>
<td>timestamp(B)=user’s timestamp, lat(B)=user’s lat, lon(B)=user’s lon</td>
<td>“what happened here 5 days ago?”</td>
</tr>
</tbody>
</table>

Here, B is a media file. A is a geographical entity (e.g. ‘MPI’) and M is a month (e.g. ‘May’) uttered as part of the query; ‘lat’ and ‘lon’ stand for GPS latitude and longitude; day and person are predicates in $w_{du}$.

Similar to Percy Liang’s DCS, we conducted experiments with 2 sets of lexical triggers $L$ and $L^+$. $L$ consisted of mapping POS tags to certain predicates as shown below:

```
([WP,WDT], [image,video]).
(NN, [atm,building,cafe,highway,parking,research_institution,restaurant,shop,sport,tourism,university]).
(JJS, [nearest]).
([NN,NNS,VB], [view2]).
(VBD, [view1]).
```

The set of POS tags referred to above are:

- **WP**: Wh-pronoun
- **WDT**: Wh-determiner
- **NN**: Noun, singular or mass
- **JJS**: Adjective, superlative
- **NNS**: Noun, plural
- **VB**: Verb
- **VBD**: Verb, past tense

Words having POS tags *WP* (e.g. “what”) and *WDT* (e.g. “which”) trigger the predicates *image* and *video* since we desire to have names of images and videos.
4.3. ADAPTING DCS TO A DYNAMIC AND EGOCENTRIC ENVIRONMENT

as answers to the queries. \textit{NN} triggers all the predicates which define geographic entities. This mapping is done to address questions such as “Which is the nearest cafe/NN?” and the predicate \textit{nearest} is triggered by the POS tag \textit{JJS}. Other superlative adjectives such as “closest”, “farthest” etc. can also be added to this mapping\textsuperscript{6}. Questions of the form “What is the view/NN of Postbank?” “Show images/NNS of Postbank.”, “What does Postbank look/VB like?” can be handled by mapping [\textit{NN, NNS, VB}] to the predicate \textit{view2} (described in Table 4.1). By mapping the POS tag \textit{VBD} to predicate \textit{view1} we successfully address questions about the past such as “What did this place look like in December?”.

We could not accommodate lexical triggers for the spatial relations (“front of”, “behind”, “left of”, “right of”) in \textit{L} solely based on POS tags. This is because of the varied POS tags attributed to the spatial relations. The following sentences with the spatial relations tagged with POS demonstrate this variety:

- “What is there left/VBN of MPI?”
- “What is there on the left/NN of MPI?”
- “What is there in front/NN of MPI?”
- “What is there behind/IN MPI?”
- “What is there right/RB of MPI?”
- “What is there on the right/NN of MPI?”

Mapping \textit{NN} to predicates \textit{frontOf}, \textit{behind}, \textit{rightOf}, \textit{leftOf} would lead to unwanted triggers for sentences such as “Which is the nearest cafe/NN?”, thus leading to bad performance of the parser. Mapping \textit{IN} (Preposition) to predicate \textit{behind} would also trigger the predicate for trivial prepositions such as ‘of’, ‘in’ etc., leading to complicated and wrong DCS trees. Hence we hard-coded these spatial relations in an augmented lexicon \textit{L+} as shown below:

\begin{Verbatim}
(front,frontOf).
(behind,behind).
(right,rightOf).
(left,leftOf).
\end{Verbatim}

As expected, the semantic parser had a better accuracy of prediction with the use of the augmented lexicon \textit{L+} (47%) than with the use of the basic lexicon \textit{L} (17.9%). The performance measures are discussed in greater detail in Chapter 6. The complete architecture of our client-server setup can be found in Section 4.5.

\textsuperscript{5}POS tags of individual words are conventionally written after a backslash as in this example: cafe/NN.

\textsuperscript{6}We have not included these relations in our study.
4.4 Example Queries and DCS Trees

With rules defined in the form of predicates, the databases, and the lexicons, the semantic parser was able to generate the DCS trees for spatio-temporal natural language questions that Xplore-M-Ego aimed to address. The denotations of those DCS trees were then used to predict names of media files as answers to the questions. Let us now see some of the DCS trees (logical forms) generated by the semantic parser along with the corresponding questions (Table 4.2). The nodes of the DCS trees represent the nature of the denotations. **image** represents denotations containing names of only images, **video** represents denotations containing names of only videos, whereas ‘*’ represents denotations containing names of both images and videos.

<table>
<thead>
<tr>
<th>Table 4.2: Example queries and DCS trees of our DCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>“what is there behind postbank?”</td>
</tr>
<tr>
<td>“what is in front of MPI-INF?”</td>
</tr>
<tr>
<td>“what is there on the right of campus center?”</td>
</tr>
<tr>
<td>“what is there on the left of mensa?”</td>
</tr>
<tr>
<td>“how does this place look in november?”</td>
</tr>
<tr>
<td>“what happened here 5 days ago?”</td>
</tr>
</tbody>
</table>

It is interesting to note that the question “what happened here 5 days ago?” triggers only one predicate **view** and does not trigger any predicate for the temporal reference (“...5 days ago...”). This is because the predicate **view** is defined in such a way so as to take into account the queried-for date without
triggering the predicate \texttt{day} (See \texttt{view1(B)} in Table 4.1). Although such a DCS tree does not impart the underlying sentence structure, this could not be avoided because of the limitation that the semantic parser is not parametrized – that is, it does not take words from sentences as arguments to predicates and infer meanings accordingly. The details of the challenges faced and the limitations are discussed in Chapter 7.

The semantic parser tries to unify\textsuperscript{7} proper nouns in sentences to unique names of the entities in the database. However contextual queries of the form “Which is the nearest cafe?” does not contain any proper nouns and so were problematic to handle. A possible solution was to add a second id to all predicates describing the type of entity it embodied. For example, in the predicate \texttt{cafe(’icoffee’,49.2557,7.0458)} an extra field was added – \texttt{cafe(’coffee’, ’cafe’,49.2557,7.0458)}. However we still get unsatisfactory results for these queries because the prediction is based on comparison of average values of GPS coordinates of the user and the entities in the database. Also, our media files are not annotated with the classes of objects that are visible in them, leading to failure in picking only those images which show the nearest cafe (for example). As a result it retrieves all images and videos that show areas geographically closest to the user at the time of sending the query.

We show a few working examples in Table 4.3. This concludes the discussion about our extensions to Percy Liang’s existing semantic parser. In the following sections we will throw light on the implementation aspects of our complete architecture of contextual media retrieval using natural language queries.

