

# INFORMATION LIFELOGGING: LEVERAGING EYE MOVEMENTS AND READING COMPREHENSION FOR EFFICIENT RETRIEVAL OF PREVIOUSLY ENCOUNTERED ON-SCREEN INFORMATION

Tu-Khiem Le, B.Sc.

A Dissertation submitted in fulfillment of the  
requirements for the award of  
Doctor of Philosophy (Ph.D.)

to the

**DCU**

Ollscoil Chathair  
Bhaile Átha Cliath  
Dublin City University

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervisors

*Asst. Prof.* Graham Healy

*Prof.* Cathal Gurrin


*Assoc. Prof.* Minh-Triet Tran (VNUHCM University of Science)

July 2024

---

# Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Sign:   
(*Tu-Khiem Le*)

Student No.: 19213727

Date: 24/07/2024

# Acknowledgements

First and foremost, I would like to express my deepest gratitude to my internal supervisors, Dr. Cathal Gurrin and Dr. Graham Healy, for their invaluable guidance, unwavering support, and constant encouragement throughout my PhD journey. Their expertise, insights, and mentorship have been instrumental in shaping my research and helping me navigate the challenges of this endeavor.

I am also profoundly grateful to my external supervisor, Dr. Minh-Triet Tran, who has been a guiding light in my academic career. His supervision during my Bachelor's thesis ignited my passion for research and motivated me to pursue a PhD degree. His continued support and belief in my abilities have been a constant source of inspiration.

I dedicate this thesis to my beloved father, who sadly passed away. His love, wisdom, and the values he instilled in me have been the bedrock of my life. Although he is no longer with us, I hope that this achievement would have made him proud.

To my dear mother, words cannot express how thankful I am for your unconditional love, care, and support. You have been my pillar of strength, always standing by my side and encouraging me to pursue my dreams. Your sacrifices and unwavering faith in me have been the driving force behind my success.

I am incredibly grateful to my elder brother, who has been my role model and mentor. His passion for programming ignited my own interest in the field, and he has generously shared his knowledge and experience with me. I am also deeply appreciative of his efforts in looking after our mother while I have been abroad.

---

pursuing this degree.

I would like to extend my heartfelt thanks to my friends, colleagues, and the various organizations that have supported me along the way. Your friendship, collaboration, and assistance have made this journey much more enjoyable and rewarding.

I am also grateful to the academic and administrative staff at Dublin City University for providing a stimulating and supportive environment for my research. The resources, facilities, and opportunities provided by the university have been essential to the successful completion of my PhD.

Finally, I would like to acknowledge the funding bodies and scholarships that have financially supported my research. Their generosity has allowed me to focus on my studies and has opened up incredible opportunities for personal and professional growth.

To everyone who has been a part of this journey, thank you from the bottom of my heart. Your support, guidance, and love have made this achievement possible.

# Table of Contents

<b>Table of Contents</b>	<b>iv</b>
<b>List of Figures</b>	<b>vii</b>
<b>List of Tables</b>	<b>ix</b>
<b>List of Abbreviations</b>	<b>xi</b>
<b>Abstract</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Lifelogging . . . . .	3
1.2 Personal Information Management . . . . .	5
1.3 Eye movement and Reading Analysis . . . . .	7
1.4 Hypothesis and Research Questions . . . . .	10
1.5 Research Contributions . . . . .	16
1.6 Research Limitations . . . . .	18
1.7 Thesis Outline . . . . .	21
<b>2 Related Work and Background</b>	<b>24</b>
2.1 Lifelogging and Lifelog Retrieval Systems . . . . .	25
2.1.1 Lifelog Challenges . . . . .	25
2.1.2 Lifelog Retrieval Systems at LSC . . . . .	29
2.2 Eye movements in Reading Comprehension . . . . .	43
2.2.1 The human eyes: Structure and Function . . . . .	43
2.2.2 Capturing eye movements: The Eye-tracker . . . . .	46
2.2.3 Basic Characteristics of Eye Movements and Application in Recognising Reading Strategies . . . . .	49
2.2.4 Eye Movements in Estimating Reading Comprehension . . . . .	51
2.3 Methods for Statistical Analysis . . . . .	57
2.3.1 Comparison of Group Means and Medians . . . . .	58
2.3.2 Validating Normality and Homogeneity of Variances . . . . .	59
2.3.3 Post-hoc Tests . . . . .	59
2.3.4 Correlation Analysis . . . . .	60
2.4 Methods for Machine Learning Analysis . . . . .	61
2.4.1 Regression Algorithms . . . . .	61
2.4.2 Ensemble Learning . . . . .	62
2.4.3 Instance-based Learning . . . . .	65
2.4.4 Probabilistic Learning . . . . .	67
2.5 Chapter Summary . . . . .	68

<b>3</b>	<b>Research Methodology and Evaluation Methods</b>	<b>70</b>
3.1	Research Methodology . . . . .	70
3.2	Operating Constraints . . . . .	75
3.3	Evaluation Metrics . . . . .	76
3.3.1	Accuracy . . . . .	77
3.3.2	Spearman’s rank correlation coefficient . . . . .	77
3.3.3	LSC score . . . . .	78
3.3.4	Precision and Recall . . . . .	79
3.4	Chapter Summary . . . . .	80
<b>4</b>	<b>State-of-the-art Lifelog Retrieval System</b>	<b>82</b>
4.1	Introduction . . . . .	82
4.2	An Overview of LifeSeeker . . . . .	83
4.2.1	A Brief History of Development . . . . .	83
4.2.2	System Design . . . . .	86
4.3	User Interface and User Interaction . . . . .	89
4.3.1	User Interface . . . . .	89
4.3.2	User Interaction . . . . .	91
4.4	Search Engine . . . . .	92
4.4.1	Indexing . . . . .	92
4.4.2	Retrieval . . . . .	100
4.5	Benchmarking Result in Lifelog Search Challenge . . . . .	104
4.6	Chapter Summary . . . . .	107
<b>5</b>	<b>Reading Comprehension Estimation using Eye Movement Measures</b>	<b>108</b>
5.1	Introduction . . . . .	108
5.2	Data Collection . . . . .	110
5.3	Methodology . . . . .	113
5.3.1	Comprehension and Measuring Comprehension . . . . .	114
5.3.2	Data pre-processing and Feature Extraction . . . . .	115
5.3.3	Analysis . . . . .	117
5.4	Results and Discussion . . . . .	121
5.4.1	Reading Condition and Comprehension Level . . . . .	121
5.4.2	Eye Movement Features and Reading Condition . . . . .	125
5.4.3	Eye Movement Features and Reading Comprehension . . . . .	132
5.5	Chapter summary . . . . .	142
<b>6</b>	<b>Longitudinal Evaluation of Reading Comprehension Estimation Model</b>	<b>145</b>
6.1	Introduction . . . . .	145
6.2	Data Collection . . . . .	146
6.3	Methodology . . . . .	149
6.3.1	Data pre-processing and Feature Extraction . . . . .	149
6.3.2	Machine Learning Analysis . . . . .	150
6.3.3	Eye Movement Features Inspection . . . . .	152
6.4	Results and Discussion . . . . .	152

6.4.1	Reading Condition Classification . . . . .	152
6.4.2	Reading Comprehension Prediction . . . . .	158
6.4.3	Eye Movement Features Inspection . . . . .	165
6.5	Chapter Summary . . . . .	168
<b>7</b>	<b>Gaze-coupled Comprehension-evidenced Interactive Infologging Retrieval System</b>	<b>170</b>
7.1	Introduction . . . . .	170
7.2	Data Collection . . . . .	172
7.2.1	Unit of Retrieval . . . . .	173
7.2.2	Data Collection Process . . . . .	174
7.2.3	Infolog Dataset Overview . . . . .	175
7.3	Methodology . . . . .	176
7.3.1	Feature Extraction and Evaluation of the Reading Comprehension Estimation Model . . . . .	176
7.3.2	Development of the interactive retrieval system . . . . .	177
7.3.3	Evaluation of InfoSeeker Retrieval System . . . . .	182
7.4	Results and Discussion . . . . .	187
7.4.1	Evaluation of Reading Comprehension Estimation Model . . . . .	187
7.4.2	Evaluation of Infoseeker . . . . .	188
7.5	Chapter Summary . . . . .	194
<b>8</b>	<b>Conclusion</b>	<b>196</b>
8.1	Summary . . . . .	196
8.2	Contributions . . . . .	201
8.2.1	Revisiting Research Contributions . . . . .	201
8.3	Limitations . . . . .	203
8.4	Future Work . . . . .	204
8.4.1	Improving the Reading Comprehension Model . . . . .	204
8.4.2	Improving the InfoSeeker System . . . . .	206
8.5	List of publications . . . . .	207
<b>A</b>	<b>Appendix</b>	<b>212</b>
A.1	Obtaining Texts for Reading Task in RCIRv1 . . . . .	212
A.1.1	Overview . . . . .	212
A.1.2	Topic-modelling . . . . .	212
A.1.3	Topic Validating . . . . .	213
A.1.4	Splitting Text Data . . . . .	213
A.2	Test Topics for Evaluation of Infolog Retrieval System . . . . .	214
	<b>Bibliography</b>	<b>219</b>

# List of Figures

1.1	The structure of the thesis. . . . .	22
2.1	User interface of VRLE (adapted from [1]) . . . . .	32
2.2	User interface of MyScéal (adapted from [2,3]) . . . . .	34
2.3	User interface of E-MyScéal (adapted from [4]) . . . . .	37
2.4	User interface of Memento (adapted from [5]) . . . . .	38
2.5	User interface of FIRST (adapted from [6]) . . . . .	39
2.6	User interface of Voxento (adapted from [7]) . . . . .	41
2.7	The anatomy of the human eye, from OpenStax College, licensed under CC BY 3.0, via Wikimedia Commons . . . . .	44
2.8	Corneal reflection and pupil detected by the eye-tracker. . . . .	49
3.1	Research process in flow chart, adapted from [8] . . . . .	72
4.1	An overview of LifeSeeker system workflow . . . . .	87
4.2	The Interactive User Interface of the LifeSeeker Retrieval System. . .	89
4.3	The image corresponding to the concepts in Listing 4.1 . . . . .	94
5.1	Visualisation of a participant’s eye movements when reading a passage.112	
5.2	Box plot of comprehension scores grouped by reading condition (pooled across participants) . . . . .	122
5.3	Pairwise relationship of reading conditions grouped by subject. . . .	123
5.4	Counts of correct, incorrect and aborted answers of each subject and grouped by reading condition. . . . .	124
5.5	Time spent on reading task grouped by reading condition. . . . .	125
5.6	Demonstration of posthoc results of top 5 feature from Table 5.5. . .	127
5.7	A two-dimensional t-SNE [9] visualisation of 254 eye movement features.128	
5.8	Performance of top three classifiers in feature selection process. . . .	130
5.9	Feature contributions to the best classifier . . . . .	133
5.10	Spearman’s rank correlation coefficient ( $\rho$ ) between the eye movement features and the reading comprehension score. . . . .	134
5.11	Feature Selection using Recursive Feature Elimination. . . . .	137
5.12	Monotonic relationship between the comprehension score and the subjective evaluation score. . . . .	139
5.13	SHAP interpretation of the ET model trained on <i>c_score</i> with the predicted reading condition as an additional feature. . . . .	141
6.1	Hyperparameters’ Importance of the Tuned LGBM Model . . . . .	157
6.2	Parallel coordinate plot showing hyperparameter settings of all tuning iterations. . . . .	158
6.3	Hyperparameters importance of the tuned RF model . . . . .	164



6.4	Parallel coordinate plot showing hyperparameter settings of all tuning iterations. . . . .	165
6.5	SHAP interpretation of the RF model trained on <i>se_score</i> with the predicted reading conditions as additional features. . . . .	169
7.1	The process of logging on-screen information. . . . .	174
7.2	The user interface of InfoSeeker. . . . .	180
A.1	Topic-modelling process . . . . .	212

# List of Tables

2.1	A statistics of the benchmarking datasets. . . . .	30
2.3	List of participating systems in LSC'21 and LSC'22 and key approaches employed by them. . . . .	31
2.4	Overview of datasets used in the existing literature that investigates reading comprehension through eye tracking data. . . . .	55
4.1	A list of feature changes in LifeSeeker from LSC'19 to LSC'22. . . . .	84
4.3	Location categories . . . . .	97
4.4	Statistics of the top-5 teams in LSC'21 . . . . .	104
4.5	The normalised score of the top-5 teams for each task in LSC'22. . . . .	105
5.1	Guideline for participants to report their comprehension . . . . .	115
5.2	Summary of ocular events and features used in my experiment . . . . .	116
5.3	Summary of encoding methods used for sequence features . . . . .	117
5.4	Posthoc test results for each pair of reading conditions. . . . .	123
5.5	The top 20 features which were significantly different between reading conditions. . . . .	126
5.6	Reading condition classification results using ML models in different training settings . . . . .	129
5.8	Re-training result of the top classifiers using selected features. . . . .	131
5.9	Reading comprehension estimation results using different combinations of machine learning classifiers and training types. . . . .	136
5.11	Re-training result of the three best models using RFE_GE and RFE_SD setting . . . . .	138
5.12	Comprehension prediction result of the ET model when integrating with identification of reading condition . . . . .	139
6.1	Summary of ocular events and features used in the analysis of RCIRv2 dataset. . . . .	150
6.2	Baseline results of the condition classification task in the GE training configuration. . . . .	153
6.3	Reading condition classification results of the model in the SD training configuration. . . . .	155
6.5	Comparison between the baseline classification model and its tuned version. . . . .	156
6.6	Spearman's rank correlation between <i>c_score</i> and <i>se_score</i> , aggregated by participants and by sessions . . . . .	159
6.7	The baseline Spearman's rank correlation coefficient score of the reading comprehension prediction models. . . . .	160
6.8	Comparison of reading comprehension prediction models when employing reading condition as additional training features. . . . .	162

6.9	Percentage of stable features for each subject, identified using the statistical testing procedure. . . . .	165
6.10	List of features that are identified as stable for most subjects . . . .	167
7.1	Example of query presentation of the test topic Q1 at multiple time-points for interactive evaluation . . . . .	186
7.2	Overall evaluation scores of the non-interactive version of the InfoSeeker system with and without gaze-coupled functionalities. . .	189
7.4	A breakdown of participants' LSC scores for each query using the baseline (BA) and gaze-coupled (GZ) InfoSeeker systems. . . . .	191
7.5	A breakdown of participants' LSC scores for each query using the baseline (BA) and gaze-coupled (GZ) InfoSeeker systems on the expanded ground truth. . . . .	193
7.6	Overall performance of the baseline (BA) and gaze-coupled (GZ) InfoSeeker systems using the initial and expanded ground truth. . . .	193
A.1	Description of the topics used in the reading dataset. . . . .	217
A.3	Illustration of text data split. . . . .	218

## List of Abbreviations

<b>BR</b>	Bayesian Ridge
<b>CI</b>	Confidence Interval
<b>CLIP</b>	Contrastive Language-Image Pretraining
<b>CNN</b>	Convolutional Neural Network
<b>EDA</b>	Exploratory Data Analysis
<b>ET</b>	Extra-Trees
<b>GE</b>	General
<b>IDF</b>	Inverse Document Frequency
<b>IR</b>	Information Retrieval
<b>KFTF</b>	Keeping Found Things Found
<b>KIS</b>	Known Item Search
<b>LGBM</b>	Light Gradient Boosting Machine
<b>LSAT</b>	Lifelog Semantic Access Task
<b>LSC</b>	Lifelog Search Challenge
<b>MCQ</b>	Multiple-Choice Question
<b>ML</b>	Machine Learning
<b>OCR</b>	Optical Character Recognition
<b>PIM</b>	Personal Information Management
<b>PR</b>	Proofreading
<b>QA</b>	Question Answering
<b>RCIR</b>	Reading Comprehension for Information Retrieval
<b>RE</b>	Reading
<b>REC</b>	Research Ethics Committee
<b>RF</b>	Random Forest
<b>RFE</b>	Recursive Feature Elimination
<b>RQ</b>	Research Question
<b>SC</b>	Scanning
<b>SD</b>	Subject Dependent
<b>SHAP</b>	SHapley Additive exPlanations
<b>SI</b>	Subject Independent
<b>SK</b>	Skimming
<b>SOTA</b>	State-of-the-art
<b>TF</b>	Term Frequency
<b>UI</b>	User Interface
<b>VR</b>	Virtual Reality

# Abstract

Tu-Khiem Le

## **Information Lifelogging: Leveraging Eye Movements and Reading Comprehension for Efficient Retrieval of Previously Encountered On-Screen Information**

The progress of lifelog research has enabled individuals to comprehensively capture their daily experiences. As a result, previous studies have primarily focused on developing tools to organise and retrieve lifelog moments effectively. However, existing lifelog data often lacks the ability to capture the lifelogger’s focal points (their attention), despite providing information-rich first-person-view lifelog images of their surroundings and activities. Consequently, this limitation hinders the lifelog retrieval systems’ utility when lifeloggers seek to retrieve specific information they have previously encountered. To address this research gap, a subjective point of view, represented through eye movements, should be incorporated as a new modality into lifelog data, thereby enhancing the retrieval performance. In pursuit of this objective, this dissertation investigates the feasibility of retrieving on-screen information by analysing lifelogger’s reading activities and comprehension level.

The primary contributions of this dissertation are as follows. Firstly, the development of LifeSeeker, an advanced interactive lifelog retrieval system, is developed and benchmarked in numerous lifelog retrieval challenges and competitions. By efficiently integrating various modality processing components (e.g., visual, text, location, biometrics) and user interaction components (e.g., search, filtering, browsing, relevance feedback) into a single interactive retrieval framework, LifeSeeker ranked among the top systems in these benchmarking activities, serving as the foundation for the rest of the thesis. Secondly, a novel reading comprehension dataset was created to explore the feasibility of recognising reading activities and estimating reading comprehension levels in daily life. Statistical tests and machine learning analyses on the dataset have revealed the strong connection between eye movement patterns, reading conditions, and reading comprehension. This led to a novel method for estimating reading comprehension with potential real-world applications. Furthermore, the longitudinal aspect of reading comprehension was investigated to examine the stability and generalisation of reading comprehension estimation models over time. Lastly, the reading comprehension estimation model was integrated into LifeSeeker as a new modality processor, resulting in a significant improvement in the system’s overall retrieval performance. In summary, this dissertation contributes to the understanding of reading activities and reading comprehension in real-world settings and showcases the potential of integrating reading comprehension estimation to enhance the retrieval of previously encountered information in lifelog data.

# Chapter 1

## Introduction

How frequently do we find ourselves seeking a piece of information that we recall having encountered previously, but cannot remember where or how to find it? This common problem, experienced in our daily lives, can be profoundly frustrating, and it has raised questions about the reliability of our memory. The inability to retrieve familiar information not only hinders our productivity but also affects our confidence in relying on our memory. Whether it is recalling critical data for a project, retrieving past research findings, or even remembering everyday details, this cognitive challenge can lead to significant setbacks and delays. The realisation that the human memory system has its limitations, resulting in lapses in information retrieval, has led us to consider the need for new mechanisms to support and enhance our capacity for remembering.

One of the approaches to support remembering vital information is by creating an external repository where such knowledge can be stored, organised, and easily retrieved from. This practice is not entirely novel, as throughout history, humans have employed various methods to externalise their memories. From ancient cave paintings and inscriptions on stones to the use of diaries, and more recently, in the digital age, we have an abundance of tools available, such as blogs, videos, photos, and social media, to capture and preserve our memories [10] However, in the digital age, the amount of information being generated has surged exponentially, rendering traditional memory-keeping methods less effective. This growing concern has given rise to a movement known as "Building a Second Brain" (BASB), aimed at addressing the limitations of our biological memory by establishing reliable

external systems [11]. Building a second brain involves the creation of a structured and easily accessible digital repository that can supplement our natural cognitive processes. This approach leverages a variety of tools and techniques to capture, organise, and link information, effectively offloading the burden of memorisation from our brains. At its core, the concept of building a second brain revolves around the idea of "Keeping Found Things Found" (KFTF), a concept that has been of interest to researchers for decades [12]. KFTF emphasises the importance of establishing reliable systems to manage and retrieve information when needed, without succumbing to the frustration of fruitless searches. However, achieving such a system requires tremendous effort and having to manually organise and maintain a repository of information can be a daunting task, especially when dealing with large volumes of data.

To address this, many desktop-based personal information management (PIM) systems have been developed to facilitate efficient information management [13–15] by automatically analysing and indexing the files stored on a computer to form a retrievable database. Since most PIM tools were designed for handling only a subset of data types (e.g. emails, PDF documents, web caches), they depend heavily on data decoders to read a new file type to index it, which makes them not generalise well to capture all on-screen information that a user saw in their daily computer usage. The emergence of the lifelogging concept, which is commonly known as the process of generating a personal archive of an individual's life experiences by passively capturing information for various sensors [10], has opened a new approach to the problem of personal information management. Instead of having to index all data that one has on their computer (which requires an excessive amount of data decoders), a continuous screen capture could essentially reveal what information is being displayed (which forms a genre of lifelog data of computer usage). Moreover, the data captured in this way illustrates how the modality of information was formatted when displayed to the user, thus allowing the user to form more detailed queries than conventional PIM tools.

Despite the ability to capture on-screen information using lifelogging techniques, the problem of efficiently indexing the content a human has consumed for future refinding remains a major challenge. Prior approaches to indexing lifelog data were content-based, which analysed the entire content of lifelog images for retrieval. However, since people often only focus on specific parts of the information that are interesting to them, indexing the entire content of an image is not an efficient way to support refinding as it may include irrelevant information that might lead to the reduction of retrieval accuracy. Eye tracking is a suitable data source to measure a human's interest, as it has been extensively researched in the psychology literature and adopted to many IR tasks [16–19]. Studying eye movement features for on-screen information retrieval is the primary focus of this dissertation since there has not been a standard approach towards indexing and retrieving on-screen information and coupling gaze features with on-screen information to enhance the retrieval outcome of viewed screen contents. In the following sections, I will give a brief overview of lifelogging (Section 1.1), personal information management (Section 1.2), and eye movement in reading analysis (Section 1.3), which are the main research areas that this dissertation is based on. After that, I will present in Section 1.4 the hypothesis and research questions that this dissertation aims to address and the contributions of this dissertation to the research community.

## **1.1 Lifelogging**

As the volume of personal data generated by individuals continues to grow, there emerges a corresponding need for effective data organisation and retrieval systems. These personal archives encompass data from various sources, including mobile and wearable devices, tablets, laptops, and social media platforms. Accompanying this surge in personal data volume is an increased interest in personal data organisation and analytics, where one pioneering area in this field is lifelogging.



Lifelogging is the process whereby individuals gather (often passively) large multimodal personal data archives from diverse sources, aggregating them into a single repository which is known as a lifelog. This practice is defined as a form of pervasive computing that generates a unified digital record of an individual's experiences, captured multimodally through digital sensors and stored as a personal multimedia archive [20]. It is also described as a phenomenon where individuals digitally record their daily lives in various levels of detail for different purposes [21]. Unlike traditional data organisation challenges, such as photo or email archives, lifelog, typically being non-curated and passively captured, presents unique challenges in multimedia analytics and information retrieval [10].

The concept of lifelogging, despite being relatively modern, has been around for decades. In 1945, Vannevar Bush proposed the *Memex* [22], a hypothetical hypermedia device for storing an individual's books, records, and communications, and creating linkable information trails. This concept laid the groundwork for lifelogging. The first practical instance that could reasonably resemble a lifelog was Richard Buckminster Fuller's Dymaxion Chronofile [23], a comprehensive physical scrapbook (of all correspondence, bills, notes, sketches and clippings from newspapers) maintained from 1920 to 1983, which now resides at Stanford University. However, it was not until Gordon Bell of Microsoft embarked on the MyLifeBits project in 2001 [24–26] that the digital lifelogging concept was fully realised. This project managed to digitally capture and store all personal data, including emails, web pages, photos, videos, and phone calls.

Historically, lifelogging was hindered by the unavailability of necessary equipment, with many required data sources being too cumbersome or expensive to capture. However, recent advancements in sensor and wearable technology have made it feasible for individuals to comprehensively track daily activities such as eating, commuting, exercising, working, and sleeping. These developments have also fostered a greater public acceptance of such technologies, enhancing participation in lifelogging, both from the perspectives of the lifelogger and the

recorded subjects in lifelogs.

Ideally, this huge amount of personal data should be securely stored in an always-on, multimodal storage service. This service should integrate various time-stamped sensor data sources, organised in a manner that enables standard data processing techniques like content analysis, information retrieval, data browsing, and summarisation. Prior to these stages, data typically undergoes cleaning, temporal alignment, normalisation of sensor outputs, and other methods of data linking and aggregation. This is done in an effort to create a consistent and comprehensive lifelog of the individual.

Due to the huge amount of lifelog data accumulated over time, it became necessary to have a system which efficiently organises data so that it could be retrieved precisely and effortlessly. The emergence of such lifelog retrieval systems also created the need for these systems to be benchmarked against each other. To serve this purpose, many lifelog research tasks were organised. The most common ones were Lifelog tasks at NTCIR [27–30], ImageCLEFlifelog [31–34], and Lifelog Search Challenge (LSC) [35–39], each of which employed various metrics for comparatively evaluating the retrieval systems. The participating systems also came in a variety of retrieval functionalities, user interfaces and user interaction methods (details in Section 2.1.2). LifeSeeker [40–44] – an interactive lifelog retrieval system developed by my colleagues and I – is one of the systems that participated in these tasks. The system is an experimental platform for exploring the components that make up a state-of-the-art lifelog retrieval system, which ultimately serves as a foundation for developing a retrieval system for on-screen information in this dissertation.

## **1.2 Personal Information Management**

Aside from generic lifelogging which aims at recording the totality of the life experience of an individual, there are many situation-specific applications of

lifelogging that mine deeper insights from a particular aspect of life. Lifelogging was used in the area of market research to measure one's exposure to advertising campaigns [45]. In 2013, Aizawa et. al. applied the concept of lifelogging to monitor diet with a system known as Foodlog [46]. Kids'Cam was a lifelog-based project which explores the environment in which children live [47]. Moreover, lifelogging is also applied in conversation analysis [48], causality detection in human activities [49], understanding human well-being [50] and sport analysis [34]. In terms of capturing computer usage, Hinbarji et al. [51] developed a MacOS-based software called Loggerman<sup>1</sup> which runs in the background and captures screenshots, mouse and keyboard inputs to form an archive of a human's on-screen experience. However, there has been no attempt to analyse the archive generated by Loggerman in terms of indexing and retrieving on-screen activities.

The concept of logging on-screen information on a day-to-day basis to form part of a human's digital archive can be linked to the personal information management (PIM) concept which focuses on analyzing the way that people manage their physical and electronic information and building tools that support such information management goals. The term Personal Information Management was first used by Lansdale [52] in 1988 which discussed the burden of manual information management and stressed the need for an automatic user-oriented personal information management system. This was followed by extensive research which has been conducted to learn the human's behaviour in keeping information and identify the area that a PIM tool can aid them in doing so [53–55]. In 2003, Dumais et al. introduced the SIS (Stuff I've Seen) system [13] which supports refinding of seen information. The system was built on top of the Microsoft Search architecture which gathers information regardless of source (e.g. webpage, files, email, books), then tokenises the text data into tokens to build an inverted index for retrieval. iMecho [14] focused on the association and semantic links between information so that the user can navigate through these links to obtain the

---

<sup>1</sup><http://loggerman.org/>

information need. Dessy [15] was another PIM system that supports mobile devices which search for files based on their content and metadata. However, these systems focused mostly on indexing emails, documents (PDF), web cache, and a few other MIME types<sup>2</sup>. They had a restriction in the supported file, meaning that these systems do not generalise to all data sources. Conversely, Reeves and Ram et al. approached the problem of digital information capture by analysing screenshots [56, 57]. Text extraction techniques like OCR and image analysis (e.g. face detection, colour histogram) were employed to understand what information was being presented on screen. This allows a searchable database for retrieving screenshots to be generated. Although Reeves and Ram et al.'s work shares similar goals with this dissertation, my approach distinguishes itself by focusing on a deeper level of on-screen information analysis, in which eye tracking is employed to understand what particular information was perceived by the user, leading to a more personalised and accurate retrieval system of previously seen information.

### 1.3 Eye movement and Reading Analysis

To incorporate eye tracking into the lifelogging system to understand what information was perceived by the user, it is necessary to investigate human reading comprehension and how eye movement can be used to estimate their level of information understanding, which in turn can be used to extract useful information from the screen to facilitate better indexing and retrieval of previously seen information.

Reading comprehension is the process of understanding written text, enabling the acquisition of information, communication with others, and successful completion of various tasks. It is a fundamental cognitive process that plays a pivotal role in our daily lives as the vast majority of the knowledge of humankind is communicated in the written form [58]. Reading entails visually perceiving written words and decoding them into meaningful units, such as phrases, sentences, and

---

<sup>2</sup>[https://en.wikipedia.org/wiki/Media\\_type](https://en.wikipedia.org/wiki/Media_type)

paragraphs [59–61]. Comprehension, on the other hand, refers to the mental process of extracting meaning from the text, making connections between ideas, and creating a coherent understanding of the overall message [62]. It involves not only understanding the literal meaning of the words but also inferring implicit information, identifying main ideas, and grasping the author’s purpose. Reading comprehension combines the skills of decoding and interpreting text, enabling individuals to engage with written material, gain knowledge, and derive meaning from what they read. Even though many individuals possess reading abilities [63], it is frequently observed that reading an identical text passage can yield varying levels of comprehension among different readers [64, 65]. Thus, understanding reading comprehension, or the ability to understand and retain written text, is of paramount importance. Previous methods of assessing reading comprehension primarily relied on a combination of assessment approaches, including interviews, questionnaires, oral retelling, freewriting, and think-aloud procedures [66]. While these methods have been effective, they are often only practical in controlled settings where specific assessments designed for measuring comprehension are available. In everyday situations where reading happens as part of routine activities like browsing websites, reading newspapers, or attending seminars, using these techniques to assess comprehension on an individual basis would be too burdensome. Consequently, there is a need for automated methods that passively estimate reading comprehension by leveraging data sources that can be unobtrusively captured in real-world settings using digital sensors and devices.

Eye movement in reading research has undergone significant development across multiple eras, each marked by advancements in sensing technology [67]. The initial era, from the 19th century to the 1920s, focused on investigating basic eye movement measures such as fixations and saccades to understand the reading process, as studied by Huey [68]. This was followed by the second era, extending until the 1970s, characterised by the work of Tinker [69] and Buswell [70]. During this period, initial attempts were made to apply eye movement research practically,

although the scope remained limited to investigating cognitive processes through eye movements. The third era witnessed a significant leap forward with the advent of digital eye-tracking systems. These enable easier measurements, which has led to a surge of eye movement studies in various aspects of reading. Rayner [67] comprehensively summarised the progress made during this era. Presently, we find ourselves in a new era of eye movement research, characterised by the development of eye movement-based applications in real-world reading scenarios [71–82]. A growing body of research has focused on exploring the eye movement characteristics associated with different reading styles, such as thorough reading, skimming, scanning, and proofreading [73–75]. These studies have demonstrated that distinct reading styles exhibit unique eye movement patterns, providing valuable insights for the development of machine learning models to identify reading styles in real-world scenarios. Furthermore, numerous studies have found the association between reading comprehension and eye movement measures [76–82]. Despite variations in language, participant profiles, and reading materials, these studies have consistently shown that eye movement measures can be used to predict an individual’s comprehension level during reading. However, some studies investigate comprehension in a question-answering manner, where comprehension questions are provided in advance, which differs from real-world reading scenarios where individuals read to gain knowledge and understanding, not to answer specific questions. Moreover, certain studies have focused solely on the sequential reading style [76, 77, 79–82], neglecting the fact that people employ different reading styles depending on their purposes and goals. We contend that different reading styles give rise to distinct eye movement patterns and levels of reading comprehension. [75] notes the relationship between eye movement measures, reading styles, and reading comprehension. They discovered significant changes in eye movement measures and comprehension levels when people employ different reading styles. Although the observed changes in eye movement characteristics were identified by comparing specific pairs of reading styles in that

work, extending this understanding to build a model which handles a more general context of predicting reading styles through a multi-class classification task, would be highly valuable. The identification of reading styles, in combination with eye movement measures, can be utilised to estimate the corresponding comprehension level of an individual during reading.

Building upon the advancements in eye movement research, my objective is to investigate deeper into the estimation of reading comprehension in real-world scenarios. To achieve this aim, I explored a novel approach that combines eye movement measures and reading styles to estimate reading comprehension. I show that by integrating the identification of reading styles alongside eye movement measures for estimating reading comprehension can enhance the accuracy of predictions within reading conditions. With this approach, on-screen information can be indexed and retrieved based on the user's level of comprehension, which can be used to facilitate better indexing and retrieval of previously seen information.

## **1.4 Hypothesis and Research Questions**

In the previous sections, I have discussed the need for a personal information management (PIM) tool that can actively index and retrieve a user's daily information intake on the computer. Most PIM tools, to the best of my knowledge, are designed to organise specific documents and files on the computer but lack the ability to generalise to the multi-modal and unstructured information displayed on a computer screen (e.g., multiple windows on the screen, one of which shows a news webpage with text, images, and advertisements). Lifelogging provides a solution to this problem by capturing on-screen information (e.g., using Loggerman [51]) and retrieving it using state-of-the-art lifelog retrieval systems [38]. However, it lacks the ability to distinguish the information a user has read from the information displayed on the screen. Consequently, the retrieval results might contain irrelevant information that the user has no interest in. To illustrate this challenge, consider a

common scenario faced by researchers. We often find ourselves in situations where we recall having read a paper that stated facts or findings relevant to our current research, but we struggle to retrieve that specific paper. Although a simple keyword search can be used, it may not always guarantee success, particularly when there are numerous papers with similar keywords, and the desired paper does not rank high in the search results due to other papers matching the keywords better. Despite knowing that we paid more attention to the paper we are looking for and merely skimmed through the others, there is currently no existing system that can leverage this information to re-rank search results based on the user's comprehension of the information. This realisation has led to the idea of incorporating eye movement measures into retrieval systems, as eye movement can provide valuable insights into where a user's attention is focused and how well they comprehend the information displayed on the screen. This adds a new dimension to the retrieval process, allowing information to be indexed based on not only the content, but how much attention the user paid to it. By doing so, more important information (i.e., information with more attention or higher comprehension) can be prioritised in the retrieval process, leading to more relevant search results. My conjecture is that eye movement measures can be employed as a new data source for lifelogs, enabling the estimation of users' comprehension of on-screen information for better indexing and ranking of retrieval results. I refer to this type of lifelog data, where on-screen information is captured in combination with eye movement measures, as **infologging**, a term I will use throughout this thesis. Based on this, I define the hypothesis for this dissertation as follows:

**Hypothesis:** *It is feasible to enhance the retrieval performance of previously perceived on-screen information by integrating users' comprehension levels, estimated from their eye gaze patterns captured through eye tracking, into a state-of-the-art interactive lifelog retrieval system.*



In order to either prove this hypothesis, a number of related research questions have been developed as follows:

- **Research Question 1 (RQ1).** What are the key design principles and components required to construct a state-of-the-art lifelog interactive retrieval system?

To develop an effective interactive retrieval system for infologging data, it is crucial to first establish a strong foundation by identifying the key design principles and components that contribute to the creation of a state-of-the-art lifelog retrieval system. By addressing this research question, I aim to gain valuable insights from the extensive body of literature within the lifelogging research domain and leverage the experience gained through participating in annual lifelog benchmarking challenges. The knowledge acquired will not only help in assessing the system's performance against contemporary standards but also serve as a guiding light in the development of a robust and user-friendly retrieval system.

While working on this research question, I have developed an interactive retrieval system called LifeSeeker and participated in multiple Lifelog Search Challenges, each year presenting a different iteration of the system with new features and improvements in terms of user interface and retrieval performance. This iterative process of design, implementation, and refinement through successive challenges has led to the creation of a system that exhibits state-of-the-art performance in lifelog retrieval tasks. By achieving this, I have shortlisted the key design principles and components that are essential for constructing a state-of-the-art system, which addresses this research question.

- **Research Question 2 (RQ2).** To what extent can machine learning models accurately estimate reading comprehension levels based on eye movement features extracted from eye-tracking data?

Estimating reading comprehension levels accurately is the core of an effective infologging retrieval system. Addressing this research question will provide valuable insights into the feasibility of using eye movement data as a reliable source for gauging users' understanding of on-screen information, ultimately contributing to the development of a more refined and user-centric retrieval system.

To develop machine learning models for estimating reading comprehension, I first needed to construct a reading dataset that contains eye movement data and reading comprehension scores, since such a dataset is not available in the literature. This dataset was gathered through a user study wherein participants were instructed to read various articles employing different reading strategies, namely sequential reading, skimming, scanning, and proofreading and then answer multiple-choice questions to assess their comprehension of the articles. Throughout this process, their eye movements were recorded using an eye tracker. Subsequently, ocular events were extracted from the eye movement data and used to compute a set of eye movement features. These features served as the foundation for training machine learning models to accomplish two primary tasks: (1) predicting the reading condition (reading styles) and (2) estimating reading comprehension levels. Experimental results show that there is a close relationship between eye movement features, reading conditions (reading styles) and reading comprehension as eye movement features can be used to predict reading conditions and reading comprehension with high accuracy and correlation scores, respectively. I also show that the prediction of reading comprehension can be improved by integrating the predicted reading condition label as extra features into the model. Ultimately, this research question is addressed by demonstrating that reading comprehension can be estimated from eye movement features.

- **Research Question 3 (RQ3).** How robust is the reading comprehension estimation model when applied to longitudinal reading data?

To ensure the real-world applicability of the reading comprehension estimation model, it is essential to examine its robustness when applied to longitudinal reading data. By exploring eye movement features and its impact on the model's performance over time, I aim to identify potential challenges and opportunities for enhancing the model's adaptability to varied reading patterns over time. Addressing this research question will provide valuable insights into the model's capacity to maintain consistent performance in real-world scenarios, where reading behavior may fluctuate over extended periods.

To explore the robustness of the reading comprehension estimation model in the context of longitudinal reading data, a longitudinal study was conducted. In this study, participants engaged in reading tasks similar to those in RQ2, but these tasks were spread over a span of six non-consecutive days. This approach was designed to simulate a more realistic, varied reading pattern over time. The experimental procedure from RQ2 was adapted for extracting eye movement features and training machine learning models to classify reading conditions and estimate reading comprehension levels. The key distinction in this study was the training and testing framework of the models: they were trained on data from the initial days and subsequently tested on data from the remaining days. The experimental results revealed consistent performance of the models across different days for both the task of classification and comprehension level estimation. Notably, it was observed that an increase in the number of days used for training correlated positively with improved model performance. This finding suggests that the models benefit from exposure to more varied data over time, enhancing their predictive accuracy. Overall, this research question is addressed.

- **Research Question 4 (RQ4).** To what extent does the integration of reading comprehension estimation improve the performance of the infologging retrieval system for on-screen information compared to a baseline system without this feature?

Integrating reading comprehension estimation into the infologging retrieval system is expected to enhance its performance by prioritising information that users have actively engaged with and comprehended. By comparing the system's performance with and without this feature, I aim to quantify the benefits of incorporating comprehension estimation in the retrieval process. Addressing this research question will not only demonstrate the effectiveness of the proposed approach but also highlight the potential for further improvements in the field of infologging retrieval systems, ultimately leading to the development of more efficient and user-centric tools for managing and accessing personal information.

In order to address this research question, another study was conducted to collect a novel infologging dataset. The data collection process involved a participant using a computer for their daily tasks while their eye movements were recorded using an eye tracker. Concurrently, the on-screen information viewed by the users was captured as images. This screen data was then processed to extract text, which was later used for indexing and retrieval purposes. In parallel, the eye-tracking data was processed through the reading comprehension estimation model to obtain the user's level of understanding of the content displayed on the screen. With this data, the state-of-the-art lifelog retrieval system was adapted to index both the screen images and the comprehension data.

To evaluate the effectiveness of this infologging retrieval system, two experiments were conducted. The first experiment was designed to evaluate the performance of the system in a non-interactive manner. The ranked lists

returned by the system with and without the integration of reading comprehension estimation were compared to determine whether the integration contributed to improved retrieval performance. Additionally, the second experiment was conducted in an interactive manner, through a user study. Similar to the first experiment, participants were asked to perform a series of search tasks using the system with and without the integration of reading comprehension estimation. The study took the format of the LSC competition, and the systems were evaluated on the accuracy of the returned results and the time taken to complete the tasks. The findings from both experiments suggest that the system’s performance is improved when reading comprehension estimation is integrated into the retrieval process. This, in turn, addresses this research question.

## 1.5 Research Contributions

In this section, the key contributions made in this thesis are outlined as follows:

- **Chapter 4 - RQ1:**
  - **Contribution 1:** I construct an interactive lifelog retrieval system (in collaboration with my colleagues) called LifeSeeker, which was evaluated across several Lifelog Search Challenges held annually. The outcomes of these challenges consistently demonstrated that LifeSeeker ranks among the leading state-of-the-art systems in lifelog retrieval. Lifeseeker is a novel combination of textual and visual search functionalities, with advanced filtering and feedback mechanisms to enhance retrieval performance. The user interface design is simple, intuitive and informative, with the integration of search history for re-finding purposes. It also possesses speed and scalability due to the use of distributed and scalable technologies and caching mechanisms.

- **Chapter 5 - RQ2:**

- **Contribution 2:** I construct a novel multi-modal reading dataset consisting of eye-tracking data and comprehension measures for four different reading conditions, enabling the investigation of the relationship between eye movements, reading strategies, and comprehension levels.
- **Contribution 3:** I show that by incorporating the identification of reading conditions with eye movement features, we can improve the machine learning model's performance in estimating reading comprehension levels, i.e. we could better predict comprehension level knowing the reading condition.
- **Contribution 4:** I provide novel insights into the importance of eye movement measures for classifying reading styles and estimating comprehension levels through a comprehensive analysis using statistical testing procedures and feature contribution analysis.

- **Chapter 6 - RQ3:**

- **Contribution 6:** I create a unique longitudinal reading dataset consisting of participant reading data collected over six non-consecutive days, enabling the exploration of the temporal robustness of reading comprehension estimation models.
- **Contribution 7:** I show that the eye movement features and the comprehension estimation model explored in RQ2 are robust when applied to longitudinal reading data, i.e. the model's performance is consistent across different days.
- **Contribution 8:** I uncover novel insights into the temporal stability of eye movement features across multiple sessions and their limited contribution to the model's performance in estimating comprehension levels, highlighting the need for further research in this area.

- **Chapter 7 - RQ4:**
  - **Contribution 9:** I introduce InfoSeeker, a novel interactive retrieval system designed for infologging data, which exploits the eye gaze data to enable filtering search results based on the infologger’s level of comprehension of the content displayed on the screen and allow users to quickly retrieve the desired information.
  - **Contribution 10:** I demonstrate the significant improvement in the performance of the InfoSeeker system through the integration of reading comprehension estimation, as evidenced by the evaluation of the system in both non-interactive and interactive settings using a user study.

## 1.6 Research Limitations

While the research presented in this thesis has made significant contributions to the fields of lifelogging, eye movement analysis, and information retrieval, it is essential to acknowledge the limitations that may impact the interpretation and generalisability of the findings. These limitations arise from various factors, including the choice of equipment, the characteristics of the datasets, and the scope of the study. By critically examining these limitations, I aim to provide a balanced perspective on the research outcomes and highlight potential areas for future investigation. In the following subsections, I will discuss two main categories of limitations: those related to the comprehension estimation model and those pertaining to the gaze-coupled retrieval system.

- **Comprehension Estimation Model:** The results of our study have demonstrated the potential of utilising eye movement features and the identification of reading conditions to predict reading comprehension levels. However, it is important to consider several limitations when interpreting these findings.

One of the primary limitations associated with the eye tracker used in our research pertains to its accuracy. Specifically, the eye tracker exhibits a vertical error, causing the gaze position to deviate from the actual position, as can be seen in the gaze visualisation in Figure 5.1. This limitation prevents us from precisely estimating the gaze position on a word level, which is crucial for analysing eye movement in conjunction with text features. However, when examining eye movement on the passage level, the error is less significant and can be disregarded, as observed in previous studies [76]. Notably, our results highlight that horizontal eye movement features were more influential than vertical eye movement features in classifying reading conditions and reading comprehension.

To address the issue of the eye tracker’s vertical shift error, various correction methods can be applied. For example, inter-trial calibration information collected in our dataset can be utilised to adjust gaze position (which we have not yet exploited in this paper), or line detection algorithms can be employed to align fixations with the corresponding text lines, as proposed by [83, 84]. It is worth noting that the error in our study is expected when using a low-cost eye tracker, as it is a trade-off for the accessibility and applicability of our research in real-world settings.

Another source of error we encountered is the presence of outliers in our dataset resulting from participants mistakenly performing wrong tasks (i.e. reading instead of scanning). Despite our efforts to mitigate this issue by including task prompts on the screen (a small window on the top-right corner of the screen as can be seen in Figure 5.1, some participants still performed the wrong task. Unfortunately, we were only made aware of these outliers after the experiment was completed, and we are unable to identify them specifically within our dataset. However, based on participants’ reports, the number of mistaken reading samples is small, and we believe that their impact on our results is minimal. This belief is further supported by the performance of our



classification model on reading conditions.

Finally, it is also crucial to acknowledge that the performance of the reading comprehension estimation model on the longitudinal reading data could be further improved, yet this was not feasible to investigate in this study due to the limit of the available data. It was concluded in Chapter 6 that the model's performance improved with an increase in the number of sessions used for training. The best model in Chapter 6 is obtained by utilising the maximum number of sessions that can be used for training (which is 4 sessions, the remaining 2 sessions are used for validation and testing) Without additional sessions, the score at which the model's performance plateaus is not yet determined, leaving room for further investigation when more data is available.

- **Gaze-coupled Retrieval System**

The development of the InfoSeeker system, currently in its early stages, represents an adaptation of a state-of-the-art lifelog retrieval system to infologging data retrieval. While the initial results are promising, there are several areas where the system could be enhanced to improve its robustness and efficiency in handling infologging data. One notable aspect for improvement is the system's current search mechanism relies on matching query text with OCR text from screenshots. This approach limits the system's capability, as it does not utilise other modalities in the retrieval process. Consequently, the system fails to address queries that require these additional modalities, such as queries that involve images or audio.

Furthermore, the system currently displays search results directly from the ranked list generated by the cosine similarity matching algorithm, without additional processing to re-organise these results. Implementing a mechanism that groups similar screenshots could significantly improve user efficiency in conducting search tasks. Such a feature would enable users to navigate through

the results more intuitively and identify relevant information quickly.

Another area for development relates to the reading comprehension model integrated into the InfoSeeker system. Although this model demonstrates acceptable performance, as detailed in Section 7.4.1, there is potential for further refinement. Specifically, the model's performance has not been assessed when re-trained exclusively on infologging data. This retraining could provide insights into the infologger's reading behaviour in real-world settings, potentially enhancing the model's predictive accuracy. However, expanding the model's training to encompass infologging data encounters a significant challenge: the limited sample size of the existing infolog dataset. Due to privacy concerns, it is impractical to collect additional infolog data from a wider range of participants.

## **1.7 Thesis Outline**

In this chapter, I have presented the motivation for this dissertation, the hypothesis, the research questions that I aim to address, and the contributions of my research. The remainder of this thesis is structured as illustrated in Figure 1.1, which can be summarised as follows:

- Chapter 2 provides the underpinning knowledge and background information about human eyes, how eye movements are recorded, and how eye movement data is processed and analysed. I also outline the existing literature on utilising eye movement in reading analysis. Related lifelog retrieval systems in Lifelog Search Challenges are also highlighted and discussed how Lifeseeker is different from them.
- Chapter 3 describes the methodology used in this dissertation for the data collection process the experimental analysis and the evaluation metrics.
- Chapter 4 presents the design and implementation of the lifelog retrieval

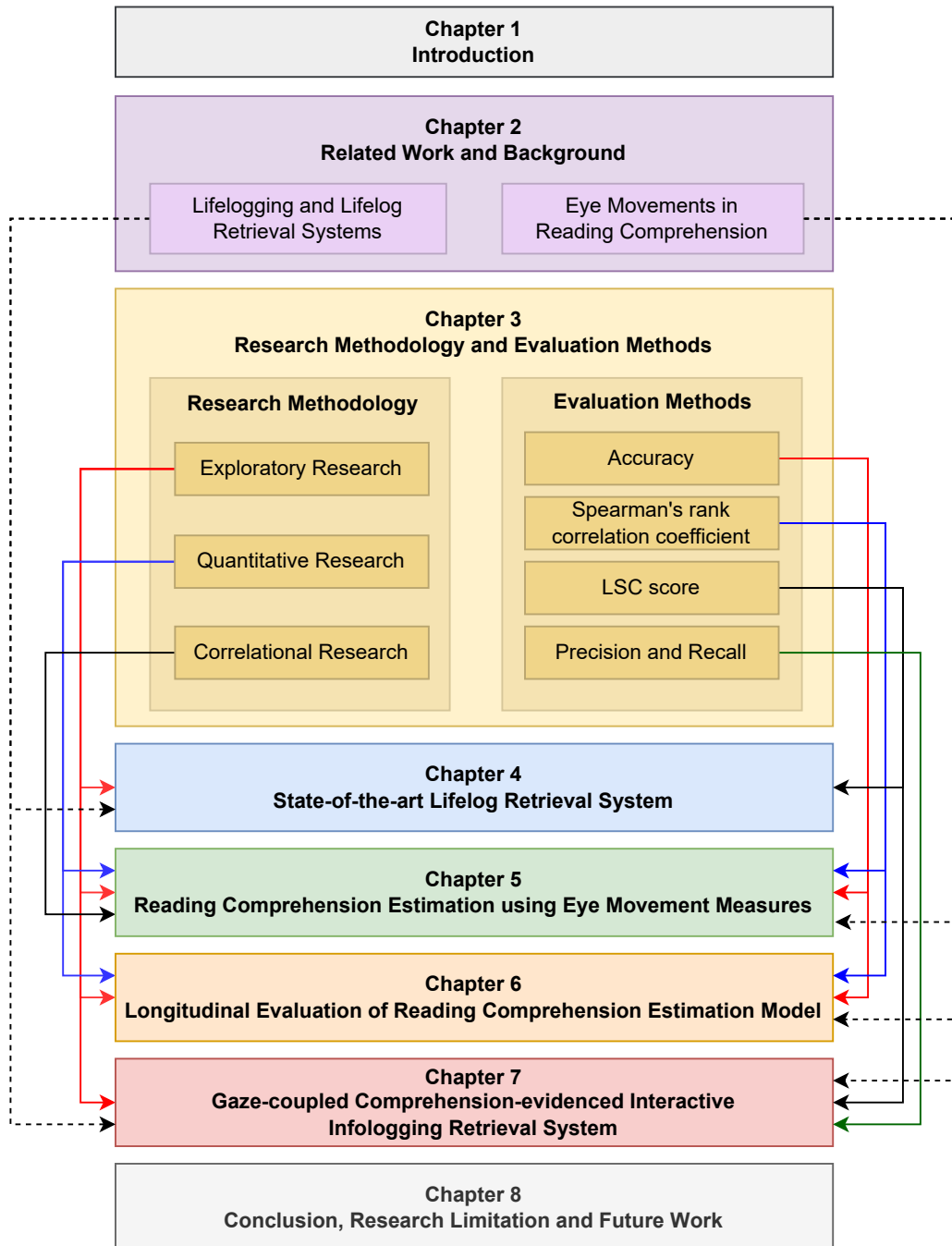


Figure 1.1: The structure of the thesis.

system, LifeSeeker, which is the foundation for the subsequent chapters.

- Chapter 5 describes the investigation into the relationship between eye movement features, reading conditions and reading comprehension. Different training configurations and their corresponding results are also compared and discussed. Feature analysis using a separate statistical testing procedure and using feature contribution analysis from SHAP is detailed to provide more insights into the features' importance.
- Chapter 6 presents the investigation into the temporal robustness of the reading comprehension estimation model. The experimental results are described and the models' hyperparameters tuning is also detailed. Feature contribution to the model's prediction is also analysed and compared against the finding the Chapter 5.
- Chapter 7 describes the design and implementation of the interactive infolog retrieval system, which is directly derived from LifeSeeker. Results on the non-interactive and interactive evaluation of the system, with and without the integration of reading comprehension estimation, are also presented and discussed.
- Chapter 8 concludes the dissertation, and discusses the limitations and future work.

## Chapter 2

# Related Work and Background

This chapter establishes the foundation for my contributions to the fields of lifelogging and eye movement research. Initially, it provides an overview of lifelog retrieval benchmarking tasks, pivotal to addressing research question 1. This is followed by a comprehensive literature review in Section 2.1, detailing the retrieval methods and functionalities of the state-of-the-art systems in the Lifelog Search Challenge (LSC). The insights from this review are utilised in developing my state-of-the-art retrieval system, thereby addressing the first research question. In Section 2.2, the chapter shifts focus to the anatomy of human eyes, the methodologies for eye movement capture, and the standard eye movement measures used in the field. This forms the basis for a subsequent literature review on using eye movement analysis to predict reading comprehension, where I identify existing research gaps and delineate my contributions, addressing research questions 2 and 3. Consequently, by integrating the insights gained from both literature reviews and my research contributions, I developed a novel interactive retrieval system for on-screen information which integrates user comprehension estimation through their eye gaze behaviour, effectively addressing research question 4. Finally, the chapter gives a brief overview of the methods and models used for addressing the research questions, setting the stage for the subsequent chapters.

