Reference Wizard: Leveraging Large Language Models for Automated Bibliography Generation

Mateusz Waloszek

Final-Year Project - BSc in Computer Science

Supervisor: Dr. Klaas-Jan Stol

Department of Computer Science

University College Cork

April 2024

Abstract

The transition from free-form references to structured BiBTeX format is a significant challenge for authors, particularly when converting large volumes of references. Reference Wizard is a comprehensive tool suite designed to aid authors in managing references for academic papers. This project aims to address the challenge of converting free-form references into the BiBTeX format, a requirement for LaTeX typesetting, through the utilization of Large Language Models (LLMs). This report provides an overview of the project, including the problem statement, background research, solution design, implementation, evaluation, and conclusions.

Declaration of Originality

In signing this declaration, you are conforming, in writing, that the submitted work is entirely your own original work, except where clearly attributed otherwise, and that it has not been submitted partly or wholly for any other educational award.

I hereby declare that:

- this is all my own work, unless clearly indicated otherwise, with full and proper accreditation;

- with respect to my own work: none of it has been submitted at any educational institution contributing in any way to an educational award;

- with respect to another's work: all text, diagrams, code, or ideas, whether verbatim, paraphrased or otherwise modified or adapted, have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or in print.

Signed: Mateusz Waloszek
Date: 23/04/2024

Acknowledgments

I would like to thank my supervisor, Dr. Klaas-Jan Stol, for his invaluable guidance and support throughout this project. His expertise and insights have been extremely helpful. I am also immensely grateful to my peers, Karol Machnik and Jaafar Merhi, for their time and effort in proofreading my report. Their feedback was crucial in refining the quality of this document, and their suggestions have greatly enhanced the clarity, integrity, and flow of the text. Thank you all for your assistance, and encouragement which have been greatly appreciated and have significantly contributed to the writing of this report.

Contents

Chapter 1

Introduction

1.1 Problem Statement

Writing in academia involves not only creating new insights but also properly acknowledging existing knowledge to uphold scholarly credibility. Authors have a range of tools and software at their disposal, for creating their work with Microsoft Word and LaTeX being among the most used options. While tools like Microsoft Word allow for relatively straightforward citation management, LaTeX users often rely on BibTeX to manage their bibliographies. BibTeX is a file format that provides a structured format for citations, which is necessary for LaTeX documents, but requires that references be formatted in a specifically formatted style. This poses a challenge, particularly when a reference is sourced from another document or database that does not provide a BibTeX-formatted citation. In these cases, authors must manually convert these references into BibTeX format, a task that becomes difficult with the increase in the number of sources [1]. Also, the varied citation formats used in disciplines introduce an additional level of intricacy resulting in a task that is not just difficult but also susceptible to errors. This process of manually converting large amounts of free-form references into their respective BiBTeX format inhibits the workflow of an author's writing process.

1.1.1 Challenges in Reference Conversion

Converting references into BibTeX format may appear simple at glance, but, it becomes more complex due to the various different formats that references can have. These formats can vary from books and journal articles to conference proceedings and online sources each having its own unique structural components. Take for instance a journal article reference:

Smith, John. ”Innovations in Artificial Intelligence.” Journal of AI Research, vol. 12, no. 4, pp. 123-135, 2020.

Converting this reference into BibTeX format involves identifying and correctly formatting various components into a structured format as shown below:

```
@article{Smith2020Innovations,
    author    = "John Smith",
    title     = "Innovations in Artificial Intelligence",
    journal   = "Journal of AI Research",
    volume    = "12",
    number    = "4",
    pages     = "123-135",
    year      = "2020"
}
```

Distinguishing between types of references, like identifying a book chapter versus a conference paper is easy for humans who understand context and subtle nuances. However it poses a challenge for automated systems which have to analyze and interpret the structure without the comprehension that humans have [2]. This challenge highlights the importance of techniques in automated reference management tools, which need to be precise and flexible, across all formats.

1.2 Introduction to the Project

The ”Reference Wizard” project aims to leverage the capabilities of Large Language Models (LLMs) in natural language understanding and generation to address the challenges of converting free-form references into BiBTeX format [3]. By automating this task, the project's goal is to create an application that can accurately comprehend the intricacies and nuances of citation styles which in return can reduce the effort needed for citation management. Utilizing LLMs provides an opportunity to achieve a level of accuracy and adaptability to human performance crucial for upholding scholarly integrity in academic writing.