\section*{4.5 Client-Server Communication}

The Google Glass (or any other wearable device/smart phones) has limited storage space and computing power. Our contextual media retrieval system however required a huge storage space for storing the collective memory and considerable amount of computing power for the working of the semantic parser. For this reason, we implemented a client-server communication model to demonstrate our system. The Google Glass is the client with which users would interact to send natural language queries and receive retrieved media. The Glass would then communicate with a Python server running the semantic parser.

An overview of the work-flow implemented in our complete query-retrieval system is shown in Figure 4.3.

To exemplify the client-server model for our query-retrieval system, we resorted to the most popular form of Inter-Process Communication (IPC) – stream sockets. Stream sockets use TCP and implement a “connection oriented” semantics allowing guaranteed passage of data from one process to another in the same order as they were sent.

In the following sections we will discuss the functionality of the client and the server.

\textsuperscript{7}The way in which Prolog matches two terms is called unification. (http://www.dai.ed.ac.uk/groups/ssp/bookpages/quickprolog/node12.html)
4.5.1 Python Server

A simple Python script was set up on the server. It opens a port and listens for client connections indefinitely (step 0 in Figure 4.3). Once a client connects to the server, it performs the following functions:

- Creates the dynamic database (as explained in the previous chapter) containing the current location of the user and the temporal reference (if any) in the query (step 5 in Figure 4.3).
- Runs the trained semantic parser on the query (step 6 in Figure 4.3).
- Fetches the result (media file) predicted by the semantic parser from the collective memory (step 7 in Figure 4.3).
- Sends the media file (image/video) to the client (step 8 in Figure 4.3).

4.5.2 Google Glass Client

The operating system of the Google Glass is Android 4.4.2. It can thus be imagined as a wearable equivalent of a modern day smart-phone running the same operating system. Developing an app for the Google Glass was hence very similar to developing an app for a smart-phone. However, an add-on to the Android SDK, the Glass Development Kit (GDK) presents more flexibility for
4.5. CLIENT-SERVER COMMUNICATION

development in the form of dedicated APIs for Glassware (apps for the Google Glass). The GDK was used to develop the client-side application for our query-retrieval system. To make the functioning of the app mostly hands-free, the app was programmed to be initiated with the voice command “Ok Glass Explore Nearby” (step 1 in Figure 4.3).

After the initiation of the app it performs the following functions:

- Starts the Glass’s Speech Recognizer. Prompts the user to speak his/her query, recognizes the speech of the user and transcribes it to text (step 2 in Figure 4.3).
- Fetches the current GPS location of the user from the Glass’s location sensor and fetches the user’s viewing direction from the Glass’s orientation sensor (step 3 in Figure 4.3).
- Connects to the server; sends the query (natural language spatio-temporal question) and metadata (GPS location and viewing direction of the user) to the server (step 4 in Figure 4.3).
- Receives the media file (image/video) returned by the server as an answer to the query; displays the image or plays the video on the GUI of the app (goal(G) in Figure 4.3).

It is worth mentioning that the in-built Speech Recognizer of the Google Glass (or any Android device) is not robust to background noise and imperfections in pronunciations. Therefore it wrongly recognizes and transcribes the utterances in many cases. All our evaluations of the system is under the assumption that the voice query is correctly recognized.

Figure 4.4 shows the GUI of our Google Glass client application.

In this chapter we described the complete framework for our egocentric media retrieval application named *Xplore-M-Ego*. In the following chapter we will discuss all the user studies that we performed to train and evaluate our system.
Table 4.3: Qualitative results of our media retrieval system

<table>
<thead>
<tr>
<th>Query</th>
<th>Retrieved Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>“What building is to the left of MPI-SWS?”</td>
<td><img src="image" alt="Retrieved Images" /></td>
</tr>
<tr>
<td>“What is behind mensa bus stop?”</td>
<td><img src="image" alt="Retrieved Images" /></td>
</tr>
<tr>
<td>“What is in front of bus terminal?”</td>
<td><img src="image" alt="Retrieved Images" /></td>
</tr>
<tr>
<td>“What is near MPI-INF?”</td>
<td><img src="image" alt="Retrieved Images" /></td>
</tr>
<tr>
<td>“What did this place (MPI-INF) look like in December?”</td>
<td><img src="image" alt="Retrieved Images" /></td>
</tr>
<tr>
<td>“What is beside unimarkt?”</td>
<td><img src="image" alt="Retrieved Images" /></td>
</tr>
<tr>
<td>“What is to the right of MMCI?”</td>
<td><img src="image" alt="Retrieved Images" /></td>
</tr>
</tbody>
</table>
4.5. CLIENT-SERVER COMMUNICATION

(a) Starting screen of the Google Glass

(b) Starting app with voice trigger

(c) The main screen of the GUI

(d) Speech Recognizer waiting for natural language utterance

(e) Voice query recognized and transcribed to text

(f) Semantic parser retrieves answer while the user waits

(g) Result retrieved and displayed to user

Figure 4.4: GUI of our Google Glass app
Chapter 5

User Study

Humans are inherently inconsistent in their perception of directions and idea of reference frames [41]. The nature of understanding/speaking English questions also has variations based on a person’s socio-cultural background. Hence, a system relying on fixed question templates and a particular set of rules to resolve spatial references does not guarantee high accuracy. A satisfactory result for one person may prove to be irrelevant for another. To better understand these perceptual biases due to swaying social conventions and yet efficiently analyze the system, a series of user studies were conducted. This chapter is dedicated to the description of these user studies.

5.1 Relevance Feedback

5.1.1 Human evaluation of model trained with synthetic data (SynthModel)

The semantic parser was trained with synthetically generated training and test question-answer pairs (Section 3.3.1) to begin with. These questions, as already discussed, followed fixed templates. The answers to spatial references were generated according to a canonical reference frame where “front of” meant “north of”, “behind” meant “south of”, “left of” meant “east of” and “right of” meant “west of”.

We tested this synthetically trained model of the semantic parser (SynthModel) on a set of real questions collected from human participants (Section 3.3.2).

In the user study, participants were asked to judge whether the predictions (results returned by the semantic parser) for this set of questions were relevant or irrelevant. The result of the evaluation is discussed in Section 6.1.

To capture the inconsistencies and ambiguities in language and reference frames associated with various social conventions, we selected our participants carefully from a wide range of background (based on country of origin, mother
CHAPTER 5. USER STUDY

tongue and the language(s) (s)he was most fluent in). The details of the participants can be found in Table 5.1.