## 2.1 Lifelogging and Lifelog Retrieval Systems

### 2.1.1 Lifelog Challenges

Lifelogging is the practice where individuals capture and accumulate vast amounts of personal data through various means, such as wearables, smartphones, and IoT devices [10]. This practice transforms everyday experiences into a digital narrative, offering a data-rich archive of one's life. While the sources of data available for constructing a lifelog dataset vary, they can be broadly categorised into six main types, according to [30]:

- **Vision:** This includes images or videos taken from the lifelogger's perspective, typically recorded using wearable cameras placed on the head or neck. As the most informative data source, it provides visual insights into the user's environment, social interactions, and activities.
- **Hearing:** This category encompasses audio recordings of the user's surroundings or summaries of listened music and sounds, capturing the auditory aspect of the lifelogger's environment.
- **Conversation:** Whether in text or audio format, this data documents the lifelogger's interactions with others. It can include content from text messages, emails, or recorded daily conversations, offering insight into social communications.
- **Biometric:** Collected automatically from smart devices like fitness bands or watches, biometric data encompasses health-related metrics such as heart rate, calories burnt, and step count, providing insights into the lifelogger's physical well-being.
- **Location:** Measured using GPS systems in smart devices, location data identifies specific addresses or semantic names of places the lifelogger has visited, adding contextual depth to the lifelog moments.

- **Activities:** Derived from sensors in smartwatches or bands, like accelerometers and gyroscopes, activity data includes estimations of actions such as sitting, standing, or running, offering an overview of the user’s physical activities throughout the day.

The increasing volume of lifelog data poses a significant challenge in efficiently navigating and extracting meaningful information from these extensive personal archives. Information needs often arise when a lifelogger wants to retrieve a specific moment or event from their lifelog data. These information needs are typically represented by queries that describe the desired moment in detail. For instance, when a lifelogger wishes to revisit a particular moment from their past, they might formulate a query such as: *"I was buying a ticket for a train in Ireland. It was from a vending ticket machine. After the purchase, I walked upstairs to the platform. I had to wait 8 minutes for the train to arrive. I had walked (for 36 minutes) to the station after eating sushi and beer."* Manually browsing through a lifelog to locate an image that corresponds to this detailed description would be extremely time-consuming, especially if the volume of lifelog data is relatively large. This is because human memory is not particularly adept at remembering specific dates [10], making it difficult to pinpoint the exact moment without the assistance of a retrieval system.

This difficulty highlights the necessity for advanced lifelog retrieval systems. Such systems are designed to process and index the huge and often unstructured multi-modal lifelog datasets to enable efficient retrieval of relevant information. The emergence of these retrieval systems creates the need for a benchmarking platform, essential for evaluating the efficacy of these systems. Consequently, a variety of benchmarking challenge tasks have been organised, each dedicated to assessing lifelog retrieval systems using different performance metrics. These benchmarking efforts play a crucial role in advancing the field, and guiding the development of more capable and user-friendly retrieval systems. The following are the most common benchmarking challenges in lifelog:

- **ImageCLEF Lifelog [31–34]:** Established in 2017, ImageCLEF Lifelog is a challenge created to recognise efficient methods for lifelog retrieval and to explore new directions in lifelog data analysis. It usually comprises two tasks: a primary task held annually for comparative evaluation of retrieval systems, and a secondary task that varies each year to introduce new lifelogging challenges to the research community. For instance, ImageCLEF Lifelog 2018 [32] focused on Activities of Daily Living (ADLs) summarisation, where participants analysed the frequency and duration of specific ADLs. The 2019 iteration [33] challenged participants to chronologically order a set of images without metadata, illustrating the diversity of tasks presented by this challenge.
- **NTCIR Lifelog [28–30, 85]:** NTCIR Lifelog shares similarities with ImageCLEF Lifelog, featuring a recurring task known as the Lifelog Semantic Access Task (LSAT). LSAT evaluates retrieval systems both interactively and automatically. Additionally, NTCIR Lifelog also investigates other aspects of lifelog analysis, such as Lifelog Event Segmentation Task (LES) in its 13th iteration (NTCIR13 [29]) and Lifelog Activity Detection Task (LADT) in the 14th iteration (NTCIR14 [30]), each addressing different facets of lifelogging.
- **Lifelog Search Challenge (LSC) [35–39, 86]:** The LSC primarily focuses on developing and evaluating interactive lifelog retrieval systems and is known for its competitive nature, attracting numerous participants annually (9 teams in the latest LSC'22). Distinct from the NTCIR Lifelog and ImageCLEF Lifelog, LSC adopts a unique evaluation format. Evaluations are conducted in a real-time environment, where queries are displayed on a screen through a series of clues (with an interval of 30 seconds between two clues). System operators are tasked with searching and submitting relevant images, with scores determined based on the accuracy and timeliness of submissions, and penalties applied for incorrect submissions.



Since the primary target of my dissertation is to develop an interactive retrieval system for infologging data, I will only focus on reviewing the state-of-the-art systems in LSC. Before discussing these systems in detail, I provide a summary of the datasets employed in the benchmarking challenges in Table 2.1. As can be seen from the table, the amount of data significantly increases over time, containing 18 months with nearly 725,000 lifelog images in the latest version of the lifelog dataset. Consequently, the retrieval systems have to be upgraded to manage the dataset to provide efficient retrieval of lifelog moments.

The main task in the Lifelog Search Challenge (LSC) is the Known-Item Search (KIS) task, where retrieval systems must find relevant images from the lifelog dataset based on a series of clues provided in a query. The evaluation of these systems is based on the accuracy and timeliness of their submissions, with penalties applied for incorrect submissions. In LSC'22, two additional tasks were introduced: the Ad-hoc Search (Ad-hoc) task and the Question Answering (QA) task. The primary focus of the Ad-hoc task is to retrieve all relevant images from the lifelog dataset based on a textual query. Retrieval systems are allowed to submit as many images as desired, and the evaluation is based on the precision and recall of the submissions. Unlike the KIS task, no penalties are applied for incorrect submissions in the Ad-hoc task. The QA task, on the other hand, requires retrieval systems to answer the query in a textual format, supported by evidence from retrieved images in the lifelog dataset. The evaluation of this task is based on the accuracy and timeliness of the submission, and the systems are allowed to submit only once. Despite having different evaluation settings, all three tasks can be assessed using a single evaluation metric called the LSC score. This score is a weighted sum of the accuracy and timeliness of the submissions, providing a unified measure of system performance across the different tasks. The details of the evaluation metrics, including the LSC score, are provided in Section 3.3.

Having introduced about the LSC, I will summarise the retrieval systems in the most recent LSC (LSC'21 and LSC'22) and describe the best-performing systems in

detail in the following sections.

### 2.1.2 Lifelog Retrieval Systems at LSC

In the Lifelog Search Challenge (LSC), participating teams primarily employ two distinct approaches for retrieving life events: concept-based retrieval and semantic-based retrieval. Concept-based retrieval is the more traditional method, relying on the analysis of both visual and non-visual content through explicit terms and keywords. This approach typically represents life events by leveraging low-level visual features, such as objects, colors, and text, integrated with associated metadata. In contrast, semantic-based retrieval systems aim to bridge the semantic gap between visual and textual content. These systems overcome the constraints of keyword-based approaches by converting lifelog data into high-dimensional joint text-visual embedding vectors. As a result, semantic-based retrieval not only deepens contextual understanding but also significantly improves the search experience, particularly for novice users.

This section will provide an overview of the various retrieval systems that have participated in the LSC, with a particular focus on those featured in the most recent challenges, LSC21 and LSC22. A summary of these systems, along with their respective approaches, is provided in Table 2.3.

#### 2.1.2.1 Concept-based retrieval systems

Concept-based retrieval has been the predominant methodology in the Lifelog Search Challenge (LSC) from 2018 to 2021, with several systems employing this approach to great success. In this section, I will discuss the systems that emerged as winners in each of these years, specifically VRLE [1], which won LSC'18, vitrivr [105], the winner of LSC'19, and MyScéal [2, 106] – the best-performing system in both LSC'20 and LSC'21. Apart from these winning systems, LSC'21 also saw the participation of various other innovative systems. Each of these systems brought unique features and methodologies to the challenge, contributing to the evolving field of lifelog retrieval.

Table 2.1: A statistics of the benchmarking datasets employed in NTCIR, ImageCLEF and LSC from 2017 to 2023.

Dataset	Version					
	<i>v1</i>	<i>v2</i>	<i>v3</i>	<i>v4</i>	<i>v5</i>	<i>v6</i>
Number of Lifeloggers	3	2	2	1	1	1
Duration	87 days	90 days	43 days	114 days	4 months	18 months
Collection Size (GB)	18.2	26.6	14.0	38.5	37.4	46.3
Number of Lifelog Images	88,124	114,547	81,474	191,439	183,299	725,226
Employed in	NTCIR12 [28] ImageCLEF 2017 [31]	NTCIR13 [29] ImageCLEF 2018 [32] LSC'18 [35]	NTCIR14 [30] ImageCLEF 2019 [33] LSC'19	ImageCLEF 2020 [34] LSC'20 [36]	NTCIR16 [85] LSC'21 [37]	LSC'22 [39] LSC'23 [86]

Table 2.3: List of participating systems in LSC’21 and LSC’22 and key approaches employed by them. This table is an extension of that in [38] to include systems in LSC’22

	Searching						Browsing		
	Concept search	Embedding	OCR	Query-by-example	Temporal query	Relevant Feedback	Event/day summary	Location visualisation	Novel interaction
MyScéal [3, 4]	✓		✓	✓	✓		✓	✓	✓
SomHunter+ [87]		✓		✓	✓	✓			
LifeSeeker [41, 42]	✓		✓	✓	✓		✓		
Voxento [7, 88]		✓							✓
CVHunter [87]		✓		✓	✓				
Memento [5, 89]		✓			✓				
FIRST [6, 90]	✓	✓		✓				✓	
NTU-ILRS [91]	✓	✓			✓	✓			
lifeXplore [92, 93]									
LifeMon [94]	✓								
vitriVr [95, 96]	✓	✓			✓	✓		✓	
vitriVr-VR [97, 98]	✓	✓				✓			✓
XQC [99]	✓					✓			✓
Exquisitor [100]	✓				✓	✓			
PhotoCube [101]	✓								✓
ViRMA [102]	✓								✓
LifeGraph [103]	✓	✓							
VRLE [1]	✓				✓				✓
MEMORIA [104]	✓				✓				

Their approaches and functionalities are also summarised at the end of this section.

### VRLE [1] - A Virtual Reality Lifelog Explorer

The Virtual Reality Lifelog Explorer (VRLE), developed by Aaron Duane et al. [1], represents an innovative step in lifelog data exploration and retrieval, utilising virtual reality (VR) technology. It is the winning system in the first LSC in 2018.

VRLE’s functionalities can be summarised as follows:

- **Data Indexing Method:** The system focuses primarily on visual concepts extracted from the lifelog data, which is provided as part of the development dataset by the organiser. Each image in the dataset is annotated with outputs from a state-of-the-art computer vision concept detector [107], providing a listing of real-world concepts (like computer, car, coffee) for each image. These concepts, in combination with additional metadata, such as dates, activities, and locations are indexed by the system for concept matching and filtering.



Figure 2.1: User interface of VRLE (adapted from [1])

- Data Retrieval Method:** VRLE enables users to construct filter queries using a VR interface by selecting relevant concepts to the query and specifying time ranges of interest. The retrieval process ranks the results by relevance of concepts and time, prioritising concept matching. The user can then browse this ranked list of images and select images for further inspection or apply more filters until the desired image is found.
- User Interface:** Employing a VR platform (specifically, HTC Vive), VRLE offers a rich, immersive experience for lifelog data interaction [1] (as shown in Figure 2.1) Users engage with the system using gesture-based or contact-based methods to select concepts and time ranges through a virtual interface. The system’s design enables a novel and intuitive mode of exploring and retrieving lifelog data, enhancing the user experience by leveraging the unique capabilities of VR technology [35].

## Vitrivr [105] - A Retrieval System for Structured and Unstructured Data

The vitivr system was initially an open-source content-based retrieval system designed for video retrieval [108]. Vitivr system was adapted to work on lifelog data, which is mainly images. It first participated in LSC in 2019 and achieved the highest result in LSC'19.

The following is a breakdown of the system's components:

- **Data Indexing Method:** The vitivr system [108] utilises a modular multimedia information retrieval stack, which supports various media types including images, video, audio, and 3D models. It employs ADAMpro [109] as the main database to store large volumes of content, which supports data distribution and various index structures for nearest-neighbour queries. For lifelog retrieval, image data is processed through deep neural networks for object classes, image captions, OCR, and action recognition. The system uses a new media type called image sequence to store and process the lifelog images as segments of one document per day. This allows the system to attach metadata to individual images as well as the entire day.
- **Retrieval Algorithm** vitivr combines Boolean retrieval and similarity-based retrieval to handle the heterogeneous lifelog data. The system allows the user to formulate Boolean expressions using the metadata attributes, values, and comparison operators. The system evaluates the expressions using dedicated feature modules and applies them as a filter to the similarity-based results. The system also uses a score-based late fusion approach to combine the results from different feature modules, which include visual, textual, and deep learning-based features.
- **User Interface** The user interface is a browser-based application implemented in Angular and TypeScript. It supports various query modes, such as Query-by-Sketch, Query-by-Example, and textual search. The system also allows the formulation of complex Boolean queries using query containers and query terms. Vitivr provides several result views that present the retrieved results in

different ways and support refinement of search results by applying additional filters

### MyScéal [2, 106] - An Experimental Interactive Lifelog Retrieval System

MyScéal is the winning system of both LSC'20 and LSC'21, which is a combination of an efficient search engine and an informative user interface with various options to view and interact with the search results.

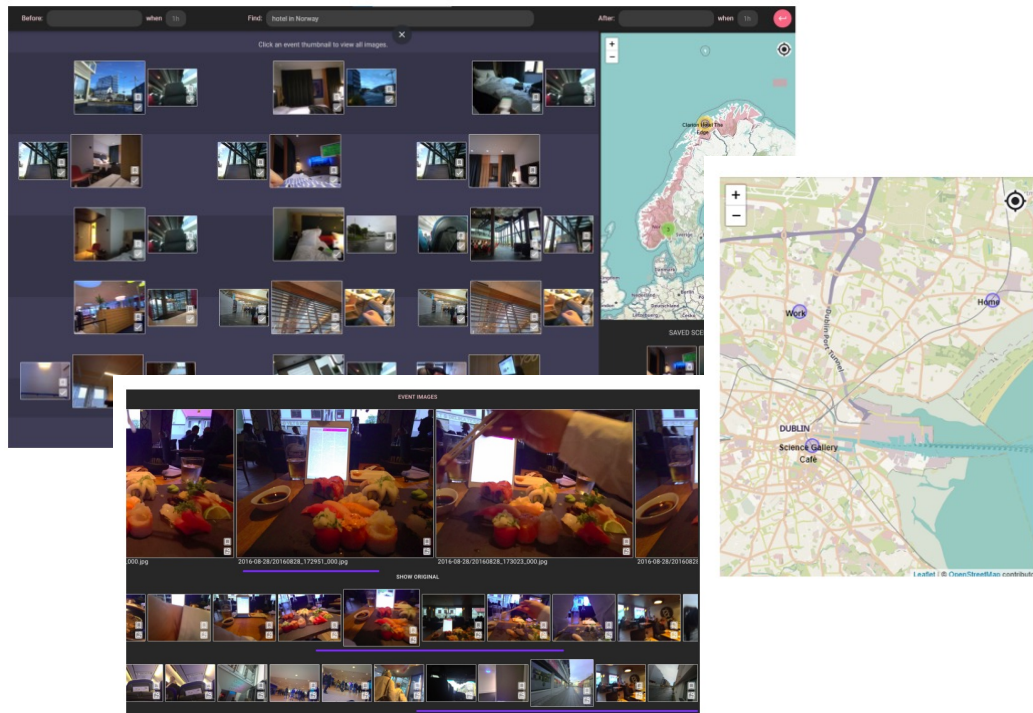


Figure 2.2: User interface of MyScéal (adapted from [2, 3])

In general, MyScéal is comprised of:

- **Data Indexing Method** Apart from the visual concepts provided by the organiser, MyScéal enriches its data indexing by incorporating additional object detection using DeepLabv3+ [110], which enhances detection accuracy. Alongside this, the system integrates colour features, text recognition, and logo detection to augment image annotations. This enriched data, coupled with metadata like GPS location, date, and activities, is indexed using

Elasticsearch, a renowned search and analytics engine.

- **Retrieval Algorithm** The retrieval process in MyScéal combines filtering and ranking methodologies. Filtering is executed based on the precise match of query terms with indexed fields, including location, time, activity, and others. The system also employs a query expansion method, leveraging Word2Vec [111] and WordNet [112], to enhance the likelihood of matching relevant concepts in the database. The ranking is conducted using a novel algorithm, aTFIDF [2,3], which reflects the relative importance of each visual concept within an image. Additionally, MyScéal supports temporal retrieval, enabling users to search for events sequentially using *before* and *after* keywords.
- **User Interface** The user interface of MyScéal, as shown in Figure 2.2, is designed for simplicity and efficiency, primarily focusing on text queries while minimising the use of faceted search. Users can input up to three parts of a query, indicating the main event and its preceding and succeeding events. Search results are displayed as a ranked list of images, with each image symbolising an event. Users can click on any image to view more images within that event. A geographic map feature allows for filtering results by drawing on the map. Additionally, the system facilitates visual similarity searches and includes utilities to assist novice users, such as reset buttons, zoomed views, pop-up reminders, and word highlighting features.

### Other systems at LSC'21

The literature on participating teams in the LSC'21 showcases a diverse set of systems, each with unique features and methodologies. **Voxento** [113] introduces a novel voice-based retrieval approach, integrating Google's web speech API for speech recognition and synthesis, enabling vocal command interactions. **FIRST** [90] experiments on a self-attention-based joint embedding model and support for multiple modalities, including textual querying and query by example.



**LifeConcept** [91] reduces the semantic gap between textual queries and images through word embeddings and relation graphs, incorporating ConceptNet [114] for concept selection. **lifeXplore** [92] provides users with chronologic day summary browsing, and interactive and combinable concepts filtering. **LifeMon** [94] employs MongoDB to store and query lifelog data as semi-structured documents, and provides a web-based user interface for filtering and exploring the results. **vitriivr-VR** [97] extends the vitriivr [95] system with a VR-based interface, offering immersive interaction with retrieval results. **Exquisitor** [100] explores interactive learning in multimedia analytics, allowing users to evolve a semantic classifier through cooperation. With the system sharing the engine with Exquisitor [100], **XQC** [99] brings a cross-platform interface to the challenge which supports lifelog retrieval through a mobile app. **PhotoCube** [101] introduces a novel approach to organise media items and metadata into a hypercube in multidimensional space and allows users to explore a lifelog via a three-dimensional exploration cube. Similarly, **ViRMA** [102] shares PhotoCube’s back-end server but introduces VR-based navigation and browsing of search results. Finally, **LifeGraph** [103] presents an experimental approach which links detected objects in images to an external knowledge base and employs graph traversal for query processing. LSC’21 also marks the shift towards semantic-based retrieval methods, as evidenced by the adoption of the CLIP model by OpenAI [115] in systems like **SomHunter+** [87] and **Memento** [5] for text-to-image matching, by comparing the cosine similarity of the query and images’ embeddings.

### 2.1.2.2 Semantic-based retrieval systems

The Lifelog Search Challenge in 2022 witnessed a significant shift in the adoption of Vision-Language models, particularly CLIP (Contrastive Language–Image Pretraining) developed by OpenAI [115], within many retrieval systems. These models are efficient in unifying various data modalities into a single vector space, thereby simplifying the overall process of information retrieval. This technological

advancement not only streamlined the system interfaces, making them more accessible and user-friendly but also notably enhanced their performance. The effectiveness of these systems, particularly those utilising CLIP in their primary search mechanism, is evident in their scores in the challenge, which will be further elaborated in Chapter 4. This section will focus on the specifics of some of the top-performing systems in LSC'22, which utilise CLIP as part of their search mechanism.

### E-MyScéal [4] - Embedding-based Interactive Lifelog Retrieval System

E-MyScéal inherits the core design and functions of MyScéal in LSC'21, with the integration of a text-image embedding model, making it a semantic-based retrieval system.

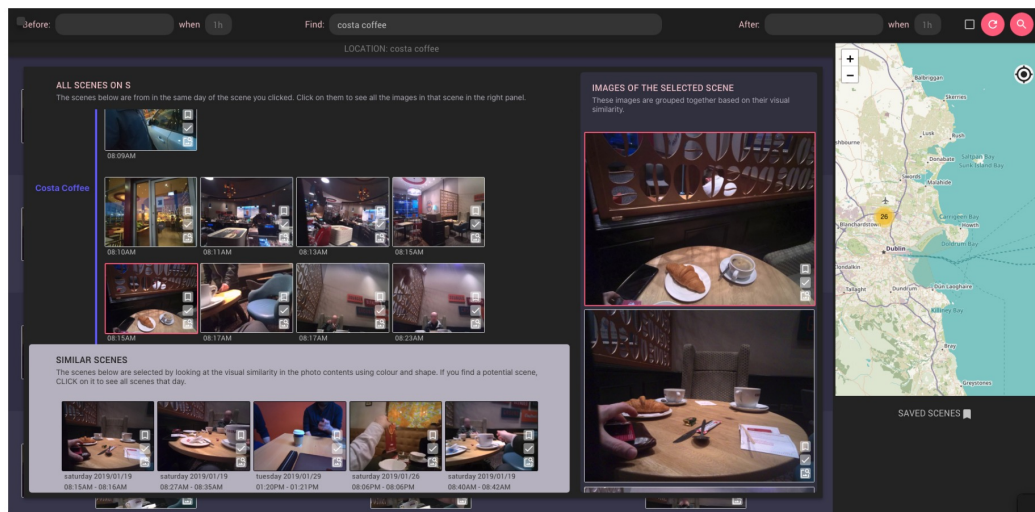


Figure 2.3: User interface of E-MyScéal (adapted from [4])

The system's functionalities are outlined as follows:

- **Data Indexing Method** E-MyScéal maintains the initial structure of the original MyScéal versions but with a critical change in its indexing strategy. By incorporating embedding models, particularly CLIP [115], the system allows for a more intuitive matching of textual queries with the rich visual data of lifelogs.

- **Retrieval Algorithm** The core of E-MyScéal’s retrieval mechanism is the embedding-based approach using the CLIP model. Apart from this, other functionalities (such as the filtering mechanism, and relevance feedback modules) from its previous iteration remain unchanged.
- **User Interface** E-MyScéal presents an interface that balances simplicity and efficiency. As depicted in Figure 2.3, the system streamlines its non-faceted interface, removing elements that previously confused to the user, while enhancing its event view browsing.

### Memento [89] - An Interactive Retrieval System for Lifelogs

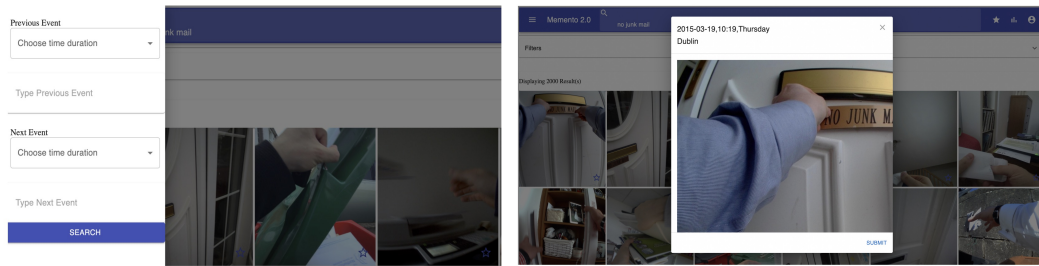


Figure 2.4: User interface of Memento (adapted from [5])

Memento’s functionalities can be summarised as follows:

- **Data Indexing Method:** Memento employs two versions of the CLIP model, ViT-L/14 [116] and ResNet-50x64 [107], to create a joint embedding space for lifelog images and their associated captions. The resulting embeddings are stored as static files to optimise the system retrieval speed [89].
- **Data Retrieval Method:** The system takes a natural language query as input and encodes it using the text encoders of the CLIP models. It then computes the cosine similarity between the query embedding and the image embeddings and ranks the images based on a weighted sum of the scores from the two models. The system also supports temporal search and navigation,

which allows the user to search for a target event in the context of a temporally close past or future event, by specifying the event and the time duration.

- **User Interface** The system has a web-based user interface that displays the ranked images in a grid layout (see Figure 2.4), allowing users to easily navigate and zoom into specific images. The interface also provides access to various functionalities, including visual data filtering, temporal search, and a repository of starred images (to save the potential results). It also offers statistics on the execution of a query, including the number of images retrieved, the query processing duration, and a confidence score for each result.

### FIRST [6] - Flexible Interactive Retrieval SysTEM for Visual Lifelog Exploration

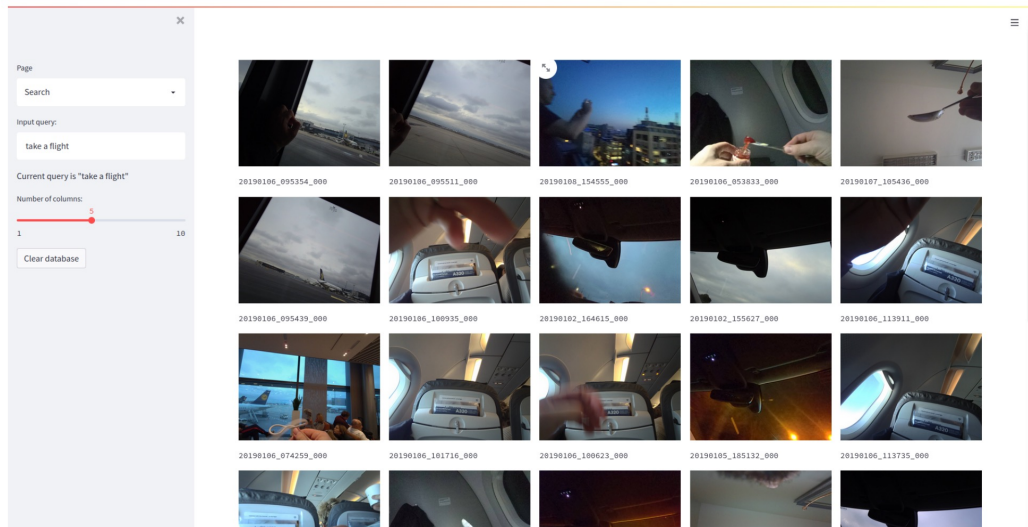


Figure 2.5: User interface of FIRST (adapted from [6])

FIRST's functionalities can be summarised as follows:

- **Data Indexing Method:** The system efficiently reduces the size of the large lifelogging data collection through pre-processing steps like filtering blurry images, normalising image orientation, and grouping similar images to

avoid overcrowding in search results. Keyframes are selected from each group of contiguous similar images (shots) and then clustered based on their GPS information and hierarchical relationships of locations. The images are also indexed in the database with metadata such as time, location, texts in the image, and visual concepts using the CLIP model and Conceptual Captions tags

- **Data Retrieval Method** The system employs CLIP [115] to extract high-dimensional representations from images. It focuses on both general and local features by encoding important regions of an image at different levels of granularity. This approach allows the system to represent an image with an adaptive semantic embedding set, which enhances its ability to match new concepts not present in a pre-defined dictionary. Moreover, by leveraging the CLIP model, the system uses similarity modelling to extend its capabilities. It defines the distance between images based on the cosine distance between their embeddings, allowing for searches using visual examples. The system also integrates external systems like Google Search to find visual examples of unfamiliar concepts, expanding the scope of search beyond its existing concepts
- **User Interface** As displayed in Figure 2.5, the system provides a user-friendly interface with multiple visualisation and interaction modules. It offers scene clustering using CLIP features and heuristics and flexible temporal navigation for quickly browsing through photos with adjustable detail levels. Local/prototype visual search in FIRST allows users to search using external image examples. This feature is particularly useful for searching unknown concepts, where users can input URLs of images for quick and intuitive searches

**Voxento [7] - A Prototype Voice-controlled Interactive Search Engine for**

## Lifelogs

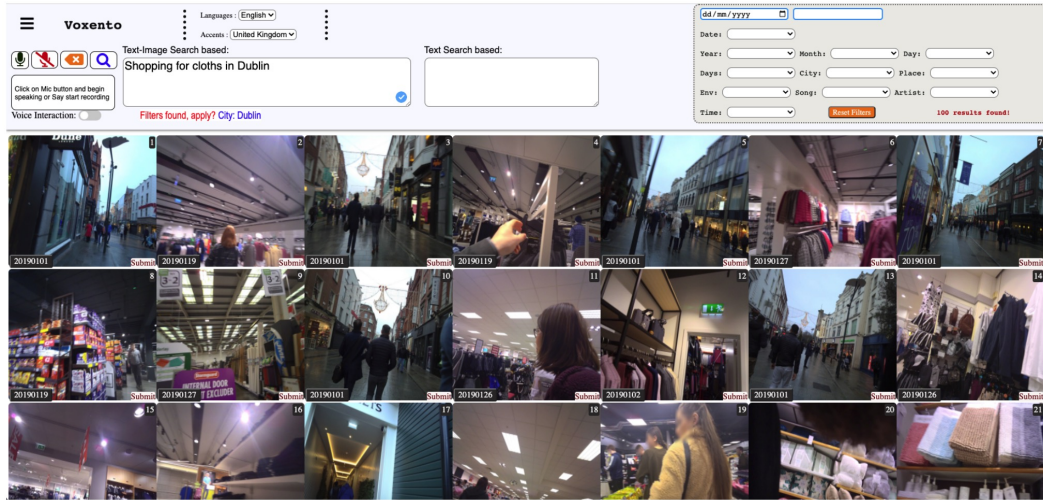


Figure 2.6: User interface of Voxento (adapted from [7])

Voxento’s functionalities can be summarised as follows:

- Data Indexing Method:** Voxento employs the CLIP model [115] to transform images into high-dimensional representations, facilitating a comparison with encoded query sentences through cosine similarity. The system enhances its indexing efficiency by implementing event segmentation, and grouping images based on activities and locations. Additionally, it refines the indexing data by excluding blurred, duplicated, or irrelevant images, ensuring that only the most relevant and clear images are indexed and retrievable.
- Data Retrieval Method:** The system has two search engines: one for text-image search and one for text-based search. The text-image search engine uses the CLIP model to rank images based on their similarity to the query. The text-based search engine uses the metadata to find images that match specific concepts such as semantic names, artists, or songs. The voice interaction feature in Voxento utilises the Google Web Speech API for speech recognition and synthesis. Users can interact using voice commands or opt for standard mouse and keyboard input. The system is designed to submit search queries

live as the user speaks, dynamically updating the results based on the evolving query

**User Interface** The system’s web-based interface is designed to support both voice interaction and conventional text-based retrieval. Users can input queries using vocal commands, initiated by the phrase "Start recording," followed by their spoken query. Alternatively, they can use traditional mouse and keyboard inputs. The interface also includes dynamic filters, accessible through a filtering menu with drop-down boxes for criteria such as time, date, location, environment, semantic name, artist, and song.

### Other systems at LSC’22

MEMORIA [104] – a concept-based retrieval system – marked its first participation at LSC’22. As a web-based, concept-based retrieval system, it allows users to upload, annotate, and search personal lifelog data using keywords, time frames, and filters. It utilises advanced computer vision methods, including Yolov5l6 [117], a model pre-trained on the Places365 dataset [118], and ResNeXt-101 [119], to extract relevant information from images efficiently.

vitivr [95] participated for the fourth time at LSC’22. Although no significant changes were made to it, the evaluation of the system at LSC’22 serves as baseline results for evaluating its VR variant – vitivr-VR [96]. vitivr-VR [96], which previously outperformed vitivr in LSC’21, enhances the user experience by offering more results viewing modes (cylindrical view, single media view and sequence view) and supporting users to formulate the multimodal query within the VR environment. lifeXplore [93] was also a returning system with improvements mainly to its user interface. The system now leverages geolocation data for improved location search and event filtering. It introduces a results ranking system based on different criteria like date, time, and location and has improved result browsing with pagination in its filter view [93].

## 2.2 Eye movements in Reading Comprehension

In this section, an introductory overview of human eye anatomy is presented in Section 2.2.1, providing a crucial understanding of the mechanisms underlying eye movements. Following this, Section 2.2.2 explores the technologies used for eye movement tracking, focusing on devices such as eye trackers. Subsequently, Section 2.2.3 summarises the fundamental characteristics of eye movements during reading, laying the groundwork for my analysis in the context of reading comprehension estimation. The section concludes with a review of existing literature in the field of investigating reading comprehension via eye movement measures, outlined in Section 2.2.4, where I identify and aim to bridge a research gap in this thesis.

### 2.2.1 The human eyes: Structure and Function

The human eye, a marvel of biological evolution, serves as the primary organ for vision [120,121]. Its intricate structure is specifically designed to capture, focus, and process light, turning it into interpretable visual information. This section provides an overview of its fundamental anatomy and the subsequent physiology of sight. Figure 2.7 shows a vertical slice of the human eye, revealing its internal structure. As illustrated, the eyeball possesses a spherical shape, optimizing it for capturing a broad field of vision. The complex components making up this essential visual organ can be categorized into three principal layers: Fibrous layer, Vascular layer, and Inner layer [121]

#### 2.2.1.1 Fibrous layer

Serving as the outer shield of the eye, the fibrous layer is fundamental to both protection and light entry. This layer is primarily composed of the *sclera* and the *cornea*. Sclera is often referred to as the "white" of the eye, the sclera is a thick, tough tissue that surrounds most of the eyeball. It provides both structural integrity and a protective barrier against foreign threats. Its opaqueness ensures that light



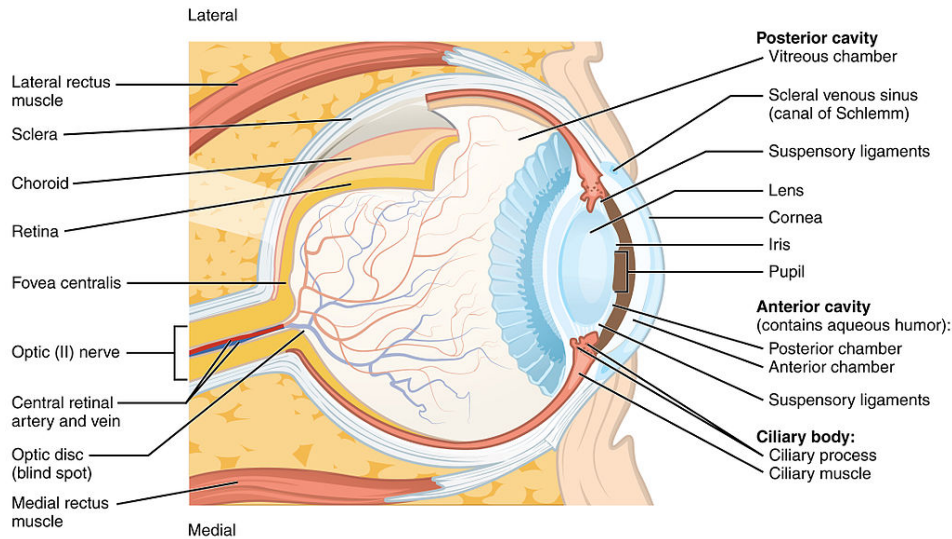


Figure 2.7: The anatomy of the human eye, from OpenStax College, licensed under CC BY 3.0, via Wikimedia Commons

can only enter the eye through the cornea. The sclera also offers attachment sites for the extrinsic muscles of the eye, enabling the movement of the eyeball in different directions. The Cornea is a transparent, dome-shaped surface that stands centrally in the front of the eye. Beyond its primary role of protecting the eye from dust and germs, its smooth and transparent properties are crucial in refracting light coming into the eye. The cornea's unique curvature and refractive index help focus light onto the retina.

### 2.2.1.2 Vascular Layer

Positioned beneath the fibrous layer, this segment encompasses the *iris*, *choroid*, and *ciliary body*. The iris, which determines our eye's color, contains a hole at its centre named the pupil. By dynamically adjusting its size, the pupil controls the amount of light entering the eye. This adjustment is orchestrated by two sets of fibers: the *sphincter* and *dilator*. The sphincter, a circular muscle triggered by bright light, contracts the pupil and is regulated by the parasympathetic nervous system. Conversely, the dilator, a radial muscle, expands the pupil when in the dark and is controlled by the sympathetic nervous system. The choroid is a vascular

layer responsible for nourishing the retina with blood. Finally, the ciliary body is a muscular ring, which dictates the shape of the *lens*. It comprises the ciliary muscle and the ciliary process. The muscle contains smooth muscles in three distinct orientations: longitudinal, circular, and radial. When these muscles contract, the size of the circular ciliary body diminishes. The ciliary process, meanwhile, connects the ciliary body to the lens through *zonular fibers*. This intricate arrangement plays a pivotal role in adjusting the lens's curvature based on viewing distances, ensuring clear vision.

### 2.2.1.3 Inner Layer

At the core of the visual system lies the *retina*, the layer tasked with detecting light. It comprises two distinctive layers: the outer *pigmented layer*, which absorbs light, preventing scattering, and the inner *neural layer*. The latter is populated with approximately 120 million photoreceptors—rods and 6 million cones [122]. Rods cater to low-light scenarios, whereas cones operate best under bright conditions and are pivotal for colour vision. The three cone subtypes, L-cones, M-cones, and S-cones, are sensitive to red, green, and blue wavelengths, respectively. The varying ratio of these cones across individuals underpins the differences in colour perception. Central to the retina is the *fovea*, a locus responsible for sharp vision. Surrounding it is the *macula*, the hub of our central and colour vision. Intriguingly, the retina also houses an area devoid of photoreceptors—the optic disc. Located nasally, this "blind spot" is where the *optic nerve* and *blood vessels* exit the eye. The absence of visual input from this region is seamlessly compensated by our brain and the continuous movement of our eyes, rendering the blind spot virtually undetectable in daily perception.

### 2.2.1.4 Eye Movement in Reading

The process of sight begins as light enters the eye through the cornea, the fibrous layer, which refracts the light onto the lens. The lens, part of the eye's vascular layer,

then adjusts the focus, directing the light onto the retina in the inner layer of the eye. Here, photoreceptors (rods and cones) convert the light into electrical signals, which are transmitted to the brain via the optic nerve, allowing us to process and interpret these signals as visual images.

During reading, light reflected from the text enters the eye through the cornea and pupil and is focused onto the retina by the lens. Although the image of the text is projected onto the retina, clear perception of this image is limited by the sparse distribution of photoreceptors across most of the retina's surface. Only the fovea, a small central area of the retina, contains a high density of photoreceptors, enabling acute visual detail necessary for activities like reading. Due to the fovea's small size, it can only cover a very small part of the visual field and we must constantly move our eyes to align different parts of the text with the fovea. This alignment ensures that we perceive the text with the greatest possible clarity. These eye movements are called saccades, the rapid movements between points of fixation. A fixation occurs when the eyes stop briefly to focus on a particular part of the text, allowing for detailed processing of the visual information. In Section 2.2.3, the characteristics of fixations and saccades during reading will be discussed in more detail.

### **2.2.2 Capturing eye movements: The Eye-tracker**

In the late 19th century, pioneers like E. Huey [123] and E.B. Delabarre [124] undertook initial attempts to study eye movements using invasive techniques. Huey developed an eye tracker that utilised a contact lens linked to an indicator to measure eye movement, while Delabarre's method involved a gypsum cap attached to the eye's surface. Both techniques were so intrusive that the authors gave their participants cocaine to alleviate their discomfort during the study. In 1901, a significant advancement was achieved by R. Dodge and T.S. Cline [125], who introduced "The Dodge Photochronograph", a non-invasive optical eye tracker, which, despite its limitations in capturing only horizontal movements and necessitating subjects to keep their heads still, highlighted that a human's

reception of information does not happen during their saccadic movements.

During the 1950s, various eye-tracking techniques emerged to study gaze patterns. The *lense system with mirrors* involved a specialized contact lens with an attached mirror, reflecting the eye's movement directly onto a recording device [126]. This method, though direct, was somewhat invasive. The *electromagnetic coil system* utilised a coil around the eye, where eye movements within a magnetic field resulted in an electric current. The current's magnitude and direction could then be used to determine the eye's position and movement [127]. Electrooculography (EOG) employs electrodes placed around the eyes, capturing the corneo-retinal standing potential to gauge the direction and magnitude of eye movements. Among these, the Dual Purkinje Systems stood out for its precision. By tracking the reflections from both the eye's surface and the lens, it offered an accurate measure of eye rotations. However, this method was known for its high-cost and difficulty in maintaining [128].

Over the 20th century, the evolution of eye-tracking technology accelerated as various sectors began recognising its potential uses. Besides academic researchers, the media and advertising sectors started using eye tracking to gauge reactions to their marketing campaigns [128]. The medical community employed eye-tracking for diagnosing and treating ocular conditions [128]. People from the field of human-computer interaction adopted it to refine user interface designs [128]. As a result, eye-tracking technology grew significantly in variety, having a wide range of options for accuracy, cost, ease of use, and invasiveness.

To date, video-based eye tracking has emerged as the predominant technique for monitoring eye movements. These systems utilise infrared-sensitive cameras, infrared lighting (illumination), and image processing algorithms. These algorithms focus on detecting the pupil's center and locating corneal reflections, which ultimately pinpoint where the individual is looking. Generally, video-based eye trackers fall into three categories:

- *Static eye-tracker*: Typically positioned on a table or desk facing the

participant. There are two main variations: the *tower-mounted*, which restricts head movement, and the *remote*, allowing for small head movements, as long as it remains within the camera's field of view.

- *Head-mounted eye-tracker*: In this configuration, both the light source and camera are mounted on the user, typically via helmets, caps, or glasses. It offers the advantage of letting participants move unencumbered and capturing their first-person perspective.
- *Head-mounted eye-tracker with an auxiliary head-tracker*: This variant integrates an additional tracking system to determine the head's spatial positioning. This supplementary data stream makes the analysis of eye-tracking data much easier

The eye tracker that I used in this thesis is a low-cost remote static eye-tracker manufactured by *Gazepoint* (model GP3 HD) [129]. Similar to most remote eye-trackers, it emits low-level infrared light (which is safe for the eyes) and this light is reflected off the cornea of our eyes. A built-in camera then captures the corneal reflections and the pupil's center and sends this to the image processing unit to analyse the relative position between the pupil and reflections. The internal tracking algorithms also validate the data quality and apply a series of filters to generate useful eye movement features (e.g., fixation, saccade, and blinks) based on the estimated gaze coordinates for each frame. The eye-tracker operates at 60Hz or 150Hz which allows real-time tracking of eye movements [129]. Its accuracy is reported to be 0.5-1.0 degrees of visual angle and allows 35cm x 22cm for horizontal and vertical tracking, respectively and +/- 15 degrees for depth movement [129].

Figure 2.8 shows the position of the eye as interpreted by the tracker. The device's algorithms have effectively identified and circled the pupil and corneal reflection, enabling precise gaze coordinate determination. To achieve optimal tracking results, the recommended distance between the participant and the eye-tracker is approximately 60-65cm [129]. This can be equated to the distance of

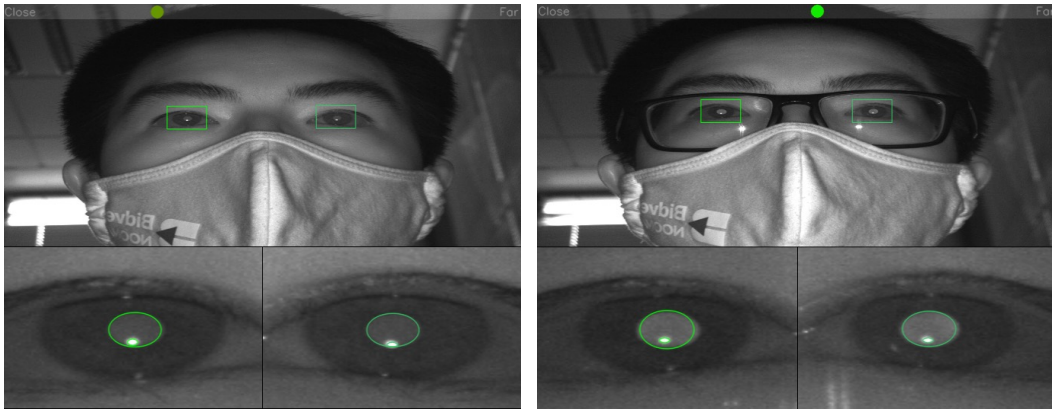


Figure 2.8: Corneal reflection and pupil detected by the eye-tracker. The figure shows the tracking in two scenarios: with and without glasses (right and left image, respectively), under the same distance from the eye-tracker.

an arm's length. Within Figure 2.8, a bar is present with indicators for *Close* and *Far* alongside a green dot. Ideal positioning is achieved when the green dot aligns centrally with the bar, as illustrated in the right image of Figure 2.8. It is important to acknowledge that the use of corrective eyeglasses can influence this optimal distance. Without glasses, the participant should be situated slightly further from the tracker. Figure 2.8 shows that under the same distance, the image on the left (without glasses) was indicated to be quite close. Moreover, proper angling of the tracker is essential when engaging participants with glasses to prevent reflections of the environment's light on glass lenses, preventing a clear view of the pupil and corneal reflection point. Following the positioning phase, calibration of the eye tracker is essential. This involves presenting the participant with one or more targets to visually track while the device collects data on eye positioning relative to these targets. Upon successful calibration, the eye tracker is ready to track the participant's eye movements.

### 2.2.3 Basic Characteristics of Eye Movements and Application in Recognising Reading Strategies

Eye movements play a fundamental role in the process of reading comprehension and I will begin with an overview of their basic characteristics. There are two

primary measures commonly used to analyse eye movements, which are: *fixations* and *saccades* [67]. Fixations are brief pauses in eye movement when the eyes focus on a specific point which generally lasts around 200-300 milliseconds [130]. During a fixation, visual information is processed and integrated. The duration of fixations is influenced by the complexity and demands of the text, with longer fixations typically observed during the processing of difficult or unfamiliar content [131]. On the other hand, saccades are rapid eye movements that shift the gaze from one fixation point to another [130]. Saccades allow us to move our eyes quickly across a text, allowing us to sample information efficiently. Many features derived from fixations and saccades have been widely applied to address many detection and recognition tasks [132–135].

Concerning reading, eye movement measures were found to be useful in recognise the reading strategies used by readers [73–75]. Biedert et al. [73] proposed a robust real-time detection method for reading and skimming activity by using a window-based technique to acquire saccades data and calculate corresponding features to perform classification. The authors obtained a result of 86% when classifying these two activities. Also, they found that the *average forward speed* and *angularity* are the most robust features for the model's generalisation and accuracy. Liao et al. [74] further extended the classification of reading strategies to five types, namely, speed reading, slow reading, in-depth-reading, skim-and-skip and keyword-spotting in using temporal (e.g. fixation duration, saccade duration) and spatial (e.g. saccade length, saccade direction) eye movement features. An average accuracy of 86.07% was obtained on a 5-fold leave-one-group-out cross-validation setting, suggesting that eye movement measures are very informative in distinguishing reading patterns. However, it is worth noting that to induce the expected reading strategies, the participants' reading process was partially controlled by the authors. For example, parts of the text were highlighted in skim-and-skip activity for the participants to follow or texts were designed to include two halves, one half is a text with clozes (a.k.a. blanks) and the other is a list of answers to fill in the clozes, in keyword-spotting

activity for the participants to find the matching keywords for the blanks [74].

To fully understand how reading activities are performed in real-world scenarios, it is necessary to allow the participants to read freely without controlling their reading process. This was addressed by Strukelj et al. [75] who investigated the characteristics of different commonly used types of reading such as regular reading, thorough reading, skimming, and spell checking, by collecting eye tracking data of participants reading a single page of text under instructed reading types and performing a comparative analysis of the eye movement features between regular reading with each of the other three reading types. The authors formed 7 to 9 hypotheses for each pair of reading types and tested them using statistical tests. According to the results, compared to regular reading, thorough reading showed higher comprehension scores with longer total reading times and more rereading. On the other hand, skimming resulted in lower comprehension scores with longer saccades, shorter *average fixation durations*, more *word skipping*, and shorter *total reading time*. Similarly, spell checking also resulted in lower comprehension scores, but with shorter saccades, longer *average fixation durations*, less *word skipping*, and longer *total reading time*.

These findings again, highlighted that eye movement measures are very informative in distinguishing reading patterns. Nonetheless, the applicability of findings to date in developing machine learning models for recognising reading strategies remains an open question since the changes in eye movement measures were found when performing pair-wise comparisons of reading types. How these changes contribute to a multi-class classification problem is unexplored.

#### **2.2.4 Eye Movements in Estimating Reading Comprehension**

Decades of eye-movement research have generated significant knowledge of how they are coupled with a human’s cognitive processes [67, 71]. When reading English text, saccades are typically 7-9 characters long and last around 20-30 milliseconds [67]. While most saccades bring the eyes forward, some saccades can



also be *regressive* which brings the eyes back to previously read words or lines. In skilled readers, regressive saccades (also called *regressions*) account for around 10%-15% of all saccades [136]. When reading a difficult text, the *number of regressions* increases as well as the *duration of fixations*, while the *saccade length* decreases [136]. These findings underlay research in estimating the level of reading comprehension from eye movement features.

While most studies found significant correlations between eye movements and reading comprehension [76–80], establishing a standard method to monitor reading comprehension remains a challenge [81,82]. Despite having different approaches and findings, these studies do share a common procedure regarding data collection and evaluation. First, participants are asked to read a given sentence or text silently. Then they are asked to answer a set of questions about the text, either multiple-choice (MCQ) or both multiple-choice and cloze (e.g. fill in the blank) questions [78]. Some studies also record participants' subjective ratings of their understanding/difficulty of the text to give an external reference to the analysis of comprehension [76,81]. Finally, the eye movements are recorded and analysed to estimate the level of reading comprehension either as a classification task (predicting classes of comprehension level, such as low, middle, and high) or as a regression task (predicting a continuous value of comprehension level, such as a score from 0 to 100).

Copeland et al. [78] explored the answer-seeking behaviour during reading by analysing the changes in eye movement from the first read-through (with no specific purpose/requirement) to the second read of the same text (to answer questions). The authors observed that proficient readers have higher reading intensity (higher *numbers of fixations* and *regressions*, longer *total fixation duration*) in the first pass reading compared to the second pass reading. This suggests that reading intensity is a good indicator of reading comprehension [78]. This work was extended by Copeland et al. [77] to employ artificial neural networks to further enhance the prediction of comprehension level. In this study, the reading behaviour was captured and assessed in four different formats: (A) text first, then text and questions presented

simultaneously; (B) text and questions presented simultaneously; (C) text first, then questions; and (D) questions first, then text and then questions again. The results revealed that the misclassification scores (MCR) were lower in formats A, B, and D (0.14, 0.11, and 0.21 respectively) compared to format C (0.51). The authors argued that it is hard to predict comprehension level in format C since the questions are presented after the text and the participants have to read the text until they are satisfied to answer questions [77]. This caused the eye movement behaviour to vary differently from the other formats where the questions were known.

Ahn et al. [81] also proposed to use neural networks to monitor the complex relationship between eye movement features to predict comprehension level when reading SAT (Scholastic Aptitude Test) passages. In this study, multiple factors related to comprehension were investigated including: (1) overall comprehension – an individual’s comprehension over all passages; (2) passage comprehension – an individual’s comprehension over a single passage; (3) reading difficulty – individual’s rating of the difficulty of the passage; and (4) first language – whether the individual’s first language is English or not. The authors adopted two neural network architectures, namely a convolutional neural network (CNN) and a recurrent neural network (RNN), to perform a binary classification task (high/low level) on each of the aforementioned factors. The results showed that the CNN model achieved 65% accuracy for overall comprehension, with an increase of 11% from baseline accuracy (54%). However, the authors concluded that the features extracted from eye movements are not sufficient for the CNN model to predict comprehension levels despite showing significant differences between high and low comprehension groups in statistical testing.

Unlike the previous studies, Southwell et al. [80] attempted to predict comprehension levels after reading long connected texts. Three eye-tracking datasets that have comprehension assessments after reading were employed in this study to address the task. Of these, two datasets have one single passage with 6500 words and the other dataset has 8 passages with 1000 words each. The

authors adopted linear models to predict comprehension levels and found that there is a strong association between eye movement measures with comprehension during reading [80]. The correlations between the observed and predicted comprehension scores on three datasets were within the range of 0.362 to 0.384. Interestingly, the authors also found that eye movement measures have generability across datasets, as models trained on one dataset can be used to predict comprehension levels on other datasets with nearly similar performance.