This report details the development and implementation of the Reference Wizard. It starts with an analysis of current solutions followed by technologies and methodologies utilized by the ”Reference Wizard”. Later sections provide an overview of the system architecture that supports the application followed by an assessment of the application's performance through accuracy and speed of conversion evaluations, followed by user feedback on their experience based on a questionnaire. Finally, the report concludes with a summary of the discoveries, proposing potential avenues for improvement.

Chapter 2

Background and Related Work

This chapter examines existing systems by looking into their strengths and weaknesses. Through these strengths and weakness, this chapter will identify key features that can be used as a foundation for the Reference Wizard. This chapter also explores recent advances in machine learning and natural language processing (NLP) that highlight the development of tools like the Reference Wizard. With a focus particularly on the use of Large Language Models (LLMs) for semantic parsing and the automation of reference management, highlighting research that contributes to the understanding and capabilities of these technologies. There will be a critical analysis of previous solutions and an exploration of the current state of automated reference management, this chapter will attempt to provide a justification for developing a tool-suite using Large Language Models.

2.1 Review of Literature

2.1.1 Semantic Parsing with Large Language Models

Semantic parsing plays a role in the field of natural language processing by converting language into a structured format that computers can comprehend. Transformer based models like GPT (Generative Pre trained Transformer) have significantly propelled this area forward. Radford et al. (2018) demonstrated the ability of trained language models to achieve top notch results across various NLP tasks without the need for task specific training [4]. In their work, Drozdov et al. (2022) introduced a method for compositional semantic parsing using large language models. Their research highlights how these models can break down expressions, into simpler components which enhances both understanding and generation capabilities [5].

2.1.2 The Impact of Large Language Models on NLP

LLMs, like the GPT series created by OpenAI, have changed the way machines interpret and produce text. Their capability to engage in zero shot learning, where the model deduces tasks without examples is especially groundbreaking for tasks like reference conversion, which involves understanding formatting standards and contexts. In a study by Brown et al. (2020), GPT 3 was introduced as the largest model of its kind at that time. The research highlighted its applicability across domains without requiring fine tuning - an important factor, for applications that demand adaptable ”all purpose” NLP solutions [6].

2.1.3 Challenges in Automated Reference Formatting

Converting free-form references into a structured format like BiBTeX through an automated system requires dealing with semantic and formatting challenges. These challenges arise from differences in citation styles, unclear author naming conventions and inconsistencies in how information is presented across sources [7].

2.1.4 Advances in Reference Management Tools

Since reference management is such an integral to academic writing and research, the shift towards digital libraries and databases has led to the development of various software solutions that assist in managing and formatting bibliographies [8]. The application of Machine Learning (ML) in this area focuses on automating the extraction and formatting of bibliographic data from unstructured text. Cohan et al. (2020) explored how deep learning models can be used for automatic citation recommendation. This shows that neural networks can effectively predict relevant references based on the context of the research paper [9].

2.2 Key Features of Effective Solutions

To become an effective solution, the application or tool should take advantage of a few key features [10]. These features ensure would that the tool is meeting the needs of it's users. The features should not only help with managing a user's bibliography, but also should streamline their academic writing.

2.2.1 Accuracy and Reliability

Tools should correctly interpret citation styles and formats so that references are consistently and reliably handled. This requires identifying and handling components of a free-form reference including author names, publication titles, dates and page numbers with, as little errors, as possible.

2.2.2 Adaptability to Various Citation Styles

With various citation styles used across different academic disciplines, an effective reference management tool must be adaptable, supporting major styles like APA, MLA, Chicago, and others. The ability to customize or add new styles would also be beneficial.

2.2.3 User-Friendly Interface

A user-friendly interface that simplifies the process of managing references is mandatory. Since ease of use encourages adoption and regular use by researchers and authors, an intuitive User Interface (UI) which allows for searching, filtering, an organization of current references and mechanisms for inserting citations into documents key [11].

2.2.4 Integration Capabilities

The tool should be able to connect with popular word processing software, academic databases and search engines. By linking with writing platforms, like LaTeX, Google Docs or Microsoft Word users can conveniently format references, in their texts making the writing process more efficient.