Table 5.1: User Study 2 (Human evaluation of SynthModel)– Participant details:

<table>
<thead>
<tr>
<th>User</th>
<th>Age</th>
<th>Gender</th>
<th>Nationality</th>
<th>Mother Tongue</th>
<th>Most fluent language</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>24</td>
<td>M</td>
<td>Indian</td>
<td>Telugu</td>
<td>Telugu</td>
</tr>
<tr>
<td>User 2</td>
<td>25</td>
<td>F</td>
<td>Iranian</td>
<td>Iranian</td>
<td>Iranian, Persian</td>
</tr>
<tr>
<td>User 3</td>
<td>24</td>
<td>F</td>
<td>German</td>
<td>German</td>
<td>German, French</td>
</tr>
<tr>
<td>User 4</td>
<td>25</td>
<td>F</td>
<td>Indian</td>
<td>Bengali</td>
<td>Bengali</td>
</tr>
<tr>
<td>User 5</td>
<td>28</td>
<td>M</td>
<td>Egyptian</td>
<td>Arabic</td>
<td>Arabic</td>
</tr>
</tbody>
</table>

5.1.2 The “human-in-the-loop” training

In an attempt to not restrict the vocabulary of the system and to bring in enough variations in the nature of questions, we collected the “real data” (Section 3.3.2) from human participants. The users also paired these questions with images and/or videos that they would expect as retrieved results. However, it was not possible to train the model automatically using this set of question-answer pairs because of the underlying inconsistencies of reference frames and structure of English sentences. Hence, we resorted to “human-in-the-loop” training of the semantic parser with these real questions.

Five users trained the model independently with 500 real-life questions. The process involved the use of a web interface to type the questions and to choose the correct answer from a list of predictions generated by the semantic parser. These five models hence trained captured the variations in language and also the variations in human perception of reference frames. To understand this with an example, for questions with the spatial reference “front of”, if a human trainer consistently chose images which were to the east of a particular entity (according to his/her direction conventions), the semantic parser would learn to associate “front of” to the predicate leftof(A,B) (since leftOf(A,B) was set to mean east of in our definitions).

Five different models of the semantic parser resulted from this training procedure. Their training accuracies and corresponding analyses are discussed in Section 6.2.

The details about the participants can be found in Table 5.2.

5.1.3 Human evaluation of the model trained with real data (RealModel)

From the five different models trained by human participants we selected the best model based on training performance. This model was tested on a set of real questions (Section 3.3.2). Human participants were asked to provide
5.1. RELEVANCE FEEDBACK

Table 5.2: User Study 3 (”human-in-the-loop” training of the semantic parser) – Participant details:

<table>
<thead>
<tr>
<th>User</th>
<th>Age</th>
<th>Gender</th>
<th>Nationality</th>
<th>Mother Tongue</th>
<th>Most fluent language</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>25</td>
<td>F</td>
<td>Indian</td>
<td>Kannada</td>
<td>English</td>
</tr>
<tr>
<td>User 2</td>
<td>28</td>
<td>M</td>
<td>Iranian</td>
<td>Iranian</td>
<td>Iranian, Persian</td>
</tr>
<tr>
<td>User 3</td>
<td>24</td>
<td>F</td>
<td>Indian</td>
<td>Tamil</td>
<td>Tamil</td>
</tr>
<tr>
<td>User 4</td>
<td>25</td>
<td>M</td>
<td>Polish</td>
<td>Polish</td>
<td>Polish, English</td>
</tr>
<tr>
<td>User 5</td>
<td>24</td>
<td>F</td>
<td>Iranian</td>
<td>Iranian</td>
<td>Iranian, Persian</td>
</tr>
</tbody>
</table>

relevance feedback on the retrieved results. The procedure of the study was similar to the one for SynthModel which was described in Section 5.1.1.

The result of the evaluation will be discussed in a later section (Section 6.2). The details about the participants are shown in Table 5.3.

Table 5.3: User Study 4 (Human evaluation of RealModel) – Participant details:

<table>
<thead>
<tr>
<th>User</th>
<th>Age</th>
<th>Gender</th>
<th>Nationality</th>
<th>Mother Tongue</th>
<th>Most fluent language</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>25</td>
<td>F</td>
<td>Indian</td>
<td>Bengali</td>
<td>Bengali</td>
</tr>
<tr>
<td>User 2</td>
<td>26</td>
<td>M</td>
<td>Iranian</td>
<td>Iranian</td>
<td>Iranian, Persian</td>
</tr>
<tr>
<td>User 3</td>
<td>25</td>
<td>M</td>
<td>Pakistani</td>
<td>Urdu</td>
<td>Urdu</td>
</tr>
<tr>
<td>User 4</td>
<td>25</td>
<td>F</td>
<td>German</td>
<td>German</td>
<td>German</td>
</tr>
<tr>
<td>User 5</td>
<td>24</td>
<td>F</td>
<td>Iranian</td>
<td>Iranian</td>
<td>Iranian, Persian</td>
</tr>
</tbody>
</table>

5.1.4 Human evaluation of temporal and contextual query-retrieval

So far all our models and the evaluations thereof consisted only of spatial relations in the training and test question-answer pairs. This was because of the following reasons:

- Temporal references are always dynamic - in that the answers vary according to the time of the query - and relative to the user’s location, hence also contextual.

Example:

“What happened here 5 days ago?”
“How did this place look in December?”

So, it was inconsequential to either generate such questions-answer pairs synthetically or to train the semantic parser with them.

- It was insignificant to evaluate the performance of the system based on temporal questions because the human participants would not have been
able to decide whether a prediction was correct or incorrect (because, it is not possible for a person to know what happened at a particular place at a specific time in the past).

- It was difficult to integrate spatial relations in question-answer pairs with user’s heading information (contextual spatial queries) since we did not have information about the user’s heading direction at the time of query. The training and test question-answer pairs followed a canonical reference frame.

Therefore, the appraisal of the system based on temporal and/or contextual queries, being highly subjective, had to be conducted in a separate module. This judgment is primarily based on common sense knowledge [21]. For example, if images retrieved for the question “How did this place look in December?” show snow covered terrains, there is a high possibility that the user would opine it to be a relevant retrieval.

Regarding the contextual spatial queries, an interesting study of user behavior was to observe the difference of opinion of the same user in two settings:

1. Taking into account the user’s heading direction. In this setting, the user was given a map of the Saarland University campus with hypothetical locations and heading directions. A sample of the map is shown in Figure 5.1.