Furthermore, the correlation between eye movements and language proficiency has been explored by Yoshimura et al. [79] in their research. The study utilised eye movement characteristics to categorise participants into three levels of TOEIC (Test of English for International Communication) scores: low, middle, and high. Additionally, the authors discovered that the combined metrics of *fixation duration* and *saccade velocity* provide valuable insights for classifying TOEIC levels. In a different study, Makowski et al. [82] proposed the integration of *scanpaths* and *lexical features* of fixated words through generative models to identify readers' identity and assess their comprehension. Although the identification of readers' identities yielded favourable outcomes, none of the examined approaches accurately predicted text comprehension. Sanches et al. [76] conducted a study where they introduced *subjective understanding*, measured through subjective ratings, as an alternative evaluation method for reading comprehension, surpassing the traditional objective comprehension assessment of answering questions. This approach was considered more natural for evaluating comprehension in real-world scenarios [76]. Notably, the authors observed a substantial 13% improvement in subjective understanding estimation through eye gaze features compared to comprehension questions, indicating a strong association between eye movements and subjective understanding.

To provide further details on these studies, I have summarised some main characteristics of the datasets used in these works in Table 2.4.

Table 2.4: Overview of datasets used in the existing literature that investigates reading comprehension through eye tracking data.

Reading Comprehension Dataset	No. Subjects	No. Texts	Avg. Length (words)	Texts' Source	Language	Subjective Evaluation (Likert scale <sup>*</sup> )	No. Questions (per text)	Assessment Point	Task type	Eye Tracker Model
<b>Sanchez et al. [76]</b>	17	19	210	JLPT textbook	Japanese	[1, 5]	no	N/A	regression	Tobii
<b>Southwell et al. [80]</b> - <i>Dataset 1</i>	104	1	6500	Book (Boys 1890)	English	**	38 MCQs	during and after reading	regression	Tobii TX300 (120Hz)
- <i>Dataset 2</i>	130	↓	↓	↓	↓	↓	12 MCQs	after reading	↓	Tobii T60 (60Hz) & Tobii TX300 (120Hz)
- <i>Dataset 3</i>	147	8	↓	↓	↓	↓	6 MCQs	↓	↓	↓
<b>Ahn et al. [81]</b>	95	4	**	SAT test	English	[1, 4]	5 MCQs	after reading	classification (high/low)	Eyelink 1000 (SR Research) (1000Hz)
<b>Copeland et al. [77]</b> - <i>Format A</i>	15	9	400	Wattle	English	[1, 10]	1 MCQ 1 Cloze	after reading (with text for reference)	regression	Seeing Machines FaceLAB (60Hz)
- <i>Format B</i>	8	↓	↓	↓	↓	↓	↓	during reading	↓	↓
- <i>Format C</i>	9	↓	↓	↓	↓	↓	↓	during reading (without text for reference)	↓	↓
- <i>Format D</i>	7	↓	↓	↓	↓	↓	↓	after reading (question showed prior to reading)	↓	↓
<b>Yoshimura et al. [79]</b>	11	10	**	TOEIC test	English	no	4 MCQs	after reading	classification (low/middle/high)	SMI RED250 (250Hz)
<b>Makowski et al. [82]</b>	62	12	158	Textbook	German	no	3 MCQs	after reading	classification (high/low)	Eyelink 1000 (SR Research) (1000Hz)
<b>My Dataset</b> - <i>RCIRv1</i> (Chapter 5)	10	96	353	RACE dataset	English	[1, 5]	3 MCQs	after reading	regression	Gazeport GP3 HD (150Hz)
- <i>RCIRv2</i> (Chapter 6)	13	144	346	↓	↓	↓	↓	↓	↓	↓

\* The smaller value the lower comprehension information is not provided by the authors  
 \*\* Abbreviation for Same as above  
 ↓ Abbreviation for Same as above

## Relevance to my present study

The compilation of previous studies demonstrates the existing progress in integrating only eye movement measures to predict reading comprehension. While some research has yielded positive results, several challenges have been identified. In particular, Copeland et al. [77] highlighted the difficulty of predicting comprehension level when people are reading purpose-free (reading without knowing what questions they will be asked). Nonetheless, this is the most common scenario in real-world reading activities, which opens up the opportunity for further research to address this challenge. During this type of reading, one might skim through the text to grasp the main idea, while in other cases, one might read the text intensively to understand the details. Depending on the way people read, the eye movement behaviour will vary and thus, the prediction of comprehension level will be affected [75]. When investigating the relationship between eye movements and comprehension, Southwell et al. [80] pointed out some characteristics that influenced comprehension level, which I found to have similarities with the characteristics of reading types as described in [75]. For instance, Southwell et al. found that making more, but short fixations is associated with a higher comprehension level, which is also an indicator of attentive reading, while fewer fixations are associated with skimming/mind-wandering and cause comprehension levels to drop [80]. Therefore, I hypothesise that eye movements, reading conditions and reading comprehension form a complex relationship, where eye movements are influenced by reading conditions and reading comprehension is influenced by both eye movements and reading conditions.

In this thesis, I aim to investigate the relationship between eye movements, reading conditions, and reading comprehension. I construct a dataset that records participants' eye movements and their comprehension levels as they engage in various reading tasks where each task is designed to induce a specific type of reading conditions. Upon thoroughly analysing the complex relationship between

eye movements, reading condition and reading comprehension, I further propose a novel approach to predict reading comprehension level by integrating information from eye movements within reading conditions. While previous work has attempted to incorporate the estimation of reading strategies when predicting comprehension, such as reading-ratio and skimming-ratio [77], the effectiveness of such integration on reading comprehension has not been extensively studied. Moreover, in addition to reading and skimming, I also investigate the effect of scanning and proofreading conditions, as these are identified as one of the most common reading types [74, 75].

I also further evaluate the temporal robustness of the proposed approach for comprehension prediction to explore the possibility of applying the approach in real-world reading activities, such as infologging. To facilitate this, another dataset is constructed to capture reading activities over a period of time. The proposed approach is then evaluated on this longitudinal dataset to investigate its temporal robustness.

## 2.3 Methods for Statistical Analysis

This section presents an overview of the key statistical methods employed throughout this dissertation, serving as a foundation for the subsequent chapters. The techniques discussed have been carefully selected to address the research questions posed and to effectively analyse the data collected during this study (in Chapter 5 and Chapter 6). These methods not only allow for the rigorous testing of hypotheses but also facilitate the exploration of underlying patterns and relationships within the data. In this research, these statistical methods are computed using the functions provided by the Python libraries called Scipy [137], which is an open-source package for mathematics and science computing.

### 2.3.1 Comparison of Group Means and Medians

To gain insights into the variations among different groups in a dataset, it is crucial to examine their central tendencies. By analysing group means and medians, it is possible to determine whether observed differences are statistically significant or simply due to random variation [138]. This comparison is essential for identifying any potential impact of independent variables on dependent variables, thereby revealing underlying patterns and relationships. This research employs both parametric and non-parametric methods to compare groups, depending on the nature of the data and the assumptions that can be made about its distribution.

Analysis of variance (ANOVA) [139], is a fundamental parametric statistical method used to compare the means of two or more groups. This method determines whether there is at least one group has a significantly different mean from the others. However, the validity of ANOVA depends on the assumption of normality and homogeneity of variances within the groups, making it essential to verify these assumptions before conducting the analysis [139]. The normality assumption means that the data should be normally distributed within each group, while the homogeneity of variances assumption requires that the variances of the groups are equal. To test these assumptions, I employ the Shapiro-Wilk test [140] and Bartlett's test [141], respectively, which are covered in Section 2.3.2.

In cases where the assumptions of ANOVA are violated, the Kruskal-Wallis test [142] can be used as a non-parametric alternative, which compares the medians across multiple groups. This method is robust against the non-normal distribution of data and unequal variances of data, as it is based on the ranks of the observations rather than their actual values [142]. Hence, Kruskal-Wallis test is particularly useful when dealing with ordinal data or when sample sizes are small.

A significant result from either ANOVA or Kruskal-Wallis test indicates that there is at least one group that is significantly different from the others. However, this does not provide information on which specific groups are different from each

other [139,142]. To identify which pairs of groups are significantly different, another test called post-hoc test is required [138]. The details of the post-hoc tests used in this research are discussed in Section 2.3.3.

### 2.3.2 Validating Normality and Homogeneity of Variances

As outlined in the previous section (Section 2.3.1), the ANOVA test assumes the normality and homogeneity of variances within the groups. Verifying these assumptions prior to conducting the analysis is crucial to ensure the validity of the test results. The Shapiro-Wilk test [140] assesses whether the data follows a normal distribution. Moreover, it is preferred for its effectiveness for small to moderate sample sizes. A non-significant result indicates that the data do not deviate significantly from a normal distribution [140]. On the other hand, to test the homogeneity of variances assumption, Bartlett's test [141] is one of the commonly used method [138]. Bartlett's test is sensitive to departures from normality [141]; thus, in this reseach, it is often conducted after a significant Shapiro-Wilk test result. If the data is not normally distributed, the Bartlett's test will not be conducted and the Kruskal-Wallis test will be used instead of ANOVA. In contrast, if the data follows a normal distribution and the Bartlett's test gives a non-significant result (meaning the variance of each group is equal), the ANOVA test will be used to compare the means of the groups.

### 2.3.3 Post-hoc Tests

Once a significant result is obtained from the ANOVA or Kruskal-Wallis test, it is crucial to determine the sepcific groups that differ [138]. Post-hoc tests facilitate this by conducting pairwise comparisons between groups to identify which pairs are significantly different from each other.

With respect to ANOVA parametric test, the t-test [143] is commonly used as a post-hoc test to examine differences between group means [138]. Conducting multiple t-tests, however, increases the risk of Type I errors. A Type I error occurs



when a true null hypothesis is incorrectly rejected, essentially finding a difference when there is none. To reduce this risk, significance level adjustments such as the Bonferroni correction [144] are applied to account for the number of comparisons made. Specifically, the original significance level (e.g., 0.05) is divided by the number of comparisons, thereby reducing the likelihood of Type I errors.

In non-parametric contexts, Conover's test [145] serves as a post-hoc analysis following a significant Kruskal-Wallis test result. This test performs pairwise comparisons using rank sums and adjusts for multiple testing, allowing the identification of significant differences between groups without relying on parametric assumptions [145].

### 2.3.4 Correlation Analysis

To explore potential associations and dependencies between variables, correlation analysis is a fundamental statistical method [138]. Correlation analysis provides insights into the strength and direction of the relationships between two continuous variables. There are two common correlation coefficients used in research: Pearson's correlation coefficient [146] and Spearman's rank correlation coefficient [147]. The correlation coefficient given by either method ranges from  $-1$  to  $1$ , where values closer to  $-1$  or  $1$  indicate strong positive or negative correlations, respectively.

Pearson's correlation [146] is used to measure the strength and direction of linear relationships between normally distributed variables. This parametric method provides insights into the degree of association between continuous variables that exhibit a linear trend. Pearson's correlation assumes that the data is normally distributed and that the relationship between the variables is linear [146]. Therefore, it is essential to verify these assumptions before conducting the analysis. For normality assumption, the Shapiro-Wilk test [140] (discussed in Section 2.3.2) can be used, while the linearity assumption can be verified by examining scatter plots of the data. In cases where the assumptions of Pearson's correlation are not met, or when measuring relationship between ordinal variables, Spearman's rank

correlation [147] is utilised. This non-parametric alternative examines the monotonic relationships between variables by evaluating the rank-order of the data. It offers a more flexible approach to correlation analysis that is less sensitive to outliers and non-linear associations.

## 2.4 Methods for Machine Learning Analysis

This research leverages a comprehensive set of machine learning techniques to extract insights and build predictive models from complex eye-tracking datasets. The following sections provide a comprehensive overview of these methods, detailing their theoretical foundations, algorithmic structures, and key characteristics.

### 2.4.1 Regression Algorithms

#### 2.4.1.1 Linear Regression

Linear regression is a statistical method which is widely used to model the relationship between a dependent variable  $Y$  (also known as the response or outcome) and one or more independent variables  $X$  (also known as predictors or features). The model tries to find the best-fitting line that describes the relationship between the dependent and independent variables, which can be expressed as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon$$

Where

- $\beta_0$  is the intercept
- $\beta_1, \beta_2, \dots, \beta_n$  are the coefficients of the independent variables,  $X_1, X_2, \dots, X_n$ ,
- $\epsilon$  is the error term.

Linear regression aims to find the coefficients that minimise the sum of squared differences between the predicted and actual values of the dependent variable. It is also particularly useful for understanding the influence of predictor variables on the dependent variable.

#### 2.4.1.2 Logistic Regression

Logistic regression extends the linear regression model to the concept of binary classification tasks. It models the probability of a given input belongs to one of two classes (e.g., 0 or 1) by applying a logistic function to the linear combination of the input features, as follows:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

Where:

- $P(Y = 1|X)$  is the probability of the input  $X$  belonging to class 1,
- $\beta_0$  is the intercept,
- $\beta_1, \beta_2, \dots, \beta_n$  are the coefficients of the independent variables,  $X_1, X_2, \dots, X_n$ .

The logistic function is also known as the sigmoid function, which maps the predicted values of the linear combination to a probability value between 0 and 1 to determine the class label. Moreover, it can be extended to multi-class classification tasks using methods such as one-vs-one (OvO) or one-vs-rest (OvR). The one-vs-one method constructs a binary classifier (i.e. the logistic regressor) for each pair of classes, while the one-vs-rest method constructs a binary classifier for each class against all other classes.

#### 2.4.2 Ensemble Learning

Ensemble learning is known as a machine learning technique that combines multiple learning models to improve the ultimate predictive performance.

#### **2.4.2.1 Random Forest**

Random Forest is an ensemble learning method that is built upon the concept of decision trees. It creates a collection of different decision trees during the training process and aggregates their predictions to make the final decision. To introduce randomness, each tree is trained on a random subset of the training data (called bagging) and a random subset of the features. This helps to reduce overfitting and enhance the generalisation of the model. Depending on the task, the trees' predictions are aggregated differently, such as majority voting for classification tasks and averaging for regression tasks. Random Forest is widely used in practice due to its robustness, ability to handle high-dimensional data, and non-linear relationships.

#### **2.4.2.2 Extra Trees (Extremely Randomised Trees)**

Extra Trees is an ensemble method which is similar to Random Forest, but with a key difference in the way the decision trees are formed. In Random Forest, the trees are built using the best split among a subset of features, while in Extra Trees, the splits are chosen randomly. This helps to diversify the trees and reduce the variance of the model, which can lead to improved generalisation performance. Hence, Extra Trees is particularly useful when dealing with high-dimensional and high variance data.

#### **2.4.2.3 Gradient Boosting Machine**

Gradient Boosting Machine is another ensemble model that is built upon a collection of weak learners (e.g., decision trees) to boost the overall predictive performance. It works by training a sequence of weak learners in multiple rounds, where each learner is trained to correct the errors made by the previous learners. Specifically, the new learner is trained to predict the errors of the previous learner, and the predictions are aggregated to make the final decision. Mathematically, the boosting process of

the model can be expressed as follows:

$$F_m(x) = F_{m-1}(x) + \gamma h_m(x)$$

Where:

- $F_m(x)$  is the prediction of the model at stage  $m$ ,
- $F_{m-1}(x)$  is the prediction of the model at stage  $m - 1$ ,
- $\gamma$  is the learning rate that controls how much the new learner contributes to the final prediction,
- $h_m(x)$  is the weak learner at round  $m$ .

#### 2.4.2.4 AdaBoost

AdaBoost is another variant of boosting technique that is designed to improve the performance of weak learners by focusing on the misclassified instances. All instances are assigned equal weights initially, then changed in each iteration based on the performance of the weak learner. In particular, the weights of the misclassified instances are increased, so that the new learner pays more attention to these instances in the next iteration. After a certain number of boosting rounds, the final prediction is made by aggregating the predictions of all weak learners. The algorithm behind AdaBoost can be summarised as follows:

$$F(x) = \sum_{m=1}^M \gamma_m h_m(x)$$

Where:

- $F(x)$  is the final prediction of the model,
- $\gamma_m$  is the weight of the weak learner at round  $m$ ,
- $h_m(x)$  is the weak learner at round  $m$ .

For each weak learner  $h_m(x)$ , its weight  $\gamma_m$  is calculated based on the error rate of the learner, which is used to update the weights of the instances.

#### 2.4.2.5 Light Gradient Boosting Machine (LightGBM)

LightGBM is an efficient and scalable gradient boosting framework with advanced implementation that is designed to handle large-scale and high-dimensional data. Comparing to traditional gradient boosting methods, LightGBM is equipped with novel techniques such as:

- Gradient-based One-Side Sampling (GOSS) which selects the instances with large gradients while randomly samples the instances with small gradients to compute gradients, enabling faster training speed while maintaining the accuracy.
- Exclusive Feature Bundling (EFB) which bundles the mutually exclusive features into a single feature, reducing the number of features and thereby improving the efficiency of the model.
- Histogram-based algorithm which puts continuous features into discrete bins and hence speeding up the training process.

Due to these advantages, LightGBM is widely used in practice for various machine learning tasks.

### 2.4.3 Instance-based Learning

#### 2.4.3.1 K-Nearest Neighbours (KNN)

The k-Nearest Neighbors algorithm is a non-parametric method used in machine learning for both classification and regression tasks. Its core principle involves making predictions for a new data point based on the characteristics of its  $k$  nearest neighbors in the feature space. For classification, KNN assigns the most common class among the  $k$  nearest neighbors, while for regression, it averages their values. There are two critical factors influence kNN's performance:

- The choice of  $k$ : A smaller  $k$  allows for more flexible decision boundaries but may be noise-sensitive, while a larger  $k$  provides smoother boundaries but might oversimplify the model.
- The distance metric: Commonly Euclidean distance, but other metrics like Manhattan or Minkowski may be more suitable depending on the data.

KNN is simple to implement and interpret, but it can be computationally expensive, especially with large datasets. However, KNN is a powerful tool for problems in which the decision boundary is complex and not easily represented by a parametric model.

#### 2.4.3.2 Support Vector Machine (SVM)

Support Vector Machine is one of the supervised learning models that are commonly used for both classification and regression tasks. SVM's main objective is to find the optimal hyperplane that best separates the data points into different classes within the maximum margin. Mathematically, SVM tries to find a hyperplane  $w^T x + b = 0$  that maximizes the margin, which is given by the formula:

$$\min_{w,b} \frac{1}{2} \|w\|^2$$

$$\text{subject to } y_i(w^T x_i + b) \geq 1 \text{ for } i = 1, 2, \dots, n$$

Where:

- $w$  is the weight vector,
- $b$  is the bias term,
- $x_i$  is the data point,
- $y_i$  is the class label,
- $n$  is the number of data points.

Moreover, SVM is also able to handle non-linearly separable data by employing different kernel functions such as linear, polynomial, and radial basis function (RBF) kernels, which map the data into a higher-dimensional space where the data points are linearly separable. To prevent overfitting, a regularisation factor is also introduced to the objective function, which controls the trade-off between the margin and the classification error. The regularised objective function is given by:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

$$\text{subject to } y_i(w^T x_i + b) \geq 1 - \xi_i \text{ for } i = 1, 2, \dots, n$$

$$\xi_i \geq 0 \text{ for } i = 1, 2, \dots, n$$

Where:

- $\xi_i$  is the slack variable that allows for some misclassification, which is given by:
  - $\xi_i = 0$  if the data point is correctly classified and lies outside/on the margin,
  - $0 < \xi_i \leq 1$  if the data point is correctly classified but lies within the margin,
  - $\xi_i > 1$  if the data point is misclassified.
- $C$  is the regularisation parameter that controls the trade-off between the margin and the classification error.

#### 2.4.4 Probabilistic Learning

In this thesis, I also employ the Bayesian Regression method which is a probabilistic learning model that is based on the Bayes' theorem. It can be viewed as an extension of the standard linear regression model, which has regularisation terms and Bayesian inference. Instead of estimating the coefficients using least squares as in linear regression, Bayesian regression treats the coefficients as random



variables with prior distributions and then update the distributions based on the observed data. The posterior distribution of the coefficients is then used to make predictions and quantify the uncertainty of the model. The Bayesian regression model can be summarised as follows:

$$y = X\beta + \epsilon$$

$$\beta \sim N(0, \sigma_p^2 I)$$

$$y \sim N(X\beta, \sigma^2 I)$$

Where:

- $y$  and  $X$  are the label and the feature matrix, respectively,
- $\beta$  is the coefficient vector,
- $\epsilon$  is the error term,
- $\sigma_p^2$  is the variance of the prior distribution,
- $\sigma^2$  is the variance of the error term.

## 2.5 Chapter Summary

This chapter lays the foundation for my research by introducing key concepts, challenges, and state-of-the-art systems in lifelogging and eye movement analysis. It begins with an overview of lifelogging and the need for efficient retrieval systems as discussed in Section 2.1. A comprehensive review of the best-performing lifelog retrieval systems in recent Lifelog Search Challenge (a popular benchmarking challenge for lifelog retrieval systems) is presented, which are categorised into concept-based and semantic-based approaches (Section 2.1.2).

The second part of the chapter explores eye movement research, covering human eye anatomy (Section 2.2.1), eye-tracking technologies (Section 2.2.2), and

the application of eye movement analysis in recognising reading strategies (Section 2.2.3 and 2.2.4). A literature review on using eye movements to predict reading comprehension is presented. Through this, I identify the research gaps and hypothesise a complex relationship between eye movements, reading conditions, and reading comprehension and propose a novel approach to predict reading comprehension by integrating information from eye movements within various reading conditions.

Finally, the remaining sections provide a brief overview of the statistical (Section 2.3) and machine learning methods (Section 2.4) employed in this research, including ANOVA, Kruskal-Wallis test, post-hoc tests, correlation analysis, and various machine learning algorithms such as regression, ensemble learning, instance-based learning, and probabilistic learning. This serves as a foundation for the subsequent chapters, where I will apply these methods to analyse the eye movement data and predict reading comprehension.

Building upon the background established in this chapter, the forthcoming chapter will present the research methodology adopted in this thesis, detailing the research process and methodologies used to address the research questions outlined in Section 1.4.

## Chapter 3

# Research Methodology and Evaluation Methods

This chapter discusses the research methodologies used in this thesis to address the proposed research questions. It is divided into three sections: Section 3.1 outlines the categories of research methodologies, the rationale for their selection for each research question, and the processes used to address these questions. Section 3.2 examines the operational constraints of the research. Finally, Section 3.3 briefly overviews the evaluation metrics for assessing the research outcomes.

### 3.1 Research Methodology

In the *Advanced Learner's Dictionary of Current English*, the term "research" is defined as "a careful investigation or inquiry specifically through search for new facts in any branch of knowledge." <sup>1</sup>. It is a process of collecting, analysing and interpreting information to answer questions about a topic or phenomenon [148] According to Clifford Woody (American philosopher, 1939), "Research comprises of defining and redefining problems, formulating the hypothesis for suggested solutions, collecting, organizing and evaluating data, making deductions and reaching conclusion and further testing the conclusion whether they fit into formulating the hypothesis." [8] For a process to be called research, it must have 6 main characteristics: be controlled, rigorous, systematic, valid and verifiable,

---

<sup>1</sup>*The Advanced Learner's Dictionary of Current English*, Oxford, 1952, p. 1069

empirical and critical. [148]: Depending on the research perspective, it can be classified into three main categories:

- **Application of Findings:**
  - **Pure Research:** This type involves developing and examining theories and hypotheses that are intellectually stimulating for the researcher but might not have immediate or future practical use.
  - **Applied Research:** Prevalent in social sciences, this type applies research methods to gather information about different facets of specific issues, situations, or phenomena.
  
- **Objectives of the Study:**
  - **Descriptive Research:** This aims to systematically depict a situation, issue, phenomenon, service, or program.
  - **Correlational Research:** Its primary goal is to identify or confirm the presence of a relationship between multiple elements of a situation.
  - **Explanatory Research:** Focuses on understanding the reasons and mechanisms behind the relationship between different aspects of a situation or phenomenon.
  - **Exploratory Research:** Conducted to explore areas with limited existing knowledge or to assess the feasibility of a specific research project.
  
- **Mode of Enquiry Used in Conducting the Study:**
  - **Quantitative Research:** This type of research is based on the collection and analysis of numerical data to explain, predict, or control phenomena of interest.
  - **Qualitative Research:** Focuses on gathering and interpreting non-numerical data, such as text, video, or audio, to explore ideas, viewpoints, or experiences.

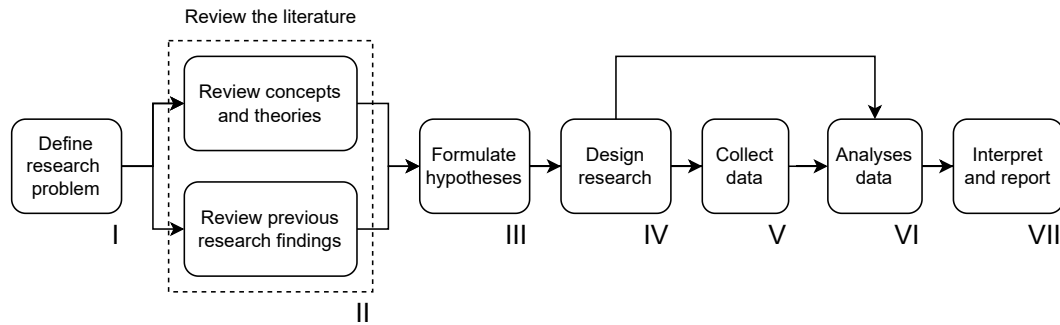


Figure 3.1: Research process in flow chart, adapted from [8]

It is worth noting that the above categories are not mutually exclusive. A research project can be classified into more than one category. For example, a research project can be both applied and explanatory [148]. Based on the above classification, I was able to identify the type of research to be conducted to address the proposed research questions in Section 1.4. Specifically, all the research questions are exploratory research, which aims to explore to feasibility of bridging the gap in the research literature. Moreover, quantitative research is used while addressing RQ2 and RQ3 since the research questions are based on the collection and analysis of eye movement measures to estimate reading comprehension. In RQ2, correlational research is also employed since it aims to investigate the relationship between eye movement measures, reading conditions and reading comprehension in detail through machine learning approaches and statistical testing procedures.

To address the research questions defined in Section 1.4, I followed the research process depicted in Figure 3.1, which is an adapted version of the research process in the flow chart from [8]. All research questions in my dissertation underwent the same process (from I to VII), except for RQ1, where data collection (step V) was unnecessary due to the availability of existing datasets for analysis. The initial stages, including research problem identification, literature review, and hypothesis formulation (steps I to III), have been previously discussed in Chapters 1 and 2. In this section, I will elaborate on the subsequent stages of the research process (steps

IV to VII) for each research question.

- **RQ1:** Since the data collection step was omitted for RQ1, there are only three main stages: IV, VI, and VII. However, it takes multiple iterations through two stages IV and VI before conclusively addressing the research question.

The research design (step IV) focused on developing LifeSeeker, the interactive lifelog retrieval system for the Lifelog Search Challenge (LSC). This involves selecting suitable technologies, search functionalities, and designing the user interface. LifeSeeker is then benchmarked in the annual LSC competitions, offering insights into system performance and user experience. Step VI entailed evaluating LifeSeeker's components, analysing strengths and weaknesses to guide improvements. This feedback loop returned the process to step IV with insight into necessary changes, leading to the development of a new LifeSeeker version for subsequent LSC competitions. Ultimately, the core components crucial for a state-of-the-art lifelog retrieval system were identified (aiding in constructing the system for RQ4), which addresses RQ1. Step VII involves documenting these findings, which can be found in Chapter 4.

- **RQ2:** This research question begins with step IV which focuses on detailing the data collection process and analysis procedure. Since this research question focuses on understanding the relationship between eye movement measures, reading conditions, and reading comprehension, and proposing a model for reading comprehension estimation, it involves two main tasks: classification of reading conditions and regression of reading comprehension, based on eye movement features. To facilitate the analyses, a reading dataset – RCIRv1 – is collected in step V, which is designed to capture participants' eye movements when they are performing reading tasks, while different reading conditions are being induced. RCIRv1 is then analysed in step VI, which involves the extraction of eye movement features and subsequent

analysis of the relationship between eye movement measures, reading conditions and reading comprehension. To tackle this, two separate processes are employed, with one focusing on statistical testing procedures and the other on machine learning approaches, for both classification and regression tasks. The results from both processes are then compared and contrasted to gain more insights into eye movement features. Upon confirming the relationship between eye movement measures, reading conditions and reading comprehension, the approach for constructing a reading comprehension estimation model is proposed and reported in Chapter 5 (step VII), which also concludes RQ2.

- **RQ3:** The process of addressing this research question is similar to RQ2, except that the data collection and analysis are conducted in a different setting, in which the data collection happens over a period of multiple days. After revising the data collection process and the data analysis procedure to introduce the longitudinal aspect (step IV), the data collection happens in step V, which generates the RCIRv2 dataset. Step VI mirrors the analytical approach of RQ2 to analyse RCIRv2, but with more emphasis on machine learning analyses since this research question focuses on evaluating the performance of the reading comprehension estimation model proposed in RQ2 on the longitudinal aspect. After verifying the model's stability and robustness, a further step is introduced which performs hyperparameter tuning to improve the model's performance and to prepare for deployment in real-world applications, which is the case in RQ4. Findings and discussions addressing RQ3 are detailed in Chapter 6, which is also the last step in RQ3's research process.
- **RQ4:** This question combines elements from RQ1, RQ2, and RQ3, aiming to develop a proof-of-concept (PoC) retrieval system for perceived on-screen information. This involved adapting the state-of-the-art lifelog retrieval

system from RQ1 and integrating the reading comprehension estimation model from RQ2 and RQ3. The design phase (step IV) included planning the data collection to simulate lifelog data creation and track on-screen content and eye movements. It also detailed the plan for the evaluation of the PoC system's performance with and without the reading comprehension model. After having the necessary tools for capturing the infologging dataset, step V begins by capturing users performing daily tasks on computer usage and their corresponding eye movements, over a period of 1 month. Participants also annotate a number of on-screen contents by giving a rating of their understanding of the perceived information and answering some MCQs that are generated based on that information to measure their reading comprehension. These annotations further evaluated the reading comprehension model before its integration into the PoC system. Upon the completion of the infolog dataset, step VI begins with a user study to evaluate the PoC system, with and without comprehension evidence. Participants are divided into two groups accordingly, and both groups are asked to perform search tasks on the PoC system, using the same set of queries. This user study mimics the LSC format, using similar evaluation metrics Finally, I discuss the results and provide insights into the PoC system in Chapter 7, which concludes the research process for RQ4.

## **3.2 Operating Constraints**

For any new research topic, I should define the operating constraints of the research to design and conduct the experiment properly. In this Ph.D. research, I identify these constraints as follows:

- The Limited number of participants for constructing the RCIRv1 and RCIRv2 datasets is one of the main constraints in this study. Recruiting participants for eye movement studies is challenging, especially when the study requires



high commitment from the participants to complete the reading experiment.

- Due to the availability of the eye tracker (one device), the data collection can not be conducted in parallel. Therefore, the longitudinal RCIRv2 dataset is limited to only 6 sessions (spanning 6 days) for each participant. As a result, the full potential of the proposed reading comprehension estimation model's performance may not have been explored.
- My research is also constrained by ethical considerations and regulatory compliance. The data collection experiments in this research were strictly designed to respect participant privacy and adhere to prevailing data governance laws.
- Since infologging is sensitive to privacy issues, the dataset for RQ4 only involves data from the researcher and is not publicly available. The researcher adheres to pre-defined steps to ensure the data is collected with no bias which influences the research outcomes.

These constraints are maintained for this Ph.D. research and act as limiting factors to focus the research effort.

### 3.3 Evaluation Metrics

In this section, I describe the evaluation metrics used in this dissertation. There are four main metrics that are employed to evaluate proposed methods for addressing the research questions, which are: Accuracy, Spearman's  $\rho$  to retrieving correlation coefficient (or correlation score in short), LSC score, Precision and Recall. Specifically, RQ1 employed the LSC score to evaluate the performance of the lifelog retrieval system, since this score is the main evaluation metric used for the LSC competition, and hence, using this metric allows for a direct comparison with other state-of-the-art lifelog retrieval systems. RQ2 and RQ3 used accuracy and correlation scores to evaluate the performance of the reading condition

classification and reading comprehension estimation model, accordingly. Accuracy score is commonly used in classification tasks, especially when the dataset is balanced, which is the case in this research. Meanwhile, correlation scores are used to evaluate the predictive power of the reading comprehension estimation model, which is also employed in previous study [80]. Finally, RQ4 used both LSC score, precision and recall to evaluate the performance of the interactive infologging retrieval system. The used of LSC score is to provide comparison with the baseline retrieval system established in RQ1, while precision and recall are used to provide more insights into the retrieval system's performance in terms of the number of relevant documents retrieved and the number of relevant documents missed, respectively. A brief description of each metric is provided in the following sections.

### 3.3.1 Accuracy

Accuracy score is a basic yet crucial metric for evaluating classification models, and it is the common choice when the dataset is balanced. It represents a measure of how well a model, system, or test correctly identifies or predicts outcomes. The accuracy score is defined as the ratio of correctly predicted instances to the total instances in the dataset. It's often expressed as a percentage. The formula for accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where  $TP$  is the number of true positives,  $TN$  is the number of true negatives,  $FP$  is the number of false positives, and  $FN$  is the number of false negatives. The accuracy score ranges from 0 to 1, with 1 being the best possible score.

### 3.3.2 Spearman's rank correlation coefficient

Spearman's rank correlation coefficient is a statistical measure used to evaluate the strength and direction of the association between two ranked variables. It's particularly useful in situations where the data does not meet the assumptions

necessary for Pearson's correlation coefficient, such as when the relationship is not linear or the data is ordinal. Spearman's rank correlation coefficient, often denoted as  $\rho$  (rho) is a non-parametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function.

To obtain  $\rho$ , each variable must be ranked. If there are ties, assign to each tied value the average of the ranks they would have received if there had been no ties. Then apply the following formula:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

where  $d_i$  is the difference between the two ranks of each observation and  $n$  is the number of observations.

The coefficient ranges from -1 to 1. A positive  $\rho$  indicates that the ranks of the two variables are positively related (i.e., when one variable increases, the other tends to increase), while a negative  $\rho$  indicates that the ranks of the two variables are negatively related (i.e., when one variable increases, the other tends to decrease). The magnitude of  $\rho$  indicates the strength of the association. A value of 0 indicates that there is no association between the two variables.

### 3.3.3 LSC score

Lifelog Search Challenge (LSC) is a benchmarking competition for lifelogging research, which has been held annually since 2018. The benchmarking process considers both the retrieval system and the person who operates it. For a given query, the operator will perform the search on their system and submit the images that they believe are relevant to the query. The ultimate goal is to retrieve the relevant lifelog image that matches a given query as fast as possible while minimising the penalty for wrong submissions. As a result, for each task, the score [35] of one LSC participant retrieving the correct answer at a time  $t$  is calculated as:

$$S_i = \max \left( 0, M + \frac{D - t}{D} (100 - M) - W * 10 \right) \quad (3.1)$$

where  $M$  refers to the minimum score earned,  $D$  denotes the query's duration and  $W$  represents the number of wrong submissions for each query. Specific to this case,  $M$  and  $D$  are set to 50 and 300, respectively. As can be seen from the formula above, the score is linearly decreased until the minimum score (50) within the 300-second period. Then the final score is taken by subtracting each negative submission by 10 points. A participant gets a zero score when the time for the query is over (300 seconds have passed) and a positive answer is not found.

The LSC score is calculated by averaging the scores of all queries, which is given by the following formula:

$$LSC \text{ score} = \frac{1}{N} \sum_{i=1}^N S_i$$

### 3.3.4 Precision and Recall

Precision and Recall are widely used metrics to evaluate the performance of retrieval systems, commonly seen in information retrieval, computer vision, and machine learning contexts, particularly for tasks like image search, document retrieval, and object detection. Precision is defined as the number of relevant documents retrieved divided by the total number of documents retrieved. Moreover, precision at  $k$  measures whether the relevant items are present in the top  $k$  positions of the result list. It is particularly used in information retrieval tasks where the order of results is significant and is defined as:

$$P(k) = \frac{TP_k}{TP_k + FP_k}$$

where  $TP_k$  is the number of true positives and  $FP_k$  is the number of false positives at the cut-off  $k$ .

Recall measures the proportion of actual positives that were identified correctly.

It is the number of true positives divided by the number of true positives plus the number of false negatives (FN), which is given by the following formula:

$$Recall = \frac{TP}{TP + FN}$$

where  $TP$  is the number of true positives and  $FN$  is the number of false negatives.

Both precision and recall are bounded between 0 and 1, with 1 being the best possible score. There is often a trade-off between precision and recall. Increasing one typically reduces the other. This is because broadening the criteria to retrieve more items (increasing recall) often includes more irrelevant items (decreasing precision), and vice versa.

### 3.4 Chapter Summary

This chapter presented the research methodologies employed in this thesis to address the proposed research questions. The chapter outlined the categories of research methodologies, the rationale for their selection, and the processes used to address each research question. Additionally, the chapter discussed the operational constraints of the research, such as the limited number of participants and ethical considerations. Finally, an overview of the evaluation metrics, including accuracy, Spearman’s rank correlation coefficient, LSC score, precision, and recall, and the rationale for their selection was provided.

With the research methodology established, the coming chapters will focus on addressing the research questions using the proposed methodologies. In particular:

- Chapter 4 will focus on RQ1, detailing the key components and construction of a state-of-the-art lifelog interactive retrieval system, LifeSeeker. The chapter will present the system’s development history, architecture, user interface, and interaction methods, followed by an evaluation of its performance in the Lifelog Search Challenge.

- Chapter 5 will address RQ2, investigating the relationship between eye movement features, reading conditions, and reading comprehension, which is crucial to building a comprehension estimation model for infolog retrieval system. The chapter will describe the construction of the RCIRv1 dataset, the extraction of eye movement features, and the analysis of their relationship with reading conditions and comprehension using machine learning models and statistical testing.
- Chapter 6 will focus on RQ3, evaluating the temporal robustness of the reading comprehension estimation model developed in Chapter 5. The chapter will detail the construction of the longitudinal RCIRv2 dataset, the adaptation of the data analysis procedure, and the evaluation of the model's performance over time.
- Chapter 7 will address RQ4, presenting the development of InfoSeeker, an interactive retrieval system for infologging data that integrates the reading comprehension estimation model. The chapter will describe the system's design, data collection process, and evaluation through non-interactive and interactive experiments.

## Chapter 4

# State-of-the-art Lifelog Retrieval System

### 4.1 Introduction

In this chapter, I address the Research Question 1, which is: **What are the key design principles and components required to construct a state-of-the-art lifelog interactive retrieval system?**

As discussed in Chapter 1, the increasing volume of lifelog data has made it difficult for users to find the desired moment in their lifelog data by manual browsing. To address this issue, lifelog retrieval systems have been developed to automate the process of searching and retrieving lifelog data. The literature review in Chapter 2 highlighted the key features and interaction methods employed by state-of-the-art lifelog retrieval systems. Building upon this foundation, this chapter focuses on the design, implementation, and evaluation of LifeSeeker, an interactive lifelog retrieval system that incorporates these state-of-the-art features and techniques. LifeSeeker was evaluated in LSC (2019, 2020, 2021 and 2022), which helped to measure efficacy in enhancing lifelog search capabilities. Throughout a series of upgrades, refinements and evaluations for each iteration of LSC, LifeSeeker managed to approach the state-of-the-art performance. As a result, I also managed to shortlist the core functionalities and interaction methods that made the success of LifeSeeker.

Furthermore, it is worth noting that LifeSeeker was developed as a collaborative

effort with my colleagues. Consequently, in this chapter, I use 'we' or 'our' to denote research activities conducted jointly with my colleagues, and 'I' or 'my' to refer exclusively to work performed independently by myself.

In the subsequent sections, I will present a concise overview of LifeSeeker's development history. This overview traces the evolution of LifeSeeker from a simple baseline system to its current state-of-the-art status. It will be followed by an in-depth description of the system's architecture, user interface, user interaction, and the underlying search engine that powers the system. Finally, the evaluation results of LifeSeeker will be presented and discussed, followed by a summary of the key findings of this chapter.

## **4.2 An Overview of LifeSeeker**

### **4.2.1 A Brief History of Development**

LifeSeeker has experienced significant development since its inception, first introduced at the Lifelog Search Challenge 2019 (LSC'19), the second iteration of the challenge. Up to the time of this dissertation, LifeSeeker has undergone several major updates, each introducing new features and improvements to meet the evolving requirements of the challenge. A comprehensive list of features and improvements is presented in Table 4.1. This section briefly summarises the pivotal modifications made to the system during each phase of its participation in the challenges.

Similar to most conventional lifelog retrieval systems, the very first version of LifeSeeker [40], is designed as a concept-based search tool that analyses both visual and non-visual content. Its primary aim was to assist users in locating specific life moments captured by a lifelog camera, using keyword queries, while also enhancing the user experience with a transparent and intuitive interface. The main search mechanism was based on concept matching between the user's query and the concepts extracted from the lifelog data. This was enhanced by extracting



Table 4.1: A list of feature changes in LifeSeeker from LSC’19 to LSC’22. Symbol + indicates a new feature added to the system, while symbol – indicates a feature removal from the system. A ✓ means the feature is kept unchanged.

Category	Features	LifeSeeker Version			
		V1 [40]	V2 [41]	V3 [42]	V4 [43]
<b>Search/Filter mechanism</b>	• Concept search				
	— Keyword matching*	+	–		
	— Using Elasticsearch [149]*		+	✓	✓
	— Using Weighted Bag-of-Words [42]			+	–
<b>Concepts enhancement</b>	• Semantic search using CLIP model* [115]				+
	• Free-text query*		+	✓	✓
	• Additional visual concepts				
	— SNIPER [150] pre-trained on COCO [151]	+	✓	✓	✓
— PlacesCNN [118] pre-trained on Place365 [118]	+	✓	✓	✓	
— Bottom-up Attention model pre-trained [152] on Visual Genome dataset [153]*		+	✓	✓	
<b>User Interface</b>	• Text recognition (OCR)				
	— Using CRAFT [154]*		+	–	
	— Using Microsoft Vision API			+	✓
	— Using Google Cloud Vision API				+
	• Locations’ semantic name (inferred through GPS)*			+	✓
<b>Additional Functionalities</b>	• Search results presentation mode				
	— Ranked list of images*	+	✓	✓	✓
	— Group by location*		+	–	
	— Group by part of day*				+
	— Pagination	+	–		
	— Single page with lazy loading function		+	✓	✓
<b>Additional Functionalities</b>	• Image zooming		+	✓	✓
	• Temporal browsing*		+	✓	✓
	• Concept expansion*	+	–		
	• Concept suggestion		+	✓	–
	• Visual similarity search				
	— Bag-of-Visual-Words using SIFT [155] features	+	✓	✓	–
	— Cosine distances of CLIP embeddings [115]*				+
	• Active search (system actively suggest concepts for filtering)*				+
	• Relevance feedback (both positive and negative)*				+
	• Caching for search results*				+
• Continued refinement (filtering applied on top of the preceding search results)*				+	
• Search timeline for reaccessing previous search results*				+	

\*These features were mainly developed by me

additional visual concepts using a pre-trained SNIPER model [150] on the MS-COCO dataset [151] for object detection and a pre-trained PlacesCNN model [118] on the Places365 dataset [118] for scene recognition. The system also expands the input adding related concepts to the original query using thesaurus lookup [40] to help novice users formulate their queries. The search results are presented in a paginated ranked-list of images, with an option for detailed view and visual similarity searches. Visual similarity matching was facilitated by employing a bag-of-visual-words approach using SIFT features [155] to find similar images.

For LSC'20, LifeSeeker 2.0 [41] saw substantial improvements in both system architecture and user interface. The architecture was restructured for greater modularity, facilitating the ease of integrating new components. This version introduced additional visual content analysis techniques, namely object detection with attributes [41] using Bottom-up Attention model [152] pre-trained on Visual Genome dataset [153] and scene text detection using CRAFT [154], which significantly enhance indexing and retrieval capabilities. The system also employed Elasticsearch [149] for indexing and searching, which provides a more scalable and efficient solution than conventional keyword matching on a relational database. The user interface was redesigned to support temporal browsing of search results and included a graph-based visualisation for location-based filtering [41]. The display of search results also shifted from a paginated list to a single page with lazy loading, which allows users to scroll through the results without having to navigate to a new page. From the user's feedback from the first system, the query expansion function was removed as it often worsened the search results [41]. Instead, concept suggestion was introduced, which allows users to actively select concepts existing in the system to perform the search task

The third iteration, LifeSeeker [42], developed for LSC'21, focused on enhancing performance and scalability. Incorporating Elasticsearch [149] as the primary storage engine improved the efficiency of indexing and data retrieval. This version also supported complex query constructions, including boolean operators and wildcards,

by leveraging Elasticsearch’s syntax-based query language. For scene text detection, we replaced CRAFT [154] with the OCR modules from Google Cloud Vision API<sup>2</sup> and Microsoft Vision API<sup>3</sup> as suggested by the organisers [37] and the top-performing team [2], which provided more accurate results.

In the most recent version, LifeSeeker 4.0, the system underwent further enhancements to support advanced search and filtering functions, an easy-to-use interface, and improved scalability. It moves from a concept-based retrieval system to a semantic-based system by employing CLIP [115] to bridge the semantic gap between the user’s query and the visual content of the lifelog data. With this, concept suggestion was no longer needed, and the system could support more complex queries and was not bound to the concepts existing in the system. Visual similarity search was also improved by using CLIP embeddings [115] instead of SIFT features [155] to find similar images. The search results were grouped into part of the day (early morning, morning, afternoon, evening, and night) to facilitate better temporal browsing. Additionally, the system also introduced many new features, including relevance feedback, caching for search results, continued refinement, and search timeline, to enhance the user experience. More details of these features are discussed in Section 4.4. These upgrades resulted in LifeSeeker 4.0 becoming the second-best-performing system in LSC’22. The remainder of this chapter will detail the design of LifeSeeker 4.0, elaborating on its key features and the rationale behind the design choices.

### 4.2.2 System Design

Figure 4.1 illustrates the workflow of LifeSeeker, which consists of two main processes: An offline process that is responsible for indexing and storing the lifelog data, and an online process that handles user queries and returns the search results.

In the offline process, lifelog images were embedded into a vector space by using the Contrastive Language-Image Pre-training [115] (CLIP) model and indexed into a vector database built for scalable similarity search called Milvus [156]. Moreover,

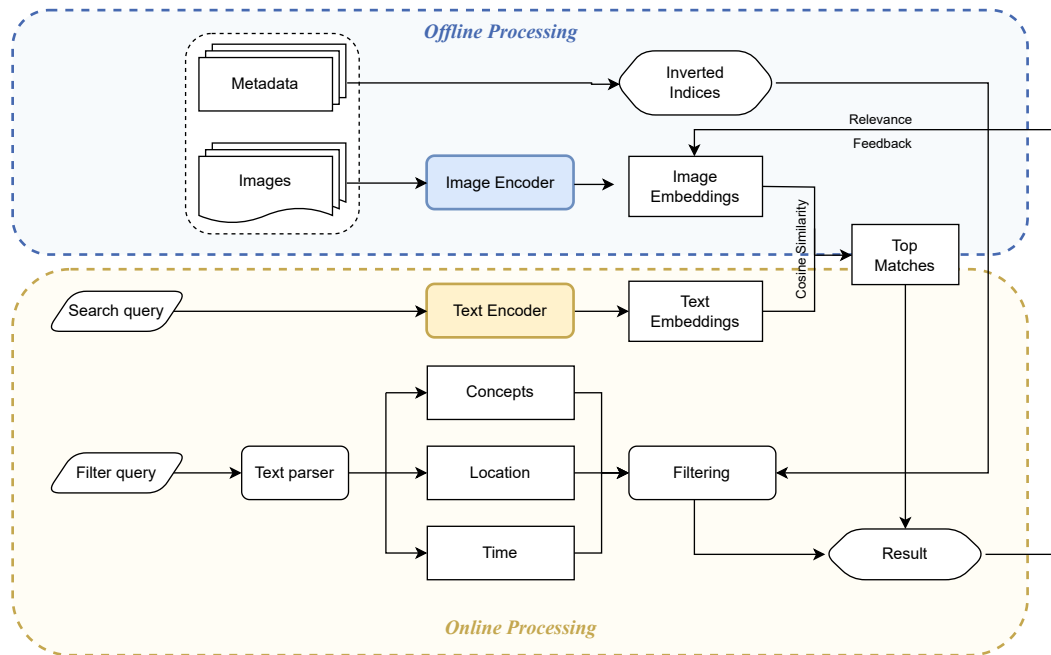


Figure 4.1: An overview of LifeSeeker system workflow

the Elasticsearch engine<sup>1</sup> was used to index and retrieve the metadata provided by the organisers combined with other metadata we extracted from the lifelog dataset. These include place categories and place attributes extracted from PlacesCNN [118], visual object concepts extracted from YOLOv4 [157] pre-trained on COCO dataset [151] and Bottom-up Attention Model [152] pre-trained on Visual Genome dataset [153], and text extraction data (OCR) with other visual concepts extracted using Google Vision API<sup>2</sup> and Microsoft Vision API<sup>3</sup>, respectively. Detailed descriptions of the structure of the indexed metadata file created for the Elasticsearch engine are detailed in Section 4.4.1. The offline process is performed only once, during the initial setup of the system. More details on this indexing process will be discussed in Section 4.4.1.

The online process is where the user interacts with the system. The user can submit queries to the system via the web interface, which can either be a search query or a filter query. The search query is used to retrieve the most relevant

<sup>1</sup><https://www.elastic.co>

<sup>2</sup><https://cloud.google.com/vision/docs/ocr>

<sup>3</sup><https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision>

images to the query, while the filter query is used to filter the search results based on the metadata. Search queries are processed by the Milvus engine, in which the CLIP model is used to embed the query into the vector space and perform a similarity search with embedded images that are already indexed in the vector database. The Elasticsearch engine is used to process filter queries, by leveraging the power of Elasticsearch query language. Search and filter results are cached in a Redis database<sup>4</sup>, enabling the system to keep track of the user's search history and provide a seamless user experience when the user needs to revisit the previous search results. On top of the search and filter functionalities, the system also supports visual similarity search, relevance feedback, and active search. Visual similarity search allows the user to search for images that are visually similar to a given image. Relevance feedback provides the user with the ability to refine the search results by annotating images that are relevant or irrelevant to the query so that the system can re-rank the search results. And finally, active search is a function that allows the system to suggest filtering concepts to the user, rather than the user having to manually come up with the appropriate filtering concepts. Details implementation of these functionalities will be discussed in Section 4.4.2.

The LifeSeeker's retrieval server is developed using the Django framework<sup>5</sup> which plays the role of a middleware layer supporting the communication between the client-side requests (user interface and interaction) and different retrieval modules. The interactive search interface of the LifeSeeker is a web-based application developed using ReactJS framework<sup>6</sup> with the support of Redux<sup>7</sup> for state management and Material-UI<sup>8</sup> for UI components. In Section 4.3, we describe the interface of LifeSeeker in detail and illustrate how a user can interact with the system to perform search tasks.

---

<sup>4</sup><https://redis.io>

<sup>5</sup><https://www.djangoproject.com>

<sup>6</sup><https://reactjs.org>

<sup>7</sup><https://redux.js.org>

<sup>8</sup><https://material-ui.com>

## 4.3 User Interface and User Interaction

### 4.3.1 User Interface

The user interface of LifeSeeker is composed of four main components: (A) the query boxes, including a free-text search box and filter box, (B) the active search's question display, (C) the search progress bar, and (D) a vertically-scrollable panel displaying the retrieved result in groups. Figure 4.2 shows the user interface of LifeSeeker with the aforementioned four main components highlighted.

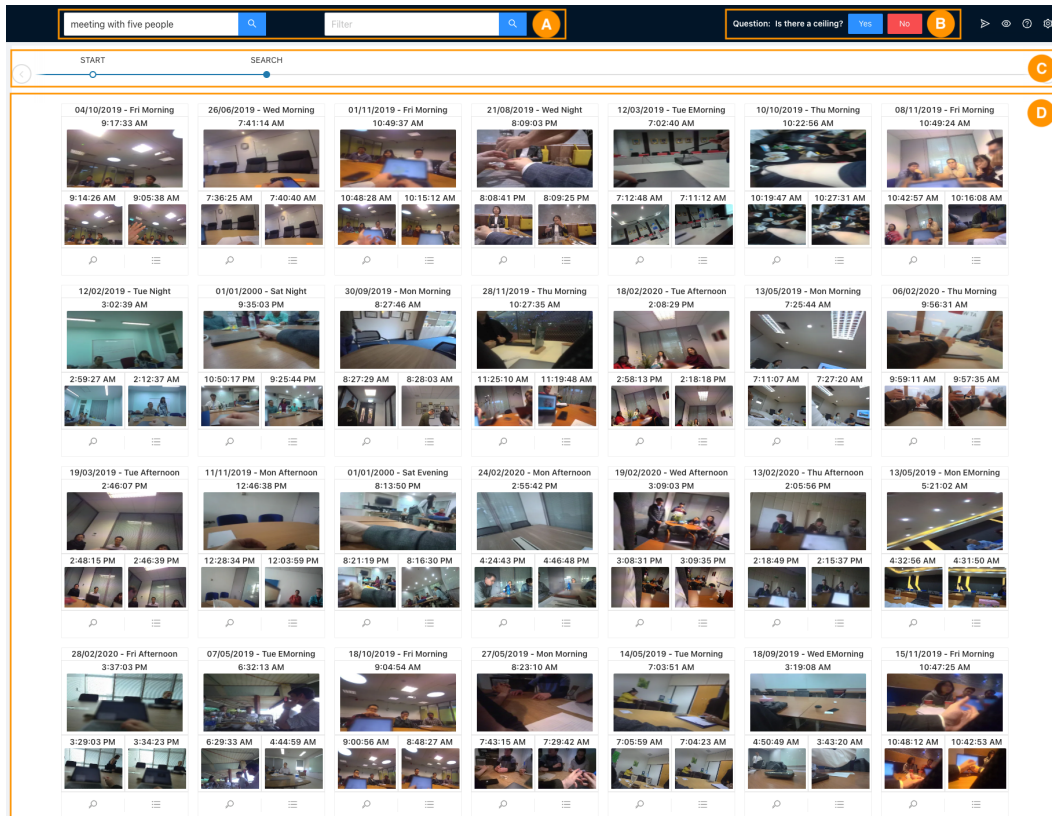


Figure 4.2: The Interactive User Interface of the LifeSeeker Retrieval System.