Figure 5.1: Map of the Saarland University campus showing hypothetical location of users (location icons) and viewing direction at each location (arrow with direction labels; N–North, S–South, E–East, W–West)
5.2. PROSPECTIVE REAL-LIFE QUESTIONS

2. Without taking into account the user’s heading direction (using a canonical/world reference frame).

Five users were made to look at retrievals for a set of 25 temporal and contextual questions in the above settings and judge them as relevant or irrelevant. The results of this user study is depicted in Section 6.3. The details of the participants are shown in Table 5.4.

Table 5.4: User Study 6 (Evaluation of the semantic parser on temporal and contextual queries)– Participant details:

<table>
<thead>
<tr>
<th>User</th>
<th>Age</th>
<th>Gender</th>
<th>Nationality</th>
<th>Mother Tongue</th>
<th>Most fluent language</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>24</td>
<td>M</td>
<td>German</td>
<td>German</td>
<td>German</td>
</tr>
<tr>
<td>User 2</td>
<td>25</td>
<td>F</td>
<td>Iranian</td>
<td>Iranian, Persian</td>
<td>German</td>
</tr>
<tr>
<td>User 3</td>
<td>24</td>
<td>M</td>
<td>German</td>
<td>German</td>
<td>German</td>
</tr>
<tr>
<td>User 4</td>
<td>23</td>
<td>M</td>
<td>Indian</td>
<td>Telugu</td>
<td>Hindi</td>
</tr>
<tr>
<td>User 5</td>
<td>25</td>
<td>F</td>
<td>Uzbek</td>
<td>Uzbek</td>
<td>Uzbek</td>
</tr>
</tbody>
</table>

5.2 Prospective real-life questions

In spite of getting satisfactory results with our query-retrieval system, the limiting factor of our application was that it was largely domain-specific (only pertaining to spatio-temporal questions). We were curious to know about what kind of domain-indefinite questions would regular users like the system answer. The following are some of the questions that we came across which had no spatio-temporal references:

- What is the fastest way to go to MPI from here?
- What is the vegetarian menu for lunch today at the university cafeteria?
- Which is the highest point near the university?
- When was this structure (user looking at a particular structure) built? What is it’s meaning?
- Is it crowded at the coffee shop?
- Are there any vegetarian sandwiches left in the cafeteria?
- What are the opening hours of the university library?
- When and where will the talk by Prof. XYZ be held?
- Where did event X take place a month ago?
- How many students attend the ABC lecture at 9:00?
• Are there any proposed Yoga class in the university sports program?

These questions range in a variety of domains and would require high level data acquisition methods and/or image understanding for successful retrieval. Also, the efficiency of any state-of-the-art semantic parser to understand these varied free-flow natural language queries is questionable. Integrating image processing or newer natural language processing techniques was out of the scope of this project. However, having studied the generic wish or need of regular users with relation to a query-retrieval system, this provides new directions for future research and development.

With this we come to an end of Part II of this thesis where we have defined our data collection procedure, elaborated on our complete architecture and discussed the various user studies we performed. Having laid the foundation for rigorous discussions, we will now proceed to present the results of our evaluations and analyze the various aspects of our media retrieval system.
Part III

Results and Conclusion
Chapter 6

Evaluation

In this chapter we would like to discuss, with graphical representations, the results of evaluations of the various models of the semantic parser.

The semantic parser can be analyzed in different modalities. These are the areas we looked into to evaluate the performance of the parser:

- Effect of beam size
- Effect of training
- Effect of restricted vocabulary
- Use of synthetically generated and real-life question-answer pairs for training and testing

The semantic parser of Percy Liang [8] that we extended in our work used beam search to generate a number of predictions for every natural language question. Beam search is an optimization of best-first-search that reduces the search space by limiting the number of nodes expanded. Quite intuitively, the performance of the semantic parser improved by increasing the beam size. An untrained model of the parser using the basic lexicon $L$ (Section 4.3) had an accuracy of 3.8% with a beam size of 100 which increased to a maximum of 6%. A beam size of 500 was found to be optimal (Figure 6.1). All further training and testing of the parser were hence carried out with a beam size of 500.

The next interesting region to investigate was how efficiently the parser understood uninhibited English natural language. Using lexicon $L$, that is, without restricting the vocabulary by hard-coding certain words into fixed predicates, the untrained model of the parser had a prediction accuracy of 6%. In an attempt to increase the prediction accuracy of the parser, we restricted the scope of the natural language queries by mapping certain words to pre-defined predicates (lexicon $L+$). For example, the phrase left of was mapped to predicate $leftOf(A,B)$, the phrase behind was mapped to predicate $behind(A,B)$ and so on. More details about the formulation of these predicates can be found in
Section 4.3. The untrained model using lexicon $L$+ had a prediction accuracy of 11.23% on a set of synthetically generated question-answer pairs.

![Effect of Beam Size on Prediction Accuracy](image)

Figure 6.1: Effect of increasing beam size on prediction accuracy (using $L$+)

We then wanted to study the effect of training on the performance of the parser. After training with 500 synthetically generated question-answer pairs, the accuracy of the parser (using lexicon $L$) was 17.9%. This kind of low performance draws a parallel with Malinowski et.al [21] who also used unrestricted natural language questions and faced challenges similar to our present work. Using the aforementioned augmented lexicon $L$+ the parser showed a remarkable improvement in prediction accuracy, elevating to about 47% (Figure 6.2). In all our further analysis of the semantic parser we use the augmented lexicon $L$+.

47% is the maximum attainable prediction accuracy of our system since further increase in the number of training examples fails to increase the accuracy. This might seem quite low at first if we happen to compare it with the Percy Liang system [8]. However, it is important to note that Liang et al. [8] had no inherent ambiguities (like spatial references and reference frames) in their queries. They worked with a static database of geographic facts. Our system, with ambiguous spatial references, varying social conventions of reference frame resolution and a dynamic egocentric environment, make the task of prediction a lot more complex, accounting for the lower performance.

We recorded the accuracy of prediction of the semantic parser in two independent cases:
6.1. SYNTHMODEL

- When trained with the synthetically generated training examples
- When trained with the real training examples

Figure 6.2: Effect of increasing number of training examples on prediction accuracy

The following sections lay out the results of our evaluations. We call the model trained with synthetic data SynthModel and the one trained with real data RealModel.

6.1 SynthModel

Out of the 500 questions in the test data set, the semantic parser model trained with synthetically generated question-answer pairs retrieved media files for 159 questions (31.8%). The rest of the questions had other types of retrieved results (for example, textual information). Let us denote the number of queries with media retrievals as $q_m$ and the number of queries with textual retrievals as $q_t$. Among $q_m$, the number of queries which had relevant retrievals will be denoted as $q_r$. From the human evaluation of this model (Section 5.1.1), 50.2% of $q_m$ were relevant. This accounted for 15.88% prediction accuracy of the semantic parser. In other words, the SynthModel had an average precision\(^1\) of 50.2% and an average recall\(^2\) of 15.88%.

\(^1\) average precision = \( \frac{q_r}{q_m} \)

\(^2\) average recall = \( \frac{q_r}{q_m + q_t} \)
The graphical representation in Figure 6.3 shows the result of the human evaluation of the performance of the parser trained with synthetic data. The mean relevance was 50.2% with a standard deviation of 12.3. The evaluation of user 5 was found to be an outlier with only 30.2% relevance. Based on our observation, the user was not consistent with his own convention of reference frame. For example, for questions of the form “What is on the right of a building?”, he sometimes considered his right hand side when facing the building while at other times he considered his right hand side with his back towards the building. This mismatch with the evaluation of the other 4 users might be because of hugely different social conventions (leading to variations in the latent notion of reference frames) or other psychological factors. Malinowski et al. [21] also point to this inconsistency in human answering of natural language questions.

![Chart showing Precision of SynthModel](chart.png)

Figure 6.3: Human evaluation of model trained with synthetic data

### 6.2 RealModel

We had discussed the necessity of “human-in-the-loop” training of the semantic parser in Section 5.1.2. The five different models trained by this process had

---

3Note that all users were moderately inconsistent in their decisions. Stephen C. Levinson explains how human population considerably vary in their spatial thinking in his book *Space in Language and Cognition* [41].
training accuracies ranging from 42.6% to 48.8%. The best model was chosen among these five models based on training performance and human participants were asked to provide relevance feedback on the results retrieved by the semantic parser.

The number of queries for which the semantic parser was able to predict media files as answers ($q_m$) increased from 31.8% (SynthModel) to 74.5%. The results of the human evaluation of the best trained model (according to training accuracy) show that the average precision is 37.38% as compared to 50.2% in case of SynthModel (Figure 6.4). However, the recall increased from 15.88% (SynthModel) to 26.67% (Figure 6.5).

![Precision of RealModel](image)

Figure 6.4: Human evaluation of model trained with real data

From the results of this human evaluation we also made an analysis of the difficulty of the problem itself and the robustness of the model. We grouped the questions in our test set in the following categories:

- **(5,0)**: judged as relevant by all 5 users
- **(4,1)**: judged as relevant by 4 users and as irrelevant by 1 user
- **(3,2)**: judged as relevant by 3 users and as irrelevant by 2 users
- **(2,3)**: judged as relevant by 2 users and as irrelevant by 3 users
- **(1,4)**: judged as relevant by 1 user and as irrelevant by 4 users
• (0,5) : judged as irrelevant by all 5 users

Figure 6.6 depicts the result of this analysis. The numbers in the middle region of the graph clearly point out the prominent disagreement between human participants. This stems from, as discussed earlier, the inherent inconsistencies with regards to reference frames and the variability in spoken language. This also hints towards the difficulty of the problem at hand since satisfactory answers for one user may be unsatisfactory for others. However, the model, trained by a human participant through relevance feedback, seems to have learned some of the inconsistencies in the process - depicted by the side-heavy graph indicating agreement between different users in most of the cases.

6.3 Temporal and Contextual Queries

In this section we will discuss the results of the user studies conducted to better understand the various notions of reference frames that individuals have and to generate a common consensus about the relevance of the retrieved results to temporal queries. The study was conducted by asking 5 users to opine about the relevance of results retrieved by the parser to 25 spatio-temporal contextual
6.3. TEMPORAL AND CONTEXTUAL QUERIES

![Relevance feedback statistics](image)

Figure 6.6: Agreement and disagreement about relevance of retrieved results among human users

questions - 5 per hypothetical location (Figure 5.1). The questions can be found in Table 6.1.

<table>
<thead>
<tr>
<th>Table 6.1: Temporal and contextual queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>What happened here one month ago?</td>
</tr>
<tr>
<td>How does this place look in December?</td>
</tr>
<tr>
<td>What is there to the right of this place?</td>
</tr>
<tr>
<td>How did this place look in November?</td>
</tr>
<tr>
<td>What is there behind this place?</td>
</tr>
</tbody>
</table>

The user study was conducted under two different settings – using a canonical reference frame and an user-centric reference frame. An user gave relevance feedback in both settings and the mean relevance scores were calculated to understand the broader picture.
Canonical frame of reference

As the canonical reference frame in our project, we used the earth’s coordinate axes or the geomagnetic reference frame. It is depicted in Figure 7.1.

User-centric frame of reference

For the user-centric reference frame we assumed that users would mean their physical “left hand side” for the spatial relation left of and so on. We modified the user queries according to this assumption and the information about which direction they were looking at the time of the query. More information as to how we integrate Egocentrism into our media retrieval system has been given in the next chapter under Section 7.1.

We show the result of this evaluation in Figure 6.7.

![Figure 6.7: Relevance Feedback on temporal and contextual queries using a Canonical reference frame and a user-centric reference frame](image)

The mean relevance was found to be 56% while using our canonical reference frame and 49.6% while using the user-centric reference frame. For some users the difference in reference frames did not lead to a difference in judgement of the relevance of the retrievals. Other users slightly preffered the retrievals according to the canonical reference frame. From this observation we revisit the hypothesis that humans are not hugely interested in accurate reference directions. They most often relate to spatial maps of places registered in their minds and these spatial maps do not conform to a particular reference frame (that is, they refer to some reference frame which is not consistent). The standard deviation of user opinions was 15% using the canonical reference frame and 10.4% while
using the user-centric reference frame. Although use of the canonical reference frame yielded more accuracy in general, there was a much higher difference in opinion among users than while using the user-centric reference frame. This finding can spark further discussion about which reference frame is best to use in applications involving the use of spatial relations.

It is interesting to note that validation of the answers to the queries is subject to the necessity that the participants are familiar with the spatial scope of the application. Earlier we mentioned that we were able to validate the questions with spatial reference while we had to resort to approximation and common sense knowledge for validating questions with temporal references. This statement is true for the limited spatial scope of our project. If we go beyond this limitation and consider the whole world as a basis, the questions with temporal references can as well be validated if they refer to a famous place. For example, if retrievals for the question “What happened at Brandenburger Tor on 31st December?” show fireworks, the user would know for certain that the results are relevant. On the other hand, if the retrievals for the question “What is behind the Taj Mahal?” show the river Jamuna, it is difficult to validate the results if one is not familiar with the fact that the river Jamuna flows behind the Taj Mahal.