The query boxes (A) are the main components of the user interface, where the user provides the query to the system. The first box on the left is the free-text search box, where the user can enter a query describing the desired life moment. This query can be of any length and in any natural language format. After the search is performed, the results will be displayed in the vertically scrollable panel (D). The

second box on the right is the filter box (**A**), where the user can specify another query to filter the results displayed in (**D**) to narrow down the search results and eventually obtain the desired lifelog moment. Once the filter is applied, the results will be updated accordingly in (**D**). It is important to note that the filter box is only available when the user performs a free-text search prior to applying the filter.

The active search's question display (**B**) is the component where the system displays questions to the user during the active search process. The questions are updated every time there is a change in the search results displayed in (**D**). One can perceive this as an alternative for filtering, but instead of the user specifying the filter, the system actively asks the user a number of Yes/No questions to narrow down the search results.

The history of the search process (the search timeline) is depicted in the search progress bar (**C**). It is a horizontal bar with many dots, each of which represents a search/filter/active search step. The user can click on any dot to go back to the corresponding step in the search process. There is no extra computation involved when the user goes back to a previous step, as the system caches the results of all the previous steps.

The vertically-scrollable panel (**D**) shows a ranked list of retrieved moments obtained from the query submitted in the query boxes (**A**) and active search function (**B**). Each item in the panel is a square box displaying the lifelog images with a date and time, grouped by part of day. Dates and times are in UTC+0 timezone and the format of `dd/mm/YYYY` and 12-hour `HH:MM:SS` respectively, where `d`, `m`, `Y`, `H`, `M`, `S` denotes the day, month, year, hour, minute and second correspondingly. This information is shown alongside the images since it is considered to be one of the most important pieces of information of a lifelog moment that cannot be recognised visually from the image. The images displayed in the rectangle boxes are the top images that match the query of a particular part of day. The vertically scrollable panel and its items' size are designed for optimal moment-scanning and browsing on a 27-inch monitor (367.69 mm × 612.49 mm). It is worth noting that this optimization

is specifically designed for the Lifelog Search Challenge to reduce the overhead time to find the correct moments to submit by scrolling up and down. According to our experience in previous Lifelog Search Challenges, viewing as many top results as possible without scrolling can result in a big gap in the score and the rank of top-performance systems in the competition. At the bottom of each square box, there are two buttons, one for showing all images in the group (returned by the search), and the other for viewing all lifelog images on the same day.

### 4.3.2 User Interaction

The flow of user interaction can be described via five steps:

1. The user inputs the query into the search box on the left in **(A)**. The query can be in the form of a full sentence describing the moment or in the form of a sequence of terms,
2. The user can either scan or browse the ranked list of relevant images displayed on the vertical-scrollable panel **(D)**,
3. Any moment for which the user wants to investigate if it is the answer to the query, the user has two options to browse it further; there are two buttons to browse for more details. Images can be enlarged for a better view by hovering on the image while pressing the **X** key,
4. If the user is not satisfied with the results, they can choose one of the following options:
  - (a) Apply a filter to the current results by specifying the filter in the filter box on the right in **(A)**
  - (b) Answer the active search's questions displayed in **(B)**
  - (c) Perform visual similarity search using any image shown in **(D)**
  - (d) Perform relevance feedback by annotating some images as relevant or irrelevant in **(D)** and submit to the system



5. The user can also go back to any previous step in the search process by clicking on the corresponding dot in the search progress bar (C).

These five steps are performed repeatedly until the user is satisfied with the results and decides to submit the answer to the system.

## 4.4 Search Engine

This section is dedicated to detailing all components of the search engine that powers LifeSeeker. As LifeSeeker was specifically designed to address the LSC challenge (which aims to retrieve lifelog moments based on cues given by a lifelogger), its search engine takes as input a text-based query and returns a list of moments (represented by images) ranked by the descending order of their relevance to the query. To achieve this, LifeSeeker is equipped with two main modules: (1) an indexing module that processes the input data (images, biometrics data, metadata) from the dataset and transforms them into a searchable representation; (2) a retrieval module which takes an input query and matches it with the data previously processed by the indexing module to return the relevant moments.

### 4.4.1 Indexing

Since the lifelog dataset is constructed by gathering data from multi-modal sensors (i.e. wearable cameras, biometric devices, GPS, phones, computers), the **Indexing** module requires various sub-modules, each responsible for processing one modality of the lifelog data. Inspired by the lifelog data analysis from NTCIR-14 Lifelog-3 task [30], we categorise the lifelog data into the following types:

1. **Time:** This is one of the most important pieces of information that helps to narrow the search space. For example, knowing when (morning, afternoon, evening) the moment happened can filter out nearly two-thirds of the original amount of images. Section 4.4.1.1 describes the process of indexing the time data in more detail.

**Listing 4.1:** A sample metadata for a lifelog moment to be indexed into Elasticsearch for filtering, corresponds to Figure 4.3

```

"_id": "20160927_140817_000",
"minute_id": "20160927_1408",
"image_path": "LSC/2016-09-27/20160927_140817_000.jpg",
"date": "2016-09-27",
"local_time": "15:08",
"day_of_week": "tuesday",
"month": "september",
"year": 2016,
"part_of_day": "afternoon",
"gps": [53.38571962, -6.258157063],
"activity_type": "walking",
"lat": 53.38572,
"lon": -6.258157,
"location_name": "work",
"location_type": "dcu, university",
"city": "Dublin",
"country": "Ireland",
"location_address": ["wad", "whitehall a ed", "dublin 9",
  "dublin", "county dublin", "leinster", "ireland"
],
"place_category": ["elevator/door", "elevator lobby"],
"microsoft_tag": ["text", "wall", "door", "indoor", "floor"],
"yolo_concept": ["tv"],
"visual_genome": ["white sign", "tiled floor",
  "black television", "wooden door", "wooden wall",
  "white table"
],
"ocr": "cademic offices first floor school office/reception
  faculty of engineering & computing dcu first floor faculty
  administration offices cngl lsim"

```

2. **Location:** Location can be viewed as a summary of a lifelogger in terms of where they were on a daily basis, which might imply the sequence of activities that the lifelogger does throughout the day. It is also useful for adding more context to the query generation process to find more relevant moments (i.e. if finding moments that the lifelogger was eating a sushi platter, the user can add "Asian restaurant" as part of the LifeSeeker input query to obtain more accurate results). The indexing pipeline for location data can be found in Section 4.4.1.2.

3. **Visual data:** Images captured from the wearable camera are information-



Figure 4.3: The image corresponding to the concepts in Listing 4.1

rich, as moments are illustrated in detail (i.e. what the surroundings look like, who appears in that moment, and which objects are seen). However, computers cannot perceive images as humans do. Therefore, in Section 4.4.1.3, we outlined several adopted approaches to convert images into a machine-searchable format.

4. **Other metadata:** Apart from the aforementioned data sources, there are other modalities provided in the dataset as listed below. However, this metadata can be indexed instantly into the search engine without further processing.

(a) **Activity:** The activity data contains two categories: walking and transport.

(b) **Biometrics:** The biometrics data that we use in our search engine includes heart rate and caloric expenditure.

#### 4.4.1.1 Time Data

When referring to time, we have different ways to describe it. For instance, “September 27, 2016 at 15:08” can be referred to as “2016/09/27 at 15:08”, “Tuesday afternoon in September 2016”, or “September 2016, after 3 pm”. Therefore, to handle input queries containing variable time formats, these different variations need to be indexed in the search engine in advance. We note that the local time gives a more intuitive view into a day in the lifelogger’s data, compared to the standard UTC time collected from wearable devices, especially when the lifelogger was traveling to another country in another hemisphere. Hence, we aligned the current time into the local timezone at the location where the lifelogger was at that time. Since lifelog data is organised on a one-minute basis, each image has a *minute\_id* that we can process as follows:

- Date: The date of the image, in the YYYY-MM-DD format;
- Month: Name of the month (e.g. January, September, December);
- Year: The year in the YYYY format;
- Local Time: The time in the lifelogger’s local timezone in 24-hour format;
- Day of Week: One of the seven days of the week expressed in the lifelogger’s local time;
- Part of the Day: Whether it is *early morning* (04:00 to 07:59), *morning* (08:00 to 11:59), *afternoon* (12:00 to 16:59), *evening* (17:00 to 20:59), or *night* (21:00 to 03:59), based on the local time.

A sample of the generated time data is illustrated in Listing 4.1 in the fields *date*, *month*, *year*, *local\_time*, *day\_of\_week*, and *part\_of\_day*.

#### 4.4.1.2 Location Data

Another important attribute in every lifelogger's life moments is the locations they have been. Knowing the correct location would give more valuable information to expedite the search process. From the geographic coordinates collected from wearable devices, we identify the detailed address of the image using Geocoding API from Google Map Platform<sup>9</sup>. Apart from the address, city and country also play a crucial role in the filtering process, especially for locations outside Ireland (where the lifelogger is based). Moreover, we also cluster the locations into 32 pre-defined place categories. Each image has information related to the location of the lifelogger at that moment as follows:

- Latitude: Angular coordinate specifies the north-south position of the image on the surface of the earth;
- Longitude: Angular coordinate specifies the east-west position of the image on the surface of the earth;
- Location's name: Semantic name of the location (i.e. Dublin Airport, DCU, ...);
- Location's type: One of the 32 predefined categories in Table 4.3;
- Location's address: Detailed address associated with the lifelogger's location;
- City: Name of the city associated with the lifelogger's location;
- Country: Name of the country associated with the lifelogger's location.

#### 4.4.1.3 Visual data

There are two main approaches to making visual data searchable: (1) extracting visual concepts from images and (2) embedding images into a vector space. The first

---

<sup>9</sup><https://developers.google.com/maps/documentation/geocoding>

Table 4.3: Location categories

ID	Name	ID	Name
1	Airport	17	Home
2	Antique store	18	Hotel
3	Apartment	19	Howth
4	Bank	20	Office
5	Bar, pub	21	Park
6	Bus stop	22	Pharmacy
7	Car	23	Plane
8	Castle	24	Restaurant
9	Church	25	Shop
10	Coffee shop	26	Shopping Center
11	Convenience store	27	Sister home
12	DCU	28	Station
13	Dental clinic	29	Store
14	Department store	30	Street
15	Embassy	31	University
16	Hall	32	Unknown

approach is straightforward, as we can employ different pre-trained models to extract visual concepts from images. These visual concepts are very useful to quickly filter out irrelevant images and narrow down the search space. The power of this approach has been proven by the success in previous participations of LifeSeeker (version 1, 2 and 3) in LSC where this has been employed as the main indexing mechanism for visual data. Some of the main visual concepts that we have used in our search engine are:

1. **Text recognition:** Texts appearing in lifelog images can help to determine not only what the lifelogger might have seen, but also the context of the associated life moment. Therefore, to convert texts in lifelog images into visual concepts, we employed the OCR tool from Google Vision API<sup>10</sup> to detect and recognise text content. The extracted texts were then aggregated into a single string (as shown in Listing 4.1 in the *ocr* field) that can be indexed by the search engine in the latter stage.
2. **Object detection:** Object tagging is an essential component for most concept-based retrieval systems. Thus, visual concepts of lifelog images,

<sup>10</sup><https://cloud.google.com/vision/docs/ocr>

obtained from object detection models, are always provided as part of the lifelog dataset in all collaborative research tasks and challenges in the lifelogging domain [36]. Besides the visual concepts shared by the lifelogger/task organisers, which were generated using Microsoft Vision API <sup>11</sup>, we also considered other object detection models (e.g. YOLOv4 [157] and Bottom-up Attention model [152]) with the aim of tagging more objects from lifelog images. The YOLOv4 [157], which was pre-trained on the COCO dataset [151], can detect 80 different categories of common objects in daily life. Meanwhile, the Bottom-up Attention model [152] is able to detect 1600 object classes along with 400 associating attribute types (e.g., black pillar, wooden floor, red car, etc.) by using multi-GPU pre-training of Faster R-CNN [158] with ResNet-101 [107]. This model not only increases the number of concepts by a significant amount but also enables the retrieval of concepts at a finer level of detail using their corresponding attributes. The fields *microsoft\_tag*, *yolo\_concept*, and *visual\_genome* in Listing 4.1 illustrate a sample result of the visual concepts generated by Microsoft Vision API, YOLOv4 and Bottom-up attention model, respectively.

- 3. Scene recognition:** In addition to text and object concepts, detecting the surroundings also gives more insight into where the lifelogger was (e.g., waiting in a lobby, exercising outdoors, working in an office). To achieve this, we utilised the PlacesCNN [118] model pre-trained on the Places365 dataset [118], which classifies images into 365 place categories. For example, the lifelog moment displayed in Figure 4.3 was recognised as "*elevator/door*" and "*elevator lobby*" as shown in the field *place\_category* in Listing 4.1).

However, the main drawback of the first approach is that it is not possible to search for images that contain a specific object or scene that is not included in the pre-trained models. Besides, the semantic meaning of the input query is not

---

<sup>11</sup><https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision>

well-handled. The second approach, on the other hand, is able to address these issues by embedding images into a vector space, where the similarity between image-image or image-query pairs can be measured by the distance between their corresponding embedding vectors. To facilitate this, we employed the Contrastive Language-Image Pre-training (CLIP) [115] model, developed by the OpenAI team, which is an embedding model that learns the relation between visual and semantic concepts of the scene. With the zero-shot transferability, CLIP has been widely used for different tasks ranging from self-supervised learning [159], action recognition [160] to image captioning [161]. As can be seen from Figure 4.1, the CLIP model acts as the *Image Encoder* which converts lifelog images into high-dimensional feature vectors. By doing so, we leverage the contextual meaning of the general image rather than using some keywords to describe the scene only. In addition, this model also acts as a *Text Encoder* to convert the input text query into the same latent space as the images, allowing us to measure the similarity between the query and images and obtain relevant images for the query.

#### 4.4.1.4 Putting It All Together

After extracting all relevant information, metadata and visual concepts, as well as obtaining the embedding vectors of all lifelog images, the indexing process begins. Images' embeddings are then indexed using Milvus [156], an open-source vector database, which is optimised for similarity search and high scalability. This database serves as the main component for handling the search queries. The remaining information, including metadata and visual concepts, are indexed using Elasticsearch <sup>12</sup>, which is responsible for executing filter queries. Listing 4.1 shows a sample of the complete metadata of a lifelog image that is indexed into Elasticsearch. This process only needs to run once for the entire dataset and will be ready to serve the search queries.

---

<sup>12</sup><https://www.elastic.co/elastic/official>



## 4.4.2 Retrieval

### 4.4.2.1 Search Query

Upon receiving a search query from the user, the query directly goes through the *Text Encoder* where the CLIP model is employed to convert the query into a high-dimensional embedding vector. The embedding vector is then passed to Milvus, where the embedding vectors of all lifelog images are stored, to find the most similar images to the query. Milvus then employs the approximate nearest neighbours search (ANNS) algorithm to find the most similar images to the query, by comparing the distance between their embedding vectors. The ranked list is then returned to the user for further browsing and filtering.

### 4.4.2.2 Filter Query

Elasticsearch was used in the main search mechanism in LifeSeeker, first introduced in the second version [41], and is currently responsible for executing filter queries only. A query into Elastic Search can be constructed by combining one or more *query clauses*<sup>13</sup> of various types, thus users can form very complex queries to define how Elastic Search retrieves data. Therefore, this search mode was intentionally integrated for expert users to compete in the LSC challenge.

In order to reduce the query analysis time and allow flexibility in controlling how each keyword should behave when retrieving lifelog moments (i.e., which should be used for matching images and which should be used for filtering purposes only), we introduced a syntax-based query mechanism as below:

$$\langle \text{CONCEPTS} \rangle ; \langle \text{LOCATION} \rangle ; \langle \text{TIME} \rangle \quad (4.1)$$

where each query part ( $\langle \text{CONCEPTS} \rangle$ ,  $\langle \text{LOCATION} \rangle$  and  $\langle \text{TIME} \rangle$ ) corresponds to a category outlined in Section 4.4.1. A syntax-based query can be formed by specifying keywords in each part in Syntax 4.1. For instance, the following query is a valid input

---

<sup>13</sup><https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html>

to LifeSeeker:

```
flower teddy bear ; bedroom home ; after 7pm on Monday
```

The Searching process in Elastic Search mode was done by employing the *query string query*<sup>14</sup> to match <CONCEPTS> and <LOCATION> keywords, while the *term query*<sup>15</sup> and *range query* mechanisms were used to filter images using the given <TIME> keywords.

#### 4.4.2.3 Active Search

Inspired by the way a decision tree functions, which repeatedly breaks down a dataset into smaller subsets based on the value of a certain attribute, we introduced the *Active Search* mechanism to LifeSeeker, which breaks down the search results (based on whether a concept presents in the image) into two smaller subsets and allows users to interactively choose which subset to continue searching. Based on this idea, we attempted to transform our passive search system into an active retrieval engine that can actively support the user during the searching process. In conventional passive search systems, the user needs to think of relevant concepts related to the information needed based on the description of the query, which depends heavily on the ability of the user and ultimately relies too much on the user to know which concepts are likely to assist in finding relevant content. In contrast, for an active retrieval engine, the mutual interaction between the user and the retrieval engine is more important. LifeSeeker can act as an assistant for the user during the search progress by asking the user some Yes/No questions on the images' visual concepts to narrow the set of relevant items.

After each search query or filter query, LifeSeeker will return a ranked list of images that are most relevant to the query. The visual concepts of all images in the

---

<sup>14</sup><https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html>

<sup>15</sup><https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-term-query.html>

ranked list are then aggregated into a single list of concepts. Counts of images that contain/do not contain each concept are then calculated and sorted in ascending order of the difference between the two counts. This helps to identify the concepts that best split the ranked list into two halves, where one of the halves will be discarded once the user answers the question of whether the desired image they are looking for contains the concept or not. The details of the Active Search mechanism are shown in Algorithm 1. The results are updated each time the user answers the Active Search question, and the process is repeated to generate new questions until the user finds the desired image or there are no more images to show.

---

**Algorithm 1** Active Search Algorithm

---

**Require:** *Images* - ranked list of images

```
1: Counter  $\leftarrow \{\}$ 
2: for Image_Id in Images do
3:   Concepts  $\leftarrow$  GetImageConcepts(Image_Id)
4:   for C in Concepts do
5:     Counter[C] $+$  = 1
6:   end for
7: end for
8:
9: Rank  $\leftarrow \{\}$ 
10: TotalImages  $\leftarrow$  Length(Images)
11: for Concept, Count in Counter do
12:   Rank[Concept]  $\leftarrow$  |Count - TotalImages|/2
13: end for
14:
15: Rank  $\leftarrow$  Sort(Rank)
16: return First key in Rank
```

---

The Active Search mechanism was first introduced in the AVSeeker [162] system, which is a variant of LifeSeeker (version 3.0) that was developed for the Video Browser Showdown (VBS) challenge in 2022 [163]. The AVSeeker system was ranked 5th out of 16 systems in VBS and was entitled the best newcomer system, which proves the effectiveness of the Active Search mechanism in supporting the user during the searching process.

#### 4.4.2.4 Visual Similarity Search

In addition to the text-based search, we also implemented a visual similarity search mechanism, which allows the user to find similar images to a given image shown in the search results. This mechanism is straightforward to implement, as all images' embeddings are already stored in Milvus. Therefore, it only requires a cosine similarity calculation between the embedding vectors of the given image and all other images in the database to obtain the final result.

#### 4.4.2.5 Relevance Feedback

To better support users when interacting with the system, we incorporate relevance feedback, where users are able to provide extra information to adjust the current result. After specifying a list of positive images (i.e., images that are relevant to the query) and negative images (i.e., images that are not relevant to the query), the system then performs a series of visual similarity searches to find similar images to the positive and negative images, respectively. The similarity scores of the negative images are then marked as negative. The final result is obtained by fusing the similarity search results of the positive and negative images using CombSUM method proposed by Edward et. al. [164], which is a state-of-the-art fusion algorithm for combining multiple ranked lists into a single ranked list. The implementation of the CombSUM method is provided by Ranx Fuse [165], which is a Python library that cumulates the implementation of state-of-the-art fusion algorithms for rank fusion. The detail of the Relevance Feedback mechanism is shown in Algorithm 2.

---

**Algorithm 2** Relevance Feedback Algorithm

---

**Require:** *Positive\_Images* - Images that are relevant to the query

**Require:** *Negative\_Images* - Images that are not relevant to the query

- 1: *Relevant\_Images*  $\leftarrow$  GetVisualSimilarImages(*Positive\_Images*)
  - 2: *Irrelevant\_Images*  $\leftarrow$  GetVisualSimilarImages(*Negative\_Images*)
  - 3: *Rank\_List*  $\leftarrow$  CombSUM(*Relevant\_Images*, *Irrelevant\_Images*)
  - 4: **return** *Rank\_List*
- 

By doing so, we are able to place images closer to positive images higher while

putting images similar to negative samples lower in the ranked list.

## 4.5 Benchmarking Result in Lifelog Search Challenge

LifeSeeker was evaluated in the annual Lifelog Search Challenge (from 2019 to 2022), along with many other lifelog retrieval systems from all around the world. The primary objective of these challenges is the speed and accurate retrieval of lifelog images that best match specific queries, with penalties imposed for incorrect submissions. The challenge format was interactive, involving users actively engaging with the system to search for and submit images they deemed most relevant to the given queries. LSC score (described in Section 3.3) was used to evaluate the performance of the systems. In this section, I present the results of LifeSeeker in the two most recent LSC challenges, namely LSC’21 and LSC’22.

Table 4.4: Statistics of the top-5 teams in LSC’21

Team name	No. queries solved	Total score	Precision*	Recall*
Myscéal [106]	19	<b>1604.31</b>	0.83	0.83
SomHunter [87]	19	1566.32	0.68	0.83
<b>LifeSeeker</b> [42]	<b>20</b>	1556.02	0.77	<b>0.87</b>
Voxento [88]	18	1466.87	<b>0.86</b>	0.78
Memento [5]	16	1238.49	0.59	0.70

\*Precision and Recall was re-defined by the challenge’s organisers. Precision is the ratio of correct images to total submissions. Recall is the proportion of solved queries to total queries.

In LSC’21, there was only one type of search task, which was the Known-Item Search (KIS) task, in which the participants were required to find a specific image based on a textual description of the image. LifeSeeker distinguished itself as the third best-scoring system in the challenge, attaining a total score of 1556.02, as detailed in Table 4.4. The table reveals that LifeSeeker resolved the highest number of queries among the top five systems, successfully addressing 20 out of 23 queries. In addition to the score, the performance of our system was also assessed through other metrics, including variants of precision and recall. According to the challenge’s organiser, precision is defined as the ratio of correct images to total submissions, and

recall is the proportion of solved queries to total queries.

LifeSeeker achieved the highest recall score in the competition at 0.87. This result signifies that our system was capable of accurately retrieving the requested information in 87% of the instances, outperforming the second-highest recall system by 4.35%. However, in terms of precision, LifeSeeker recorded a score of 0.77. This was 8.8% lower than that of Voxento [88], the system with the highest precision. A primary factor contributing to this lower precision score was the number of incorrect submissions made by the user of LifeSeeker during the challenge. Despite this, the overall performance indicators suggest that LifeSeeker is a robust system, demonstrating a high capacity for retrieving desired information efficiently in a competitive environment.

Table 4.5: The normalised score of the top-5 teams for each task in LSC'22. Detailed metrics of original scores and metrics were not released by the organisers.

Team name	Task		
	Ad-hoc	KIS	QA
Myscéal [4]	98	<b>100</b>	<b>100</b>
<b>LifeSeeker</b> [43]	<b>100</b>	88	96
Memento [89]	66	92	79
FIRST [6]	51	95	75
Voxento [7]	49	87	56

Unlike LSC'21, LSC'22 featured three different types of search tasks, namely Ad-hoc, KIS and QA. The Ad-hoc search task required users to find, within a time limit, as many relevant images as possible that matched the description. Compared to KIS, the query in the Ad-hoc task is more general and less specific (e.g. "Find all moments that I had a burger"). Conversely, the Question-Answering task demanded the identification of a single image that answered a specific question, with only one submission attempt allowed.

As indicated in Table 4.5, LifeSeeker excelled in the Ad-hoc search task, achieving the highest score. This performance underscores the effectiveness of the improvements applied to the system, particularly in achieving high recall scores.

However, in the KIS task, LifeSeeker’s score of 88 was the fourth-highest score among the top five systems. In contrast, the Question-Answering task showed LifeSeeker’s effectiveness, as evidenced by its second-highest score of 96/100 in this category. One might question why LifeSeeker performed better in the QA task than in the KIS task, given that the two tasks are similar (i.e. both require the system to find a specific image based on a textual description). The answer lies in the difference in the number of submissions allowed. In the KIS task, there are multiple submissions allowed, and the system operator has the option to risk submitting an uncertain result for a chance to obtain a higher score if the submission is correct. And this, in our case, led to a lower precision score. Conversely, in the QA task, only one submission is allowed, most systems’ operators would be more cautious and only submit when they are confident in the result (when receiving more hints), leading to a higher score.

The benchmarking results of LifeSeeker in LSC’21 and LSC’22 demonstrate the system’s effectiveness in retrieving relevant images based on textual descriptions. In particular, LifeSeeker has a competitive performance with other state-of-the-art systems in competitions, achieving the third-highest score in LSC’21 and the second-highest score in LSC’22, and hence, LifeSeeker can be considered as one of the current state-of-the-art lifelog retrieval systems. Through the development of LifeSeeker, I have gained valuable insights into the components and techniques that are essential for the construction of a state-of-the-art system. From Table 4.1, I have identified the following key components for building a state-of-the-art system, which are: (1) a robust semantic search engine which supports free-text query in natural language, (2) an effective concept-based filtering mechanism, which works for all metadata (e.g. visual concepts, scene texts, location, time), (3) a simple user interface with enhancement in result presentation and visualisation to provide a comprehensive overview of the search results, and (4) functionalities to support the refinement of search results, including filtering, visual similarity search, and relevance feedback. The identification of these key components has provided me with a solid foundation

for developing the next retrieval system to allow on-screen information in infologging data to be retrieved, which I will further elaborate in Chapter 6

## **4.6 Chapter Summary**

In this chapter, I addressed Research Question 1 by presenting the design, implementation and evaluation of LifeSeeker, an interactive retrieval system for the multi-modal personal lifelog data, which allows users to search and filter for lifelog moments based on textual queries. To achieve this, the system utilises many state-of-the-art techniques (including image-text embeddings model and along with visual concepts extractors) and advanced engineering solutions for efficient indexing and retrieval (such as Milvus for large-scale vector similarity calculation, Elasticsearch for distributed and scalable text search, and Redis for caching). This combines with an intuitive and easy-to-use user interface, which is designed to facilitate fast and effective search and exploration of the lifelog data, making LifeSeeker among the state-of-the-art retrieval systems for lifelog data.

Overall, research question 1 is answered as I have developed a state-of-the-art interactive lifelog retrieval system, which shows competitive performance in LSC'21 and LSC'22. Additionally, I have identified four main key factors for developing an effective interactive retrieval system, which I have summarised at the end of Chapter 4.5. Learning of these key factors forms a solid foundation for the development of a subsequent retrieval system designed for efficiently retrieving on-screen information in infologging data. The specifics of this development will be discussed in greater detail in Chapter 6.



## Chapter 5

# Reading Comprehension Estimation using Eye Movement Measures

### 5.1 Introduction

In this chapter, I address Research Question 2: **To what extent can machine learning models accurately estimate reading comprehension levels based on eye movement features extracted from eye-tracking data?**

To investigate this, I first create a reading dataset that captures participants' eye movements while reading a set of passages using one of four pre-defined reading conditions. Then, ocular events are detected from the eye-tracking data on which a set of eye movement features are extracted. The relationship between eye movement features, reading conditions and reading comprehension is then investigated using two separate approaches: (1) statistical testing procedures and (2) machine learning analyses. In the machine learning approach, I have further divided the problem into two smaller tasks, which are reading condition classification and reading comprehension prediction. I show that by integrating the identification of reading styles alongside eye movement measures for estimating reading comprehension can enhance the accuracy of predictions within reading conditions.

In this chapter, I obtained an average classification accuracy of reading styles

of 75.3% in a subject-dependent setting (training and testing on the same subject) and 68.9% in a general setting (training and testing on all subjects). Furthermore, my analysis revealed that reading and skimming styles are associated with higher comprehension levels, whereas scanning and proofreading styles are linked to lower comprehension levels. As a result, applying the reading conditions classification model to predict reading style prior to estimating reading comprehension yielded an 8.9% improvement in correlation coefficient compared to the model which was trained with reading styles (with coefficients of 0.697 and 0.608, respectively). The correlation coefficient between the predicted and actual comprehension levels can be further boosted up to 0.708 when the true reading condition labels are (i.e. a perfect classification model) used to train the model. Furthermore, I want to highlight that the study was conducted using a low-cost eye tracker. This further advances possibilities for what can be done in every day reading scenarios.

The remainder of this chapter is structured as follows: In Section 5.2, I describe the dataset utilised in my study, including the data collection protocols and experiment setup. Section 5.3 outlines the methodology employed in my research, covering the data analysis techniques, feature extraction methods, and model development. I investigate the integration of eye movement measures and reading styles for estimating reading comprehension. The results of my experiments and analysis are presented in Section 5.4. I report the results obtained for the machine learning models in classifying reading conditions and estimating reading comprehension levels. Finally, I conclude this chapter in Section 5.5 by summarising the key contributions of my research and discussing the implications of my findings. We also identify potential directions for future studies in the field of estimating reading comprehension in real-world scenarios.

## 5.2 Data Collection

Data collection was carried out with approval from Dublin City University's Research Ethics Committee (DCUREC/2021/138). In selecting participants, I adhered to the following inclusion criteria: (1) no history of reading difficulties, (2) normal or corrected-to-normal vision, (3) be able to maintain a relatively steady head position for the duration of the experiment, and (4) comply with the instructions provided during the experiment. A total of  $N = 10$  participants, 6 males and 4 females were recruited for the study. Of these, 5 were non-native English speakers (from  $S_0$  to  $S_4$ ) where remainder were native English speakers (from  $S_5$  to  $S_9$ ). The study involved 96 trials, with each trial consisting of reading a passage (with an average length of 353 words) and answering multiple-choice questions (MCQs) related to the passage. For each MCQ, participants were presented with five choices, with four options directly related to the passage (with exactly one correct answer) and one option labelled as "I don't know the answer" to minimise random guessing. The participants were instructed on how to read the passage using one of four reading conditions (sequential reading, skimming, scanning, and proofreading) prior to the experiment. The maximum allowed reading time for each trial was 60 seconds. The study was designed such that there were an equal number of trials (24) for each reading condition.

The study utilised passages selected from the RACE dataset [166] (see Appendix A.1), which consists of 12 frequently-occurring topics such as university/education, transportation, nature and animals, music, art, energy and climate change, sleep, stress, and mental health. The choice of this dataset was motivated by the fact that the texts within it were scraped from various online sources such as news articles, blogs, leaflets, and advertisements. This diverse range of text formats closely mirrors the types of texts that individuals encounter in their daily lives. Each participant read an equal number of passages for each condition and topic. The set of passages were sampled so that all passages are different and there are passages that is unique

to one participant and passages that are common to all participants (more details in Appendix A.1). The MCQs used to assess comprehension were chosen from the set of MCQs provided for each passage in the dataset, excluding cloze-type questions. The passage sampling process involved topic modelling using TF-IDF (Term Frequency - Inverse Document Frequency [167]) and NMF (Non-negative Matrix Factorization [168]). This is detailed in Appendix A.1. Passages were presented on a 24-inch Phillips LCD monitor (model 240V5QDAB/00) with  $1920 \times 1080$  resolution and controlled by a Dell Optiplex 5060 PC powered by the Windows 10 operating system. Experimental participants sat approximately 60cm from the screen and no chin rest was used. Eye movements were captured using the Gazepoint GP3 HD Eye Tracking device<sup>1</sup> with a sampling rate of  $150Hz$  (one sample per 6.67 milliseconds). The data-gathering process was driven by software written in Python using Psychopy [169].

During the study, the participants were seated in a chair facing a computer monitor, with the eye tracker positioned below the monitor to capture eye movements. Calibration was performed using a 5-point grid, and the accuracy was checked with a 12-point grid using the eye tracker's software. This process takes approximately 5 minutes. The data collection process consisted of 96 trials divided into 4 sessions. Participants were allowed to rest for a maximum of 15 minutes after each session was completed. Within one session, 24 passages were presented, and after each passage, the participant was required to give a subjective evaluation (Likert scale from 1 to 5, see Table 5.1) and answer 3 MCQs. Each trial began with a short guideline indicating the required reading condition (i.e. read the following text carefully, skim quickly through the following text). This was followed by the presentation of the passage on the screen. After reading the passage, the participant was asked to provide a subjective evaluation. Finally, the participant was required to answer three multiple-choice comprehension questions related to the passage. I refer to the dataset constructed in this study as the first version of the *Reading Comprehension for Information Retrieval* dataset (RCIRv1).

---

<sup>1</sup><https://www.gazept.com/product/gp3hd/>

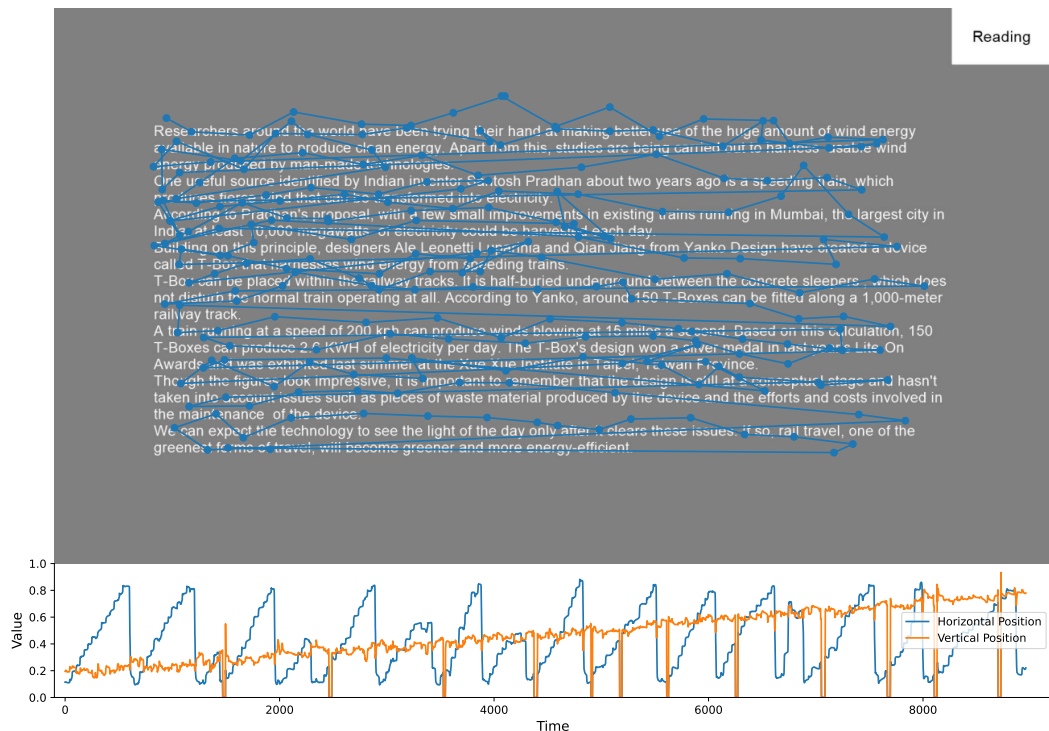


Figure 5.1: Upper: Visualisation of a participant’s eye movements when reading a passage. The dots represent fixations and the lines represent saccades. Lower: The corresponding eye movement captured by the eye-tracker, decomposed into horizontal and vertical components. A value of 0 represents the leftmost or topmost position of the screen, while the value of 1 represents the rightmost or bottommost position of the screen.

In the RCIRv1 dataset collection process, consistent instructions and guidelines were provided to all participants. The participants were also reminded that the study was not focused on competition with others or achieving a high MCQ score, but rather on properly completing the tasks at hand. This was intended to minimize the impact of extraneous variables on the study's results and allowed the participants to express their natural behaviour. During the study, the experimenter monitored and recorded any unusual occurrences, such as participant yawning, sudden loud noises, disruptions, or large movements, in order to provide additional context for each trial and aid in the analysis of the data at a later stage for data cleaning. Additionally, the participants were asked to provide verbal responses to subjective evaluation questions and multiple-choice questions (MCQs) rather than entering their answers on their own using a keyboard, in order to reduce the potential for large head displacements and disruptions to the eye tracker calibration. Figure 5.1 shows an example of a participant's eye movements when reading a passage.

### **5.3 Methodology**

This study aims to explore the relationship between eye movements, reading conditions, and comprehension levels, within the context of different reading strategies. To accomplish this, I investigate the pairwise associations among these three main data sources under two distinct angles: statistical testing procedures and machine learning analyses. Outcomes from both of these two approaches were kept separate and served primarily to highlight and explain their differences and similarities. In the subsequent sub-sections, I introduce the specific type of comprehension that I studied and it can be measured. We then proceed to outline the preprocessing procedure and feature extraction pipeline employed for analysing eye movement data. Finally, I elaborate on the methodologies used to explore the relationships among the aforementioned factors and come up with a pipeline for predicting comprehension levels based on eye movements.

### 5.3.1 Comprehension and Measuring Comprehension

There are many types of comprehension, such as literal comprehension, inferential, reorganisation, prediction and evaluation [170]. The focus of this chapter is on literal comprehension – the simplest form of comprehension, which refers to the understanding of the explicit meaning of the text and the ability to answer questions based on direct evidence from the text’s content. The experiment protocol is designed to induce this type of comprehension by asking participants to answer multiple-choice questions based on what they have read.

To measure comprehension, a comprehension score, denoted as  $c\_score$ , was generated for each text, based on the participant’s answer to the multiple-choice questions. This score is a weighted sum of the total correct, incorrect and unknown answers (as outlined in experiment protocol in Section 5.2). The computation of the comprehension score is defined as follows:

$$c\_score = 1 \times N_c + 0.5 \times N_i + 0 \times N_u$$

where  $N_c$ ,  $N_i$ , and  $N_u$  represent the number of correct, incorrect and unknown answers, respectively. The chosen weighting approach is rooted in the strict guidelines provided to volunteers regarding how to handle multiple-choice questions. In particular, participants were instructed to pick an answer (1, 2, 3 or 4) if and only if they found supporting evidence within the text they just read, otherwise, they were instructed to choose option 5 – *"I don't know the answer"*. This guideline was provided so that participants would provide accurate responses based on the evidence in the text, reducing the likelihood of false comprehension estimations resulting from random guessing.

In addition to the comprehension score, I also collected participants’ subjective comprehension evaluations as an additional reference for comprehension, as described in the experiment protocol in Section 5.2. This evaluation was obtained using a 5-point Likert scale, ranging from 1 (very poor) to 5 (very good). To establish a

baseline for participants to evaluate their comprehension, I provided them with a set of guidelines to follow, which can be summarised as follows:

Table 5.1: Guideline for participants to report their comprehension

Score	Description
1	I followed very little of the text. Besides a few keywords I noticed, I wouldn't be able to say what the text is about. My comprehension is very low.
2	I got one or two points/pieces of information from the texts but overall I wouldn't be able to summarise what the text is about.
3	I got multiple points from the text and would be able to say what the text is about but I know I missed a lot of specific information.
4	I got most points of information in the text and would be able to comfortably summarise what the text is about, and answer questions on the text
5	I got (nearly) all points from the text, and feel confident I could answer any reasonable question asked about the text.

### 5.3.2 Data pre-processing and Feature Extraction

A commonly referenced set of features in the literature includes fixation, saccades, and blinks, which are frequently employed in studies involving eye-movement data [67]. However, the extraction of oculomotor events from eye-tracking data involves numerous algorithms proposed to address this task. A comparison of various well-known algorithms is summarised in [171]. Interestingly, no single algorithm stands out as the definitive choice for detecting ocular events, as their selection often depends on the specific task at hand [171].

In this experiment, I utilised the ocular event detection algorithm that was already integrated into the eye-trackers software, as it has undergone manufacturer testing and was widely used in relevant studies conducted with the same eye tracker. Building upon the identified ocular events, I have further derived additional features to better characterise the reading process, including measures such as moving distance, velocity, angle of movement, and rate of regressive movement. A comprehensive summary of these ocular events and features can be



found in Table 5.2

Table 5.2: Summary of ocular events and features used in my experiment

Name	Description	Type
<i>nfx</i>	Number of fixations	scalar
<i>nbk</i>	Number of blinks	scalar
<i>fxdur</i>	Fixation durations	sequence
<i>scdur</i>	Saccade durations	sequence
<i>scdir</i>	Saccade directions (angles)	sequence
<i>bkdur</i>	Blink durations	sequence
<i>dist</i>	L2 distances between two consecutive fixations (i.e. one saccade)	sequence
<i>dist_v</i>	L1 distance between two consecutive fixations on vertical axis	sequence
<i>dist_h</i>	L1 distance between two consecutive fixations on horizontal axis	sequence
<i>velo</i>	Velocity of movement	sequence
<i>velo_v</i>	Velocity of movement on vertical axis	sequence
<i>velo_h</i>	Velocity of movement on horizontal axis	sequence
<i>nregr</i>	Number of regressions	scalar
<i>regr_rate</i>	Ratio between the number of regressions and number of fixations	scalar

Given that the reading process encompasses various ocular events, many extracted features constitute sequence data. To facilitate analysis and leverage machine learning algorithms, I transformed these sequence data into a standardized dimension. Two methods were employed for this purpose: *statistical encoding* and *histogram encoding*. In *statistical encoding*, I computed several statistics of the data sequence, including trimmed max, trimmed min, mean, standard deviation, interquartile range, skewness, and kurtosis. The use of trimmed max and min aimed to mitigate the influence of outliers and extreme values within the data sequence. On the other hand, in *histogram encoding*, I generated the histogram of the data sequence and employed it as the feature vector. With these two encoding methods, I was able to transform the reading samples into a fixed-length feature vector, which can be easily analysed and fed into machine learning algorithms. In Table 5.3, I describe the encoding methods used in my experiment. The new features generated from these encoding methods will inherit the same name as the

original feature, with the addition of a suffix to indicate the encoding method used.

Table 5.3: Summary of encoding methods used for sequence features, along with their abbreviation which will be appended to the end of the corresponding feature's name.

Suffix Name	Description
<i>tr_max</i>	Trimmed Max
<i>tr_min</i>	Trimmed Min
<i>std</i>	Standard Deviation of the distribution
<i>mean</i>	Mean value of the distribution
<i>argmin</i>	Element that has trimmed min value
<i>argmax</i>	Element that has trimmed max value
<i>tr_range</i>	Trimmed Range ( $tr\_max - tr\_min$ )
<i>iqr</i>	Inter-quartile Range ( $Q3 - Q1$ )
<i>kurtosis</i>	Kurtosis value of the distribution
<i>skewness</i>	Skewness of the distribution
<i>bin_1 to bin_n</i>	Histogram bins

### 5.3.3 Analysis

To understand how eye movement reflects one's level of comprehension, I studied the triad of eye movement features, reading condition and comprehension level since the interaction between these is key to this study. To accomplish this, I employed several statistical and machine learning techniques to monitor the interconnected relationships between each pair of these factors. The proposed methods which were used to investigate each pairwise relationship were described in the following subsections.

Prior to these details, I would like to give a brief overview of some experiment settings that I used in this study. Since my dataset was constructed from reading samples of many participants, I could approach the dataset in three different ways which I referred to as experiment settings:

- **General (GE):** The dataset is treated as a whole. All samples from all participants are aggregated together to carry out the investigations and analyses. This setting allows us to have an insight into the common characteristics of eye movement while reading across all participants.

- **Subject-Dependent (SD):** Only samples from one participant are used to analyse at a time (e.g. a machine learning model is trained and tested on the same participant). Analysis done in this setting would reveal the unique eye movement features which characterise a participant’s reading.
- **Subject-Independent (SI):** This setting takes the entire dataset except the samples from one participant to analyse. It explores the generalisability of eye movement features to an unseen participant. This process can be repeated across participants.

Moreover, to ensure the reliability of the findings, the dataset is split into training and testing sets based on the topics obtained from the topic modelling process as mentioned in Section 5.2 and detailed in Appendix A.1. A pooled train set contains 720 samples (72 from each participant) and a pooled test set contains 240 samples (24 from each participant). In addition, cross-validation methodology was employed throughout the analyses on the training set, to provide a more robust evaluation of the models’ performance. Specifically, the training data was split into 10 distinct combinations of training and validation sets using the stratified shuffle split technique, with a validation size of 22.22%. For conducting significance tests, the number of splits was increased to 100 to enhance the robustness of the results.

### **5.3.3.1 Reading Condition and Comprehension Level**

To address the question of whether different reading conditions could lead to different levels of comprehension, I conducted some statistical analyses under the GE and SD settings. In both settings, the comprehension scores were treated as a continuous dependent variable and reading condition was the independent variable (Reading, Scanning, Skimming and Proofreading). We adopted the Shapiro-Wilk test [140] as described in Section 2.3.2 to check the normality of the comprehension scores to choose the appropriate test for determining whether there are statistically significant differences between reading conditions on comprehension scores (either ANOVA [139]

or Kruskal-Wallis test [142] as outlined in Section 2.3.1). This was then followed by a post hoc test (t-test [143] or Conover's test [145] as discussed in 2.3.3), with p-value adjustment for multiple comparisons [144], to further identify which pairs of reading conditions are significantly different from each other.

### **5.3.3.2 Eye Movement Features and Reading Condition**

In this experiment, I aimed to investigate the relationship between eye movement features and reading conditions to gain insights into how eye movements are influenced by different reading strategies.

#### **Statistical Testing Procedures**

We started by conducting exploratory data analysis (EDA) using statistical tests to identify eye movement features that exhibited significant differences across reading conditions. To ensure the appropriateness of the tests, I conducted preliminary checks such as the Shapiro-Wilk test [140] ( $\alpha = 0.05$ ) for normality and Bartlett's test [141] ( $\alpha = 0.05$ ) for homoscedasticity. For the features that met the ANOVA [139] assumptions of normality and homoscedasticity, I applied a one-way ANOVA test to determine if there were statistically significant differences between reading conditions. Features that did not meet the assumptions underwent a Kruskal-Wallis test [142] instead. Post-hoc tests, such as t-tests [143] or Conover's test [145], were performed following a significant result to identify specific pairs of reading conditions that differed significantly from each other.

#### **Machine Learning Analyses**

We also explored the potential of eye movement measures as indicators of reading strategies. We approached this as a classification task, with reading conditions as the output classes and eye movement measures and other metadata (excluding comprehension scores) as the input features. We implemented a baseline approach using various machine learning classifiers, including Random Forest, Extra Trees,

Ada-Boost, ElasticNet, Ridge, Bayesian Ridge, K Nearest Neighbor, Gradient Boosting, Light Gradient Boosting Machine, and Logistic Regression. The top-3 best-performing models were further tuned and feature selection was performed using the Recursive Feature Elimination (RFE) method.

Feature selection was conducted in two settings: General (GE) and Subject Dependent (SD), similar to the model training setting. The GE setting aimed to identify common features shared by all participants, while the SD setting aimed to identify subject-specific features unique to each participant. The selected features were then used to retrain the models. Among the models, the one with the best performance was selected to conduct feature analysis to gain a deeper understanding of how the features contributed to the model's predictions. For this regard, the SHAP (Shapley Additive exPlanations) method was employed.

### **5.3.3.3 Eye Movement Features and Comprehension Level**

In this experiment, my goal was to examine the efficacy of eye movement features in predicting comprehension levels.

#### **Statistical Testing Procedures**

Following a similar procedure to the previous experiment (Section 5.3.3.2), I conducted an exploratory data analysis (EDA) to identify eye movement features strongly correlated with comprehension scores. Our EDA aimed to address two main questions. Firstly, I sought to identify the eye movement features that exhibited a strong correlation with comprehension scores for each participant. Utilising the Spearman's correlation test [147], I identified the top positively or negatively correlated features. Secondly, I investigated whether a common set of eye movement features emerged among a subset of participants. To answer this question, I compiled the top 10 features with the strongest Spearman's correlation coefficients [147] from each subject, created a feature set, and determined the frequency of each feature in the set.

## Machine Learning Analyses

To investigate how eye movement features could predict comprehension scores, I constructed baseline models employing various regression models such as Extra Trees, Random Forest, Gradient Boosting, Bayesian Ridge, Adaboost, and Light Gradient Boosting Machine. Model training was conducted under GE and SD settings, while excluding the SI setting based on previous findings (Section 5.3.3.2). To reduce dimensionality, I employed a feature reduction process and retrained the top 3 performing models using the reduced features to evaluate their impact on model performance. The model with the best performance was selected for further investigation, particularly the incorporation of reading condition prediction. We assessed this by adding the reading condition as an additional feature obtained through two approaches: (1) using the predicted reading condition from the model trained in Section 5.3.3.2 and (2) using the actual reading condition from the dataset. Lastly, I conducted an in-depth analysis of the selected model using SHAP to gain insights into the contribution of features to the model's predictions.

## 5.4 Results and Discussion

### 5.4.1 Reading Condition and Comprehension Level

Since the comprehension scores ( $c\_score$ ) were found to deviate from normal distribution based on the Shapiro-Wilk's test ( $p < 0.0001$ ), which violates the assumption of normality for the One-way ANOVA, I opted for the non-parametric Kruskal-Wallis H-test. In the GE configuration, I had a total of 960 samples evenly divided into four groups representing different reading conditions. The Kruskal-Wallis test was performed to determine whether there were differences in  $c\_score$  between groups of samples that differed in reading conditions: Reading, Scanning, Skimming and Proofreading (240 for each condition, across participants). Distributions of  $c\_score$  were not similar for all conditions, as assessed by visual

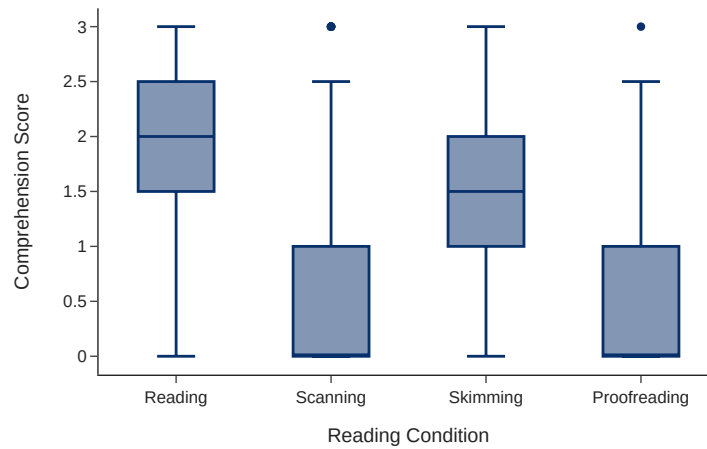


Figure 5.2: Box plot of comprehension scores grouped by reading condition (pooled across participants)

inspection of a boxplot (Figure 5.2). We found that  $c\_score$  was statistically significantly different between different reading conditions,  $H(3) = 430.07$ ,  $p < 0.0001$ , with a mean rank of 698.77 for Reading, 304.65 for Scanning, 617.71 for Skimming, and 300.87 for Proofreading. Subsequently, pairwise comparisons were performed using Dunn's procedure. A Bonferroni correction for multiple comparisons was made with statistical significance accepted at the  $p < 0.05$  level. This post hoc analysis result was demonstrated in Table 5.4 which revealed statistically significant differences in  $c\_score$  between all group combinations except the Scanning-Proofreading pair.