The following tables summarizes the quantitative results of our evaluations.

**Table 6.2: Use of Lexicons $L$ and $L+$**

<table>
<thead>
<tr>
<th>Lexicon Type</th>
<th>Untrained Model</th>
<th>Trained Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Lexicon $L$</td>
<td>6%</td>
<td>17.9%</td>
</tr>
<tr>
<td>Augmented Lexicon $L+$</td>
<td>11.23%</td>
<td>47%</td>
</tr>
</tbody>
</table>

**Table 6.3: Average Precision and Average Recall of semantic parser models**

<table>
<thead>
<tr>
<th>Model</th>
<th>Average Precision</th>
<th>Average Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>SynthModel</td>
<td>50.2%</td>
<td>16%</td>
</tr>
<tr>
<td>RealModel</td>
<td>37.38%</td>
<td>28%</td>
</tr>
</tbody>
</table>

**Table 6.4: Relevance feedback using different reference frames**

<table>
<thead>
<tr>
<th>Reference Type</th>
<th>Canonical</th>
<th>User-centric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>56%</td>
<td>49.6%</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>15%</td>
<td>10.4%</td>
</tr>
</tbody>
</table>
Constrained by the limited scope of our project, the user studies we conducted proved that spatio-temporal question-answering is a challenge not only in the area of natural language processing but also for holistic architectures aiming to bind language and perception. In the following chapter we would engage in a deeper discussion about these challenges and limitations and come up with new research directions to overcome them.
Chapter 7

Discussion

So far we have seen the theoretical background and implementation details of our work. Although we have given an initial analysis of the challenges faced in the course of the project and the limitations of the semantic parser, a thorough discussion about the same was omitted for later. In this chapter we bring forward a detailed discussion about how we overcame certain challenges and the drawbacks of the semantic parser leading to the restricted scope of our application. We also propose some future directions to the present and related research works.

7.1 Challenges

Converting a dynamic world to a static world

The main challenge of this work was to adapt a semantic parser programmed to work with a static world to handle a dynamic egocentric environment that was a basic requirement for our project. This was done by making the overall architecture dynamic while keeping the world of the semantic parser static per query. As described in Section 4.3, we implemented a dynamic database \( w_{du} \) which is re-written every time a query is made from the wearable device, thus updating the location of the user, his/her heading direction and the temporal reference in the query. After this pre-processing step where \( w_{du} \) is reset, the semantic parser works with a static database of facts \( w = w_s + w_d \) (where \( w_d = w_{du} + w_{dm} \) (Section 4.3)).

Integrating Egocentrism into media retrieval

Taking into account the context of the user was the next big challenge. Contextual questions which just referred to the present location (for example “how did this place look in November?”) was easier to handle because it involved mapping the user’s current GPS location (that is being written in the dynamic database \( w_{du} \)) to those of the existing media files. However, contextual questions which involved an egocentric reference frame (for example, spatial relations
like “front of”, “behind”, “left of” and “right of”) was impossible to handle by the semantic parser. This is because humans do not adhere to any consistent reference frame and most often go by their physical ‘left hand’ for “left of” etc. Naturally it is impossible to understand the hidden intent of such a human natural language question by a machine. This was tackled by programming the semantic parser to follow a canonical allocentric reference frame (the geomagnetic reference frame), assuming that the direction in which the human user faces is the local north, and changing the spatial reference of the query internally in the pre-processing step. The following figure explains this concept more clearly.

![Diagram](image)

Figure 7.1: The canonical reference frame used in our work and how we integrate egocentrism in media retrieval

From the above figure, if the user faces east and queries for “What is there in front of MPI-INF?”, the question would be changed during pre-processing to “What is there on the right of MPI-INF?”. The semantic parser would predict answer for this changed question. For simplicity we have narrowed down to only the four basic heading directions - north, south, east and west. Also, in our augmented lexicon $L+$ we only have prototype predicates for the spatial relations “front of”, “behind”, “left of” and “right of”. However, questions collected from real users also have spatial relations like “opposite to”, “ahead of”, “beside” etc. During the human-in-the-loop training of the semantic parser, the parser learns to associate these relations to one of our spatial relation predicates (Table 4.1). Working with spatial relations also proved to be a challenging problem in natural language research in the past [42, 43, 44, 21].

**Increasing the coverage of the static database**

The static database contains information like name and GPS location of geo-
graphic entities in our spatial scope. We have extracted this information from the OpenStreetMap as described in Section 3.1. However, an entity (for example, a building) can be referred to by a number of names - its full name, an acronym of its name, its popular name or its name in a different language. For example, the database entry corresponding to Max Planck Institute for Informatics contained the German name of the institution. The data from OpenStreetMap did not contain this variation in names of the entities. This led to failure of the semantic parser to answer questions which referred to a building by a name different from that by which it is registered in the database. Since the semantic parser fails at reference resolution, it cannot map an acronym like ‘MPI’ to the actual entity. We had to account for this missing data by employing a human annotator to add all possible common names for each entity in the database. However, such human annotation may still be incomplete.

Scalability

In semantic parsing a natural language question \(x\), program induction is the process of finding a logical form \(z\) over an exponentially large space of possibilities that produces the target answer \(y\) [8]. The exponential space of possibilities arise due to a large number of predicates (either entities in the database or rules that define relationships between nodes in DCS trees). In order to stay within the scalable limits of the semantic parser we reduced our spatial scope to only the Saarland University campus. We also considered a restricted number of geographic entities as representative of our static database \(w_s\). For example, we left out less significant entities like ‘benches’, ‘recycling’, ‘vending machines’ etc.

Handling temporal queries

Questions of the form “what happened here 5 days ago?” , “how did this place look 1 month ago?” etc., in the absence of any parametrization mechanism (Section 7.2), had to be parsed in the pre-processing step to infer the temporal reference in the utterances. The temporal reference was stored in the dynamic database in the form of a timestamp (for example, \(\text{day(20140511)}\)). Also, the phrase “what happened” was mapped to trigger the predicate \(\text{view(A)}\), where \(A\) is the list of media files having the same timestamp as that in the predicate \(\text{day()}\). Clearly, we do not rely on the parser at all for ‘parsing’ the meaning of the question in this case apart from triggering the predicate \(\text{view(A)}\), which is trivial. However, this was the only way to handle these temporal queries with the present capabilities of the semantic parser.

7.2 Limitations of the Semantic Parser

Scalability

Despite bypassing it in our project by reducing the geographic scope, scalability remains a serious limitation of the semantic parser of Percy Liang. Increase in the number of predicates – by increasing the number of facts in the database, increase in constraint satisfaction rules, or using big complicates sentences –
renders the parser incapable of producing the logical forms and hence leads to failure in predicting answers.

**Non-parametrization**
The semantic parser does not support parametrization of predicates. This means we cannot define a predicate which takes a parameter from the natural language query and evaluates the constraint in light of this parameter. For example, queries of the form “what happened here 5 days ago?” would better generalize if we could define a parameter, say, $\text{time}(p,q)$ where $p \in \{1, 2, 3,...\}$ and $q \in \{\text{hours}, \text{days}, \text{weeks}, \text{months}, \text{years}\}$.

**Reference Resolution**
Reference resolution is the determination of which word(s)/phrase(s) refer to some other word/phrase [45]. Not being adept in reference resolution is a major limitation of the semantic parser. Human beings by nature refer to the same entities with various names, the most common being acronyms. A built-in method to recognize and resolve acronyms in English language utterances would prove to be a performance boost for the semantic parser. In the absence of such a functionality, NLP research would have to depend on manual annotations.

**Incorrect POS tags**
The assignment of POS tags to individual words in a sentence is sometimes faulty, resulting in triggering of wrong predicates and hence prediction of wrong logical forms. For example, in absence of hard-coding, the word ‘front’ in the question “what is in front of ...?” is identified as a noun resulting in the mapping $\text{front} \rightarrow \text{image/video}$ (according to our lexicon $L$, POS noun triggers the predicates $\text{image}$ and $\text{video}$). The DCS tree shown below demonstrates this fact:

The predicted logical form (DCS tree) for the example query is incorrect. Such incorrect derivations may occasionally have correct answers from which the semantic parser learns wrong associations. This learning naturally is derogatory for the general performance since faulty DCS trees will not give correct answers for all questions.

### 7.3 Accuracy of Performance

The low accuracy of our DCS can be accounted for by a lot of factors. Among those which were already mentioned before are inclusion of spatial references in English questions, innate inconsistencies and ambiguities in reference frame determination and reference resolution among human beings and inclusion of temporal references which surpassed any possible performance measure.

Adding to the already existing fuzziness of our architecture, the use of GPS coordinates for the retrieval of relevant results proved to be discordant. There were discrepancies in the readings of the GPS coordinates from location sensors.
of different android devices and most often they did not match those extracted from the OpenStreetMap. This resulted in swaying localization of the media files in our collective memory leading to incorrect retrievals. Moreover, media files visually representing, say, entity A, was captured from a little far away (location B) from A thus registering the GPS coordinate of B in the metadata of the media files showing A. However, querying for media files showing A tried to match the GPS location of A with the existing media files. This most often resulted in irrelevant retrievals.

Since matching the exact GPS coordinates for retrievals proved to be a failure, we devised a scheme for a rough localization by rounding off the GPS coordinates to the first 6 significant digits. For example, the GPS coordinate (45.257915,7.045783) was rounded off to (49.2579,7.0459). Although we loose information about the precise location of the entity/media file/user by this method, this yielded better retrieved results for the queries. Yet, since the retrievals for a question consisted of media files encompassing a diameter around the true location, most often they were less satisfactory. The following example demonstrates this problem:

Table 7.1: Example of a failure case caused due to matching GPS coordinates

<table>
<thead>
<tr>
<th>Query</th>
<th>Retrieved Images</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;What does MPI-SWS look like?&quot;</td>
<td><img src="image1.jpg" alt="Image 1" /> <img src="image2.jpg" alt="Image 2" /> <img src="image3.jpg" alt="Image 3" /></td>
</tr>
</tbody>
</table>
CHAPTER 7. DISCUSSION

In the example shown above, the retrieved images show areas around MPI-SWS rather than the building itself. This problem may be approached by applying scene understanding algorithms on the retrieved results for further refinement.

Despite the complexities leading to lower quantitative results of our system, general comments from human evaluators indicate that endeavors were taken in the right direction. In the absence of a suitable benchmark, we conjecture that the performance of our system is quite satisfactory.

7.4 Future Work

The future scope of this work can be broadly classified into 3 areas:

- Integration of image processing and computer vision methods for scene understanding (similar to Malinowski et al. [21]).

- Development of a better semantic parser in light of our discussions about its limitations.

- Development of more robust location sensors in devices used for capturing media.

- Generation of a consensus about reference frames for applications involving the use of spatial relations.

Integration of scene understanding would facilitate answering questions about the real world more efficiently. This would help in fine tuning the retrieved results by ignoring those images and videos which do not visually represent the true answer of the query. Scene understanding would also facilitate answering questions like “how many vegetarian sandwiches are left in the university cafeteria?” (Section 5.2). It could also be used to answer more complicated spatio-temporal questions like “Was there an ATM in front of the university cafeteria one year ago?”

A semantic parser which is scalable, supports parametrization and can parse spatio-temporal references in natural language utterances would help resolve most of the limitations of this and related research works.

We have noticed that one of the reasons for the low performance of our system is due to inaccurate localizations of media on the map. This is because, at the same location, different devices register different values of GPS coordinates. A more synchronized procedure for global positioning and more robust location sensors in the devices may prove to alleviate this problem.

It is perhaps impossible to eliminate the ambiguities that human beings have (because of various social conventions) with regards to reference frames. One way to tackle the problem is to implement a calibration algorithm in the query-retrieval system by means of which the parser would learn the direction conventions of individual users. However, since individual users are themselves inconsistent, this method would not increase the robustness of the system by a large margin. For the purpose of research and development, the scientific
community can come to a common consensus about the use of a canonical reference frame. This would in turn build a kind of benchmark against which past and future works can be compared.

For our present project, all the human evaluations were conducted in a lab setting because of limited infrastructure. We conjecture that human opinions about relevance of the retrieved results may vary if participants were to actually test the application in the open. The notion of reference frames can also then be studied more efficiently. Therefore we would like to conduct further experiments for testing our architecture in a real-world scenario where human participants will be allowed to interact with a wearable device (for example, the Google Glass) while walking around in the Saarland University campus and give relevance feedback on the results retrieved by the semantic parser. This study will also be important from the HCI point of view and may spark ideas about similar HCI applications.
Chapter 8

Conclusion

In this thesis we have proposed a novel architecture for media retrieval with natural language voice queries in a dynamic setting – Xplore-M-Ego. Our work brings forth a new direction to this paradigm by the introduction of egocentrism or user’s context in query-retrieval. Where traditional media retrieval systems used computer vision algorithms and retrieved media pertaining to scene contents, we have presented a system which retrieves media pertaining to questions about physical arrangements of entities in the user’s locality (outdoors). For this we have used a rough localization of the media files with the help of GPS coordinates. Constructively integrating the user-centric frame of reference (as desirable in an egocentric application) remained a problem and will probably always be ambiguous (as pointed out by various scientific articles about human behavioral studies with respect to reference frames). However, our research and user studies throw further light on the problem and may inspire future works in this area.