Similar procedures were carried out in the Subject Dependent (SD) configuration, but only data samples from the same individual were used each time. Figure 5.3 summarises the results of the post hoc test for each subject in the dataset. In comparison to the GE configuration results, I observed that, apart from the Scanning-Proofreading pair, there was an additional pair, Reading-Skimming, which did not exhibit significant differences in  $c\_score$ . A significant relationship between Reading and Skimming was only observed in this particular subject's test results. Generally, most subjects displayed a consistent pattern where the  $c\_score$

Table 5.4: Posthoc test results for each pair of reading conditions. Each row tests the null hypothesis that the Condition 1 and Condition 2 distributions are the same. Asymptotic significances (2-sided tests) are displayed. The significance level is .050.  
<sup>a</sup>. Significance values have been adjusted by the Bonferroni correction for multiple comparison tests.

Condition 1 - Condition 2	Test Statistic	Std. Error	Std. Test Statistic	Sig.	Adj. Sig. <sup>a</sup>
<i>Proofreading-Scanning</i>	3.785	24.555	0.154	0.877	1.000
<i>Proofreading-Skimming</i>	316.848	24.555	12.903	<0.001	<0.001
<i>Proofreading-Reading</i>	397.900	24.555	16.204	<0.001	<0.001
<i>Scanning-Skimming</i>	-313.063	24.555	-12.749	<0.001	<0.001
<i>Scanning-Reading</i>	394.115	24.555	16.050	<0.001	<0.001
<i>Skimming-Reading</i>	81.052	24.555	3.301	0.001	0.006

of four pairs, Reading-Scanning, Reading-Proofreading, Skimming-Scanning, and Skimming-Proofreading, were statistically significantly different. However, subjects S2, S5, and S6 did not show a significant difference in the Skimming-Scanning pair.

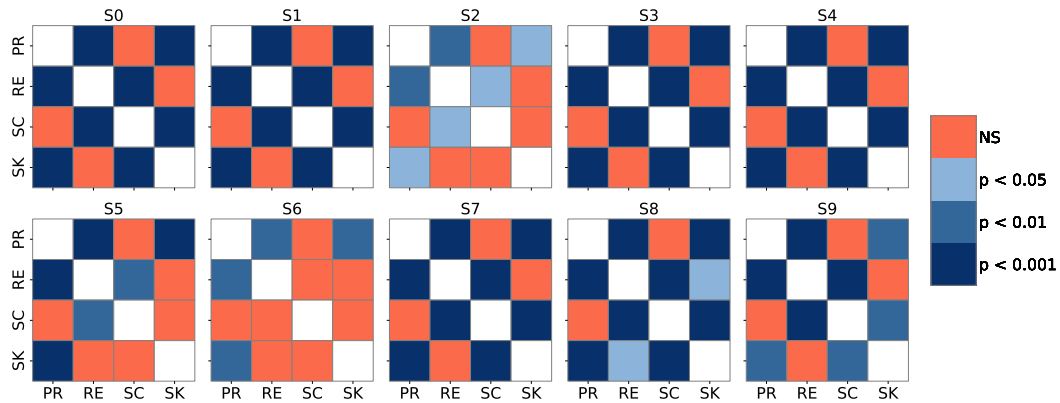


Figure 5.3: Pairwise relationship of reading conditions grouped by subject (RE: Reading, SK: Skimming, SC: Scanning, PR: Proofreading). P-values were adjusted using Bonferroni Correction for multiple comparison tests.

These observations suggest that Reading and Skimming involve a more comprehensive engagement with the text which requires readers to focus on understanding the content as a whole, allowing them to extract meaning and make accurate inferences. On the other hand, Scanning and Proofreading tasks require a more targeted and fragmented approach, which may lead to a lower comprehension level as readers may overlook important contextual information. The inspection of



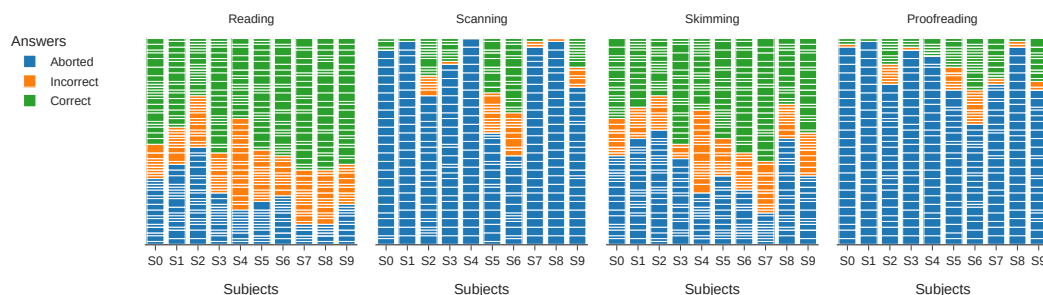


Figure 5.4: Counts of correct, incorrect and aborted answers of each subject and grouped by reading condition.

the number of correct/incorrect answers also supports this pattern. As shown in Figure 5.4, Reading tasks have a higher number of answer attempts compared to Skimming, and both Reading and Skimming have more answer attempts than Scanning and Proofreading. This suggested that participants are more likely to attempt the questions in the Reading and Skimming tasks as they have a thorough examination of the text, while in the Scanning and Proofreading task, the emphasis on seeking specific details or errors may divert participants' attention from understanding the content. Regarding time spent on the reading task as displayed in Figure 5.5, I observed a pattern that time spent on Reading was the longest (except S4, S6 and S9), followed by Proofreading, then Skimming and Scanning. From this, I could see that native English speakers (*S5* to *S9*) are time-efficient in reading tasks, especially in Reading and Skimming. This could explain why they have more answer attempts, as well as correct answers, in these tasks compared to the non-native group (*S0* to *S4*). In the case of slow readers, who got cut off by the time limit for many tasks such as S4 and S6, there is a contrast in their comprehension levels. Specifically, S4 had the most incorrect answers in Reading and Skimming compared to the remaining participants, while S6 maintained a good ratio between correct and incorrect answers which is similar to the other participants.

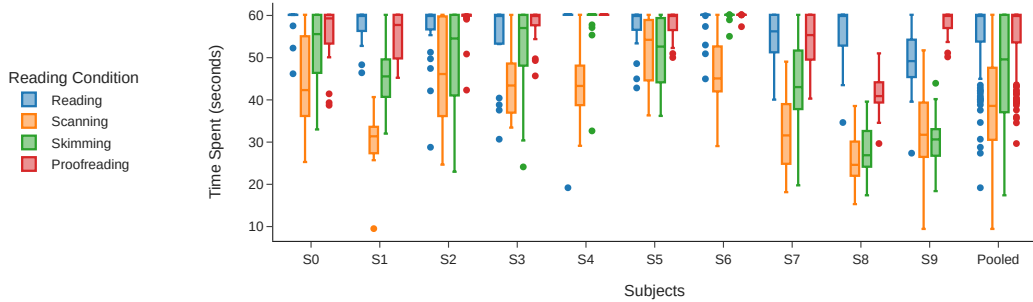


Figure 5.5: Time spent on reading task of each subject ( $S_0$  to  $S_9$ ) and all participants ( $Pooled$ ), grouped by reading condition. The maximum time allowed for each task is 60 seconds.

## 5.4.2 Eye Movement Features and Reading Condition

### 5.4.2.1 Statistical Testing Procedures

In the EDA, I observed that all eye movement features did not pass the one-way ANOVA test's assumption except for two fixation duration features ( $fxdur\_bin\_6$  and  $fxdur\_bin\_8$ ) and three velocity features ( $velo\_v\_bin\_0$ ,  $velo\_v\_bin\_3$  and  $velo\_v\_bin\_4$ ). Therefore, the one-way ANOVA test ( $\alpha = 0.05$ ) was used to test these five features while the remaining features were tested using the Kruskal-Wallis test ( $\alpha = 0.05$ ). There were 175 out of 254 input features that showed significant differences between the reading conditions and were passed to the post-hoc test (either t-test or Conover's test, depending on the preceding significance results were obtained from an ANOVA test or Kruskal-Wallis test, respectively). The significance level was adjusted using Bonferroni Correction to control the Family-Wise Error(FWER) rate at 0.05. Due to the large number of features which made posthoc results difficult to interpret, I have ranked the features based on the rate at which they showed significant differences in reading condition pairs (i.e. number of pairs that the feature showed significant difference divided by the total number of pairs). This was calculated on the subject level in order to provide a more detailed breakdown of the features' performance. Table 5.5

Table 5.5: The top 20 features which were significantly different between reading conditions. The significance level for multiple tests was adjusted using the Bonferroni Correction method. Features were ranked by the number of reading condition pairs that they showed significant differences over the total number of reading condition pairs (16 pairs)

Features	Subjects									
	<i>S0</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>S4</i>	<i>S5</i>	<i>S6</i>	<i>S7</i>	<i>S8</i>	<i>S9</i>
<i>dist_bin_4</i>	0.75	0.75	0.63	0.63	0.63	0.38	0.63	0.63	0.75	0.75
<i>dist_iqr</i>	0.63	0.75	0.63	0.63	0.75	0.50	0.63	0.63	0.75	0.50
<i>dist_h_bin_4</i>	0.75	0.75	0.63	0.50	0.63	0.38	0.63	0.63	0.75	0.63
<i>dist_h_iqr</i>	0.50	0.75	0.63	0.75	0.63	0.50	0.63	0.63	0.75	0.50
<i>velo_h_bin_4</i>	0.75	0.75	0.50	0.63	0.63	0.38	0.63	0.63	0.75	0.63
<i>dist_skewness</i>	0.63	0.75	0.75	0.75	-	0.50	0.63	0.75	0.75	0.63
<i>dist_h_skewness</i>	0.75	0.63	0.75	0.75	-	0.50	0.63	0.63	0.75	0.63
<i>velo_bin_4</i>	0.75	0.75	0.50	0.63	0.38	0.38	0.63	0.63	0.75	0.63
<i>dist_kurtosis</i>	0.63	0.63	0.75	0.75	-	0.50	0.63	0.75	0.75	0.63
<i>dist_h_bin_3</i>	0.75	0.75	0.63	0.63	0.63	-	0.63	0.75	0.63	0.63
<i>velo_h_kurtosis</i>	0.75	0.63	0.63	0.50	0.63	0.50	0.38	0.63	0.63	0.63
<i>velo_h_skewness</i>	0.63	0.63	0.75	0.50	0.63	0.63	0.38	0.50	0.63	0.63
<i>fxdur_std</i>	0.38	0.63	0.38	0.75	0.63	0.38	0.75	0.63	0.50	0.75
<i>dist_h_kurtosis</i>	0.63	0.63	0.75	0.75	-	0.50	0.63	0.50	0.75	0.63
<i>velo_skewness</i>	0.63	0.63	0.75	0.50	0.63	0.50	0.38	0.50	0.63	0.63
<i>velo_bin_5</i>	0.63	0.63	0.63	0.38	0.63	0.38	0.50	0.63	0.75	0.63
<i>dist_bin_5</i>	0.63	0.63	0.63	-	0.50	0.50	0.50	0.63	0.75	0.63
<i>velo_mean</i>	0.75	0.75	0.63	0.50	0.50	-	0.38	0.50	0.75	0.63
<i>dist_h_tr_max</i>	0.75	0.75	0.63	0.63	-	0.38	0.38	0.50	0.75	0.63
<i>velo_kurtosis</i>	0.75	0.63	0.63	0.63	0.38	-	0.38	0.63	0.75	0.63
Mean Score	0.58	0.58	0.58	0.53	0.52	0.45	0.46	0.57	0.62	0.56

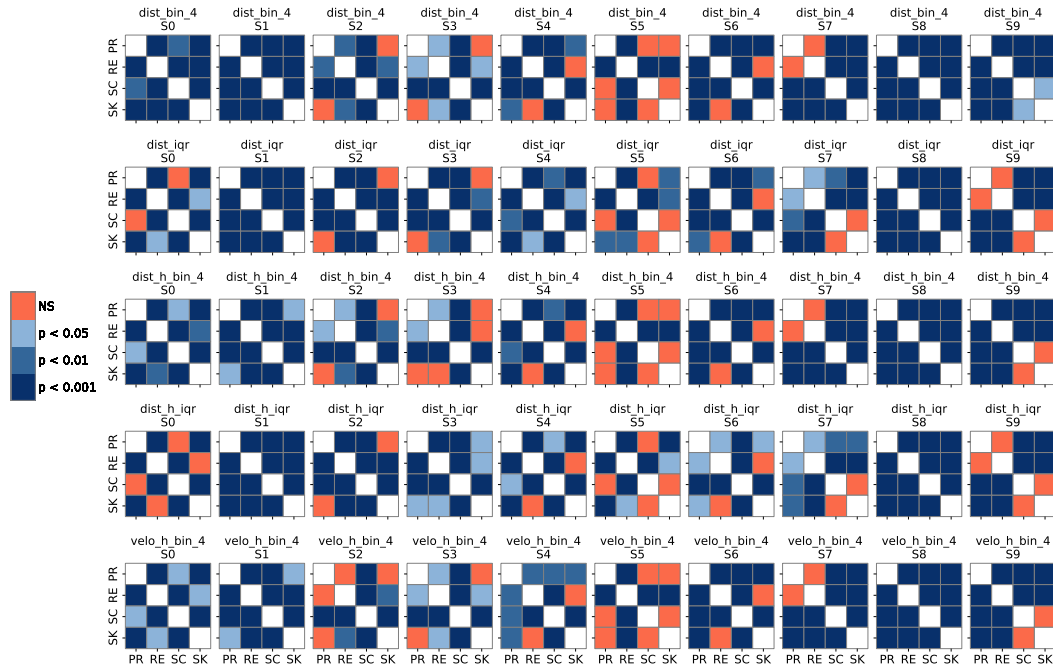


Figure 5.6: Demonstration of posthoc results of top 5 feature from Table 5.5. P-values were adjusted for multiple tests using Bonferroni Correction. The abbreviations for Reading, Skimming, Scanning and Proofreading are RE, SK, SC and PR, respectively.

displayed the top 20 features that showed significant differences in most pairs of reading conditions. It is worth highlighting that velocity and distance features are most sensitive to reading conditions as their statistical features were ranked among the top 20. Moreover, 9 out of these 20 features were derived from the horizontal eye movement only, which suggests that by analysing the speed of left-right eye movement, I can predict, to some extent, the reading condition.

For the subject-level analysis (i.e. SD configuration), I visualised the test results of the 5 highest ranked features (as shown in Figure 5.6) to gain more insights into the pairwise relationship of reading conditions reflected by these features. Although some subjects share similar test results (e.g. subject S1 and S8 one 0.01 significance level; or subject S4 and S6 on all features except *dist\_h\_iqr* at significance level 0.05), there were no common patterns across all subjects. This indicates that the relationship between the features and reading conditions is subject-specific. Among all subjects, I spotted a noticeably low score in subject S5 (with a mean score to

be 0.45). From Figure 5.6, I observed that most features did not show significant differences in Scanning-Skimming and Scanning-Proofreading pairs in subject S5. Referencing back to Figure 5.3, it is also true for subject S5's  $c\_score$  in these two pairs, which suggests that this subject performed the Scanning task differently compared to other subjects.

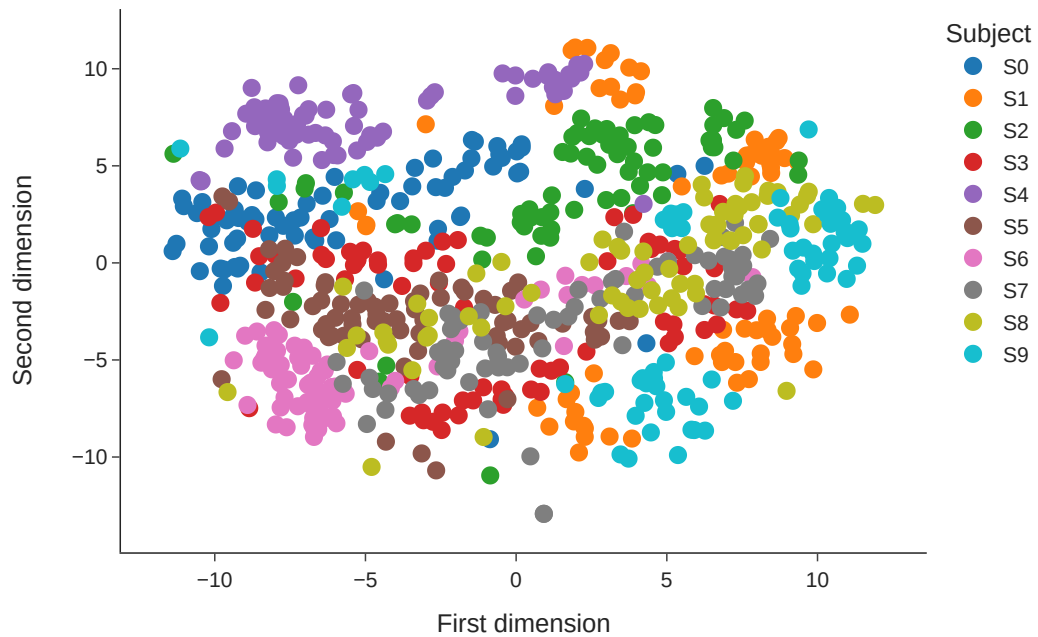


Figure 5.7: A two-dimensional t-SNE [9] visualisation of 254 eye movement features.

#### 5.4.2.2 Machine Learning Approaches

We also approached the problem from a machine learning perspective. A variety of machine learning classifiers were adopted to classify the reading conditions using the same feature set. Table 5.6 illustrated the baseline result in which I observed that the tree-based model achieved high accuracy scores compared to the others since they can capture the complex non-linear relationship between features. In particular, LGBM achieves the highest mean accuracy scores of 0.651 and 0.463 in both GE and SI settings, respectively, while being slightly outperformed by ET in the SD setting

Table 5.6: Reading condition classification results using different combinations of machine learning classifiers and training types on the initial feature set (254 features)\*. The best result for each subject is highlighted in bold. Values in bold and underlined indicate the best result obtained from a classifier for a specific training type (GE, SD and SI).

Model	Type	Mean Accuracy	Subject Breakdown																	
			S0	S1	S2	S3	S4	S5	S6	S7	S8	S9								
<i>LGBM</i>	GE	<b>0.651</b>	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SD	0.726	<b>0.794</b>	0.925	0.731	0.675	0.669	0.600	0.656	0.600	0.656	0.706	0.875	0.631						
	SI	<b>0.463</b>	0.486	0.528	0.486	0.569	0.472	0.542	0.389	0.431	0.431	0.292								
<i>RF</i>	GE	0.614	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SD	0.734	0.775	0.919	0.794	0.606	0.638	0.619	0.656	0.731	<b>0.925</b>	0.675								
	SI	0.442	0.611	0.403	0.500	0.389	0.417	0.458	0.389	0.444	0.444	0.361								
<i>ET</i>	GE	0.612	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SD	<b>0.738</b>	<b>0.794</b>	0.913	<b>0.825</b>	0.619	0.650	<b>0.631</b>	<b>0.663</b>	0.719	<b>0.925</b>	0.644								
	SI	0.431	0.569	0.375	0.472	0.417	0.417	0.431	0.389	0.444	0.444	0.347								
<i>SVM</i>	GE	0.609	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SD	0.736	0.788	<b>0.931</b>	0.806	0.588	<b>0.694</b>	0.625	0.638	0.700	0.881	<b>0.713</b>								
	SI	0.443	0.583	0.458	0.472	0.597	0.306	0.444	0.361	0.500	0.417	0.292								
<i>KNN</i>	GE	0.586	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SD	0.688	0.756	0.875	0.731	0.606	0.631	0.581	<b>0.663</b>	0.594	0.819	0.625								
	SI	0.386	0.528	0.361	0.528	0.361	0.319	0.403	0.306	0.389	0.347	0.319								
<i>LR</i>	GE	0.552	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	SD	0.737	0.763	0.925	0.800	<b>0.700</b>	0.650	0.613	0.625	<b>0.750</b>	0.856	0.688								
	SI	0.440	0.639	0.458	0.486	0.583	0.347	0.431	0.403	0.431	0.306	0.319								

\*The mean accuracy for a dummy classifier is 0.25 for all training types and all subjects

achieving 0.738 compared to 0.726 of LGBM. Besides that, I observed that SD models yielded higher accuracy scores than GE and SI models, which was expected since the models were trained on data from the same subject, thus, allowing these models to generalise better to the unique characteristic of the subject. The low accuracy score of the SI models might be due to the variation of the eye movement feature from subject to subject, which makes it difficult to find a general pattern that can be applied to all. This was indeed reflected in Figure 5.7 in which I projected the high dimensional eye movement features of the reading instances from all subjects into a two-dimensional Euclidean space using the t-SNE [9] method. The visualisation showed that instances from the same subject tend to cluster close to each other, which generates difficulties for the SI models to generalise to these subjects when they are being held out during training.

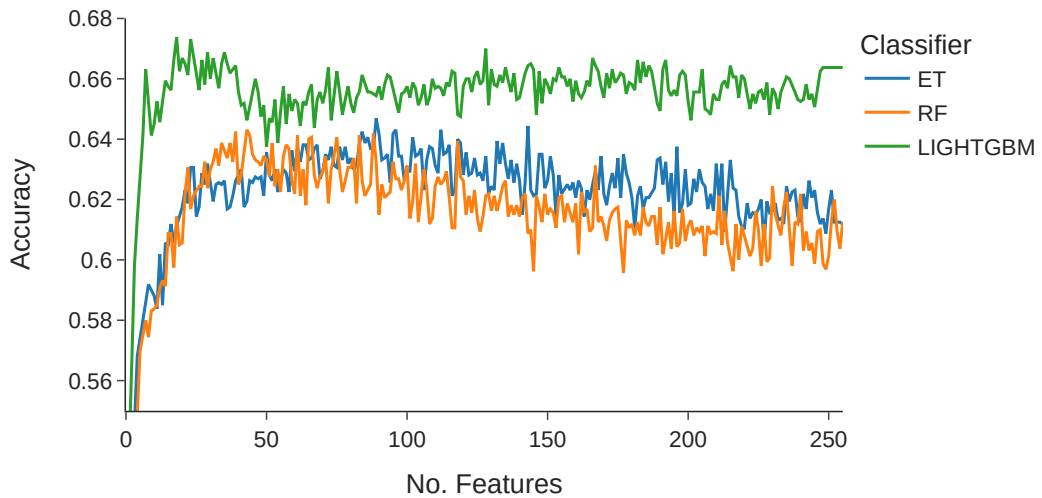


Figure 5.8: Performance of three classifiers (ET, RF and LGBM) in the feature selection process using Recursive Feature Elimination method under the RFE\_GE setting

As outlined in the Method section, three top-performing classifiers (highest average mean accuracy across all training settings) were picked for further analysis, namely LGBM, ET, and RF. A recursive feature elimination (RFE) method was

applied to these models to identify the most important features and the result is shown in Figure 5.8. In RFE\_GE configuration, three models are compared against each other, I found that LGBM peaked at 18-39 features and achieved the highest accuracy of 0.674 when there were 19 features, while ET and RF fluctuated in the accuracy range from 0.630 to 0.646 and showed a slight downward trend as the number of features increased. Since LGBM outperformed the other two models regardless of the number of features, the 19 features at which LGBM achieved the highest accuracy score were selected to form the final feature set to re-train and re-evaluate the three best models (in both GE and SD settings). Furthermore, I also compared the performance of the three models in the RFE\_SD configuration, where the feature selection was performed separately for each subject. The features at which the model achieved the optimal accuracy were used to re-train that model on the subject’s data to obtain the final result. In Table 5.8, I reported the mean accuracy score of the three models when re-trained on the selected features from two different feature selection approaches (RFE\_GE and RFE\_SD).

Table 5.8: Re-training result of the three best models on the selected features using RFE\_GE (from LGBM’s best 19 features) and RFE\_SD (from models’ optimal features) settings.

Model	Feature Selection Setting		Training Setting		Accuracy
	GE	SD	GE	SD	
<i>ET</i>	✓		✓		0.667 ± 0.031
	✓			✓	0.748 ± 0.091
		✓		✓	0.738 ± 0.093
<i>LGBM</i>	✓		✓		0.689 ± 0.037
	✓			✓	0.724 ± 0.090
		✓		✓	0.726 ± 0.103
<i>RF</i>	✓		✓		0.643 ± 0.031
	✓			✓	<b>0.753 ± 0.084</b>
		✓		✓	0.734 ± 0.084

We observed that three models performed better when trained on the SD settings compared to GE settings, which was aligned with the observation in Table



5.6. Moreover, a GE feature selection process followed by an SD re-training yielded a higher accuracy score than the SD feature selection with an SD re-training (except LGBM which was slightly lower by 0.002). This result suggested that the original feature set can be condensed to a smaller set of features that can be used to train a personalised model for each subject, which ultimately avoids the curse of dimensionality and improves the generalisation ability of the model.

Examining the feature contribution to the prediction of the RF model (which achieved the highest accuracy score after the feature selection process), using SHAP [172], I found that the most important features were the mean velocity (*velo\_mean*) and interquartile range of the movement distance (*dist\_iqr*). Considering the top 5 features with the biggest average impact on model output, I found that the *dist\_iqr* is among the top 5 features in 10 out of 10 subjects, while the *velo\_mean* is among the top 5 features in 9 out of 10 subjects. This result is consistent with the findings from the statistical approach in Section 5.4.2.1 as I found that distance and speed of movement features are among the top features having significant differences between reading conditions (more details in Table 5.5). Moreover, the *velo\_mean* and *dist\_iqr* are ranked among the top 20 features in Table 5.5, with the *dist\_iqr* being ranked second. This further confirmed the importance of these two features in the prediction of reading conditions.

### 5.4.3 Eye Movement Features and Reading Comprehension

#### 5.4.3.1 Statistical Testing Procedures

Figure 5.10 displays Spearman’s correlation coefficient between eye movement features and reading comprehension scores. As detailed in Section 5.3.3.3, the figure presents a feature set that is the union of the top 10 features with the strongest correlation with *c\_score* from each subject, sorted by frequency (i.e. the number of subjects that the feature was selected by taking the top-10). The figure reveals that the regression rate (*regr\_rate*) and the distance interquartile range (*dist\_iqr*) had the highest frequency of 7 and were negatively correlated with

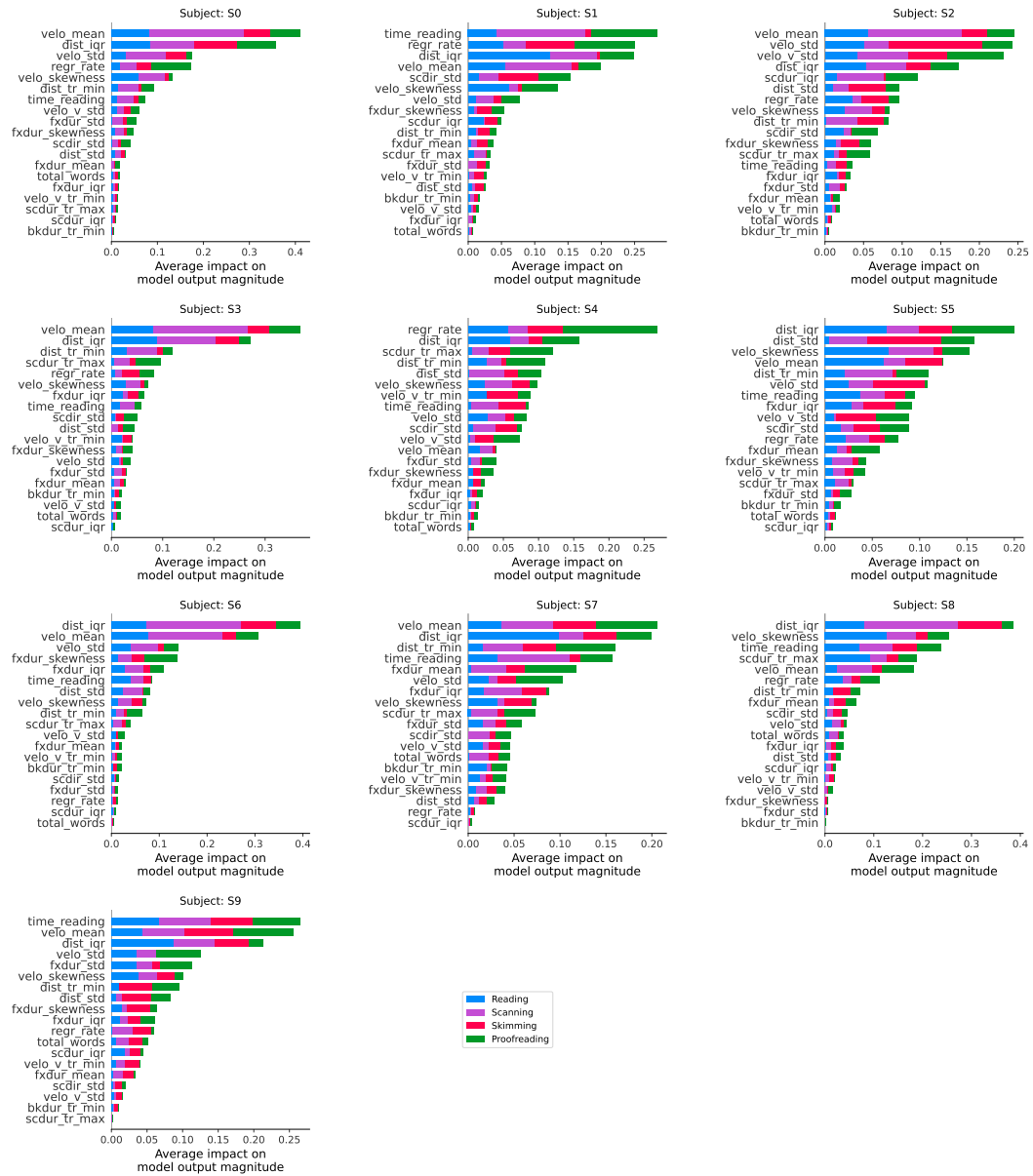


Figure 5.9: Feature contributions to the best classifier (RF, as shown in Table 5.8) results, explained by the SHAP method. The features are sorted by their importance in the model. Since the classifier was trained on the SD setting, there are 10 corresponding SHAP plots for each subject.

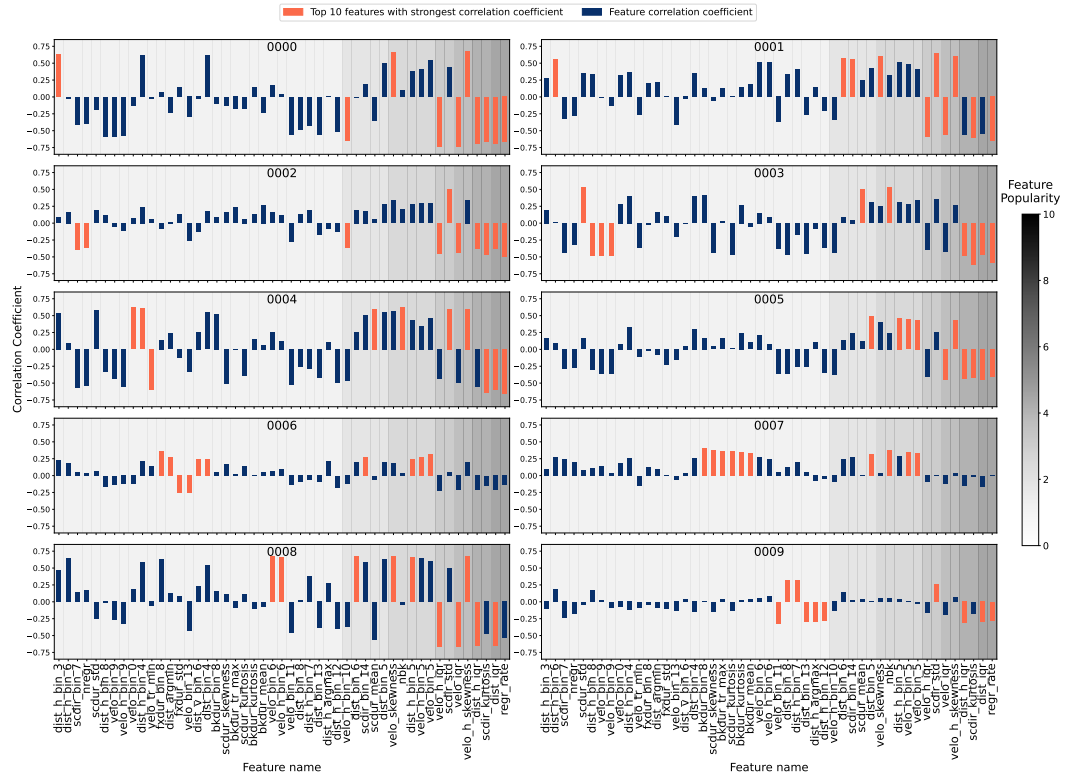


Figure 5.10: Spearman’s rank correlation coefficient ( $\rho$ ) between the eye movement features and the reading comprehension score. The displayed feature set is the union of the top 10 features with the highest  $\rho$  from each participant. The top 10 features of each participant is highlighted in orange. The features are sorted in ascending order of their commonality across participants, which is indicated by the gradient of the background.

$c\_score$ . This suggests that there is a common pattern in the eye movement features that are associated with reading comprehension across subjects. However, this pattern was not observed in subjects S6 and S7. While other subjects had top features with frequencies ranging from 4 to 7, subjects S6 and S7 had top features with frequencies ranging only from 1 to 3, with 6 features having a frequency of 1 (meaning that they were unique to the subject only). This indicates that these subjects may have used a different strategy to carry out the reading tasks.

Furthermore, I found that features with a frequency of four or more were all features obtained from *statistical encoding* rather than *histogram encoding*. This observation implies that the feature value range, as captured by histogram bins, might not play a significant role in determining the reading comprehension score.

To verify this, I experimented with the machine learning approach to investigate whether the presence of histogram features would affect the performance of the models.

#### **5.4.3.2 Machine Learning Approaches**

Table 5.9 displayed results of the machine learning approaches described in Section 5.3.3.3. Tree-based models showed their superiority over others in both GE and SD settings, with ET achieving the highest mean correlation score of 0.595 in the GE setting and the highest mean correlation score of 0.500 in the SD setting. This was followed by BR which achieved a slightly lower correlation score in the GE setting (0.569) but had a nearly identical correlation score as ET in the SD setting (0.499). Considering the subject breakdown in Table 5.9, I observed that most of the highest and second highest correlation scores (on each subject) were achieved by ET and BR. Moreover, these high scores were observed when the histogram features were excluded from the training data, which confirmed that the histogram features were not as important as the statistical features in determining the reading comprehension score. In addition, it can be seen that the mean correlation scores of the models trained on the SD setting were lower than those trained on the GE setting, which was opposite to the insight obtained from the reading condition classification task. However, the subject breakdown showed that the previous conclusion still holds since the low correlation scores in S6, S7 and S9 were the main reason for the low mean correlation scores in the SD setting. EDA has previously suggested that these subjects may have used a different strategy to carry out the reading tasks, which explains why the model failed to predict these subjects' comprehension scores.

Taking the top-3 best-performing regressors (i.e. ET, BR and RF), I conducted a feature selection experiment to investigate whether the performance of the models could be further improved by selecting a subset of features from the original feature set using the Recursive Feature Elimination (RFE) method. Figure 5.11 shows

Table 5.9: Reading comprehension estimation results using different combinations of machine learning classifiers and training types. The values are mean Spearman’s rank correlation coefficients ( $\rho$ ) between predicted comprehension scores and true comprehension scores. *Hist Feats* means the utilisation of histogram features. The highest mean  $\rho$  scores for each subject are highlighted in **bold**, while the second highest values are underlined. For *Mean Scores* column, the highest and second highest values are highlighted for each training type.

Model	Type		Hist Feats	Mean Scores	Subject Breakdown													
	GE	SD			S0	S1	S2	S3	S4	S5	S6	S7	S8	S9				
ET	✓		✓	0.588	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	0.472	0.768	0.652	0.430	0.554	0.694	0.522	0.249	0.079	0.586	0.186	-	-	-	-
	✓			<b>0.595</b>	-	-	-	-	-	-	-	-	-	-	-	-	-	-
BR				<b>0.500</b>	0.780	<b>0.731</b>	<b>0.475</b>	<u>0.621</u>	<u>0.713</u>	0.404	0.310	0.188	<b>0.663</b>	0.115	-	-	-	-
	✓		✓	0.569	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	0.407	0.660	0.661	0.354	0.414	0.678	0.351	0.137	<b>0.221</b>	0.550	0.046	-	-	-	-
RF				0.544	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	<u>0.499</u>	0.733	0.698	0.387	<b>0.673</b>	0.707	0.443	<b>0.381</b>	<u>0.209</u>	<u>0.642</u>	0.120	-	-	-	-
	✓		✓	0.570	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GBR		✓	✓	0.483	0.785	0.620	0.379	0.568	0.712	<b>0.595</b>	0.300	0.101	0.570	<u>0.203</u>	-	-	-	-
	✓			0.573	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	0.483	<u>0.786</u>	0.675	<u>0.467</u>	0.585	0.694	0.390	0.329	0.170	0.633	0.103	-	-	-	-
ADA				0.544	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	0.428	0.747	0.585	0.344	0.446	0.700	0.451	0.254	0.050	0.538	0.163	-	-	-	-
	✓			0.537	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LGBM		✓	✓	0.450	0.754	0.652	0.424	0.519	<b>0.714</b>	0.335	<u>0.339</u>	0.156	0.558	0.050	-	-	-	-
	✓		✓	0.561	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	0.476	0.762	0.622	0.370	0.526	0.691	0.527	0.306	0.200	0.537	<b>0.214</b>	-	-	-	-
LGBM				0.545	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	0.469	<b>0.792</b>	<u>0.702</u>	0.446	0.562	0.678	0.372	0.323	0.132	0.595	0.086	-	-	-	-
	✓		✓	0.546	-	-	-	-	-	-	-	-	-	-	-	-	-	-
LGBM		✓	✓	0.430	0.708	0.619	0.390	0.490	0.696	<u>0.589</u>	0.175	0.121	0.545	-0.03	-	-	-	-
	✓			0.535	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		✓	✓	0.427	0.758	0.660	0.349	0.503	0.665	0.361	0.273	0.030	0.614	0.057	-	-	-	-

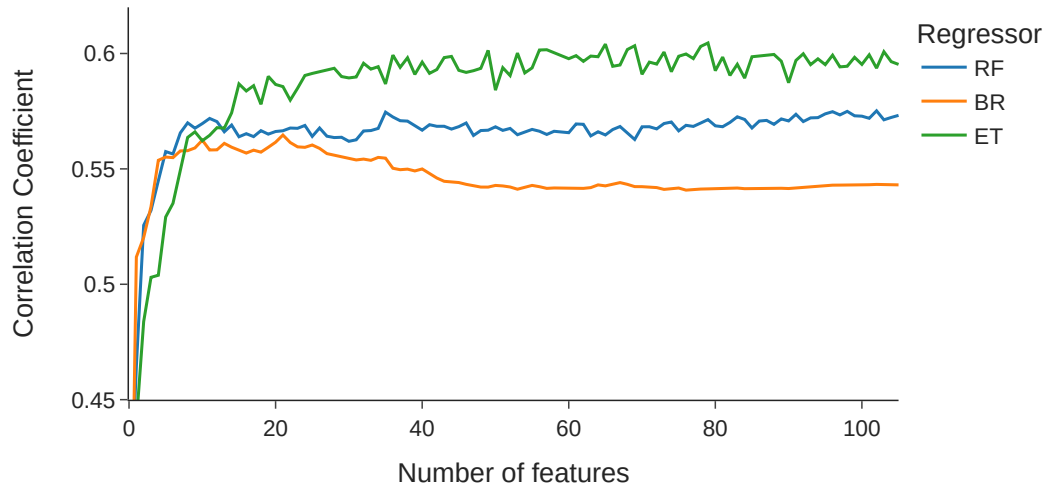


Figure 5.11: Feature Selection using Recursive Feature Elimination.

the mean correlation scores of the models trained on the GE setting with different numbers of features selected by RFE. ET achieved the highest mean correlation score when the number of features surpassed 16. RF shares a similar trend with ET, but its mean correlation score was slightly lower than ET. BR, on the other hand, witnessed a decrease in the mean correlation score when the number of features surpassed 20. The spot where ET peaked was at 80 features and these features were selected for re-training the top-3 models. In Table 5.11, I reported the re-training results in which I observed that the ET achieved the highest mean correlation score of 0.599, which was slightly higher than its original mean correlation score of 0.595 when trained on a full feature set. To determine whether there was a statistically significant mean difference between the correlation scores of ET estimator pre- and post-feature-selection, I generated 100 stratified shuffle split iterations for each setting and performed a paired t-test on the mean correlation scores. The assumption of normality was not violated, as assessed by Shapiro-Wilk's test ( $p = 0.265$ ). We found that the mean correlation score of ET estimator post-feature-selection ( $0.608 \pm 0.042$ ) was statistically significantly higher than its mean correlation score pre-feature-selection ( $0.600 \pm 0.044$ ) by 0.008 (95% *CI*, 0.005 to 0.011),  $t(99) = 5.388$ ,  $p < 0.0001$ .

Table 5.11: Re-training result of the three best models using RFE\_GE and RFE\_SD setting

Model	Feature Selection Setting		Training Setting		Correlation Coefficient
	<i>GE</i>	<i>SD</i>	<i>GE</i>	<i>SD</i>	
<i>BR</i>	✓		✓		$0.542 \pm 0.050$
	✓			✓	$0.505 \pm 0.199$
		✓		✓	$0.505 \pm 0.199$
<i>ET</i>	✓		✓		<b><math>0.599 \pm 0.048</math></b>
	✓			✓	$0.502 \pm 0.153$
		✓		✓	$0.502 \pm 0.153$
<i>RF</i>	✓		✓		$0.575 \pm 0.052$
	✓			✓	$0.499 \pm 0.165$
		✓		✓	$0.499 \pm 0.165$

Based on the results from Section 5.3.3.1 in which I observed that there was a relationship between reading condition and reading comprehension score, I hypothesized that the reading condition could be used as an additional feature to improve the performance of the models. The reading condition was obtained by incorporating the model trained in Section 5.3.3.2 to predict each reading sample. Next, one-hot encoding was applied to generate the reading condition feature, which was then added to the previously selected feature set to re-train the ET model. Table 5.12 shows the re-training results in which I observed a statistically significant increase in mean correlation score by 0.089 (95% *CI*, 0.082 to 0.096) from  $0.608 \pm 0.042$  to  $0.697 \pm 0.036$ ,  $t(99) = 24.760$ ,  $p < 0.0001$ . In an ideal scenario where the reading condition classification model was perfect, I could expect the model to predict the comprehension score reaching a mean correlation up to  $0.708 \pm 0.033$ .

Thus far, my investigation has revealed the relationship between eye movement features, reading conditions, and reading comprehension scores, demonstrating the predictability of reading comprehension levels based on eye movement features. In my subsequent analysis, I will shift my focus to a different prediction target: the

Table 5.12: Comprehension prediction result of the ET model when integrating with identification of reading condition (*none*: no information about the condition, *predicted*: predicted condition obtained from my best model obtained in Section 5.3.3.2, *actual*: actual condition labels obtained from the dataset). The table shows the mean  $\rho$  scores on the validation and held-out test set, under two prediction targets: *c\_score* calculated from MCQs and *se\_score* obtained from participant’s subjective judgement of their own understanding.

Target	Condition	Correlation Coefficient	
		<i>Validation</i>	<i>Test</i>
<i>c_score</i>	none	0.608 ± 0.042	0.564 ± 0.014
	predicted	0.697 ± 0.036	0.614 ± 0.012
	actual	0.708 ± 0.033	0.693 ± 0.011
<i>se_score</i>	none	0.684 ± 0.040	0.608 ± 0.011
	predicted	<b>0.785 ± 0.026</b>	<b>0.689 ± 0.008</b>
	actual	0.799 ± 0.024	0.772 ± 0.008

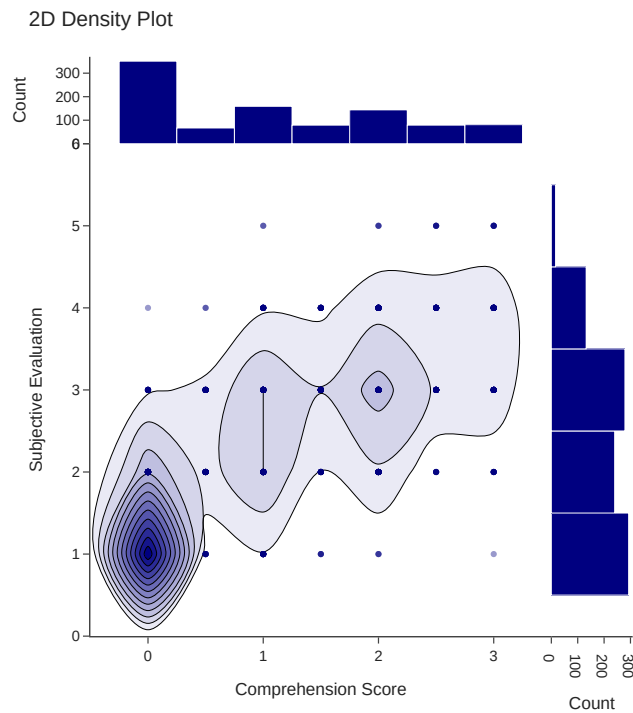


Figure 5.12: Monotonic relationship between the comprehension score and the subjective evaluation score.



subjective evaluation score (*se\_score*). As previously mentioned in Section 5.3.1, this score was collected from participants after they interacted with a text and serves as an alternative measure of reading comprehension level, distinct from the reading comprehension test (the MCQs). Preliminary analysis, visualised in Figure 5.12, revealed a monotonic relationship between *c\_score* and *se\_score*, further substantiated by a statistically significant and strong positive correlation using Spearman’s rank-order correlation ( $r_s(718) = 0.740, p < 0.0001$ ). Therefore, I anticipated that the model would achieve similar performance in predicting *se\_score* as it did with *c\_score*. Notably, my findings remained consistent, with the lowest correlation score  $0.684 \pm 0.040$  obtained when using only eye movement features as input, which improves to  $0.785 \pm 0.026$  when the predicted reading condition is added as an additional feature, and further increases to  $0.799 \pm 0.024$  when the ground-truth reading condition is employed. Interestingly, the mean correlation scores on *se\_score* surpass those on defined *c\_score*, indicating that the model performed better on *se\_score* than on *c\_score*, despite having the same set of eye movement features as input. This suggested that *se\_score* could reflect participants’ reading comprehension levels better than *c\_score*, which can be attributed to the fact that *se\_score* was formed by participants’ internal judgment of understanding, as guided by my instructions on scoring (see Section 5.3.1), while *c\_score* was determined externally by the correctness of participants MCQ answers. In addition to that, the time constraints of the experiment restricted us to only include three MCQs per text, which may not fully capture participants’ reading comprehension levels. Nevertheless, Table 5.12 shows that the correlation score gap between reading comprehension scores and subjective evaluation scores is not substantial (0.091, in the setting where the reading condition is derived from ground truth), with a 95% confidence interval of [0.084, 0.098],  $t(99) = 25.758, p < 0.0001$ . This suggests that reading comprehension scores remain a valid measure and can be enhanced in future studies by exploring alternative approaches to constructing them.

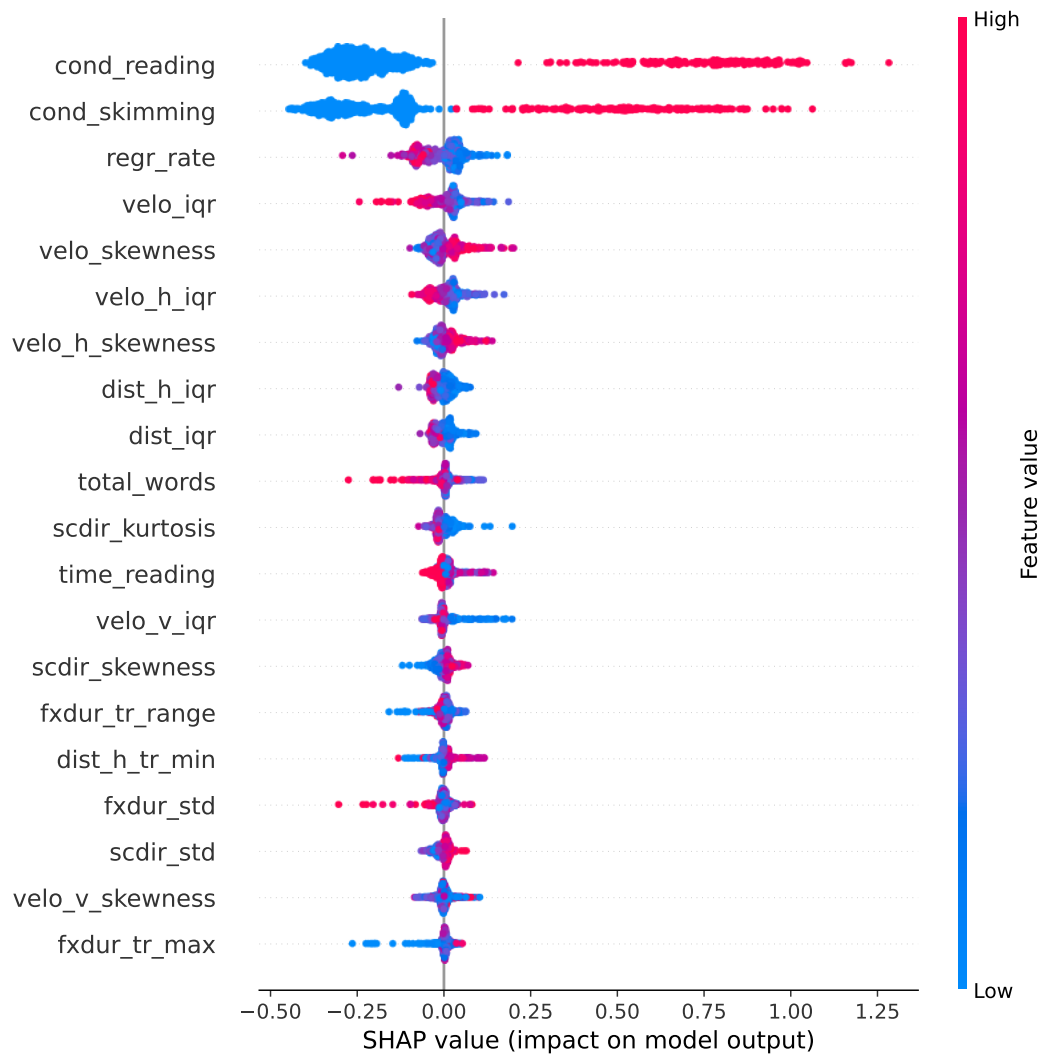


Figure 5.13: SHAP interpretation of the ET model trained on *c\_score* with the predicted reading condition as an additional feature. Features are ordered by their importance. The colour represents the value of the feature where red is the highest and blue is the lowest. A positive SHAP value means that the feature contributes to a higher prediction score, while a negative SHAP value means that the feature contributes to a lower prediction score.

To gain insights into the factors influencing the model's predictions of comprehension levels, I conducted a feature importance analysis using the ET model trained on the *c\_score* target with the predicted reading condition as an additional feature. The SHAP values of the top 20 most important features are visualised in Figure 5.13. Notably, the reading condition, referred to as *Reading* (abbr. *cond\_0*), and *Skimming* (abbr. *cond\_2*) were found to be the most influential factors impacting the model's decision. Instances where the reading sample was classified as either *Reading* or *Skimming* resulted in higher predicted comprehension scores. These findings align with my earlier observations in Section 5.4.1, where I noted that reading and skimming conditions were associated with higher comprehension scores compared to scanning and proofreading.

Furthermore, several eye movement features that exhibited high correlations with comprehension scores during my exploratory data analysis (EDA), such as *regr\_rate*, *dist\_iqr*, *sdir\_kurtosis*, *dist\_h\_iqr*, *velo\_h\_skewness*, and *velo\_iqr*, were identified as among the most important features for the model's predictions. A higher regression rate, indicating instances of re-reading due to missed or misunderstood words or sentences, was found to be associated with lower comprehension levels. Similarly, a narrow velocity range (*velo\_iqr*) was linked to higher comprehension scores, suggesting that a consistent reading speed correlates with better comprehension compared to fluctuating reading speeds.

## 5.5 Chapter summary

In conclusion, this study aimed to advance the estimation of reading comprehension in real-world scenarios by leveraging eye movement measures and reading conditions. By integrating the identification of reading conditions into the utilisation of eye movement measures, I have demonstrated improved performance in predicting reading comprehension levels. My findings highlight the intricate relationship between eye movement measures, reading conditions, and reading

comprehension.

Through a series of statistical tests and machine learning approaches, I achieved an average classification accuracy of 75.3% for reading conditions in the subject-dependent setting and 68.9% in the general setting. These promising outcomes were obtained by employing a feature selection method that condensed the initial set of 254 input features down to a concise set of 19 features using the LGBM model. Since the feature selection was conducted under the general setting, the selected features provide generalisability across participants and serve as valuable features for future studies on classifying reading conditions. Furthermore, I discovered an interesting trend regarding the higher accuracy observed in the subject-dependent setting compared to the general setting. This can be attributed to the distinctive nature of eye movement features for each individual, even though there is a shared feature set among participants obtained from the feature selection process. Consequently, the model's applicability to other participants was somewhat limited due to the personalised nature of eye movement characteristics.

Additionally, my analysis revealed that reading and skimming styles are associated with higher comprehension levels while scanning and proofreading styles are linked to lower comprehension levels. Hence, applying the reading conditions classification model to predict reading style before estimating reading comprehension resulted in an 8.9% improvement in the correlation coefficient compared to the model trained with reading styles alone (from 0.608 to 0.697). Furthermore, when the true reading condition labels were utilised to train the model, the correlation coefficient between the predicted and actual comprehension levels increased to 0.708, which is the case of obtaining a perfect reading condition classification model. Interestingly, my model performed better when predicting subjective understanding (using participants' reports on their comprehension levels) compared to objective understanding (using comprehension scores obtained from the comprehension test). We obtained a correlation coefficient of 0.785 when predicting subjective understanding using reading conditions labels from the

classification model and 0.799 using true reading conditions labels, which is an 8.8% and 9.0% improvement respectively compared to predicting objective understanding. This discrepancy suggests a gap between participants' subjective understanding and their actual comprehension levels, providing a new research direction to explore the relationship between these two measures and develop a more comprehensive approach to measuring reading comprehension.

Overall, Research Question 2 is addressed as I showed that reading comprehension levels can be estimated from eye movement measures effectively with the predicted reading comprehension levels having a strong positive correlation with the actual comprehension levels ( $\rho = 0.785$  when predicting subjective understanding and  $\rho = 0.697$  when predicting objective understanding).

## Chapter 6

# Longitudinal Evaluation of Reading Comprehension Estimation Model

### 6.1 Introduction

In this chapter, I address Research Question 3, which is: **How robust is the reading comprehension estimation model when applied to longitudinal reading data?**

Since my ultimate goal in this dissertation is to obtain a reading comprehension estimation model that is effective in capturing a human's comprehension during their daily reading on a computer, which is a longitudinal process, it is important to investigate the robustness of models when applied to longitudinal reading data, hence the formulation of Research Question 3. To address this research question, I adopt a similar research process to Chapter 5, in which I collect a longitudinal reading dataset, then extract eye movement features from the eye-tracking data, and perform machine learning analyses on two tasks, which are reading condition classification and reading comprehension estimation. I also aim to confirm the finding in Chapter 5, which stated that when incorporating the predicted reading condition as extra features to eye movement features, the reading comprehension estimation model can achieve a better performance.

The experimental results show that both reading condition classification and

reading comprehension estimation models can achieve a good and stable performance when predicting future reading data. Notably, I show that the more training data these models have, the better performance they can achieve. I was also able to confirm the findings in the previous chapter (Chapter 5), which show that the reading condition classification model can be used to improve the reading comprehension estimation model. This results in an overall Spearman’s rank correlation of 0.594 and 0.516 between the predicted value and the true label on validation and test sets, respectively.

The remainder of this chapter is organised as follows. Section 6.2 details the data collection process, which is followed by Section 6.3 where I describe the methods used to analyse this longitudinal reading dataset. Experimental results are presented in Section 6.4, in which I also discuss the findings and implications of this chapter. Finally, Section 6.5 concludes this chapter.

## **6.2 Data Collection**

This section outlines the process of creating the longitudinal reading dataset. While the procedure largely mirrors that of Chapter 5, modifications were made to investigate the longitudinal aspect of reading. A detailed list of these changes is listed at the end of this section to provide a clear description of the differences between the two datasets. The process of data gathering received approval from Dublin City University’s Research Ethics Committee under the reference number DCUREC/2021/147. In selecting participants, I adhered to the following inclusion criteria: (1) is not the participant in RCIRv1 dataset, (2) has no history of reading difficulties, (3) has normal or corrected-to-normal vision, (4) be able to maintain a relatively steady head position for the duration of the experiment, (5) comply with the instructions provided during the experiment, and (6) commit to finishing all experiment sessions, which spans 6 non-consecutive days.

We recruited a total of  $N = 13$  participants: 5 males and 8 females. Among them,

6 individuals (*S0*, *S1*, *S2*, *S5*, *S11*, and *S12*) were native English speakers, while the remainder were non-native speakers. These participants either had natural vision or wore corrective lenses to achieve normal vision. The study involved 6 sessions, each of which was conducted on a different day. There was a 1-3 days gap between consecutive sessions. Participants were required to complete all 6 sessions, failing to do so would result in their data being excluded from the dataset. The participants received instructions on how to complete the reading task on the first day (first session). The following sessions were conducted directly without any reminder of the instructions or additional training.

In each session, participants read 24 passages (with average length of 346 words), spanning across 4 different reading conditions (reading, scanning, skimming and proofreading). The passages were sampled so that they are different from the passages in the RCIRv1 dataset and the set of passages is different each day (i.e., each session). There are different maximum times allowed for performing each reading condition, specifically, 60 seconds for scanning, 45 seconds for skimming, 120 seconds for reading, and 90 seconds for proofreading. The introduction of these limits was based on the observation on time spent on reading tasks in the RCIRv1 dataset as shown in Figure 5.5, where most of the participants failed to finish reading and proofreading tasks within the 60-second time limit. Moreover, this adjustment is also in line with participants' feedback in the previous study in which they felt that the time limit was too short for reading and proofreading tasks. Even though the information about the time limit is kept hidden from the participant so that different behaviours can be induced for different reading tasks, there are two participants reported that they had a grasp of the time limit being the same for all tasks, but they were not sure about it and did not take advantage of it.

Upon finishing reading a passage, participants were asked to give a subjective judgement of their own understanding (from 1 to 5, with 1 being the lowest comprehension level), and answer 3 multiple-choice questions. There was a short break of up to 10 minutes after reading the first 12 passages for participants to rest



their eyes. The total duration to complete a session was approximately 1 hour (and 1.5 hours for the first session, due to the training session).

Similar to the RCIRv1 dataset (Section 5.2), passages from the RACE dataset [166] were utilised as reading material, with the exception that there was no topic modelling process (i.e. the passages were not clustered into topics). The MCQs were also sampled using the same method as Chapter 5. To optimise readability, I used a font size of 28.5pt and line spacing of 1, which was informed by findings from Luz et al. [173] which suggested using a font size of 18pt or larger and line spacing of 1 or larger. We also used yellow as the background colour and set the text colour to black, which was recommended by [174]. Passages were presented on a 24-inch Phillips LCD monitor (model 240V5QDAB/00) with  $1920 \times 1080$  resolution and controlled by a Dell Optiplex 5060 PC powered by the Windows 10 operating system. Experimental participants sat approximately 60cm from the screen and no chin rest was used. Eye movements were captured using the Gazepoint GP3 HD Eye Tracking device<sup>1</sup> with a sampling rate of  $150Hz$  (one sample per 6.67 milliseconds). The data gathering process was driven by software written in Python using Psychopy [169].

Compared to the RCIRv1 dataset in Chapter 5, these are the key differences:

- The data collection happened over several days (6 days in total), with a gap of 1-3 days between consecutive days.
- The participants who took part in this study did not take part in the previous study in Chapter 5 (i.e. they are not the same participants in the RCIRv1 dataset).
- There are different time limits for each reading condition, compared to the fixed time limit of 60 seconds in RCIRv1.
- There are no topic modelling and all participants read the same set of passages.
- Background color, font size and line spacing are controlled to increase readability.

---

<sup>1</sup><https://www.gazept.com/product/gp3hd/>

## **6.3 Methodology**

This chapter aims to explore the longitudinal aspect of reading comprehension through eye movements. The preceding chapter laid the groundwork by examining the complex relationship between reading conditions, reading comprehension and eye movement and developing models for reading condition classification and reading comprehension prediction. However, these models were trained and evaluated on the RCIRv1 dataset, which was collected in a single day for each participant. With the constructed RCIRv2 dataset, I focus on analysing the temporal robustness of the proposed method in Chapter 5. In addition, I also aim to explore eye movement measures that characterise human reading behaviour over multiple sessions, in order to identify stable (between-session) eye movement features and examine whether they have an impact on predicting reading comprehension. In the subsequent sections, I will first describe a revised feature extraction process. Then, I will present the method used to investigate the reading condition classification and reading comprehension prediction over multiple sessions. Finally, I will conduct a separate statistical testing procedure to examine the stability of eye movement features over multiple sessions and compare this with the features that have significant contributions to Machine Learning models' prediction performance.

### **6.3.1 Data pre-processing and Feature Extraction**

Similar to the procedure described in Section 5.3.2, the oculomotor events (fixations, saccades and blinks) were extracted from the raw eye-tracking data using the built-in algorithm shipped with the manufacturer's software. The same set of additional features were also derived from the detected ocular events, including moving distances, velocity, angle of movement, and rate of regressive movement. However, there is a difference in some features for cases where they are normalised by the total time spent on reading. The normalisation step was

introduced to account for the introduction of different time limits for each reading condition in this longitudinal dataset. Consequently, the total time spent on reading is no longer being used as a feature in this study. Table 6.1 displays a summary of the features used in this study and indicates which features were calculated differently compared to the previous study.

Table 6.1: Summary of ocular events and features used in the analysis of RCIRv2 dataset. Value in *Diff.* column indicates whether the feature was calculated differently compared to the previous study. These features were normalised by the total time spent on reading.

Name	Description	Type	Diff.
<i>nfx_norm</i>	Normalised number of fixations	scalar	Yes
<i>nbk_norm</i>	Normalised number of blinks	scalar	Yes
<i>fxdur_norm</i>	Normalised fixation durations	sequence	Yes
<i>scdur_norm</i>	Normalised saccade durations	sequence	Yes
<i>smdir</i>	Saccade directions (angles)	sequence	No
<i>bkdur_norm</i>	Normalised blink durations	sequence	Yes
<i>dist</i>	L2 distances between two consecutive fixations (i.e. one saccade)	sequence	No
<i>dist_v</i>	L1 distance between two consecutive fixations on vertical axis	sequence	No
<i>dist_h</i>	L1 distance between two consecutive fixations on horizontal axis	sequence	No
<i>velo</i>	Velocity of movement	sequence	No
<i>velo_v</i>	Velocity of movement on vertical axis	sequence	No
<i>velo_h</i>	Velocity of movement on horizontal axis	sequence	No
<i>nregr_norm</i>	Normalised number of regressions	scalar	Yes
<i>regr_rate</i>	Ratio between the number of regressions and number of fixations	scalar	No

The features that are sequences (as indicated in Table 6.1) were encoded into scalar values using *statistical encoding* and *histogram encoding* methods, as described in Section 5.3.2.

### 6.3.2 Machine Learning Analysis

Building upon the findings and machine learning pipelines developed in Chapter 5, this chapter investigates the temporal robustness of the proposed method under two ML tasks: reading condition classification and reading comprehension prediction.

For both tasks, I approach the problem by training and evaluating the models on different subsets of reading sessions in the RCIRv2 dataset. Let  $m$  to be the total number of sessions. Beginning with *session 0*, I train the models on the first  $n$  sessions (i.e. *session 0* to *session*  $(n - 1)^{th}$ ), evaluate them on the  $n^{th}$  session, the remainders (*session*  $(n + 1)^{th}$  to *session*  $(m - 1)^{th}$ ) are used as the test set. The process is repeated for  $n$  from 1 to  $m - 2$  (to guarantee that there is at least one session for validation and one session for testing).

Since I have shortlisted the best-performing machine learning models in Chapter 5 which are Light Gradient Boosting Machine (LGBM), Random Forest (RF) and Extra Trees (ET), these will be the main models to be examined in this study. Furthermore, the analysis is also conducted on two training configurations as in the preceding chapter, including General (GE) training and Subject Dependent (SE) training, to further compare how the models perform in these settings over multiple sessions. After identifying the best-performing baseline model, a hyperparameter tuning process is conducted to find the best set of hyperparameters for them. To provide a rationale for the hyperparameter decision, I also conducted a sensitivity analysis to examine the impact of the hyperparameters on the model's performance.

For the reading comprehension prediction task, there are two extra experiments that are added to the aforementioned procedure. The first experiment is to confirm the finding in Chapter 5 that the subjective evaluation *se\_score* (i.e. the self-reported reading comprehension score) has a strong correlation with the objective comprehension score *c\_score* (i.e. the score calculated based on participants' answers to MCQs) and can be used as an alternative to *c\_score* in the prediction task. The second one is to examine the integration of reading conditions as additional features to train the prediction model. As a result, the best baseline classifier and the best-tuned classifier from the reading condition classification task will be employed for predicting reading comprehension scores.

### **6.3.3 Eye Movement Features Inspection**

This section investigates the stability of eye movement features over multiple sessions and compares this with the features that have significant contributions to the machine learning models' prediction performance to check whether the features that are important to the models are also stable over time. To identify the stable features, I conduct a statistical testing procedure to compare the features' distributions over multiple sessions. First, the reading samples are grouped by participant. Next, a Shapiro-Wilk test is used to check whether the features are normally distributed. Then Bartlett's test is used to check whether the variances of the features are equal. If the features are normally distributed and have equal variances, the one-way ANOVA test is used to compare the features' means over multiple sessions, otherwise, the Friedman test is used. The significance level is set to 0.05 and the Bonferroni correction is applied to ensure that the family-wise error rate is controlled at 0.05 for multiple comparisons. The procedure is repeated for each feature. After that, the eye movement features have no significant difference over multiple sessions and are identified as stable features. Finally, the stable features are compared with the features that have significant contributions to the models' prediction performance (obtained by using SHAP method) to check whether the features that are important to the models are also stable over time.

## **6.4 Results and Discussion**

### **6.4.1 Reading Condition Classification**

Table 6.2 shows the baseline results of the condition classification task using the General training configuration. It can be seen that for all models, the more sessions that are used for training, the better the performance of the models. The best-performing model is LGBM trained on the first 4 sessions, reaching the highest mean accuracy of 0.651 with (an accuracy of 0.705 on the fifth session and 0.596

Table 6.2: Baseline results of the condition classification task in the GE training configuration. The T0 to T5 are abbreviations for *session 0* to *session 5*. Each row displays the configuration for training, validation and testing sessions. The white cells with letter *t* indicate that these sessions are used for training, while cells with symbol - represent ignored sessions. The session which is immediately after the training sessions (i.e., the cells in white which contain a real value) displays the classification accuracy when evaluating on the validation session. The remaining cells in grey color are accuracy scores when evaluating on testing sessions, which are displayed to provide an insight into the model’s performance over time.

Classifier	Sessions						Mean Accuracy
	T0	T1	T2	T3	T4	T5	
ET	<i>t</i>	<b>0.516</b>	0.497	0.558	0.587	0.519	0.535
RF	<i>t</i>	0.503	0.542	0.561	0.567	0.529	0.540
LGBM	<i>t</i>	0.420	0.330	0.487	0.417	0.429	0.417
ET	-	<i>t</i>	0.506	0.545	0.554	0.545	0.538
RF	-	<i>t</i>	0.516	0.545	0.554	0.548	0.541
LGBM	-	<i>t</i>	0.282	0.292	0.487	0.510	0.393
ET	<i>t</i>	<i>t</i>	0.532	0.603	0.599	0.558	0.573
RF	<i>t</i>	<i>t</i>	<b>0.542</b>	0.625	0.583	0.574	0.581
LGBM	<i>t</i>	<i>t</i>	0.353	0.439	0.535	0.622	0.487
ET	-	<i>t</i>	<i>t</i>	0.545	0.577	0.513	0.545
RF	-	<i>t</i>	<i>t</i>	0.545	0.596	0.532	0.558
LGBM	-	<i>t</i>	<i>t</i>	0.436	0.590	0.465	0.497
ET	<i>t</i>	<i>t</i>	<i>t</i>	0.574	0.580	0.545	0.566
RF	<i>t</i>	<i>t</i>	<i>t</i>	<b>0.590</b>	0.615	0.551	0.585
LGBM	<i>t</i>	<i>t</i>	<i>t</i>	0.465	0.631	0.535	0.544
ET	-	<i>t</i>	<i>t</i>	<i>t</i>	0.606	0.571	0.588
RF	-	<i>t</i>	<i>t</i>	<i>t</i>	0.628	0.554	0.591
LGBM	-	<i>t</i>	<i>t</i>	<i>t</i>	0.654	0.599	0.627
ET	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>	0.660	0.580	0.620
RF	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>	0.638	0.606	0.622
LGBM	<i>t</i>	<i>t</i>	<i>t</i>	<i>t</i>	<b>0.705</b>	0.596	<b>0.651</b>

on the sixth session). This aligns with the results we obtained in Chapter 5 (Table 5.6) where LGBM also achieved the best performance on the condition classification task on the General setting with the mean accuracy of 0.651. This indicates that the model shows a generalisation ability across multiple sessions, as it approached a similar performance when trained on a single session as in the case of the preceding chapter. Moreover, I found that when excluding the first session from the training set, the performance of the model worsens. This indicates that data from the first session is useful for the models to learn the patterns in the data and help them to generalise to subsequent sessions. This also shows that participants' reading behaviour in the first session in which they received instruction is not different from the subsequent sessions in which they did not receive any instruction or reminder of how to perform the tasks. We can observe this by looking at the first two iterations, where the first one is trained on T0 and the second one is on T1. There is not much difference in the performance of the models when comparing the two iterations on mean accuracy, the difference is only 0.3% for ET and 0.1% for RF and, except for LGBM where the difference is 2.4% (but accuracy score when training on T0 is higher than when training on T1).

When examining the models when trained using the SD configuration, as displayed in Table 6.3, we found that their performance is worse than when trained using the GE configuration. The best-performing model (ET) only achieved a mean accuracy of 0.628, which is 2.3% away from that of the LGBM model trained on the GE configuration. This shows a contrast to the findings in Chapter 5 where the models trained on the SD configuration achieved better performance than those trained on the GE configuration. One of the reasons for this could be due to the changes in eye movement features from one session to another. Another possible factor would be due to the amount of data used for training the GE configuration in this study being larger than that in the preceding chapter.

Based on the baseline results on both GE and SD training configurations, we found that training the models on the first 4 sessions is the best option for the

Table 6.3: Reading condition classification results of the model in the SD training configuration. Models are trained on T0 to T3, validated on T4 and tested on T5. The results are highlighted based on the best score for each subject.

Subject	Classifiers								
	ET			LGBM			RF		
	T4	T5	Mean	T4	T5	Mean	T4	T5	Mean
0000	0.375	0.458	0.417	0.500	0.500	<b>0.500</b>	0.292	0.542	0.417
0001	0.333	0.375	0.354	0.500	0.250	<b>0.375</b>	0.292	0.375	0.333
0002	0.500	0.542	0.521	0.542	0.583	<b>0.563</b>	0.542	0.583	<b>0.563</b>
0003	0.667	0.625	0.646	0.708	0.750	<b>0.729</b>	0.750	0.583	0.667
0004	0.833	0.792	<b>0.813</b>	0.750	0.833	0.792	0.875	0.667	0.771
0005	0.667	0.500	<b>0.583</b>	0.667	0.458	0.562	0.583	0.500	0.542
0006	0.625	0.625	0.625	0.667	0.625	<b>0.646</b>	0.625	0.667	<b>0.646</b>
0007	0.708	0.792	0.750	0.750	0.625	0.688	0.708	0.875	<b>0.792</b>
0008	0.667	0.583	<b>0.625</b>	0.625	0.542	0.583	0.667	0.583	<b>0.625</b>
0009	0.875	0.750	<b>0.813</b>	0.792	0.708	0.750	0.833	0.708	0.771
0010	0.750	0.583	<b>0.667</b>	0.583	0.583	0.583	0.625	0.500	0.563
0011	0.708	0.542	<b>0.625</b>	0.625	0.500	0.563	0.667	0.542	0.604
0012	0.708	0.750	<b>0.729</b>	0.750	0.667	0.708	0.708	0.792	0.750
Mean	0.647	0.609	<b>0.628</b>	0.651	0.587	0.619	0.628	0.609	0.619



models to achieve the best performance. Moreover, the best-performing model is LGBM which reaches an accuracy of 0.705 on the validation set (*session\_4*) and 0.651 on the test set (*session\_5*), making a mean accuracy of 0.651. To further increase the model performance, we will use the best-performing model to perform hyperparameter tuning using Optuna. The tuning process was done with 3000 iterations with objective functions to maximise the accuracy score on the validation set, which is *session\_4* in this case. The list of hyperparameters and their corresponding value ranges which were used in the tuning process is shown in Listing 6.1.

Table 6.5: Comparison between the baseline classification model and its tuned version. The model is trained on T0 to T3 and tuned to maximise the accuracy score on T4, which is the validation set.

Classifier	Status	Sessions		Mean Accuracy
		T4	T5	
LGBM	Baseline	0.705	0.596	0.651
	Tuned	<b>0.782</b>	<b>0.628</b>	<b>0.705</b>

As shown in Table 6.5, I was able to maximise the accuracy score on the validation set to 0.782, which is 7.7% higher than the baseline model. This tuned model also achieved an accuracy score of 0.628 on the test set, which is also an improvement of 3.2% from the score of 0.596 of the baseline model. Overall, the tuned model boosted the mean accuracy score by 5.4% from 0.651 to 0.705. The hyperparameters of the tuned model are shown in Listing 6.2.

**Listing 6.1:** Hyperparameters space used for tuning process

```
'num_leaves': [10 : 50]
'bagging_freq': [0 : 5]
'n_estimators': [5 : 200]
'feature_fraction': [0.1 : 1.0]
'bagging_fraction': [0.1 : 1.0]
'drop_rate': [0.1 : 0.9]
'learning_rate': [0.01 : 0.2]
```

**Listing 6.2:** Hyperparameters of the best iteration for LGBM

```
'num_leaves': 11,
'bagging_freq': 1,
'n_estimators': 129,
'feature_fraction': 0.9304680799819736,
'bagging_fraction': 0.6603602015327825,
'drop_rate': 0.22176058999161752,
'learning_rate': 0.031872640733368415
```

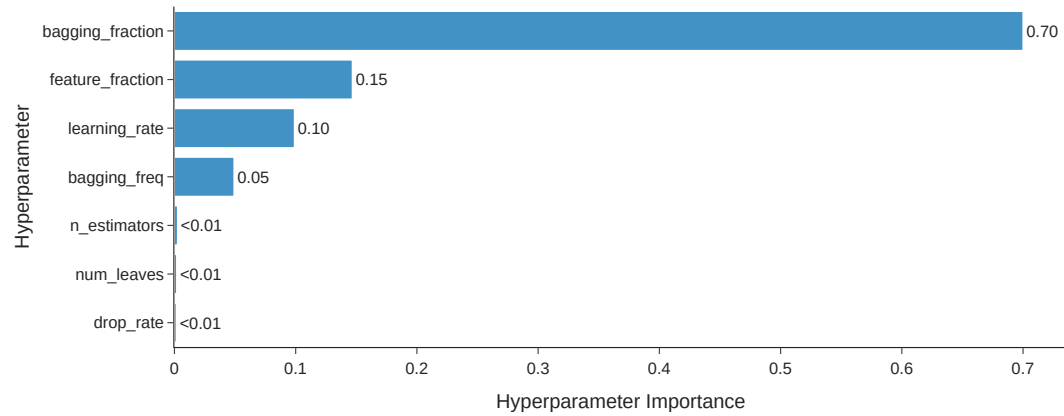


Figure 6.1: Hyperparameters' Importance of the Tuned LGBM Model

To make sense of the hyperparameters obtained from the tuning process, I examined the importance of these hyperparameters to the performance of the model throughout the tuning process. As displayed in Figure 6.1, the most important hyperparameters are *bagging\_fraction*, *feature\_fraction*, *learning\_rate*, which accounts for 69%, 15%, and 10% of the total importance, respectively.

The *bagging\_fraction* parameter specifies the fraction of data to be used for each training iteration and is generally used to speed up the training and avoid overfitting. Similarly, the *feature\_fraction* parameter controls the percentage of features to be used. According to the parallel coordinate plot in Figure 6.2, the *bagging\_fraction* between 0.5 and 0.8 with a *bagging\_freq* (bagging frequency) of 1 (i.e. performs

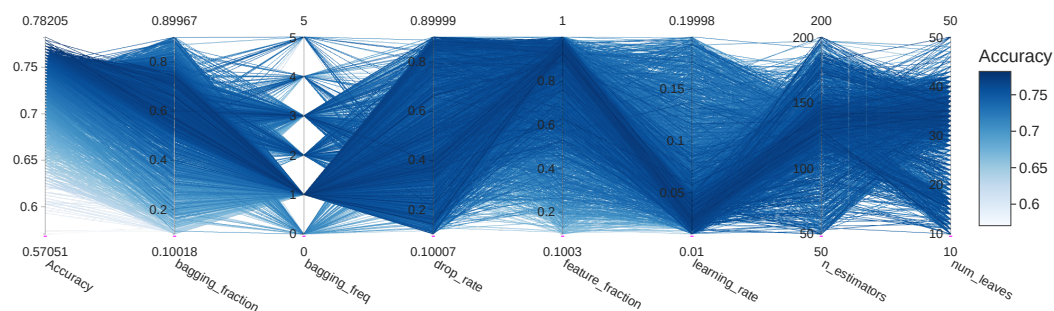


Figure 6.2: Parallel coordinate plot showing hyperparameter settings of all tuning iterations. Each line represents a single iteration while each axis represents a single hyperparameter (and its range/options for tuning as detailed in Listing 6.1). The colour of the line represents the accuracy score of the iteration. The darker the colour, the higher the accuracy score.

bagging for every training iteration) work best for the model. Moreover, most of the tuning iterations achieving high accuracy ( $> 0.75$ ) have *feature\_fraction* between 0.9 and 1.0. This suggests that most eye movement features are relevant and contribute positively to the model’s prediction power. Lowering the *feature\_fraction* could result in losing the important interaction between features and thus, reducing the model’s performance. The best tuning iteration resulted in a *bagging\_fraction* of approximately 0.66 and a *feature\_fraction* of approximately 0.93 (see Listing 6.2), which are within the ranges mentioned above. All in all, I have obtained a tuned model which predicts reading conditions based on participants’ eye movements with an overall accuracy of 0.705 on the validation set and 0.628 on the test set.

### 6.4.2 Reading Comprehension Prediction

Prior to the prediction of reading comprehension, I first re-examined the correlation between the comprehension score (*c\_score*) and the subjective evaluation score (*se\_score*), as the previous chapter (Chapter 5) has shown that there is a strong correlation between these two variables. Moreover, I also found that using the *se\_score* as the target variable for the prediction of reading comprehension is more effective than using the *c\_score*.

The Spearman’s rank correlation between *c\_score* and *se\_score*

Table 6.6: Spearman’s rank correlation between  $c\_score$  and  $se\_score$ , aggregated by participants and by sessions

Aggregate by subjects			Aggregate by sessions		
Subject	$\rho$	$p$ -value	Session	$\rho$	$p$ -value
0000	0.468	<0.001	Trial 0	0.651	<0.001
0001	0.185	0.027	Trial 1	0.586	<0.001
0002	0.568	<0.001	Trial 2	0.556	<0.001
0003	0.650	<0.001	Trial 3	0.557	<0.001
0004	0.844	<0.001	Trial 4	0.618	<0.001
0005	0.401	<0.001	Trial 5	0.561	<0.001
0006	0.505	<0.001			
0007	0.490	<0.001			
0008	0.383	<0.001			
0009	0.629	<0.001			
0010	0.780	<0.001			
0011	0.417	<0.001			
0012	0.470	<0.001			
Mean	0.522	-	Mean	0.588	-

( $r_s(1870) = 0.588, p < 0.0001$ ) over all samples in the RCIRv2 dataset show a strong positive correlation between these two variables. The same relationship can also be observed when the correlation is calculated for each participant and each session (see Table 6.6). For correlation aggregated by participants, the mean correlation is 0.522 ( $p < 0.0001$ ). While most participants have a correlation between 0.4 and 0.6, the correlation for participant 0001 is 0.185 ( $p = 0.027$ ), which is significantly lower than the mean correlation. This means that there is a small number of mismatches between participant 0001’s performance on MCQs and their evaluation of their performance. However, when applying the correlation to the session level, the mean correlation is 0.588 ( $p < 0.0001$ ), which equals to the correlation over all samples. The correlation coefficients of these sessions vary between 0.55 and 0.65 with no significant outliers. This shows that the correlation between  $c\_score$  and  $se\_score$  is stable over time. Hence, this also confirms the finding in Chapter 5 that the  $se\_score$  is a good proxy for the  $c\_score$ .

Based on the results in Section 5.4.3 in the preceding chapter, I used the set of

Table 6.7: The baseline Spearman’s rank correlation coefficient score of the reading comprehension prediction models. Session 0 to session 5 are denoted as T0 to T5, accordingly. The white cells with letter  $t$  indicate that these sessions are used for training. The session which is immediately after the training sessions (i.e., the cells in white which contain a real value) displays the correlation score when evaluating on the validation session. The remaining cells in grey color are scores when evaluating on testing sessions. The models are trained on two different regression target, which are  $c\_score$  and  $se\_score$ . The highlighted scores are the best value within each session.

Regressor	Target	Sessions						Mean $\rho$
		T0	T1	T2	T3	T4	T5	
LGBM	$c\_score$	$t$	0.301	0.242	0.330	0.296	0.249	0.284
ET	$c\_score$	$t$	0.307	0.239	0.389	0.338	0.345	0.324
RF	$c\_score$	$t$	0.314	0.249	0.390	0.315	0.329	0.319
LGBM	$c\_score$	$t$	$t$	0.215	0.379	0.390	0.270	0.314
ET	$c\_score$	$t$	$t$	0.340	0.374	0.371	0.283	0.342
RF	$c\_score$	$t$	$t$	0.354	0.369	0.374	0.333	0.357
LGBM	$c\_score$	$t$	$t$	$t$	0.397	0.348	0.246	0.330
ET	$c\_score$	$t$	$t$	$t$	0.404	0.394	0.297	0.365
RF	$c\_score$	$t$	$t$	$t$	0.396	0.400	0.300	0.365
LGBM	$c\_score$	$t$	$t$	$t$	$t$	0.429	0.315	0.372
ET	$c\_score$	$t$	$t$	$t$	$t$	0.434	0.365	0.399
RF	$c\_score$	$t$	$t$	$t$	$t$	0.450	0.370	0.410
LGBM	$se\_score$	$t$	0.302	0.287	0.427	0.371	0.339	0.345
ET	$se\_score$	$t$	<b>0.339</b>	0.336	0.543	0.430	0.476	0.425
RF	$se\_score$	$t$	0.327	0.338	0.527	0.408	0.468	0.414
LGBM	$se\_score$	$t$	$t$	0.344	0.510	0.489	0.393	0.434
ET	$se\_score$	$t$	$t$	0.391	0.521	0.482	0.439	0.458
RF	$se\_score$	$t$	$t$	<b>0.409</b>	0.534	0.494	0.483	0.480
LGBM	$se\_score$	$t$	$t$	$t$	<b>0.554</b>	0.511	0.485	0.516
ET	$se\_score$	$t$	$t$	$t$	0.546	0.535	0.470	0.517
RF	$se\_score$	$t$	$t$	$t$	0.536	0.539	0.494	0.523
LGBM	$se\_score$	$t$	$t$	$t$	$t$	0.550	0.521	0.536
ET	$se\_score$	$t$	$t$	$t$	$t$	<b>0.584</b>	0.502	0.543
RF	$se\_score$	$t$	$t$	$t$	$t$	0.545	<b>0.544</b>	<b>0.545</b>

best-performing regressors to conduct analysis for this longitudinal dataset. The analysis took a similar approach as in the previous Section 6.4.1, where regressors were trained on the increasing number of sessions (beginning from session 0), validated the next session and tested on the remainders. In addition, both the  $c\_score$  and  $se\_score$  were used to evaluate the performance of the regressors, in order to show the difference in models' performance between these two measures.

The results are shown in Table 6.7. Similar to what was observed in the previous section, the performance of the regressors increased as the number of training sessions increased. Considering the best configuration, which is training with the first 4 sessions, the best-performing regressor for  $c\_score$  is RF (validation score at 0.450 on *session 4*) and for  $se\_score$  is ET (validation score at 0.584 on *session 4*). When considering the mean correlation coefficient across all sessions, the best-performing regressor is RF for both  $c\_score$  and  $se\_score$  (mean score at 0.410 and 0.545 respectively).

Since the performance of the regressors is the highest when trained in the first 4 sessions, I decided to use this configuration to conduct subsequent analysis, which compares the regressors' performance when introducing reading conditions as additional features. In this analysis, information about reading conditions can be obtained in three different ways: (1) using the true label of the reading condition in the dataset, (2) using the predicted label of the reading condition from the baseline classifier (i.e. the classifier without tuning), and (3) using the predicted label of the reading condition from the tuned classifier. Table 6.8 shows the results of this analysis. I observed that there is a significant increase in the performance of the regressors when using the true label of the reading condition as additional features, compared to the baseline (i.e. without reading condition information). The  $\rho$  score increases from 0.434 to 0.488 on the validation set (*session 4*) on the ET model for  $c\_score$ , and from 0.545 to 0.626 on the validation set (*session 4*) on RF model for  $se\_score$ . This shows that having reading condition information as additional features can help the regressors to better predict the  $c\_score$  and  $se\_score$ , which

Table 6.8: Comparison of reading comprehension prediction models when employing reading condition as additional training features. *None* means no reading condition feature is used, while *Baseline classifier* and *Tuned classifier* indicates the condition was predicted using the baseline classifier and the tuned classifier (described in previous Section 6.4.1), respectively. The rows in *italic* display the comprehension prediction performance when using the true label of reading condition (i.e. similar to having a perfect reading condition classification model), which is used for reference purpose only.

Regressor	Reading Condition	Target	Sessions		Mean $\rho$
			T4	T5	
RF	None	c_score	0.450	0.370	0.410
ET	None	c_score	0.434	0.365	0.399
LGBM	None	c_score	0.429	0.315	0.372
RF	Baseline classifier	c_score	0.449	0.346	0.398
ET	Baseline classifier	c_score	0.456	0.350	0.403
LGBM	Baseline classifier	c_score	0.430	0.299	0.365
RF	None	se_score	0.545	<b>0.544</b>	0.545
ET	None	se_score	0.584	0.502	0.543
LGBM	None	se_score	0.550	0.521	0.536
RF	Baseline classifier	se_score	0.563	0.495	0.529
ET	Baseline classifier	se_score	0.515	0.423	0.469
LGBM	Baseline classifier	se_score	0.527	0.462	0.495
RF	Tuned classifier	se_score	0.590	0.497	0.543
ET	Tuned classifier	se_score	0.546	0.451	0.498
LGBM	Tuned classifier	se_score	0.565	0.478	0.521
Tuned RF	Tuned classifier	se_score	<b>0.594</b>	0.516	<b>0.555</b>
Tuned ET	Tuned classifier	se_score	0.573	0.473	0.523
<i>RF</i>	<i>True label</i>	<i>c_score</i>	<i>0.458</i>	<i>0.354</i>	<i>0.406</i>
<i>ET</i>	<i>True label</i>	<i>c_score</i>	<i>0.488</i>	<i>0.381</i>	<i>0.434</i>
<i>LGBM</i>	<i>True label</i>	<i>c_score</i>	<i>0.425</i>	<i>0.320</i>	<i>0.373</i>
<i>RF</i>	<i>True label</i>	<i>se_score</i>	<i>0.626</i>	<i>0.546</i>	<i>0.586</i>
<i>ET</i>	<i>True label</i>	<i>se_score</i>	<i>0.568</i>	<i>0.527</i>	<i>0.548</i>
<i>LGBM</i>	<i>True label</i>	<i>se_score</i>	<i>0.591</i>	<i>0.530</i>	<i>0.560</i>

aligns with what I have concluded in the preceding chapter. However, this increase is only realised if a perfect reading condition classifier is available. When using the predicted label from the baseline classifier, the performance of the regressors is somewhat similar to models trained without reading condition information. However, when using the predicted label from the tuned classifier, I managed to achieve a  $\rho$  (Spearman's rank correlation coefficient) score of 0.59 on the validation set (*session 4*) on RF model for *se\_score*. Despite that, it can be seen that the  $\rho$  score of 0.497 on the test set (*session 5*) is lower than that of the baseline model (without reading condition information), which is 0.544 on the RF model.

To further increase the performance of the regressors, a hyperparameter tuning is conducted on the two best-performing regressors, which are RF and ET, with *se\_score* as the target and *session 4* as the validation. Since these two regressors share similar hyperparameters, I used the same hyperparameter space for both of them (which is detailed in Listing 6.3) and tuned for 3000 iterations using Optuna.

**Listing 6.3:** Hyperparameters space used for tuning process

```
'criterion': ['squared_error', 'absolute_error', 'friedman_mse', 'poisson']
'n_estimators': [1000 : 4000]
'max_depth': [10 : 2000]
'min_samples_split': [2 : 40]
'min_weight_fraction_leaf': [0 : 0.5]
'min_impurity_decrease': [0 : 5]
'ccp_alpha': [0 : 0.5]
'max_samples': [0 : 1]
```

As reported in Table 6.8, the performance of the regressors is increased after tuning. Despite tuned RF only shows a slight increase of 0.004 from its best score of 0.590 before tuning to 0.594 on the validation set. The tuned RF achieved a  $\rho$  score of 0.516 on the test set, which is an increase of 0.190 from the score of 0.497 before tuning. It also achieved the highest mean  $\rho$  score of 0.555 overall. More details on the hyperparameters for this tuned RF model can be found in Listing 6.4.



Listing 6.4: Hyperparameters of the best iteration for RF

```
'criterion': 'friedman_mse'
'n_estimators': 3297
'max_depth': 1761
'min_samples_split': 13
'min_weight_fraction_leaf': 0.009992585036153842
'min_impurity_decrease': 4.2580393330611885
'ccp_alpha': 0.0005973019759799459
'max_samples': 0.7578648164895252
```

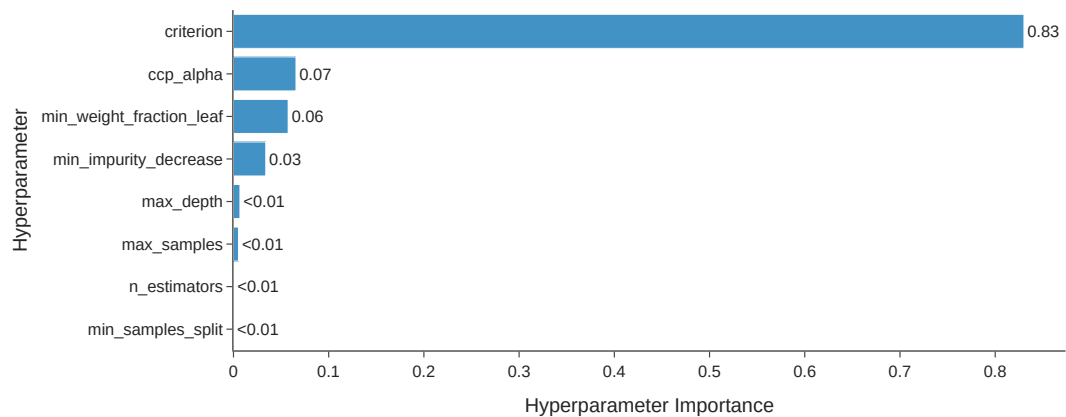


Figure 6.3: Hyperparameters importance of the tuned RF model

Inspecting the contribution of these hyperparameters to the models' performances, I found that the most important hyperparameter is *criterion*, which is found to be *friedman\_mse* (Figure 6.4). This criterion is based on the approach proposed by Jerome Friedman in his work on gradient-boosting machines. Unlike the regular mean squared error, which simply calculates the average of the squared differences between the predicted and actual values, Friedman's MSE incorporates additional statistical techniques to improve the quality of the splits in a decision tree, which in this case also boosted the performance of the RF regressor. Moreover, the next most important hyperparameters are *ccp\_alpha*, *min\_weight\_fraction\_leaf* and *min\_impurity\_decrease*, which are the

hyperparameters that control the complexity of the tree to avoid overfitting/underfitting.

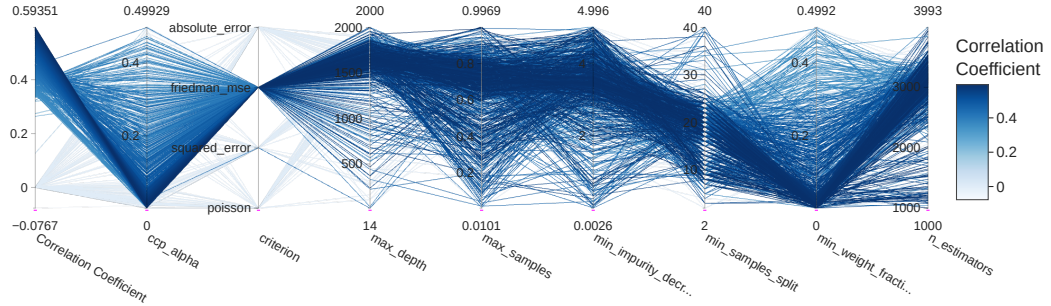


Figure 6.4: Parallel coordinate plot showing hyperparameter settings of all tuning iterations. Each line represents a single iteration while each axis represents a single hyperparameter (and its range/options for tuning as detailed in Listing 6.3). The color of the line represents the  $\rho$  score of the iteration. The darker the colour, the higher the  $\rho$  score.

Table 6.9: Percentage of stable features for each subject, identified using the statistical testing procedure.

Subject	Stable features (%)	Unstable features (%)
<i>S0</i>	38.5	61.5
<i>S1</i>	58.7	41.3
<i>S2</i>	28.8	71.2
<i>S3</i>	28.8	71.2
<i>S4</i>	39.4	60.6
<i>S5</i>	42.3	57.7
<i>S6</i>	44.2	55.8
<i>S7</i>	47.1	52.9
<i>S8</i>	48.1	51.9
<i>S9</i>	52.9	47.1
<i>S10</i>	42.3	57.7
<i>S11</i>	59.6	40.4
<i>S12</i>	43.3	56.7

### 6.4.3 Eye Movement Features Inspection

By adopting the statistical testing procedure as described in Section 6.3.3, I identify for each subject, the set of eye movement features that are stable over time (i.e.

features that show no significant difference across all sessions), with a confidence level of 95%. As shown in Figure 6.9, most of the subjects have 38.5% to 59.6% of their eye movement features that are stable over time, while only two subjects have only 28.8% stable features. When investigating the stable features that are commonly observed across subjects (displayed in Table 6.10), I found that the top stable features are the encodings derived from the fixation duration, saccade duration and blink duration features in eye movement. Decades of research have shown that these features are the most important features in eye movement analysis and identified their duration range to be 150-300ms for fixation, 20-40ms for saccade and 100-400ms for blink [67]. It is not surprising that my findings also align with what has been reported in the literature. Considering the second and third most common stable features, which are not encodings of the fixation, saccade and blink duration features, there are *dist\_v\_iqr*, *nfx\_norm*, *nregr\_norm*, *nbk\_norm*, *smdir\_tr\_min*, *smdir\_tr\_range*, *velo\_v\_iqr*, *velo\_v\_mean*, *velo\_v\_tr\_max*. We started to see the emergence of distance and velocity features, which we have found to be important in the previous chapter in determining reading conditions. However, these distance and velocity features are only stable on vertical movements and not horizontal movements, which is unexpected since horizontal movements are more important for estimating reading comprehension (as found in the previous chapter).

When applying SHAP to explain the feature contribution to the best-performing reading comprehension prediction model (the Tuned RF in Table 6.8), I was able to conclude that the stable features do not contribute much to the prediction of reading comprehension, as none of the top most common stable features are listed in the most important features for the model. Instead, I observed that the reading condition feature plays a significant role in the prediction of reading comprehension as *cond\_reading* being top 1 on the list. Besides, *regr\_rate* is also among the top most important features of which a higher value leads to worse reading comprehension scores. This further confirms and strengthens the findings in Chapter 5 that regression rate is a good indicator of

Table 6.10: List of features that are identified as stable for most subjects

Count	Feature names		
13	bkdur_norm_mean fxdur_norm_std fxdur_norm_tr_range scdur_norm_tr_min	fxdur_norm_iqr fxdur_norm_tr_max scdur_norm_mean	fxdur_norm_mean fxdur_norm_tr_min scdur_norm_tr_max
12	dist_v_iqr scdur_norm_iqr	nfx_norm scdur_norm_std	nregr_norm scdur_norm_tr_range
11	bkdur_norm_tr_max smdir_tr_range velo_v_tr_max	nbk_norm velo_v_iqr	smdir_tr_min velo_v_mean
10	bkdur_norm_tr_min dist_v_tr_max	dist_h_mean smdir_std	dist_v_mean velo_v_tr_range
9	bkdur_norm_iqr dist_h_tr_min smdir_skewness velo_v_std	bkdur_norm_std dist_v_tr_range scdur_norm_kurtosis velo_v_tr_min	bkdur_norm_tr_range smdir_iqr scdur_norm_skewness
8	dist_v_tr_min	smdir_mean	velo_h_mean
7	dist_mean	smdir_tr_max	velo_h_tr_min

reading comprehension.

## **6.5 Chapter Summary**

This chapter investigates the robustness of the reading comprehension estimation model when applied to longitudinal reading data, which is collected over six days from 13 participants. I adopted a similar research process as in Chapter 5, which extracts eye movement features from eye-tracking data and feeds into the training machine learning methods to perform two tasks: reading condition classification and reading comprehension estimation. The results show that both tasks can achieve good and stable performance over time, with a mean accuracy of 0.705 on reading condition classification and a mean Spearman's rank correlation coefficient of 0.555 for reading comprehension prediction. In addition, this chapter also examines the stability of eye movement features over multiple sessions and compares them with the features that have significant contributions to the models' prediction. Experimental results show that despite the high stability of the features, they do not contribute much to the prediction of reading comprehension. Notably, the top most important features align with what was found in the previous chapter, which is also another indicator of the robustness of the reading comprehension estimation model.

Overall, research question 3 is addressed since I have shown that the reading comprehension estimation model is robust when applied to longitudinal reading data. The model, when trained on the first four sessions, can predict reading comprehension on the last two sessions with a mean Spearman's rank correlation coefficient of 0.555 between the predicted and actual subjective reading comprehension scores (specifically,  $\rho = 0.594$  for validation session – *session 4*, and 0.516 for testing session – *session 5*).

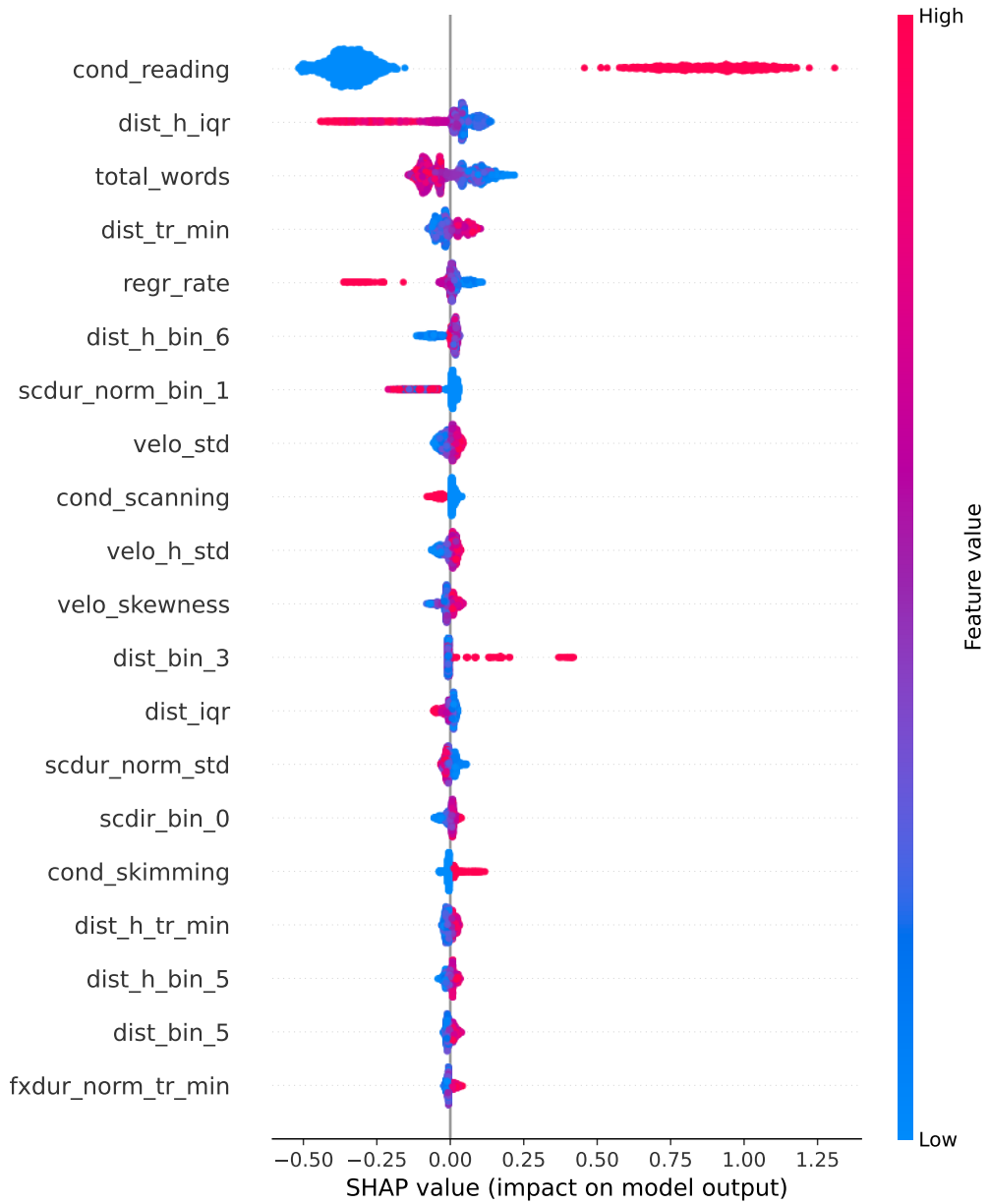


Figure 6.5: SHAP interpretation of the RF model trained on *se\_score* with the predicted reading conditions as additional features. Features are ordered by their importance. The colour represents the value of the feature where red is the highest and blue is the lowest. A positive SHAP value means that the feature contributes to predicting a higher target value, while a negative SHAP value means that the feature contributes to predicting a lower target value.

## Chapter 7

# Gaze-coupled

# Comprehension-evidenced

# Interactive Infologging Retrieval

# System

## 7.1 Introduction

In this chapter, I address Research Question 4: **To what extent does the integration of reading comprehension estimation improve the performance of the infologging retrieval system for on-screen information compared to a baseline system without this feature?**

To answer this question, I propose a novel retrieval system, InfoSeeker, that indexes on-screen information and leverages the user's eye gaze data and reading comprehension level to improve the retrieval performance of previously viewed information. To facilitate the evaluation of the proposed system, I collected a dataset, which I refer to as an *infolog*, of on-screen information and eye gaze data of a participant for daily computer usage over 15 days. This infolog dataset can be viewed as a form of lifelogging that focuses on capturing on-screen information (text) perceived by a user allowing this information to be retrieved later. The novelty of this work lies in the integration of eye gaze data as an additional data

source to enhance the utility of the infologging images (also lifelogging images), enabling the infologger's on-screen attention to be captured to aid the retrieval of on-screen information. However, existing lifelog retrieval systems do not support the retrieval of infologging data, nor do they exploit the rich information contained in the eye gaze data.

In this chapter, I present the design and development of InfoSeeker, an interactive retrieval system for infologging data. InfoSeeker integrates two novel functionalities: a reading comprehension filter and a gaze heatmap visualisation. The reading comprehension filter allows users to retrieve information based on the infologger's comprehension level, estimated from the eye movement patterns of the infologger when engaging with the on-screen information. The estimation is obtained by employing a comprehension estimation model that was trained on a reading dataset which I have described in Chapter 6. The gaze heatmap visualisation provides an intuitive representation of the infologger's eye gaze distribution on the information, highlighting the areas of interest and attention on the screen. These functionalities aim to enhance the retrieval performance and user experience of InfoSeeker.

To evaluate whether the introduction of the aforementioned functionalities improves the retrieval performance of infologging data, two experiments were conducted: a non-interactive retrieval experiment and an interactive retrieval experiment via a user study. The non-interactive experiment tests the system's ability to retrieve and rank relevant results based on the user's query and comprehension level filter, compared to a baseline system that does not exploit the eye gaze data. The interactive experiment evaluates the system in a realistic setting, where users perform a series of retrieval tasks using both the baseline system and the gaze-coupled system. Performance of both systems was measured using the LSC score – a common metric in evaluating lifelog retrieval systems – which is a single value that takes submission accuracy, task completion time, and penalties for incorrect submissions into account. The two proposed experiments are



inspired by the evaluation methodology of the lifelog retrieval task at NTCIR [28–30, 85] and LSC [35–39]. Since infologging data is a form of lifelogging data, the evaluation methodology of lifelogging retrieval systems is suitable for evaluating the proposed system.

The results of the experiments show that InfoSeeker, with the integration of a reading comprehension estimation filter, outperforms the baseline system in both non-interactive and interactive settings, demonstrating the effectiveness of gaze-coupled functionalities in enhancing the retrieval of infologging data. The feedback from the users also confirms the utility and usability of the reading comprehension filter and the gaze heatmap visualisation in the retrieval process. This chapter contributes to the advancement of lifelog research by introducing a novel retrieval system for infologging data and by exploiting eye gaze data as a valuable source of information.