We have also presented a constructive criticism of the state-of-the-art semantic parser designed by Percy Liang (having extended the same for our work), the imminent challenges and the limitations due to it. We hope that such a discussion will trigger the development of more efficient and domain-independent semantic parsers. Although the limited scope of our work restricts us from calling it a general-purpose architecture, we feel we have nurtured the idea in the right direction and brought out satisfactory performance (with scope for minor improvements). Moreover, our discussions about the future scope of the work may be seed to more interesting applications designed for greater public aid and convenience.

We hope to have incited interest in the reader towards the importance and potential of our proposed collective memory, a dynamic-egocentric media retrieval system, the subjectivity and variability of human natural language, and the open problem of reference frame resolution among human beings.
CHAPTER 8. CONCLUSION
Appendix A

List of Abbreviations

**POS** Part-of-Speech
**CSP** Constraint Satisfaction Problem
**DCS** Dependency-based Compositional Semantics
**EM** Expectation Maximization
**RDF** Resource Description Framework
**OWL** Web Ontology Language
**SPARQL** SPARQL Protocol and RDF Query Language
**DOM** Document Object Model
**SDK** Standard Development Kit
**GDK** Glass Development Kit
**GPS** Global Positioning System
**GUI** Graphical User Interface
**IPC** Inter Process Communication
**TCP** Transmission Control Protocol
**HCI** Human Computer Interaction
Figure A.1: How Google Glass Works
## Appendix B

### List of Figures

1.1 Life-logging devices .................................................. 4
1.2 The Google Glass ....................................................... 5
1.3 Black-box representation of Xplore-M-Ego ......................... 6

2.1 Example of 3D reconstructed view of the location of images and the underlying scene (left) and the user interface (right) presented by Snavely et al. (figure taken from [9]) ........................................ 10
2.2 Overview of PhotoScope (figure taken from [9]) .................. 11
2.3 Overview of the Vidicontexs system (figure taken from [12]) . 11
2.4 User interface of the Vidicontexts system (figure taken from [12]) .................................................. 12
2.5 Overview of Videoscapes computation; portals are shown in green (figure taken from [11]) ............................................. 12
2.6 The interactive exploration mode of Videoscapes (figure taken from [11]) .................................................. 13
2.7 Probabilistic framework proposed by Malinowski et al. (figure taken from [21]) .................................................. 15
2.8 Examples of human-generated question-answer pairs illustrating inconsistencies (figure taken from [21]) .................................................. 16
2.9 Semantic net representation of the sentence “Stephen drives a blue car” (figure taken from [22]) .................................................. 17
2.10 An example of SPARQL query for finding events represented in RDF-OWL format (figure taken from [26]) .................................................. 17
2.11 Definition of coordinate axes with respect to motion in video frames (figure taken from [27]) .................................................. 18
2.12 An example of image retrieval with structured object queries (figure taken from [29]) .................................................. 18
2.13 An example of recursive merging of words (figure taken from [28]) .................................................. 19
2.14 Mapping of images and sentences into the same space by Socher et al. (figure taken from [30]) .................................................. 19

3.1 Saarland University campus from OpenStreetMap ............ 25
APPENDIX B. LIST OF FIGURES

3.2 GUI of DataCollectionApp ........................................... 26
4.1 Example of a DCS tree; denotations refer to a set of feasible solutions of a particular CSP instance (figure taken from [8]) .... 35
4.2 Probabilistic model of Liang et al. (figure taken from [8]) .... 36
4.3 Overview of the client-server architecture of our query-retrieval system ................................................................. 46
4.4 GUI of our Google Glass app ............................................ 49
5.1 Map of the Saarland University campus showing hypothetical location of users (location icons) and viewing direction at each location (arrow with direction labels; N–North, S–South, E–East, W–West) .......................................................... 54
6.1 Effect of increasing beam size on prediction accuracy (using $L+$) 60
6.2 Effect of increasing number of training examples on prediction accuracy ................................................................. 61
6.3 Human evaluation of model trained with synthetic data .... 62
6.4 Human evaluation of model trained with real data .......... 63
6.5 Percentage of relevant retrievals among all test queries for RealModel and SynthModel ............................................. 64
6.6 Agreement and disagreement about relevance of retrieved results among human users ............................................. 65
6.7 Relevance Feedback on temporal and contextual queries using a Canonical reference frame and a user-centric reference frame .... 66
7.1 The canonical reference frame used in our work and how we integrate egocentrism in media retrieval ........................................... 70
7.2 Example of faulty POS tagging by the semantic parser .... 73
A.1 How Google Glass Works ............................................. 80
## Appendix C

### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Related Work in a nutshell</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Natural Language Questions and collective memory</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>User Study 1 (Data Collection)– Participant details</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Definitions of predicates in our DCS</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>Example queries and DCS trees of our DCS</td>
<td>44</td>
</tr>
<tr>
<td>4.3</td>
<td>Qualitative results of our media retrieval system</td>
<td>48</td>
</tr>
<tr>
<td>5.1</td>
<td>User Study 2 (Human evaluation of SynthModel)– Participant details</td>
<td>52</td>
</tr>
<tr>
<td>5.2</td>
<td>User Study 3 (&quot;human-in-the-loop&quot; training of the semantic parser)–</td>
<td>53</td>
</tr>
<tr>
<td></td>
<td>Participant details</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>User Study 4 (Human evaluation of RealModel)– Participant details</td>
<td>53</td>
</tr>
<tr>
<td>5.4</td>
<td>User Study 6 (Evaluation of the semantic parser on temporal and</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>contextual queries)– Participant details</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Temporal and contextual queries</td>
<td>65</td>
</tr>
<tr>
<td>6.2</td>
<td>Use of Lexicons $L$ and $L+$</td>
<td>67</td>
</tr>
<tr>
<td>6.3</td>
<td>Average Precision and Average Recall of semantic parser models</td>
<td>67</td>
</tr>
<tr>
<td>6.4</td>
<td>Relevance feedback using different reference frames</td>
<td>67</td>
</tr>
<tr>
<td>7.1</td>
<td>Example of a failure case caused due to matching GPS coordinates</td>
<td>73</td>
</tr>
</tbody>
</table>
Bibliography