## 7.2 Data Collection

In this section, I will describe the process of creating the on-screen information lifelogging dataset, or the **infolog** dataset in short. Similar to how most lifelog datasets are created, which involves the use of devices to passively capture the user’s data (e.g., wearable cameras that capture images, GPS devices that capture location information), the infolog dataset is created by employing a software tool that passively captures the user’s on-screen information (as screenshots) and their corresponding eye gaze data (through an eye tracker). Inspired by the design of Loggerman [51], a software that aims to capture as many aspects of computer usage as possible (e.g. screenshots, keystrokes, keyboard, mouse events, clipboard, apps transition), I came up with a design of a simpler version of Loggerman, which serves the need of capturing on-screen information alongside eye gaze data. In the subsequent sections, I will first re-define the unit of retrieval for the infolog dataset, and then describe the process of creating the infolog dataset.

### 7.2.1 Unit of Retrieval

In contrast to other Information Retrieval (IR) tasks such as web or blog search, lifelogging does not have a universally accepted smallest unit of retrieval or even a concept of a document [10]. The choice of the retrieval unit in lifelogging is highly use-case specific [10, 175]. In existing lifelog research, various units of retrieval have been employed, such as the life event [10], a minute as a time-unit [27], or individual data points like images, temperature readings, and location data [26]. Notably, the minute is often the preferred unit of retrieval in many lifelog retrieval benchmarking challenges (i.e. NTCIR [28–30], ImageCLEF Lifelog [31–34], and LSC [35–39]).

Loggerman [51], a system with functionalities similar to the infolog system, also utilises minute as a unit of retrieval. In Loggerman, screenshots are captured at intervals chosen by the user (every 5, 10, or 30 seconds) or by default every minute, a mode referred to as 'smart-shooting' to balance capture frequency and storage use. For the infolog dataset, however, adopting a minute as the unit of retrieval is not practical due to the inclusion of eye gaze data. Capturing screenshots at fixed intervals, such as every minute, can lead to misalignment between the eye gaze data and the information displayed in the screenshot, as the user might have viewed multiple pieces of information on the screen during that time. This misalignment would diminish the dataset's utility for recalling previously viewed information.

To address this, the infolog dataset employs an event-based interval for screenshot capture, with an *event* defined as any change in on-screen information presentation. This approach ensures that each screenshot is accompanied by eye gaze data specific to the period of that information's display. Such alignment not only facilitates the retrieval of information viewed by the user but also allows for an estimation of the user's reading comprehension of that information, thereby enhancing the efficiency of information retrieval. Consequently, the chosen unit of retrieval for the infolog dataset is each instance of change in on-screen information presentation (i.e. a screenshot).

### 7.2.2 Data Collection Process

The data collection process for this study involves an individual, referred to as the *infologger*, engaging in their daily activities on a computer. During this time, the infologging software passively records both on-screen information and the infologger’s eye movements. Eye movements of the infologger are captured using a Gazepoint GP3 HD eye tracker, operating at a sampling frequency of 150Hz. This device is positioned beneath the computer monitor and connected to the infologger’s computer via a USB 3.0 port.

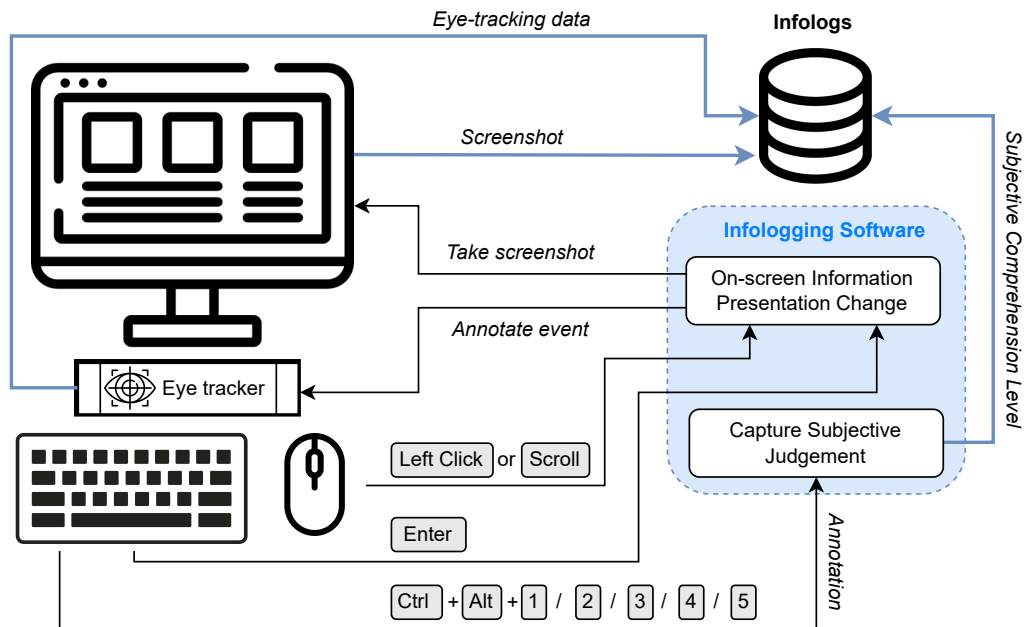


Figure 7.1: The process of logging on-screen information. The lines in blue highlight the data sources that are presented in the infolog dataset.

Additionally, the infolog software, developed in Python, runs simultaneously on the same computer. Its primary function is to monitor specific computer events that signal changes in the on-screen content, capture screenshots, and log the continuous stream of eye gaze data. For the sake of simplicity, the software is configured to respond to a limited set of triggers: mouse clicks, mouse scrolls, and the pressing of the enter/return key, which are considered indicators of changes in screen presentation. Each time one of these events occurs, the infolog software

captures a screenshot of the entire computer screen after a brief delay of 200 milliseconds, allowing time for the screen content to render fully. Simultaneously, it marks an annotation in the stream of eye gaze data, signifying the commencement of a new event. This process is depicted in Figure 7.1, providing a visual representation of the data collection process.

Moreover, to assist in evaluating the reading comprehension estimation model (detailed in Section 7.3.1), the infolog software includes a feature that allows the infologger to subjectively annotate their comprehension level of the information displayed on the screen instantly. This annotation is done by pressing *Ctrl + Alt + Number* on the keyboard, where *Number* ranges from 1 to 5. Here, 1 represents the lowest level of comprehension and 5 the highest (the same scale as the previous study in Chapter 5 and Chapter 6). While these annotations are not compulsory, the infologger is encouraged to annotate as frequently as possible to generate a robust dataset for evaluation purposes.

### 7.2.3 Infolog Dataset Overview

Given the infologging data aims to capture the user's everyday computer interactions, the infologger is permitted to engage in any typical computer tasks on their personal computer. This approach, however, presents a privacy challenge as it may involve sensitive information that the infologger may prefer to keep private, including details they might not wish to disclose to the researcher. To address this concern and protect privacy, I assumed the role of the infologger and conducted the data collection on my personal computer. This practice is not unusual in lifelogging research, as many lifelog datasets only involve a single participant, who is also the researcher themselves [10, 34, 36–39, 85]. This method is generally seen as the most viable approach to gathering research data while upholding strict privacy standards.

The infologging dataset constructed in this study contains a total of 6,825 screenshots, each paired with corresponding eye gaze data. These were collected over a span of 15 days, with an average daily usage of approximately 40 minutes.

Among these screenshots, 118 were annotated with the infologger’s subjective comprehension judgments, serving as a key dataset for evaluating the comprehension estimation model.

Prior to the data analysis, I followed the principles as recommended by [10,176] to filter out any screenshots containing personal or sensitive information. This included images showing personal messages, private documents, and banking details of the infologger, for instance. Consequently, 424 screenshots were identified and removed from the dataset, leaving 6,401 screenshots available for subsequent analysis.

### **7.3 Methodology**

In this section, I will first describe the process of extracting eye movement features from the eye gaze data, on top of which the comprehension estimation model is employed to estimate the infologger’s comprehension of the information displayed on the screen. Then, I will describe the changes made to the SOTA lifelog retrieval system – LifeSeeker – to enable the retrieval of information based on the infologging data, which I refer to as InfoSeeker. Finally, the evaluation of InfoSeeker through two subsequent experiments will be described, with one focusing on a non-interactive evaluation and the other focusing on an interactive setting through a user study, to show the effectiveness of including gaze data in retrieving previously viewed information.

#### **7.3.1 Feature Extraction and Evaluation of the Reading Comprehension Estimation Model**

Eye-tracking data is first split into small segments, each of which corresponds to a screenshot. Each of these segments can be viewed as a reading instance as in the RCIRv2 dataset (from Chapter 6). From these segments, various ocular events such as fixations, saccades, and blinks are extracted. The set of eye movement features derived from these events is consistent with those utilised in Chapter 6 (for a full

list of features, please refer to Section 6.3.1).

These extracted eye movement features are inputted into the pre-trained reading condition model (the Tuned LGBM model outlined in Table 6.5 from Section 6.4.1). This model predicts the reading strategy employed by the infologger. As previously discussed in Chapters 5 and 6, the identified reading condition is then used as an additional feature alongside the eye movement features in the reading comprehension estimation model to predict comprehension levels. Applying the Tuned RF model from Table 6.8 in Section 6.4.2 to these features allows for the estimation of the comprehension level for each screenshot. This comprehension level is integrated into the metadata of each screenshot to enhance the retrieval process, which will be elaborated upon in the following section (Section 7.3.2).

To assess the reliability of this additional metadata (i.e., reading comprehension level) in aiding the retrieval process, an evaluation is conducted to capture its correlation with the infologger’s actual comprehension level. The correlation coefficient should be reasonably close to the score obtained in Table 6.8 in Chapter 6 to be considered reliable. As outlined in the data collection methodology (Section 7.2), the infologger provides subjective annotations of their comprehension level on some screenshots, serving as ground truth for this evaluation. Spearman’s rank correlation coefficient is utilised to facilitate the evaluation. The findings from this experiment will be presented in Section 7.4.1

### 7.3.2 Development of the interactive retrieval system

In this section, the focus is on the development of an interactive retrieval system for infologging data, named *InfoSeeker*. This system is an extension of LifeSeeker [44], a state-of-the-art interactive lifelog retrieval system that my colleagues and I developed (described in Chapter 4) The decision to base InfoSeeker on the LifeSeeker system was due to two main reasons. Firstly, there is a notable similarity between the infologging and lifelogging data, as both are multimodal digital archives that represent aspects of a human’s daily life. Both

datasets' structures are also similar, comprising of a set of images, each of which is paired with a set of metadata. Secondly, given LifeSeeker's state-of-the-art performance in lifelog retrieval, its adaptation for infologging data is not only a straightforward process but also allows for a potential to achieve state-of-the-art performance in this new domain – the infologging data.

Additionally, screenshots data (captured by Loggerman [51]) was once a component of the lifelog dataset used in LSC'18 [35], in which lifelog retrieval systems were evaluated. However, screenshots data was omitted in the later versions of lifelog datasets and subsequent benchmarking challenges. While the reason for this exclusion is not explicitly stated by LSC's organisers, it could be due to privacy concerns, as the screenshots may contain sensitive information [10, 176]. Nonetheless, I envision that there is a possibility that screenshots data would be re-introduced in future lifelog datasets once privacy-preserving methods are developed to address this concern. Therefore, the development of InfoSeeker can also be seen as a step towards the development of a future lifelog retrieval system that supports both lifelogging and infologging data.

The structure of this section mirrors the approach in Chapter 4, where I will initially discuss InfoSeeker's user interface and interaction, followed by an in-depth description of the system's architecture and retrieval unit.

### 7.3.2.1 User Interface and Interaction

InfoSeeker's user interface, as depicted in Figure 7.2, closely resembles that of LifeSeeker, with a key addition being the functionality to view on-screen information enhanced by gaze heatmaps. Each screenshot in the system is linked to a specific instance of eye-tracking data, enabling the generation of a gaze heatmap for individual screenshots. Users can toggle the heatmap on or off by clicking the *Eye* icon located in the top right corner of the menu bar (refer to Figure 7.2). These visualisations allow users to effortlessly identify moments when the infologger was focused on particular screen areas, as the heatmap typically

concentrates on regions containing the information that was being read.

An additional modification to the user interface is the incorporation of a comprehension level filter, highlighted in an orange box in Figure 7.2.B. This filter offers three levels—*low*, *medium*, and *high*—corresponding to the infologger’s comprehension level as determined by the comprehension estimation model described in Section 7.3.1. The model’s comprehension scores, ranging from 1 to 5, are categorised into three groups for this filter: *low* (1-2), *medium* (3), and *high* (4-5). Users can apply this filter based on their query context to retrieve the most relevant results.

### 7.3.2.2 Search Engine

As outlined in Chapter 4, the search engine is the pivotal component of our retrieval system, tasked with returning relevant results based on user queries. This engine operates via two primary processes: an offline data indexing process executed once prior to deployment, and an online query processing and retrieval process. The following sections detail both the indexing (offline) process and the retrieval (online) process.

### 7.3.2.3 Indexing

The indexing process for the infolog data, owing to its similarities with lifelog data, largely mirrors that used in LifeSeeker, as discussed in Section 4.4.1. In LifeSeeker, four main categories of lifelog data are indexed: time, location, visual content, and others (e.g., activity and biometric data). For InfoSeeker, the indexing process simplifies to process the following data:

1. **Time Data:** Adopting the same procedure from LifeSeeker, I extracted various temporal features from the time data. Based on the image ID format `YYYYmmdd_HHMMSS`, I parsed the timestamp to derive features such as part of the day, day of the week, month name, and year. Given that the current infolog data is within a single timezone, timezone and local time data are not





Figure 7.2: The user interface of InfoSeeker. (A) The default interface shows the list of screenshots that match the query. (B) The interface for viewing screenshots alongside the infologger’s gaze heatmap. (C) The search results after applying a *high* filter on the comprehension level.

indexed but can be included for future datasets which span multiple timezones.

- 2. Visual Data:** The primary focus here is on extracting textual information displayed on the computer screen. For this purpose, I employed the state-of-the-art Optical Character Recognition (OCR) model from Meta, Tesseract [177]. Tesseract is an open-source engine known for its accuracy and support for a wide range of languages. It uses a combination of line finding, feature extraction and neural network classification to recognise text in images. In this work, Tesseract was used with its default settings as it provided satisfactory results for the infolog data. The recognised texts returned by Tesseract engine are indexed as part of the screenshot's metadata. Additionally, for efficient retrieval, I performed text embedding on the recognised texts, transforming them into vector representations. These vectors (also known as embeddings) are then indexed using Milvus [156], an open-source vector database that supports rapid vector searches, as was previously used in LifeSeeker. These embeddings are generated using Sentence Transformers [178], which provides a wide range of pre-trained models for generating dense vector representations of text. The specific model chosen was *msmarco-distilbert-base-v3*<sup>1</sup>, which is a DistilBERT [179] model fine-tuned on the Microsoft Machine Reading Comprehension (MS MARCO) dataset [180]. This model was selected due to its strong performance on semantic textual similarity tasks and its computational efficiency compared to larger models. The MS MARCO dataset, on which the model was fine-tuned, consists of over 500k questions and their corresponding answers, making it well-suited for generating embeddings that capture semantic meaning. Furthermore, other modalities (e.g., images) present in the data are not focused on in this experiment but can be indexed in the future using LifeSeeker's approach.

---

<sup>1</sup><https://huggingface.co/sentence-transformers/msmarco-distilbert-base-v3>

3. **Reading Comprehension Data:** This is a novel data source, not present in traditional lifelog data. The comprehension level, estimated from the infologger's gaze data and the screenshot's visual content (as described in Section 7.3.1), is indexed as part of the screenshot's metadata to aid retrieval based on comprehension levels.

#### 7.3.2.4 Retrieval

The retrieval process closely aligns with the one described in Section 4.4.2 in Chapter 4. Both search and filter queries function similarly to LifeSeeker. Specifically, search queries involve embedding query text into vector representations, followed by performing a vector search on the indexed data using Milvus. Filter queries, meanwhile, are processed through Elasticsearch. A notable addition in InfoSeeker is the comprehension level filter. When applied, it refines the search results to include only those screenshots that match the specified comprehension level.

### 7.3.3 Evaluation of InfoSeeker Retrieval System

In this evaluation, I aim to explore the effectiveness of reading comprehension level filtering gaze visualisation and in enhancing the retrieval performance of InfoSeeker. In this section, I will first describe the process of creating the test topics for evaluating the system in Section 7.3.3.1 and then present the detail of the evaluation experiments in Section 7.3.3.2.

#### 7.3.3.1 Generation of Test Topics

As discussed in Section 2.1, the information needs arise when the lifelogger wants to recall a past event. Similarly, the information needs in infologging are created as the infologger needs a specific piece of information, which they know that they have seen it before. To facilitate the evaluation of the system, a test set of 10 topics was created, with each topic consisting of a query which matches an information

need and an associated screenshot which contains the information relevant to the query. In lifelog, such a test set is typically generated by the owner of that lifelog to include topics that best represent their real-life information needs [28]. However, in this study, I – the infologger – am also the system’s developer, and thus, the process of generating the test set has to be modified to bring in rules and guidelines that would reduce the bias. Since the main source of bias is selecting screenshots that favour the proposed system, I have decided to involve a third party in the process of selecting the screenshots. The choice of the third party also needs to follow certain criteria to ensure the quality of the test set. As a result, I have defined the following criteria for the third party:

- The person has to be a trusted individual who the infologger is comfortable with sharing the infolog data without any concern of privacy breach. The third party will be given access to all screenshots in the infolog data (with are organised in folders by date) and is free to select any screenshots that they deem suitable for the test set.
- The person has to be an expert in the field of lifelogging and lifelog retrieval so that they can select screenshots that are challenging to the system but not impossible to retrieve.
- The person has no prior knowledge of the proposed system so that they would not be biased towards selecting screenshots that favour the proposed system
- The person has to select screenshots that best represent the infologger’s information needs. In particular, since this study focuses on retrieving the information that the infologger has previously seen, the selected screenshots should be those that contain information that the infologger has engaged with. To aid the third party in this process, the dataset with the infologger’s gaze visualisation is also made available to them. Other than that, no extra information is provided to the third party, including the infologger’s comprehension levels of the screenshots.

After considering the above criteria, I have decided to involve my colleague as the third party, who in turn has selected 10 screenshots as test topics. From these, I begin to formulate the query for each topic. The query formulation process also adheres to the following rules:

- The infologger has to formulate the query based on the content inside the screenshot only. No external information is allowed to be used in the query formulation process, including other screenshots in the infolog data and the gaze visualisation data. Having access to only the screenshot makes the query formulation process more similar to the real-life scenario, where the infologger only remembers part of the information and has to recall the moment when they saw the information.
- The infologger is not allowed to use any type of retrieval system during the query formulation process to avoid bias.
- The query has to include cues about the infologger's comprehension of the information in the screenshot (i.e., "I remember this very well", "I did not pay much attention", "I only skimmed through this", etc.). This is to facilitate the evaluation of the comprehension level filtering feature, which is the main focus of this study.
- The query has to be formulated in a way that it is not too easy for the system to retrieve the screenshot. Instead, the query should begin with a general description of the screenshot and gradually become more specific. This is the practice that most lifelog benchmarking challenges employ, especially in LSC [35–39].

Following the above rules, 10 queries were formulated and the test set was finalised. The list of queries in the test set is provided in Appendix Section A.2. Of these, there are 5 queries associated with high comprehension levels (queries Q1, Q2, Q5, Q6, Q8), 3 queries with medium comprehension levels (queries Q4, Q9), and 2 queries

with low comprehension levels (queries Q3, Q7). It is important to acknowledge that the aforementioned comprehension levels are estimated by the infologger when formulating the query and may not align with the comprehension level predicted by the system.

### 7.3.3.2 Experimental Setup

The main target of this study is to compare InfoSeeker's performance with and without these gaze-coupled functionalities. The version of InfoSeeker without gaze enhancement serves as the *baseline system* (denote as *BA*), while the version with gaze-coupled functionalities is referred to as *gaze-coupled system* (denote as *GZ*). In order to evaluate the BA and GZ systems, two distinct experiments were conducted:

- **Experiment 1 - Non-interactive retrieval:** Adopting the evaluation methodology from the LSAT (Lifelog Semantic Access Task) in NTCIR challenges [28–30], this experiment tests the system's ability to retrieve and rank relevant results. This experiment is conducted in an automatic manner, in which there is no user interaction involved. For each test topic, the entire query text is inputted into the system at once, and the system returns a ranked list of screenshots as the results. After obtaining ranked lists from both BA and GZ systems, precision (at 1, 5, 10, 25, 50 and 100) and recall are calculated for both lists, on which the performance of the two systems is compared.
- **Experiment 2 - Interactive retrieval:** Inspired by LSC's evaluation methodology of interactive retrieval systems, this experiment adopts a similar approach to evaluate InfoSeeker, since infologging and lifelogging data are highly similar (as discussed in Section 7.3.2). Consequently, a user study was conducted, which mirrors the LSC format. Participants in this study perform a series of retrieval tasks using the test topics generated in Section 7.3.3.1. In particular, they will perform half of the test topics using the BA system and

Table 7.1: Example of query presentation of the test topic Q1 at multiple time-points for interactive evaluation

Time	Query text
0s	I clearly remember that I came across the news stating that Nvidia has a new collaboration...
30s	I clearly remember that I came across the news stating that Nvidia has a new collaboration in the field of artificial intelligence...
60s	I clearly remember that I came across the news stating that Nvidia has a new collaboration in the field of artificial intelligence. This project involves the use of Large Language Models (LLM) for chip design...
90s	I clearly remember that I came across the news stating that Nvidia has a new collaboration in the field of artificial intelligence. This project involves the use of Large Language Models (LLM) for chip design, and it's referred to as ChipNeMo...
120s	I clearly remember that I came across the news stating that Nvidia has a new collaboration in the field of artificial intelligence. This project involves the use of Large Language Models (LLM) for chip design, and it's referred to as ChipNeMo. Unfortunately, I can't remember the specifics regarding the number of parameters it was trained with...
150s	I clearly remember that I came across the news stating that Nvidia has a new collaboration in the field of artificial intelligence. This project involves the use of Large Language Models (LLM) for chip design, and it's referred to as ChipNeMo. Unfortunately, I can't remember the specifics regarding the number of parameters it was trained with. I recall seeing this earlier this month in the evening.

the other half using the GZ system. Since each query is formulated so that it begins with a general description and gradually becomes more specific (see Section 7.3.3.1), it can be divided into 6 incremental clues revealed at intervals (every 30 seconds), with the full query is presented at the 150-second mark. Table 7.1 provides an illustrative example of this query presentation process. It is important to acknowledge that this type of query presentation is adapted from the LSC competition. Additionally, I also employed the same metric used in the LSC to evaluate the retrieval performance, which focuses on submission accuracy, task completion time, and penalties for incorrect submissions (described in Section 3.3.3).

## 7.4 Results and Discussion

### 7.4.1 Evaluation of Reading Comprehension Estimation Model

As described in Section 7.3.1, the predicted comprehension level is compared to the infologger’s subjective comprehension level to assess the reliability of using the output from the reading comprehension model as additional metadata to aid the retrieval of on-screen information. In the infologging data, the subjective comprehension levels of 118 screenshots are provided by the infologger.

The correlation between these subjective annotations and the predicted comprehension levels was analysed using Spearman’s rank correlation coefficient. The results yielded a correlation coefficient of  $r_s(116) = 0.485$  with a significance level of  $p < 0.001$ . This correlation, when compared to the mean correlation score of 0.555 reported in Table 6.8 from the RCIRv2 dataset in Chapter 6, shows a modest difference of only 0.07. This variance is relatively minor, especially considering the substantial differences in dataset settings and formats.

The RCIRv2 dataset was collected under controlled conditions with text presented in a standardised format. In contrast, the infologging dataset was compiled in a real-world environment where text formats vary widely and often



appear alongside images. Given these contextual disparities, a 0.07 deviation in the correlation score is deemed acceptable. This finding suggests that the reading comprehension model can be effectively applied to the infologging dataset to enhance the retrieval process, enabling the use of the existing pre-trained reading comprehension estimation model (in Chapter 6) without having to collect additional data for model training specific to the infologging context.

## 7.4.2 Evaluation of Infoseeker

### 7.4.2.1 Experiment 1 - Non-interactive Retrieval

Table 7.2 presents the evaluation scores for each query in the non-interactive InfoSeeker retrieval system, comparing the performance of the system with and without gaze-coupled functionalities. Generally, the gaze-coupled system (GZ) demonstrates better performance over the baseline system (BA), which retrieves the correct results for 9 out of 10 queries, surpassing the BA system, which retrieves 8 out of 10. In terms of precision, the GZ system has four queries with the correct screenshot ranked at the top position ( $N_{P@1 \neq 0} = 4$ ) and six queries where the correct screenshot is among the top 5 ( $N_{P@5 \neq 0} = 6$ ). In contrast, the BA system only has one query with the correct screenshot in the top 5 ( $N_{P@1 \neq 0} = 0$  and  $N_{P@5 \neq 0} = 1$ ).

Analysis of the queries where the GZ system achieved high P@1 scores indicates that the system is most effective in retrieving screenshots associated with a high level of comprehension. Applying a high comprehension level filter often results in the correct screenshot being ranked at the top of the search results. This outcome aligns with the system's primary objective, which is to retrieve information that the user recalls having previously seen and comprehended well. However, the system's performance is less effective when filtering for 'low' or 'medium' comprehension levels. This disparity can be attributed to the large volume of screenshots in the dataset that are tagged with low and medium comprehension levels. For instance, a screenshot captured while an infologger is

Table 7.2: Overall evaluation scores of the non-interactive version of the InfoSeeker system with and without gaze-coupled functionalities.

Query	P@1		P@5		P@10		P@25		P@50		Precision P@100		Recall	
	BA	GZ	BA	GZ	BA	GZ	BA	GZ	BA	GZ	BA	GZ	BA	GZ
1	0.00	1.00	0.20	0.20	0.10	0.10	0.04	0.04	0.02	0.02	0.01	0.01	1.00	1.00
2	0.00	1.00	0.00	0.20	0.00	0.10	0.00	0.04	0.00	0.02	0.01	0.01	1.00	1.00
3	0.00	0.00	0.00	0.20	0.10	0.10	0.04	0.04	0.02	0.02	0.01	0.01	1.00	1.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.02	0.00	0.01	0.00	1.00	0.00
5	0.00	1.00	0.00	0.20	0.00	0.10	0.04	0.04	0.02	0.02	0.01	0.01	1.00	1.00
6	0.00	1.00	0.00	0.20	0.10	0.10	0.04	0.04	0.02	0.02	0.01	0.01	1.00	1.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	1.00
8	0.00	0.00	0.00	0.20	0.00	0.10	0.00	0.04	0.00	0.02	0.01	0.01	1.00	1.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.02	0.00	0.01	0.01	1.00	1.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	1.00

merely scrolling through a webpage might be classified as having a low comprehension level. This classification makes it challenging to differentiate such a screenshot from one where the infologger actively reads content but still has low comprehension. This issue opens up a new direction for future research to enhance the system's ability to accurately retrieve information based on varying levels of comprehension. Currently, the most feasible method to distinguish between these scenarios involves manually inspecting the screenshots, with an emphasis on analysing the heatmap visualisation of gaze data. This approach underlines the importance of the interactive evaluation of the InfoSeeker system, where the integration of gaze data plays a crucial role in enhancing the retrieval process.

#### **7.4.2.2 Experiment 2 - Interactive Retrieval through a User Study**

In this section, I discuss the outcomes of a user study conducted to evaluate the InfoSeeker system's performance in an interactive manner. To ensure the experimental results' reliability, the study was constrained to participants with prior experience in operating lifelog retrieval systems. This approach was essential to minimize biases that could arise from the participants' unfamiliarity with the LSC-style evaluation format and the complexities of the retrieval system. Consequently, the study was able to concentrate on assessing the impact of gaze-coupled functionalities on the InfoSeeker system, without the learning curve associated with the system. With that, five experienced users were recruited for the study.

As mentioned in Section 7.3.3, each of the 10 queries generated for this experiment was based on specific screenshots from the dataset. Participants were tasked with locating the exact moment depicted in these screenshots using either the BA or GZ system, depending on their assigned setting. To control for the potential bias introduced by the varying difficulty levels of queries, the queries were divided into two groups: Group A (queries with IDs from 1 to 5) and Group B (queries with IDs from 6 to 10). Two users first interacted with Group A using the

Table 7.4: A breakdown of participants’ LSC scores for each query using the baseline (BA) and gaze-coupled (GZ) InfoSeeker systems.

System	Setting 1			Setting 2			
	Query ID	$U0$	$U1$	Query ID	$U2$	$U3$	$U4$
<i>BA</i>	1	73.61	4.17	6	0.00	0.00	0.00
	2	0.00	0.00	7	0.00	0.00	27.22
	3	0.00	0.00	8	0.00	0.00	0.00
	4	0.00	0.00	9	0.00	0.00	0.00
	5	0.00	19.44	10	0.00	0.00	0.00
<i>GZ</i>	6	48.06	56.11	1	42.50	76.39	64.17
	7	0.00	0.00	2	26.11	52.50	52.78
	8	65.28	85.83	3	48.33	62.78	0.00
	9	0.00	82.50	4	40.56	65.28	21.39
	10	0.00	0.00	5	0.56	65.83	73.33

BA system, and then with Group B using the GZ system (Setting 1). The other three users started with Group B on the BA system before switching to Group A on the GZ system (Setting 2). Prior to the study, participants received an introduction to the InfoSeeker system and its functionalities, along with an explanation of the infologging data and the types of data sources present in the dataset. They were also given a sample query for practice on the system to familiarise themselves with the interface and the retrieval process.

Table 7.4 shows a detailed breakdown of the results from the user study. The GZ system not only achieved higher average LSC scores but also successfully solved more queries compared to the BA system. This is attributed to the integration of gaze data, which refined the search results by filtering out irrelevant screenshots and prioritizing those aligned with the user’s comprehension levels. The gaze heatmap visualisation provided an intuitive interface, guiding users to focus on areas of high gaze activity and bypass those with minimal engagement.

Feedback from participants further confirmed the utility of the reading comprehension filter and gaze heatmap visualisation in enhancing the retrieval process. Compared to the BA system where users had to rely on other modalities to identify the correct moment (e.g. time data), the GZ system’s heatmap feature

allowed for quicker and more efficient retrieval. However, two out of five participants also found the requirement of pinpointing the exact moment in the BA system to be challenging. These users, despite being able to identify screenshots that match the query, failed to locate the precise moment of the lifelogger's actual engagement with this information. This is particularly difficult when multiple visually identical screenshots are present, but only one of them is deemed relevant to the query.

While including all content-matching screenshots in the ground truth might seem more inclusive for evaluation, it diverges from the system's core objective: to accurately retrieve the specific information the infologger has viewed in the past. Take this example query: "It was a paper which reviews the 20-year of eye movement research by Keith Rayner that I often refer to during my writing. I wonder when the first time I read it was.", simply including every instance where the paper appears on screen would not suffice. The infologger's interest lies in identifying the very first engagement, not the subsequent times the paper was possibly revisited for minor references. Thus, the necessity of identifying the exact moment becomes apparent. This requirement underscores the importance of precision in retrieval systems, ensuring they deliver not just relevant but contextually accurate results.

However, to not overlook the scenarios where the infologger might be more interested in retrieving content irrespective of their past engagement, an ablation study was conducted to evaluate the performance of the system when the requirement to locate the exact moment is removed. For this purpose, the ground truth for each query was expanded to include all screenshots in which the content is relevant to the query, regardless of whether they represented the exact moment of the infologger's engagement. The participants' submissions from the initial user study were re-evaluated against this expanded ground truth. Table 7.5 details the outcomes of this re-assessment. The results indicated that the gaze-coupled system (GZ) continued to exhibit superior performance compared to the baseline system

Table 7.5: A breakdown of participants’ LSC scores for each query using the baseline (BA) and gaze-coupled (GZ) InfoSeeker systems on the expanded ground truth.

System	Setting 1			Setting 2			
	Query ID	$U_0$	$U_1$	Query ID	$U_2$	$U_3$	$U_4$
<i>BA</i>	1	73.61	50.28	6	0.00	0.00	0.00
	2	60.83	0.00	7	2.78	0.00	27.22
	3	0.00	65.56	8	83.89	60.56	75.83
	4	34.44	0.00	9	39.72	50.83	75.83
	5	21.11	32.22	10	80.00	0.00	31.39
<i>GZ</i>	6	48.06	56.11	1	42.50	76.39	64.17
	7	51.39	0.00	2	26.11	52.50	52.78
	8	65.28	85.83	3	48.33	62.78	0.00
	9	38.06	82.50	4	56.39	65.28	31.67
	10	0.00	0.00	5	22.50	65.83	73.33

(BA), in both experimental settings and across all measured metrics, including the average LSC score and the total number of queries successfully solved. This further confirms that with gaze-coupled functions, the GZ system is able to achieve better performance than the BA system.

Table 7.6: Overall performance of the baseline (BA) and gaze-coupled (GZ) InfoSeeker systems using the initial and expanded ground truth. The values presented are the mean LSC scores of the user’s submissions.

User	Initial		Expanded	
	Ground-truth		Ground-truth	
	BA	GZ	BA	GZ
0	14.72	<b>22.67</b>	38.00	<b>40.56</b>
1	4.72	<b>44.89</b>	29.61	<b>44.89</b>
2	0.00	<b>31.61</b>	<b>41.28</b>	39.17
3	0.00	<b>64.56</b>	22.28	<b>64.56</b>
4	5.44	<b>42.33</b>	42.06	<b>44.39</b>
Mean Score	4.98	<b>41.21</b>	34.64	<b>46.71</b>

Table 7.6 shows the overall performance of the baseline systems (BA) and the gaze-coupled (GZ) systems in both settings, with two different versions of ground truth. The evaluation clearly demonstrates that the gaze-coupled system (GZ) outperforms the baseline system (BA) in both settings, regardless of the ground

truth used. Specifically, the GZ system shows a substantial improvement of 36.23% over the BA system under the initial ground-truth setting, which focuses on identifying only one correct screenshot. Moreover, in the expanded ground-truth setting, where all content-relevant screenshots are considered, the GZ system still maintains a notable lead of 12.07% over the BA system. The findings in this section, alongside the results from the non-interactive evaluation in the preceding section, provide strong evidence that integrating the reading comprehension filter and gaze heatmap visualisation into the conventional retrieval system has a significant positive impact on the system's performance. This, in turn, addresses research question 4 of this thesis.

## **7.5 Chapter Summary**

This section summarises the main contributions and findings of this chapter, which addresses the research question 4. It establishes a novel approach for capturing on-screen information alongside gaze data, resulting in the creation of a new type of lifelogging dataset, referred to as infolog dataset. This dataset is unique in its ability to facilitate the retrieval of previously viewed visual content. A key development in this research is the redefinition of the unit of retrieval for the infolog dataset. Moving away from traditional time-based units, this new approach focuses on event-based changes in information presentation on the screen, ensuring a total capture of on-screen content and a more accurate mapping of associated gaze data.

The main contribution of this chapter is the introduction of InfoSeeker, an interactive retrieval system designed for infologging data. This system exploits the eye gaze data to enhance the retrieval of on-screen information. Utilising pre-trained models from earlier chapters, InfoSeeker estimates the user's comprehension level of the displayed information, integrating this value into the retrieval process as a filter. This allows information that the infologger previously engaged with to be ranked higher in the retrieval results. Besides, gaze heatmap

visualisation is also introduced to aid users in quickly identifying information that has high gaze activity on the screen and eliminating the content that the infologger did not engage with. This functionality is particularly useful to speed up the search process.

The performance of InfoSeeker was tested through two key experiments: a non-interactive retrieval experiment and an interactive retrieval experiment conducted via a user study. These experiments were designed to compare the system's performance with and without the integration of gaze-coupled functionalities. The findings from these experiments were clear and consistent. The gaze-enhanced version of the system demonstrated superior performance across various metrics, including precision, recall, and LSC scores, when compared to the version without gaze data.

In conclusion, research question 4 is addressed since I have demonstrated that the integration of gaze-coupled functionalities into the retrieval system has a significant positive impact on the system's performance, which is evaluated using both non-interactive and interactive experiments.



## Chapter 8

# Conclusion

In this thesis, I proposed the hypothesis that it is possible to estimate the comprehension level of content displayed on a computer screen and enhance the performance of the state-of-the-art lifelog retrieval system by employing this estimated comprehension level as a result filtering mechanism for the retrieval of previously perceived on-screen information. To validate this, I determined four research questions which I addressed through a series of evaluations to either prove or disprove the hypothesis. In this final chapter, I provide a summary of how the research questions are addressed, which ultimately shows that my hypothesis is upheld.

### 8.1 Summary

For the first research question, I asked how a state-of-the-art lifelog retrieval system can be constructed. This question is formed because I aim to explore what components are essential for building a state-of-the-art retrieval system and apply the insights to develop a retrieval system for the on-screen information. The choice to focus on developing a lifelog retrieval system was driven by the similarities between lifelogging and the proposed concept of infologging data, coupled with the active research community in lifelogging that frequently organises retrieval benchmarking competitions. These competitions provided a platform to develop and benchmark a system, facilitating the evaluation of what constitutes a state-of-the-art system. Addressing this research question involved the

development, implementation, and evaluation of LifeSeeker, an interactive system for multi-modal personal lifelog data. LifeSeeker’s design allows users to conduct efficient searches and filter lifelog moments using free-text queries. The system derives its search capabilities from the combination of a variety of SOTA techniques, including an image-text embedding model and visual concept extractors, crucial for precise and relevant data retrieval. LifeSeeker also incorporates advanced engineering solutions for indexing and retrieval, utilising tools like Milvus for fast vector similarity calculations, Elasticsearch for complex filtering mechanism search and scalability, and Redis for effective caching. A notable feature of LifeSeeker is its user-centric interface, specifically designed to optimise the search and exploration experience within lifelog data. This includes clustering methods for quick results browsing, a range of filtering options (both active and passive), relevance feedback functionalities for refining search results, and a search history timeline for easy reaccess of previous queries and search results. These functionalities are the direct result of the continued improvement throughout a series of participation in the Lifelog Search Challenge (from 2019 to 2022). LifeSeeker’s performance in these benchmarking challenges has consistently ranked it among the top systems, affirming its status as a state-of-the-art lifelog retrieval system. The development of LifeSeeker and the identification of the key components contributing to its success have not only contributed to the field of lifelogging but also laid a solid foundation for the development of a retrieval system for infolog data. Consequently, the first research question has been addressed through the creation and refinement of LifeSeeker.

For the second research question, I asked how reading comprehension can be estimated through eye movement measures using machine learning models. This question was driven by the need to investigate the extent to which human comprehension levels can be inferred from eye movements. The ultimate goal was to apply these findings in estimating the comprehension level of on-screen information, thereby enhancing the retrieval of previously perceived information.

To address this question, an experimental study was undertaken involving 10 participants. The study involved reading a series of static passages on a computer while engaging in four distinct reading strategies: sequential reading, scanning, skimming, and proofreading. Post-reading, participants were required to answer multiple-choice questions and provide subjective assessments of their comprehension, serving as labels for machine learning model evaluation. The eye movement data collected from this experiment was then analysed using two separate procedures: machine learning approaches and statistical testing. Initial statistical analysis revealed a complex relationship between eye movement features, reading conditions, and comprehension levels. On the other hand, machine learning models were trained using these features to solve two primary tasks: (1) predicting the reading strategy employed by participants, and (2) estimating their reading comprehension levels. The classification models, applied in both subject-dependent and general contexts, demonstrated an accuracy of 75.3% and 68.9%, respectively. Importantly, the inclusion of predicted reading conditions as additional features alongside eye movement features in training comprehension prediction models resulted in a significant improvement in predictive performance. A Spearman's correlation coefficient ( $\rho$ ) of 0.697 was obtained when correlating predicted comprehension levels with actual levels derived from participants' responses to multiple-choice questions. In addition, I also found that when participants' subjective assessments of understanding were used as the training target instead of their responses to the multiple-choice questions, the correlation coefficient increased to 0.785, indicating a stronger alignment with participants' perceived comprehension. This outcome demonstrates the potential of utilising eye movement measures in training machine learning models to accurately estimate reading comprehension levels. As a result, I conclude that it is viable to infer reading comprehension through eye movement data, and hence, the second research question has been addressed.

The third research question explored the temporal robustness of the reading

comprehension estimation model developed in response to the second research question. This investigation was crucial to confirm the model’s applicability and consistency over time, particularly for use with longitudinal datasets like infologging data. To address this question, a longitudinal study was conducted, gathering eye movement data from 13 participants engaged in reading tasks over six non-consecutive days. The experimental design closely mirrored that of the previous study, with the primary distinction being the extension of the experiment over multiple days. This setup allowed for the assessment of the model’s stability and reliability over time. Eye movement features were extracted, and the evaluation was conducted in a similar manner to the previous study. The results from this longitudinal study were positive, demonstrating that the model retained a consistent level of performance throughout the duration of the experiment. Specifically, the model achieved a mean accuracy of 0.705 in classifying reading conditions and a mean Spearman’s rank correlation coefficient of 0.555 in predicting reading comprehension levels. In this study, I also found that the model’s performance improved with an increase in the number of sessions used for training. This suggests that the model benefits from exposure to a broader range of data over time, enhancing its predictive capabilities. Furthermore, the study examined the consistency of eye movement features across different sessions and compared these with the features most influential in the model’s predictions. The findings indicated more than half of the features are stable, yet these stable features did not significantly influence the prediction of reading comprehension. Notably, the features with the most significant contribution to the model prediction were consistent with those identified in the previous study, which confirms the model’s validity when training on longitudinal data. These findings collectively demonstrate the temporal robustness of the reading comprehension estimation model. This robustness is crucial for applying the model to longitudinal datasets like infologging data. Thus, the third research question is addressed.

The fourth research question centered on evaluating the performance of a

state-of-the-art lifelog retrieval system, specifically developed for infologging data, both with and without the integration of a reading comprehension estimation model. This question was crucial for synthesising the findings from the preceding research questions to develop a retrieval system for infologging data. It also focuses on assessing the performance of the proposed system. To answer this question, a prototype retrieval system, named InfoSeeker, was developed. This system was built upon the foundation of the state-of-the-art lifelog retrieval system outlined in the first research question. Concurrently, an infolog dataset was created to serve the evaluation of InfoSeeker, which captures the on-screen activities and corresponding eye movement data of an infologger over a 15-day period. The evaluation of the InfoSeeker system on the constructed infolog dataset was conducted in both non-interactive and interactive settings and focused on comparing the system's performance with and without the incorporation of the reading comprehension estimation model (from research question 3). In the non-interactive setting, the evaluation involved comparing the top-100 results retrieved by the systems against the established ground truth, utilising precision and recall metrics. This approach provided a quantitative measure of the system's ability to rank relevant results high in the ranked list (allowing the user to find the correct result quickly in the interactive mode). Meanwhile, the interactive setting entailed a user study, structured in the format of the Lifelog Search Challenge (LSC). Participants were asked to conduct a series of search tasks using InfoSeeker, and their performance was evaluated based on the LSC score (see Chapter 3 for the metric definition). This metric considers both the accuracy of the submissions and the time taken to complete the search tasks. The findings from two evaluations show that InfoSeeker, when enhanced with the reading comprehension estimation model, consistently outperformed its counterpart without this integration in both settings. This outcome confirms the substantial benefit of incorporating the reading comprehension estimation model into the retrieval system to retrieve previously perceived information. In summary, the successful development and

evaluation of the InfoSeeker system, particularly with the integration of the reading comprehension estimation model, conclusively addressed the fourth research question.

As I have addressed the four primary research questions, I can discuss the validity of my hypothesis. Given the limitations of this research, which are described in the following section, I proved that it is feasible to estimate reading comprehension levels through eye movement measures. With this, the comprehension level of on-screen information can be estimated, which can then be used to facilitate the retrieval of previously perceived information by employing this as a filtering mechanism in the retrieval system. Hence, I consider my proposed hypothesis defined in Section 1.4 to be upheld.

## **8.2 Contributions**

### **8.2.1 Revisiting Research Contributions**

In Chapter 1.4, I outlined the main contributions of this research. As the thesis concludes, it is essential to revisit these contributions and demonstrate how they have been successfully addressed throughout the work presented in this dissertation.

Chapter 4 focused on addressing RQ1 and made a significant contribution by constructing LifeSeeker, an interactive lifelog retrieval system that has consistently ranked among the leading state-of-the-art systems in the Lifelog Search Challenge. This contribution laid the foundation for the subsequent research questions and the development of the infologging retrieval system, as the key design principles and components that make a system achieve state-of-the-art performance were identified.

In Chapter 5, I addressed RQ2 and made several key contributions. Firstly, I constructed a novel multi-modal reading dataset, enabling the investigation of the relationship between eye movements, reading strategies, and comprehension levels. Secondly, I demonstrated that incorporating reading condition identification with eye movement features can improve the performance of machine learning models in

estimating reading comprehension levels. Lastly, I provided novel insights into the importance of eye movement measures for classifying reading styles and estimating comprehension levels through comprehensive statistical and feature contribution analyses.

Chapter 6 focused on RQ3 and contributed to the field by creating a unique longitudinal reading dataset, which allowed for the exploration of the temporal robustness of the reading comprehension estimation model. The findings from this chapter demonstrated the robustness of the eye movement features and the comprehension estimation model when applied to longitudinal reading data. Additionally, novel insights were uncovered regarding the temporal stability of eye movement features and their limited contribution to the model's performance in estimating comprehension levels over time.

Finally, in Chapter 7, I addressed RQ4 by introducing InfoSeeker, a novel interactive retrieval system designed for infologging data. This system leverages eye gaze data to enable filtering of search results based on the user's level of comprehension, facilitating quick retrieval of desired information. The evaluation of InfoSeeker, conducted through both non-interactive and interactive user studies, demonstrated the significant improvement in the system's performance achieved by integrating reading comprehension estimation into the retrieval process.

The contributions made in this thesis have advanced our understanding of the relationships between eye movements, reading strategies, and comprehension levels, while also showcasing the potential for integrating this knowledge into practical applications, such as lifelog and infologging retrieval systems. By revisiting these contributions, I aim to highlight the coherence and significance of the research presented in this dissertation, demonstrating how each piece contributes to the overall narrative and addresses the research questions posed in Chapter 1.4.

### **8.3 Limitations**

In Chapter 1.6, I discussed the limitations of this research, focusing on two main categories: the comprehension estimation model and the gaze-coupled retrieval system. It is crucial to revisit these limitations to provide context for the interpretation of the findings and to guide future research efforts.

Regarding the comprehension estimation model, the vertical error exhibited by the eye tracker used in this study posed a challenge in precisely estimating the gaze position at the word level. However, as the analysis was conducted at the passage level, the impact of this error was minimized. Additionally, the presence of outliers in the dataset, resulting from participants occasionally performing the wrong reading tasks, may have slightly influenced the results. Despite these limitations, the strong performance of the classification model on reading conditions suggests that the impact of these issues was not substantial.

Furthermore, the limited number of sessions available for training the comprehension estimation model on the longitudinal reading data restricted the exploration of the model's full potential. As discussed in Chapter 6, increasing the number of training sessions led to improved model performance. However, due to the constraints of the available data, the upper limit of the model's performance could not be determined, presenting an opportunity for future research when more data becomes available.

Turning to the gaze-coupled retrieval system, the InfoSeeker system, being in its early stages of development, has several areas for improvement. The current search mechanism primarily relies on matching query text with OCR text from screenshots, limiting its ability to handle queries that require other modalities. Additionally, the system's search results are directly displayed from the ranked list generated by the cosine similarity matching algorithm, without further processing to re-organize the results. Implementing a mechanism to group similar screenshots could enhance user efficiency in conducting search tasks. Moreover, the reading comprehension



model integrated into the InfoSeeker system has not been re-trained specifically on infologging data, potentially limiting its predictive accuracy in real-world settings.

Despite these limitations, the research presented in this thesis has made significant contributions to the fields of lifelogging, eye movement analysis, and information retrieval. The insights gained from this work lay the foundation for future research efforts, which can build upon the findings and address the identified limitations. By continuing to explore the integration of eye movement data and reading comprehension estimation in information retrieval systems, we can move closer to the development of truly user-centric and efficient tools for managing and accessing personal information.

## 8.4 Future Work

The research carried out in this thesis aimed at investigating the feasibility of estimating reading comprehension levels through eye movement measures and the potential of employing this model to facilitate the retrieval of previously perceived information. As a result, there is future work that can be carried out to further improve the proposed model and the retrieval system. In the subsequent sections, I envision potential areas of study that can be carried out in the future.

### 8.4.1 Improving the Reading Comprehension Model

There are many directions in the proposed reading comprehension estimation model can be improved.

- **Addressing Current Model Limitations:** A primary focus should be on addressing the recognised limitations of the current model. This includes investigating and rectifying the vertical shift error in the eye tracker, as previously discussed. Implementing a method to correct these errors is crucial for enhancing data quality. Additionally, exploring inter-trial calibration, where calibration is conducted before each reading trial, could significantly

refine the adjustment of eye gaze data, leading to more precise analysis.

- **Exploring Feature Engineering Methods:** The current feature set, globally derived from ocular events in eye-tracking data, could be expanded through more localised methods. For instance, word-level features like first fixation duration on a word, number of word visits or number of refixations could offer more granular insights. These features, however, require accurate alignment between gaze positions and text, which is currently hindered by the eye tracker's vertical shift error. Another promising direction is extracting general time-series features from eye gaze positions, which could reveal trends in eye movements, aiding in reading condition classification and, subsequently, comprehension estimation.
- **Incorporation of Additional Modalities:** Expanding the model to include other modalities such as electrooculography (EOG) signals, facial expressions, and textual features could offer a more holistic understanding of reading comprehension. EOG signals, in conjunction with eye gaze data, could assist in identifying and rectifying gaze estimation errors. Facial expressions provide insights into the reader's emotional state during reading, potentially influenced by the text content. Textual features like text length, word count, and sentence and paragraph numbers could reflect the complexity of the text and be used alongside eye-tracking data to predict reading comprehension.
- **Exploring Deep Learning Models:** While the current research primarily employed machine learning models for their simplicity and interpretability, deep learning models offer a more powerful alternative, especially in tasks like natural language processing. Investigating deep learning models for reading comprehension estimation could uncover more sophisticated relationships in eye movement features and potentially lead to more accurate predictions.

### 8.4.2 Improving the InfoSeeker System

Improvements and future developments for the InfoSeeker system can be outlined as follows:

- **Enhancing InfoSeeker’s Modality Indexing and Result Presentation:** One of the primary areas for improvement involves expanding the system’s capability to index additional modalities found in screenshots, such as images. This enhancement would enable InfoSeeker to address queries recalling both images and text from the screenshots. Furthermore, optimising the presentation of search results to group similar screenshots could provide users with a more coherent and comprehensible overview, facilitating easier navigation through the results.
- **Re-training the Reading Comprehension Estimation Model:** Re-training the reading comprehension estimation model using samples directly from infologging data could potentially increase the performance. Given that real-world information in the infolog is presented in various formats, re-training the model with infologging data would allow it to adapt to diverse text formats and new eye movement patterns in accordance with these formats, enhancing its robustness and applicability.
- **Refining Text Embedding with Gaze Information:** The current approach to text indexing in screenshots, which involves embedding texts into a vector space using deep learning models, could be further refined by integrating gaze information. This proposed gaze-coupled text embedding would involve co-registering gaze positions with word bounding boxes in the text, assigning varying weights to words based on gaze visits. Such a method would ensure that words receiving more attention are emphasized in the indexing process, potentially enhancing the retrieval accuracy.
- **Incorporating Large Language Models for Question-Answering:**

With the rapid advancements in large language models (LLMs) [181–185], there is a unique opportunity to leverage these models for a question-answering system for infologging data. Instead of users scrolling through a ranked list of results, this list could be fed directly into an LLM to generate precise answers to the user’s queries. This integration could significantly elevate the user experience by streamlining the retrieval process and delivering more direct, concise answers.

## 8.5 List of publications

This section summarises all publications that are produced during the course of this PhD research.

**The following lists the publications that I am the first author / co-first of**

1. **Le, Tu-Khiem**, Ninh Van-Tu, Zhou Liting, Duc-Tien Dang-Nguyen, and Gurrin Cathal. "DCU team at The 2019 Insight for Wellbeing Task: Multimodal personal health lifelog data analysis." (2019).
2. **Le, Tu-Khiem**, Van-Tu Ninh, Liting Zhou, Minh-Huy Nguyen-Ngoc, Luca Piras, Michael Alexander Riegler, Pål Halvorsen et al. "Organiser Team at ImageCLEFlifelog 2020: A Baseline Approach for Moment Retrieval and Athlete Performance Prediction using Lifelog Data." (2020).
3. **Le, Tu-Khiem**, Manh-Duy Nguyen, Ly-Duyen Tran, Van-Tu Ninh, Cathal Gurrin, and Graham Healy. "DCU team at the NTCIR-15 Micro-activity Retrieval Task." NTCIR, 2020.
4. **Le, Tu-Khiem**, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Minh-Triet Tran, Liting Zhou, Pablo Redondo, Sinead Smyth, and Cathal Gurrin. "Lifeseeker: Interactive lifelog search engine at lsc 2019." In Proceedings of the ACM

- Workshop on Lifelog Search Challenge, pp. 37-40. 2019.
5. **Le, Tu-Khiem**, Van-Tu Ninh, Minh-Triet Tran, Thanh-An Nguyen, Hai-Dang Nguyen, Liting Zhou, Graham Healy, and Cathal Gurrin. "Lifeseeker 2.0: Interactive lifelog search engine at lsc 2020." In Proceedings of the Third Annual Workshop on Lifelog Search Challenge, pp. 57-62. 2020.
  6. **Le, Tu-Khiem**, Van-Tu Ninh, Mai-Khiem Tran, Graham Healy, Cathal Gurrin, and Minh-Triet Tran. 2022. AVSeeker: An Active Video Retrieval Engine at VBS2022. In MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6–10, 2022, Proceedings, Part II. Springer-Verlag, Berlin, Heidelberg, 537–542.
  7. Ninh, Van-Tu, **Tu-Khiem Le**, Liting Zhou, Graham Healy, Kaushik Venkataraman, Minh-Triet Tran, Duc-Tien Dang-Nguyen, S. Smith, and Cathal Gurrin. "A baseline interactive retrieval engine for the NTICR-14 Lifelog-3 semantic access task." In The Fourteenth NTCIR Conference (NTCIR-14). 2019.
  8. Healy, Graham, **Tu-Khiem Le**, Minh-Triet Tran, Thanh-Binh Nguyen, Boi Mai Quach, and Cathal Gurrin. "Overview of the NTCIR-16 RCIR Task." In Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-16). Tokyo, Japan. 2022.
  9. Nguyen, Thao-Nhu, **Tu-Khiem Le**, Van-Tu Ninh, Minh-Triet Tran, Nguyen Thanh Binh, Graham Healy, Annalina Caputo, and Cathal Gurrin. "LifeSeeker 3.0: An Interactive Lifelog Search Engine for LSC'21." In Proceedings of the 4th annual on lifelog search challenge, pp. 41-46. 2021.
  10. Nguyen, Thao-Nhu, **Tu-Khiem Le**, Van-Tu Ninh, Annalina Caputo, Graham Healy, Sinéad Smyth, Minh-Triet Tran, and Nguyen Thanh Binh. "LifeSeeker: an interactive concept-based retrieval system for lifelog data." *Multimedia Tools and Applications* 82, no. 24 (2023): 37855-37876.

11. Thao-Nhu Nguyen, **Tu-Khiem Le**, Van-Tu Ninh, Minh-Triet Tran, Thanh Binh Nguyen, Graham Healy, Sinéad Smyth, Annalina Caputo, and Cathal Gurrin. 2022. LifeSeeker 4.0: An Interactive Lifelog Search Engine for LSC'22. In Proceedings of the 5th Annual on Lifelog Search Challenge (LSC '22). Association for Computing Machinery, New York, NY, USA, 14–19
12. Nguyen, Thao-Nhu, **Tu-Khiem Le**, Van-Tu Ninh, Cathal Gurrin, Minh-Triet Tran, Thanh Binh Nguyen, Graham Healy, Annalina Caputo, and Sinead Smyth. "E-LifeSeeker: An Interactive Lifelog Search Engine for LSC'23." In Proceedings of the 6th Annual ACM Lifelog Search Challenge, pp. 13-17. 2023.
13. Hordvik, Maria Tysse, Julie Sophie Teilstad Østby, Manoj Kesavulu, Thao-Nhu Nguyen, **Tu-Khiem Le**, and Duc-Tien Dang-Nguyen. "LifeLens: Transforming Lifelog Search with Innovative UX/UI Design." In Proceedings of the 6th Annual ACM Lifelog Search Challenge, pp. 1-6. 2023.

**The following lists the publications that I contributed partially**

1. Gurrin, Cathal, Hideo Joho, Frank Hopfgartner, Liting Zhou, Van-Tu Ninh, **Tu-Khiem Le**, Rami Albatal, Duc-Tien Dang-Nguyen, and Graham Healy. "Advances in lifelog data organisation and retrieval at the NTCIR-14 Lifelog-3 task." In NII Testbeds and Community for Information Access Research: 14th International Conference, NTCIR 2019, Tokyo, Japan, June 10–13, 2019, Revised Selected Papers 14, pp. 16-28. Springer International Publishing, 2019.
2. Gurrin, Cathal, Hideo Joho, Frank Hopfgartner, Liting Zhou, V-T. Ninh, T-K. Le, Rami Albatal, D-T. Dang-Nguyen, and Graham Healy. "Overview of the NTCIR-14 lifelog-3 task." In Proceedings of the 14th NTCIR conference, pp. 14-26. NII, 2019.

3. Gurrin, Cathal, **Tu-Khiem Le**, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn Þór Jónsson, Jakub Lokoč, Wolfgang Hürst, Minh-Triet Tran, and Klaus Schoeffmann. "Introduction to the third annual lifelog search challenge (LSC'20)." In Proceedings of the 2020 International Conference on Multimedia Retrieval, pp. 584-585. 2020.
4. Healy, Graham, **Tu-Khiem Le**, Hideo Joho, Frank Hopfgartner, and Cathal Gurrin. "Overview of ntcir-15 mart." In Proceedings of the 15th NTCIR Conference on Evaluation of Information Access Technologies, pp. 299-303. National Institute of Informatics, 2020.
5. Dang Nguyen, Duc Tien, Luca Piras, Michael Riegler, Liting Zhou, Mathias Lux, Minh Triet Tran, **Tu-Khiem Le**, Van-Tu Ninh, and Cathal Gurrin. "Overview of ImageCLEFlifelog 2019: solve my life puzzle and lifelog moment retrieval." CEUR Workshop Proceedings, 2019.
6. Tran, Minh-Triet, Thanh-An Nguyen, Quoc-Cuong Tran, Mai-Khiem Tran, Khanh Nguyen, Van-Tu Ninh, **Tu-Khiem Le** et al. "FIRST-Flexible Interactive Retrieval SysTem for visual lifelog exploration at LSC 2020." In Proceedings of the Third Annual Workshop on Lifelog Search Challenge, pp. 67-72. 2020.
7. B. Ionescu, H. Muller, R. P'eteri, A. Ben Abacha, V. V. Datla, S. A. Hasan, D. Demner-Fushman, S. Kozlovski, V. Liauchuk, Y. D. Cid, V. Kovalev, O. Pelka, C. M. Friedrich, A. G. S. de Herrera, V. Ninh, **T.-K. Le**, L. Zhou, L. Piras, M. Riegler, P. Halvorsen, M. Tran, M. Lux, C. Gurrin, D. Dang-Nguyen, J. Chamberlain, A. Clark, A. Campello, D. Fichou, R. Berari, P. Brie, M. Dogariu, L. Stefan, and M. G. Constantin. Overview of the imageclef 2020: Multimedia retrieval in medical, lifelogging, nature, and internet applications.
8. Ninh, Van-Tu, **Tu-Khiem Le**, Duc-Tien Dang-Nguyen, and Cathal Gurrin.

- "Replay detection and multi-stream synchronization in CS: GO game streams using content-based Image retrieval and Image signature matching." (2019).
9. Ninh, Van-Tu, **Tu-Khiem Le**, Liting Zhou, Luca Piras, Michael Alexander Riegler, Pål Halvorsen, Mathias Lux, Minh-Triet Tran, Cathal Gurrin, and Duc Tien Dang Nguyen. "Overview of imageclef lifelog 2020: lifelog moment retrieval and sport performance lifelog." (2020).
10. Van, Tu-ninh, **Tu-Khiem Le**, Liting Zhou, Luca Piras, Michael Riegler, Mathias Lux, Minh-Triet Tran, Cathal Gurrin, and Duc Tien Dang Nguyen. "LIFER 2.0: Discovering Personal Lifelog Insights using an Interactive Lifelog Retrieval System." (2019)
11. Nguyen, Thao-Nhu, **Tu-Khiem Le**, Van-Tu Ninh, Ly-Duyen Tran, Manh-Duy Nguyen, Minh-Triet Tran, Binh T. Nguyen et al. "DCU and HCMUS at NTCIR-16 Lifelog-4." In Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies. NTCIR, 2022.



# Appendix A

## Appendix

### A.1 Obtaining Texts for Reading Task in RCIRv1

#### A.1.1 Overview

Texts and comprehension questions used in RCIR are extracted from the RACE dataset [166]. The dataset contains articles collected from online websites which span many different topical domains. The comprehension questions were constructed by the dataset experts to assess individuals comprehension of each text. The questions are in the form of multiple choices of two types: normal answers and embedded answers (cloze). In addition, the RACE dataset is divided into 2 levels (middle and high school text content). In this study, we only focus on the high-school level as our targeted participants are undergraduates, postgraduate students, and staff members within the department.

#### A.1.2 Topic-modelling

Since the texts in RACE dataset are not categorised, we have employed a topic-modelling process (as shown in Figure A.1) to group the texts into topics.

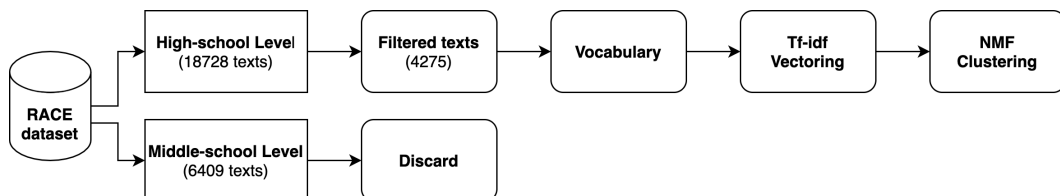


Figure A.1: Topic-modelling process

Firstly, we filtered the high-school level texts to select the candidate texts that have at least three questions in the normal answer style (questions without cloze). A vocabulary was then built for the filtered texts using the five most frequent words from each text. Next, all candidate texts were TF-IDF [167] vectorised to fit an NMF [168] clustering model to group these texts into multiple clusters. Twelve clusters were selected from these and formed into topics based on the texts within the cluster.

### A.1.3 Topic Validating

Prior to the data collection process, we also conducted a topic validation process for the text, where we had two annotators confirm for each text that it aligned with the generated topic. A summary of the texts falling into each topic is described in Table A.1.

### A.1.4 Splitting Text Data

We divided the twelve selected topics into **six topics** (1, 11, 16, 19, 24, 41) for training data and **six topics** for testing data (2, 7, 9, 29, 37, 40). Texts in the training topics were further organised into two groups: consistent group and inconsistent group. The consistent group, in contrast to the inconsistent one, comprised of texts that would be read by all participants (i.e. a consistent 24 texts). The texts in the testing topics were in the inconsistent group only (i.e., each text was unique).

In particular, for one training topic, a participant is expected to read:

- (A) A set of four texts in the consistent group.
- (B) A set of four texts in the inconsistent group.
- (C) Another set of four texts in the inconsistent group (but shared with one other participant in the experiment).

For one testing topic, a participant will read a set (D) of four texts in the inconsistent group. Four texts in each set correspond to the four aforementioned reading conditions (reading, scanning, skimming, proofreading). Therefore, in total, a participant read:

$$\begin{aligned}
 \text{No. text to read} &= N_{\text{train}} * (S_A + S_B + S_C) + N_{\text{test}} * S_D \\
 &= 6 * (4 + 4 + 4) + 6 * 4 \\
 &= 96
 \end{aligned} \tag{A.1}$$

where  $N_{\text{train}}$  is the number of training topics and  $N_{\text{test}}$  is the number of testing topics.  $S_A$ ,  $S_B$ ,  $S_C$ , and  $S_D$  are the number of texts in each set (A, B, C, and D).

As illustrated in Table A.3, the participant S0 will need to read 24 consistent texts in purple (representing set A), 48 texts in two inconsistent sets B and C for training topics (green and read highlighting correspondingly) and 24 inconsistent texts – set D – for testing topics (highlighted in yellow). It is important to note that while a certain proportion of texts were reused between participants for comparison purposes in the training dataset, all texts in the testing set are unique and from an independent set of 6 topics.

## A.2 Test Topics for Evaluation of Infolog Retrieval System

**Q1** I clearly remember that I came across a news stating that Nvidia has a new collaboration in the field of artificial intelligence. This project involves the use of Large Language Models (LLM) for chip design, and it's referred to as ChipNeMo. Unfortunately, I can't remember the specifics regarding the number of parameters it was trained with. I recall seeing this earlier this month in the evening.

- Q2** I want to find the article about investing, where I picked up a new concept called meme stock and went googling it after reading that article. In that article, the author talked about an investing strategy that is slow yet efficient. I understand the strategy very well, but could not recall its name. I read that on a Friday night.
- Q3** I was exploring a Github project. It was something related to LLMs. I was just skimming quickly through the text, and went through the figure on its architecture and pseudo-code. I think LLM was used for visual encoding in that project. It was on a Friday night.
- Q4** I was looking at a news about an AI startup called Anthropic having a partnership with Google. I did not read it thoroughly. Just knowing that the company gained access to a new chip on Google Cloud to deploy its chatbot named Claude. It was on a Thursday afternoon.
- Q5** It was a very cool finding about light that interests me a lot. Two chinese researchers found that light could affect water evaporation, through a solar simulation. I remembered most of the simulation, except the material used. It spells similar to Jelly? I recall reading it on Saturday.
- Q6** One of the biggest tech event this year – OpenAI Devday, in which a new API was released for developers to create their own assistant. These assistants are now having access to call many new tools. I recall reading through those tools and understanding them very well, and I just want to check them back again.
- Q7** I recall seeing this person name while doing daily quiz on Bing. He is an actor and all three questions in the quiz are related to his movies. I had to go through his wikipedia page to find the answers. There was so much information on the page that I cannot comprehend given a short time. It was nearly midnight when I attempted the quiz"
- Q8** I remember very clearly that I read an article about solana project, which I

have been investing in for a long time, announcing that it will partner with a milk tea store. Solana price went up after that. I want to recall the store name. I read that on a Friday night.

**Q9** I remember seeing a post about Huggingface. No, it's not just huggingface but also also pages which release dataset. There was a concern about licensing. I can't recall the details since I only paid medium attention at that time. I'm trying to recall the percentage of dataset with unspecified license on huggingface. I read that on an evening.

**Q10** Reading daily news in my Gmail. The theme was about frontend development (CSS, javascript, ...). I recall clearly that I saw a significant improvement was introduced to Next.js, which enhances local deployment and production cold start. What was the feature's name? I read that in the afternoon.

Table A.1: Description of the topics used in the dataset. Topics showing *No* in *Train* column belong to the test set in the analysis in Section 5.3.3

Topic	Description	Top 5 Keywords	Train
1	The texts mainly focus on the different topics related to university, students and education.	students, college, education, student, university	Yes
2	The texts are about the students' school life, teaching and learning.	school, high, teacher, teachers, schools	No
7	The texts are related to animals (e.g. their life, their abilities)	animals, animal, elephants, wild, zoo	No
9	The texts mainly focus on the public transportation, especially trains.	train, london, station, travel, bus	No
11	The texts are about musics related, in which most of the articles describe bibliography of singers, composers, and bands.	music, songs, song, festival, listening	Yes
16	The texts are related to energy in general (e.g. green energy, clean energy, source of energy).	energy, pollution, air, oil, wind	Yes
19	The texts mainly discussing our sleep with most of the articles are about the study conducted to research sleep.	sleep, night, sleeping, hours, bed	Yes
24	The texts are the story around cars and driving cars.	car, cars, road, driving, traffic	Yes
29	The texts are mainly related to arts, spanning different genres of art, history, galleries and exhibitions.	art, paintings, artists, painting, artist	No
37	The texts mainly focus on discussing climate change and global warming, and how the wildlife is affected.	ice, sea, scientists, antarctica, climate	No
40	The texts are mainly about stress, mental health and emotional health.	stress, health, mental, anxiety, life	No
41	The texts are related to pets, the stories about them and their abilities.	dog, dogs, cat, pet, pets	Yes

Table A.3: Illustration of text data split. Each cell in the table represents a set of 4 texts, which correspond to 4 reading conditions (reading, scanning, skimming, proofreading).

Split	Topic	Text group											
		Cons.			Incons.			Incons.			Incons.		
Train	1	All	S0 + S1	S1 + S2	S2 + S3	S3 + S4	S4 + S5	S5 + S6	S6 + S7	S7 + S8	S8 + S9	S9 + S0	
	2	All	S0 + S1	S1 + S2	S2 + S3	S3 + S4	S4 + S5	S5 + S6	S6 + S7	S7 + S8	S8 + S9	S9 + S0	
	3	All	S0 + S1	S1 + S2	S2 + S3	S3 + S4	S4 + S5	S5 + S6	S6 + S7	S7 + S8	S8 + S9	S9 + S0	
	4	All	S0 + S1	S1 + S2	S2 + S3	S3 + S4	S4 + S5	S5 + S6	S6 + S7	S7 + S8	S8 + S9	S9 + S0	
	5	All	S0 + S1	S1 + S2	S2 + S3	S3 + S4	S4 + S5	S5 + S6	S6 + S7	S7 + S8	S8 + S9	S9 + S0	
	6	All	S0 + S1	S1 + S2	S2 + S3	S3 + S4	S4 + S5	S5 + S6	S6 + S7	S7 + S8	S8 + S9	S9 + S0	
Test	7	None	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	
	8	-	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	
	9	-	S0	S1	S2	S2	S4	S5	S6	S7	S8	S9	
	10	-	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	
	11	-	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	
	12	-	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	

# Bibliography

- [1] Aaron Duane, Cathal Gurrin, and Wolfgang Hürst. Virtual reality lifelog explorer: Lifelog search challenge at acm icmr 2018. In *LSC '18*, 2018.
- [2] Ly-Duyen Tran, Manh-Duy Nguyen, Nguyen Thanh Binh, Hyowon Lee, and Cathal Gurrin. Myscéal: An experimental interactive lifelog retrieval system for lsc'20. *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*, 2020.
- [3] Ly-Duyen Tran, Manh-Duy Nguyen, Nguyen Thanh Binh, Hyowon Lee, and Cathal Gurrin. Myscéal 2.0: a revised experimental interactive lifelog retrieval system for lsc'21. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, pages 11–16. 2021.
- [4] Ly-Duyen Tran, Manh-Duy Nguyen, Binh Nguyen, Hyowon Lee, Liting Zhou, and Cathal Gurrin. E-myscéal: Embedding-based interactive lifelog retrieval system for lsc'22. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, pages 32–37. 2022.
- [5] Naushad Alam, Yvette Graham, and Cathal Gurrin. Memento: A prototype lifelog search engine for lsc'21. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, LSC '21, page 53–58, New York, NY, USA, 2021. Association for Computing Machinery.
- [6] Nhat Hoang-Xuan, Hoang-Phuc Trang-Trung, E-Ro Nguyen, Thanh-Cong Le, Mai-Khiem Tran, Tu-Khiem Le, Van-Tu Ninh, Cathal Gurrin, and Minh-Triet Tran. Flexible interactive retrieval system 3.0 for visual lifelog exploration at lsc 2022. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, pages 20–26. 2022.



- [7] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. Voxento 3.0: A prototype voice-controlled interactive search engine for lifelog. In *Proceedings of the 5th Annual on Lifelog Search Challenge, LSC '22*, page 43–47, New York, NY, USA, 2022. Association for Computing Machinery.
- [8] Chakravanti Rajagopalachari Kothari. *Research methodology*. new Age, 2004.
- [9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [10] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. Lifelogging: Personal big data. *Found. Trends Inf. Retr.*, 8(1):1–125, jun 2014.
- [11] Tiago Forte. *Building a Second Brain: A Proven Method to Organise Your Digital Life and Unlock Your Creative Potential*. Profile Books, 2022.
- [12] William Jones. *Keeping Found Things Found: The Study and Practice of Personal Information Management*. Morgan Kaufmann, 2010.
- [13] Susan T. Dumais, Edward Cutrell, Jonathan J. Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff i’ve seen: A system for personal information retrieval and re-use. volume 49, pages 28–35, 2015.
- [14] Jidong Chen, Hang Guo, Wentao Wu, and Wei Wang. imecho: an associative memory based desktop search system. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009)*, pages 731–740, November 2009.
- [15] Eemil Lagerspetz, Tancred Lindholm, and Sasu Tarkoma. Dessy: Towards flexible mobile desktop search. In *Proceedings of the DIALM-POMC International Workshop on Foundations of Mobile Computing, Portland, Oregon, USA, August 16, 2007, CD-ROM*. ACM, 2007.

- [16] Georg Buscher, Andreas Dengel, Ralf Biedert, and Ludger V. Elst. Attentive documents: Eye tracking as implicit feedback for information retrieval and beyond. 1(2), January 2012.
- [17] Zhen Liang, Hong Fu, Yun Zhang, Zheru Chi, and Dagan Feng. Content-based image retrieval using a combination of visual features and eye tracking data. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, page 41–44, New York, NY, USA, 2010. Association for Computing Machinery.
- [18] Ying Zhou, Jiajun Wang, and Zheru Chi. Content-based image retrieval based on eye-tracking. COGAIN '18, New York, NY, USA, 2018. Association for Computing Machinery.
- [19] Qingyong Li, Mei Tian, Jun Liu, and Jinrui Sun. A novel image retrieval system with real-time eye tracking. In *Proceedings of International Conference on Internet Multimedia Computing and Service*, ICIMCS '14, page 101–106, New York, NY, USA, 2014. Association for Computing Machinery.
- [20] Martin Dodge and Rob Kitchin. ‘outlines of a world coming into existence’: pervasive computing and the ethics of forgetting. *Environment and planning B: planning and design*, 34(3):431–445, 2007.
- [21] Cathal Gurrin, Rami Albatat, Hideo Joho, and Kaori Ishii. *A privacy by design approach to lifelogging*, pages 49–73. IOS Press, 2014.
- [22] Bush Vannevar. As we may think. *ACM Sigpc Notes*, 1979.
- [23] Wikipedia. Dymaxion chronofile — Wikipedia, the free encyclopedia, 2023. [Online; accessed 13-November-2023].
- [24] Gordon Bell. A personal digital store. *Commun. ACM*, 44:86–91, 2001.
- [25] Jim Gemmell, Gordon Bell, Roger Lueder, Steven M. Drucker, and Curtis Wong. Mylifebits: fulfilling the memex vision. In *MULTIMEDIA '02*, 2002.

- [26] Jim Gemmell, Gordon Bell, and Roger Lueder. Mylifebits: a personal database for everything. *Commun. ACM*, 49:88–95, 2006.
- [27] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. Ntcir lifelog: The first test collection for lifelog research. pages 705–708, 07 2016.
- [28] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. Overview of ntcir-12 lifelog task. In *NTCIR Conference on Evaluation of Information Access Technologies*, 2016.
- [29] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, Duc-Tien Dang-Nguyen, Rashmi Gupta, and Rami Albatal. Overview of ntcir-13 lifelog-2 task. In *NTCIR Conference on Evaluation of Information Access Technologies*, 2017.
- [30] Cathal Gurrin, H. Joho, Frank Hopfgartner, Liting Zhou, Tu Ninh, Tu-Khiem Le, Rami Albatal, Duc-Tien Dang-Nguyen, and Graham Healy. Overview of the ntcir-14 lifelog-3 task. 06 2019.
- [31] Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Liting Zhou, Matthias Lux, Minh-Triet Tran, Tu-Khiem Le, Van-Tu Ninh, and Cathal Gurrin. Overview of imagecleflifelog 2019: Solve my life puzzle and lifelog moment retrieval. In *CLEF*, 2019.
- [32] Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Liting Zhou, Matthias Lux, and Cathal Gurrin. Overview of imagecleflifelog 2018: Daily living understanding and lifelog moment retrieval. In *CLEF*, 2018.
- [33] Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Liting Zhou, Mathias Lux, Minh-Triet Tran, Tu-Khiem Le, Van-Tu Ninh, and Cathal Gurrin. Overview of imagecleflifelog 2019: Solve my life puzzle and lifelog moment retrieval. In *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*, volume 2380 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.

- [34] Van-Tu Ninh, Tu-Khiem Le, Liting Zhou, Luca Piras, Michael Riegler, Pål Halvorsen, Mathias Lux, Minh-Triet Tran, Cathal Gurrin, and Duc-Tien Dang-Nguyen. Overview of imageclef lifelog 2020: Lifelog moment retrieval and sport performance lifelog. In *Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020*, volume 2696 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020.
- [35] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Andreas Leibetseder, Liting Zhou, Aaron Duane, Duc Tien Dang Nguyen, Michael Riegler, Luca Piras, Minh-Triet Tran, Jakub Lokoč, and Wolfgang Hürst. [invited papers] comparing approaches to interactive lifelog search at the lifelog search challenge (lsc2018). *ITE Transactions on Media Technology and Applications*, 7:46–59, 04 2019.
- [36] Cathal Gurrin, Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn Þór Jónsson, Jakub Lokoč, Wolfgang Hürst, Minh-Triet Tran, and Klaus Schöffmann. Introduction to the third annual lifelog search challenge (lsc'20). In *Proceedings of the 2020 International Conference on Multimedia Retrieval, ICMR '20*, page 584–585, New York, NY, USA, 2020. Association for Computing Machinery.
- [37] Cathal Gurrin, Björn Þór Jónsson, Klaus Schöffmann, Duc-Tien Dang-Nguyen, Jakub Lokoč, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Graham Healy. Introduction to the fourth annual lifelog search challenge, lsc'21. New York, NY, USA, 2021. Association for Computing Machinery.
- [38] Ly-Duyen Tran, Manh-Duy Nguyen, Duc-Tien Dang-Nguyen, Silvan Heller, Florian Spiess, Jakub Lokoč, Ladislav Peška, Thao-Nhu Nguyen, Omar Shahbaz Khan, Aaron Duane, Björn Þór Jónsson, Luca Rossetto, An-Zi Yen, Ahmed Alateeq, Naushad Alam, Minh-Triet Tran, Graham Healy, Klaus Schoeffmann, and Cathal Gurrin. Comparing interactive retrieval approaches at the lifelog search challenge 2021. *IEEE Access*, 11:30982–30995, 2023.

- [39] Cathal Gurrin, Liting Zhou, Graham Healy, Björn Þór Jónsson, Duc-Tien Dang-Nguyen, Jakub Lokoć, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Klaus Schöffmann. Introduction to the fifth annual lifelog search challenge, lsc'22. In *Proceedings of the 2022 International Conference on Multimedia Retrieval, ICMR '22*, page 685–687, New York, NY, USA, 2022. Association for Computing Machinery.
- [40] Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Minh-Triet Tran, Liting Zhou, Pablo Redondo, Sinead Smyth, and Cathal Gurrin. Lifeseeker: Interactive lifelog search engine at lsc 2019. *LSC '19*, page 37–40, New York, NY, USA, 2019. Association for Computing Machinery.
- [41] Tu-Khiem Le, Van-Tu Ninh, Minh-Triet Tran, Thanh-An Nguyen, Hai-Dang Nguyen, Liting Zhou, Graham Healy, and Cathal Gurrin. *LifeSeeker 2.0: Interactive Lifelog Search Engine at LSC 2020*, page 57–62. Association for Computing Machinery, New York, NY, USA, 2020.
- [42] Thao-Nhu Nguyen, Tu-Khiem Le, Van-Tu Ninh, Minh-Triet Tran, Nguyen Thanh Binh, Graham Healy, Annalina Caputo, and Cathal Gurrin. Lifeseeker 3.0: An interactive lifelog search engine for lsc'21. In *Proceedings of the 4th Annual on Lifelog Search Challenge, LSC '21*, page 41–46, New York, NY, USA, 2021. Association for Computing Machinery.
- [43] Thao-Nhu Nguyen, Tu-Khiem Le, Van-Tu Ninh, Minh-Triet Tran, Thanh Binh Nguyen, Graham Healy, Sinéad Smyth, Annalina Caputo, and Cathal Gurrin. Lifeseeker 4.0: An interactive lifelog search engine for lsc'22. In *Proceedings of the 5th Annual on Lifelog Search Challenge, LSC '22*, page 14–19, New York, NY, USA, 2022. Association for Computing Machinery.
- [44] Thao-Nhu Nguyen, Tu-Khiem Le, Van-Tu Ninh, A. Caputo, Graham Healy, Sinéad Smyth, Minh-Triet Tran, and Nguyen Thanh Binh. Lifeseeker: an

- interactive concept-based retrieval system for lifelog data. *Multimedia Tools and Applications*, 82:37855 – 37876, 2023.
- [45] Mark Hughes, Eamonn Newman, Alan F. Smeaton, and Noel O’Connor. A lifelogging approach to automated market research. In *Proceedings of the SenseCam Symposium, Oxford, UK*, 2012.
- [46] Kiyoharu Aizawa, Yuto Maruyama, He Li, and Chamin Morikawa. Food balance estimation by using personal dietary tendencies in a multimedia food log. *IEEE Trans. Multim.*, 15(8):2176–2185, 2013.
- [47] Louise N. Signal, Moira B. Smith, Michelle Barr, James Stanley, Tim J. Chambers, Jiang Zhou, Aaron Duane, Gabrielle L.S. Jenkin, Amber L. Pearson, Cathal Gurrin, Alan F. Smeaton, Janet Hoek, and Cliona Ni Mhurchu. Kids’cam: An objective methodology to study the world in which children live. *American Journal of Preventive Medicine*, 53(3):e89–e95, 2017.
- [48] Seyed Ali Bahreinian. *Just-in-time information retrieval and summarization for personal assistance*. PhD thesis, 2019.
- [49] Liting Zhou, Jianquan Liu, Shoji Nishimura, Joseph Antony, and Cathal Gurrin. Causality inspired retrieval of human-object interactions from video. In *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2019.
- [50] Minh-Son Dao, Peijiang Zhao, Tomohiro Sato, Koji Zettsu, Duc-Tien Dang-Nguyen, Cathal Gurrin, and Ngoc-Thanh Nguyen. Overview of mediaeval 2019: Insights for wellbeing taskmultimodal personal health lifelog data analysis. In *Working Notes Proceedings of the MediaEval 2019 Workshop, Sophia Antipolis, France, 27-30 October 2019*, volume 2670 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019.
- [51] Zaher Hinbarji, Rami Albatal, Noel E. O’Connor, and Cathal Gurrin. Loggerman, a comprehensive logging and visualization tool to capture

- computer usage. In *MultiMedia Modeling - 22nd International Conference, MMM 2016, Miami, FL, USA, January 4-6, 2016, Proceedings, Part II*, volume 9517 of *Lecture Notes in Computer Science*, pages 342–347. Springer, 2016.
- [52] Mark Lansdale. The psychology of personal information management. *Applied Ergonomics*, 19(1):55–66, 1988.
- [53] William Jones, Susan T. Dumais, and Harry Bruce. Once found, what then? A study of "keeping" behaviors in the personal use of web information. In *Information, Connetcitons and Community - Proceedings of the 65th ASIS&T Annual Meeting, ASIST 2002, Philadelphia, PA, USA, November 18-21, 2002*, volume 39 of *Proceedings of the Association for Information Science and Technology*, pages 391–402. Wiley, 2002.
- [54] Deborah Barreau and Bonnie A. Nardi. Finding and reminding: File organization from the desktop. *SIGCHI Bull.*, 27(3):39–43, July 1995.
- [55] Steve Whittaker and Candace L. Sidner. Email overload: Exploring personal information management of email. In *Conference on Human Factors in Computing Systems: Common Ground, CHI '96, Vancouver, BC, Canada, April 13-18, 1996, Proceedings*, pages 276–283. ACM, 1996.
- [56] Byron Reeves, Nilam Ram, Thomas N. Robinson, James J. Cummings, C. Lee Giles, Jennifer Pan, Agnese Chiatti, Mj Cho, Katie Roehrick, Xiao Yang, Anupriya Gagneja, Miriam Brinberg, Daniel Muise, Yingdan Lu, Mufan Luo, Andrew Fitzgerald, and Leo Yeykelis. Screenomics: A framework to capture and analyze personal life experiences and the ways that technology shapes them. *Human-Computer Interaction*, 36(2):150–201, 2021. PMID: 33867652.
- [57] Nilam Ram, Xiao Yang, Mu-Jung Cho, Miriam Brinberg, Fiona Muirhead, Byron Reeves, and Thomas N. Robinson. Screenomics: A new approach for observing and studying individuals' digital lives. *Journal of Adolescent Research*, 35(1):16–50, 2020.

- [58] Keith Rayner, Alexander Pollatsek, Jane Ashby, and Charles Clifton Jr. *Psychology of reading*. Psychology Press, 2012.
- [59] Walter Kintsch. *Comprehension: A paradigm for cognition*. Cambridge university press, 1998.
- [60] Denis Pelli and Katharine Tillman. The uncrowded window of object recognition. *Nature neuroscience*, 11:1129–35, 11 2008.
- [61] Linnea C. Ehri. Learning to read words: Theory, findings, and issues. *Scientific Studies of Reading*, 9(2):167–188, 2005.
- [62] Walter Kintsch. The role of knowledge in discourse comprehension: a construction-integration model. *Psychological review*, 95(2):163, 1988.
- [63] Max Roser and Esteban Ortiz-Ospina. Global education. *Our World in Data*, 2016. <https://ourworldindata.org/global-education>.
- [64] Danielle S. McNamara and Walter Kintsch. Learning from texts: Effects of prior knowledge and text coherence. *Discourse Processes*, 22(3):247–288, 1996.
- [65] Richard C. Anderson and P. David Pearson. *A schema-theoretic view of basic processes in reading comprehension*, page 37–55. Cambridge Applied Linguistics. Cambridge University Press, 1988.
- [66] Janette K. Klingner. Assessing reading comprehension. *Assessment for Effective Intervention*, 29(4):59–70, 2004.
- [67] Keith Rayner. Eye movements in reading and information processing: 20 years of research. *Psychological bulletin*, 124 3:372–422, 1998.
- [68] 1870-1913 Huey, Edmund Burke. *The Psychology and pedagogy of reading : with a review of the history of reading and writing and of methods, texts, and hygiene in reading*. Macmillan, 1908.



- [69] Miles A Tinker. The study of eye movements in reading. *Psychological Bulletin*, 43(6):387–422, 1946.
- [70] Guy Thomas Buswell. How people look at pictures: A study of the psychology of perception in art. *The Elementary School Journal*, 37(1):15–26, 1935.
- [71] Ralf Biedert, Georg Buscher, and Andreas Dengel. The eye book. *Informatik-Spektrum*, 33(3):272–281, 2009.
- [72] Erik D Reichle, Keith Rayner, and Alexander Pollatsek. The ez reader model of eye-movement control in reading: Comparisons to other models. *Behavioral and brain sciences*, 26(4):445–476, 2003.
- [73] Ralf Biedert, Jörn Hees, Andreas R. Dengel, and Georg Buscher. A robust realtime reading-skimming classifier. *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2012.
- [74] Wen-Hung Liao, Chin-Wen Chang, and Yi-Chieh Wu. Classification of reading patterns based on gaze information. In *2017 IEEE International Symposium on Multimedia (ISM)*, pages 595–600, 2017.
- [75] Alexander Strukelj and Diederick Christian Niehorster. One page of text: Eye movements during regular and thorough reading, skimming, and spell checking. *Journal of Eye Movement Research*, 11, 2018.
- [76] Charles Lima Sanches, Olivier Augereau, and Koichi Kise. Estimation of reading subjective understanding based on eye gaze analysis. *PLOS ONE*, 13(10):1–16, 10 2018.
- [77] Leana Copeland, Tom Gedeon, and B. Sumudu U. Mendis. Predicting reading comprehension scores from eye movements using artificial neural networks and fuzzy output error. *Artif. Intell. Res.*, 3:35–48, 2014.

- [78] Leana Copeland and Tom Gedeon. Measuring reading comprehension using eye movements. In *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*, pages 791–796, 2013.
- [79] Kazuyo Yoshimura, Koichi Kise, and Kai Kunze. The eye as the window of the language ability: Estimation of english skills by analyzing eye movement while reading documents. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 251–255, 2015.
- [80] Rosy Southwell, Julie M. Gregg, Robert Earl Bixler, and Sidney K. D’Mello. What eye movements reveal about later comprehension of long connected texts. *Cognitive science*, 44 10:e12905, 2020.
- [81] Seoyoung Ahn, Conor Kelton, Aruna Balasubramanian, and Gregory J. Zelinsky. Towards predicting reading comprehension from gaze behavior. In Andreas Bulling, Anke Huckauf, Eakta Jain, Ralph Radach, and Daniel Weiskopf, editors, *ETRA ’20: 2020 Symposium on Eye Tracking Research and Applications, Short Papers, Stuttgart, Germany, June 2-5, 2020*, pages 32:1–32:5. ACM, 2020.
- [82] Silvia Makowski, Lena A. Jäger, Ahmed Abdelwahab, Niels Landwehr, and Tobias Scheffer. A discriminative model for identifying readers and assessing text comprehension from eye movements. In Michele Berlingerio, Francesco Bonchi, Thomas Gärtner, Neil Hurley, and Georgiana Ifrim, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 209–225, Cham, 2019. Springer International Publishing.
- [83] Stephen Bottos and Balakumar Balasingam. Tracking the progression of reading through eye-gaze measurements. In *2019 22th International Conference on Information Fusion (FUSION)*, pages 1–8, 2019.

- [84] Jon W Carr, Valentina N Pescuma, Michele Furlan, Maria Ktori, and Davide Crepaldi. Algorithms for the automated correction of vertical drift in eye-tracking data. *Behavior Research Methods*, 54(1):287–310, 2022.
- [85] Liting Zhou, Cathal Gurrin, Graham Healy, Hideo Joho, Thanh-Binh Nguyen, Rami Albatal, Frank Hopfgartner, and Duc-Tien Dang-Nguyen. Overview of the ntcir-16 lifelog-4 task. In *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies*, pages 130–135. National Institute of Informatics, 2022.
- [86] Cathal Gurrin, Björn Pór Jónsson, Duc Tien Dang Nguyen, Graham Healy, Jakub Lokoc, Liting Zhou, Luca Rossetto, Minh-Triet Tran, Wolfgang Hürst, Werner Bailer, and Klaus Schoeffmann. Introduction to the sixth annual lifelog search challenge, lsc’23. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval, ICMR ’23*, page 678–679, New York, NY, USA, 2023. Association for Computing Machinery.
- [87] Jakub Lokoč, František Mejzlik, Patrik Veselý, and Tomáš Souček. Enhanced somhunter for known-item search in lifelog data. In *Proceedings of the 4th Annual on Lifelog Search Challenge, LSC ’21*, page 71–73, New York, NY, USA, 2021. Association for Computing Machinery.
- [88] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. Voxento 2.0: A prototype voice-controlled interactive search engine for lifelogs. In *Proceedings of the 4th Annual on Lifelog Search Challenge, LSC ’21*, page 65–70, New York, NY, USA, 2021. Association for Computing Machinery.
- [89] Naushad Alam, Yvette Graham, and Cathal Gurrin. Memento 2.0: An improved lifelog search engine for lsc’22. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, pages 2–7. 2022.
- [90] Hoang-Phuc Trang-Trung, Thanh-Cong Le, Mai-Khiem Tran, Van-Tu Ninh, Tu-Khiem Le, Cathal Gurrin, and Minh-Triet Tran. Flexible interactive

- retrieval system 2.0 for visual lifelog exploration at lsc 2021. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, LSC '21, page 81–87, New York, NY, USA, 2021. Association for Computing Machinery.
- [91] Wei-Hong Ang, An-Zi Yen, Tai-Te Chu, Hen-Hsen Huang, and Hsin-Hsi Chen. Lifeconcept: An interactive approach for multimodal lifelog retrieval through concept recommendation. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, pages 47–51. 2021.
- [92] Andreas Leibetseder and Klaus Schoeffmann. Lifexplore at the lifelog search challenge 2021. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, pages 23–28. 2021.
- [93] Andreas Leibetseder, Daniela Stefanics, and Klaus Schoeffmann. Lifexplore at the lifelog search challenge 2022. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, LSC '22, page 48–52, New York, NY, USA, 2022. Association for Computing Machinery.
- [94] Alexander Christian Faisst and Björn Þór Jónsson. Lifemon: A mongodb-based lifelog retrieval prototype. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, pages 75–80, 2021.
- [95] Silvan Heller, Ralph Gasser, Mahnaz Parian-Scherb, Sanja Popovic, Luca Rossetto, Loris Sauter, Florian Spiess, and Heiko Schuldt. Interactive multimodal lifelog retrieval with vitrivr at lsc 2021. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, LSC '21, page 35–39, New York, NY, USA, 2021. Association for Computing Machinery.
- [96] Silvan Heller, Luca Rossetto, Loris Sauter, and Heiko Schuldt. Vitrivr at the lifelog search challenge 2022. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, LSC '22, page 27–31, New York, NY, USA, 2022. Association for Computing Machinery.

- [97] Florian Spiess, Ralph Gasser, Silvan Heller, Luca Rossetto, Loris Sauter, Milan van Zanten, and Heiko Schuldt. Exploring intuitive lifelog retrieval and interaction modes in virtual reality with vitrivr-vr. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, LSC '21, page 17–22, New York, NY, USA, 2021. Association for Computing Machinery.
- [98] Florian Spiess and Heiko Schuldt. Multimodal interactive lifelog retrieval with vitrivr-vr. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, LSC '22, page 38–42, New York, NY, USA, 2022. Association for Computing Machinery.
- [99] Emil Knudsen, Thomas Holstein Qvortrup, Omar Shahbaz Khan, and Björn Þór Jónsson. Xqc at the lifelog search challenge 2021: Interactive learning on a mobile device. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, pages 89–93. ACM, 2021.
- [100] Omar Shahbaz Khan, Aaron Duane, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. Exquisitor at the lifelog search challenge 2021: Relationships between semantic classifiers. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, pages 3–6. ACM, 2021.
- [101] Jihye Shin, Alexandra Waldau, Aaron Duane, and Björn Þór Jónsson. Photocube at the lifelog search challenge 2021. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, pages 59–63. ACM, 2021.
- [102] Aaron Duane and Bjorn Þór Jónsson. Virma: Virtual reality multimedia analytics at lsc 2021. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, LSC '21, page 29–34, New York, NY, USA, 2021. Association for Computing Machinery.
- [103] Luca Rossetto, Matthias Baumgartner, Ralph Gasser, Lucien Heitz, Ruijie Wang, and Abraham Bernstein. Exploring graph-querying approaches in lifegraph. In *Proceedings of the 4th Annual on Lifelog Search Challenge*,

- LSC '21, page 7–10, New York, NY, USA, 2021. Association for Computing Machinery.
- [104] Ricardo Ribiero, Alina Trifan, and Antonio J. R. Neves. Memoria: A memory enhancement and moment retrieval application for lsc 2022. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, LSC '22, page 8–13, New York, NY, USA, 2022. Association for Computing Machinery.
- [105] L. Rossetto, R. Gasser, S. Heller, Mahnaz Parian, and H. Schuldt. Retrieval of structured and unstructured data with vitrivr. In *LSC '19*, 2019.
- [106] Ly-Duyen Tran, Manh-Duy Nguyen, Nguyen Thanh Binh, Hyowon Lee, and Cathal Gurrin. Myscéal 2.0: A revised experimental interactive lifelog retrieval system for lsc'21. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, LSC '21, page 11–16, New York, NY, USA, 2021. Association for Computing Machinery.
- [107] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [108] Luca Rossetto, Ivan Giangreco, Claudiu Tanase, and Heiko Schuldt. Vitrivr: A flexible retrieval stack supporting multiple query modes for searching in multimedia collections. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, page 1183–1186, New York, NY, USA, 2016. Association for Computing Machinery.
- [109] Ivan Giangreco and Heiko Schuldt. Adam pro: database support for big multimedia retrieval. *Datenbank-Spektrum*, 16:17–26, 2016.
- [110] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Computer Vision – ECCV 2018: 15th*

*European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*, page 833–851, Berlin, Heidelberg, 2018. Springer-Verlag.

- [111] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [112] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, nov 1995.
- [113] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. Voxento 2.0: A prototype voice-controlled interactive search engine for lifelogs. In *Proceedings of the 4th Annual on Lifelog Search Challenge, LSC '21*, page 65–70, New York, NY, USA, 2021. Association for Computing Machinery.
- [114] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- [115] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.
- [116] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.
- [117] Ultralytics. YOLOv5. <https://github.com/ultralytics/yolov5>, 2021.
- [118] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE*

- transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.
- [119] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.
- [120] M. Bear, B. Connors, and M.A. Paradiso. *Neuroscience: Exploring the Brain, Enhanced Edition: Exploring the Brain, Enhanced Edition*. Jones & Bartlett Learning, 2020.
- [121] Nicholas R. Galloway, Winfried M. K. Amoaku, Peter H. Galloway, and Andrew C. Browning. *Basic Anatomy and Physiology of the Eye*, pages 7–18. Springer International Publishing, Cham, 2022.
- [122] Meng Zhao and Guang-Hua Peng. Regulatory mechanisms of retinal photoreceptors development at single cell resolution. *International Journal of Molecular Sciences*, 22(16):8357, 2021.
- [123] B. Huey Edmund. Preliminary experiments in the physiology and psychology of reading. *American Journal of Psychology*, 9(4):575–586, 1898.
- [124] Edmund Burke Delabarre. A method of recording eye-movements. *American Journal of Psychology*, 9:572, 1898.
- [125] Raymond Dodge and Thomas Sparks Cline. The angle velocity of eye movements. *Psychological Review*, 8(2):145–157, 1901.
- [126] R. W. Ditchburn and B. L. Ginsborg. Vision with a stabilized retinal image. *Nature*, 170:36–37, 1952.
- [127] Han Collelijn, F. van der Mark, and T. C. Jansen. Precise recording of human eye movements. *Vision Research*, 15:447–IN5, 1975.
- [128] Kenneth Holmqvist. *Eye Tracking : A Comprehensive Guide to Methods and Measures*. Oxford University Press, United Kingdom, 2011.



- [129] Gazepoint gp3 hd eye tracker 150hz, 2023. Accessed: 2023-10-13.
- [130] Keith Rayner. Eye movements and attention in reading, scene perception, and visual search. *The Quarterly Journal of Experimental Psychology*, 62(8):1457–1506, 2009. PMID: 19449261.
- [131] Marcel Adam Just and Patricia A. Carpenter. A theory of reading: from eye fixations to comprehension. *Psychological review*, 87 4:329–54, 1980.
- [132] Kai Kunze, Hitoshi Kawaichi, Kazuyo Yoshimura, and Koichi Kise. The wordometer - estimating the number of words read using document image retrieval and mobile eye tracking. In *12th International Conference on Document Analysis and Recognition, ICDAR 2013, Washington, DC, USA, August 25-28, 2013*, pages 25–29. IEEE Computer Society, 2013.
- [133] Kai Kunze, Masai Katsutoshi, Yuji Uema, and Masahiko Inami. How much do you read? counting the number of words a user reads using electrooculography. In *Proceedings of the 6th Augmented Human International Conference, AH '15*, page 125–128, New York, NY, USA, 2015. Association for Computing Machinery.
- [134] Shoya Ishimaru, Kai Kunze, Koichi Kise, and Andreas Dengel. The wordometer 2.0: Estimating the number of words you read in real life using commercial eog glasses. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, UbiComp '16*, page 293–296, New York, NY, USA, 2016. Association for Computing Machinery.
- [135] Andreas Bulling, Jamie A. Ward, Hans Gellersen, and Gerhard Tröster. Eye movement analysis for activity recognition using electrooculography. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):741–753, 2011.

- [136] Keith Rayner, Kathryn H. Chace, Timothy J. Slattery, and Jane Ashby. Eye movements as reflections of comprehension processes in reading. *Scientific Studies of Reading*, 10(3):241–255, 2006.
- [137] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [138] William L Carlson and Betty Thorne. Applied statistical methods: for business, economics, and the social sciences. (*No Title*), 1997.
- [139] Ronald Aylmer Fisher. Statistical methods for research workers. In *Breakthroughs in statistics: Methodology and distribution*, pages 66–70. Springer, 1970.
- [140] Samuel Sanford Shapiro and Martin B Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.
- [141] Maurice Stevenson Bartlett. Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 160(901):268–282, 1937.
- [142] William H Kruskal and W Allen Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260):583–621, 1952.
- [143] Student. The probable error of a mean. *Biometrika*, pages 1–25, 1908.

- [144] Carlo Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R istituto superiore di scienze economiche e commerciali di firenze*, 8:3–62, 1936.
- [145] William J Conover and Ronald L Iman. Rank transformations as a bridge between parametric and nonparametric statistics. *The American Statistician*, 35(3):124–129, 1981.
- [146] Karl Pearson. Vii. mathematical contributions to the theory of evolution.—iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, (187):253–318, 1896.
- [147] Charles Spearman. The proof and measurement of association between two things. 1961.
- [148] Ranjit Kumar. Research methodology: A step-by-step guide for beginners. *Research methodology*, pages 1–528, 2018.
- [149] Oleksii Kononenko, Olga Baysal, Reid Holmes, and Michael W Godfrey. Mining modern repositories with elasticsearch. In *Proceedings of the 11th working conference on mining software repositories*, pages 328–331, 2014.
- [150] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. *Advances in neural information processing systems*, 31, 2018.
- [151] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [152] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

- [153] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017.
- [154] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoon Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374, 2019.
- [155] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [156] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, and Charles Xie. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data, SIGMOD '21*, page 2614–2627, New York, NY, USA, 2021. Association for Computing Machinery.
- [157] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-YOLOv4: Scaling cross stage partial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13029–13038, June 2021.
- [158] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15*, page 91–99, Cambridge, MA, USA, 2015. MIT Press.

- [159] Norman Mu, Alexander Kirillov, David A. Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. *ArXiv*, abs/2112.12750, 2021.
- [160] Evgeny Izutov. Ligar: Lightweight general-purpose action recognition. *ArXiv*, abs/2108.13153, 2021.
- [161] Ron Mokady, Amir Hertz, and Amit H. Bermano. Clipcap: Clip prefix for image captioning. *ArXiv*, abs/2111.09734, 2021.
- [162] Tu-Khiem Le, Van-Tu Ninh, Mai-Khiem Tran, Graham Healy, Cathal Gurrin, and Minh-Triet Tran. Avseeker: an active video retrieval engine at vbs2022. In *International Conference on Multimedia Modeling*, pages 537–542. Springer, 2022.
- [163] Jakub Lokoč, Stelios Andreadis, Werner Bailer, Aaron Duane, Cathal Gurrin, Zhixin Ma, Nicola Messina, Thao-Nhu Nguyen, Ladislav Peška, Luca Rossetto, et al. Interactive video retrieval in the age of effective joint embedding deep models: lessons from the 11th vbs. *Multimedia Systems*, 29(6):3481–3504, 2023.
- [164] Edward A. Fox and Joseph A. Shaw. Combination of multiple searches. In *TREC*, volume 500-215 of *NIST Special Publication*, pages 243–252. National Institute of Standards and Technology (NIST), 1993.
- [165] Elias Bassani and Luca Romelli. Ranx.fuse: A python library for metasearch. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22*, page 4808–4812, New York, NY, USA, 2022. Association for Computing Machinery.
- [166] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*,

- pages 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [167] Claude Sammut and Geoffrey I. Webb, editors. *TF-IDF*, pages 986–987. Springer US, Boston, MA, 2010.
- [168] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.
- [169] John Peirce, Jeremy R. Gray, Sol Simpson, Michael R. MacAskill, Richard Höchenberger, Hiroyuki Sogo, Erik K. Kastman, and Jonas Kristoffer Lindeløv. Psychopy2: Experiments in behavior made easy. *Behavior Research Methods*, 51:195 – 203, 2019.
- [170] Richard R. Day and Jeong suk Park. Developing reading comprehension questions. *Reading in a foreign language*, 17:60–73, 2005.
- [171] Richard Andersson, Linnea Larsson, Kenneth Holmqvist, Martin Stridh, and Marcus Nyström. One algorithm to rule them all? an evaluation and discussion of ten eye movement event-detection algorithms. *Behavior Research Methods*, 49(2):616–637, 2017.
- [172] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [173] Luz Rello, Martin Pielot, and Mari-Carmen Marcos. Make it big! the effect of font size and line spacing on online readability. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI ’16, page 3637–3648, New York, NY, USA, 2016. Association for Computing Machinery.
- [174] Luz Rello and Jeffrey P. Bigham. Good background colors for readers: A study of people with and without dyslexia. In *Proceedings of the 19th International*

- ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '17, page 72–80, New York, NY, USA, 2017. Association for Computing Machinery.
- [175] Rashmi Gupta. Considering documents in lifelog information retrieval. In *2018 International Conference on Multimedia Retrieval (ICMR)*, pages 4–4. IEEE, 2018.
- [176] Duc-Tien Dang-Nguyen, Liting Zhou, Rashmi Gupta, Michael Riegler, and Cathal Gurrin. Building a disclosed lifelog dataset: Challenges, principles and processes. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, CBMI '17*, New York, NY, USA, 2017. Association for Computing Machinery.
- [177] Ray Smith. An overview of the tesseract ocr engine. In *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pages 629–633, Washington, DC, USA, 2007. IEEE Computer Society.
- [178] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
- [179] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.
- [180] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *choice*, 2640:660, 2016.
- [181] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

- [182] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.
- [183] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [184] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [185] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.