2.2.5 Comprehensive Database and Search Features

An extensive database that allows users to search for and import references directly from within the tool enhances productivity. Features that enable searching by keyword, author, title, or a Digital Object Identifier (DOI) help users quickly find and incorporate relevant literature into their work.

2.2.6 Collaboration and Sharing Functions

The tool should have collaboration features that would allow users to share resources with one another which would enhance the effectiveness teams writing academic papers. This involves functions like leaving comments, adding notes or organizing libraries.

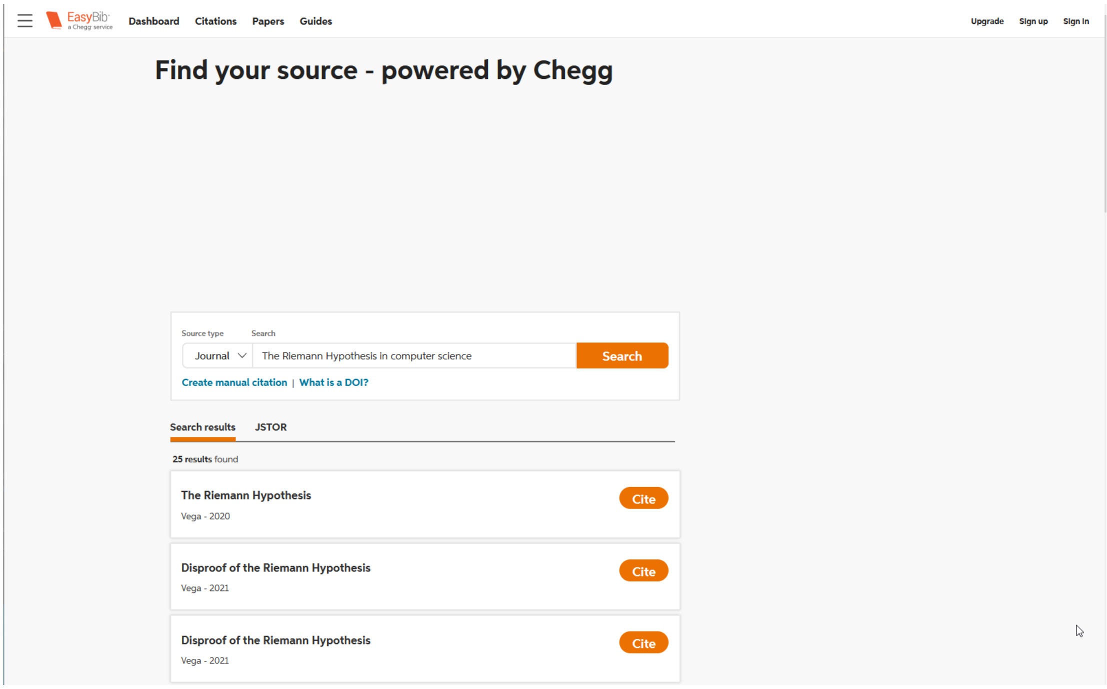
2.3 Analysis of Current Solutions

This section will conduct an examination of the methodologies employed by current tools in parsing and converting references. It will detail the technologies these systems utilize and their integration into academic workflows, providing insight into the current state of reference management systems.

In the realm of reference management tools, a variety of solutions exist, each with its unique strengths and limitations. The comparison below is based on several key criteria: User-Friendliness, Citation Styles, Integration, Database, and Collaboration capabilities. These criteria are crucial for evaluating the effectiveness of each tool in facilitating academic writing and research.

EasyBib

EasyBib is a comprehensive citation management tool that simplifies the process of generating and managing bibliographic data. EasyBib allows users to manually input citation details or search for sources using a title, ISBN, or URL. It then generates citations in various formats, including BibTeX, which is particularly useful for LaTeX users. This feature makes EasyBib a practical tool for quickly creating accurate citations without manually formatting each entry. While EasyBib provides robust support for a variety of citation styles, its integration with document creation tools and collaboration features are somewhat limited compared to more specialized reference management software. However, its user-friendly interface and the ability to quickly generate citations make it a valuable tool for individual researchers and students [12].



Find your source - powered by Chegg

Source type: Journal
Search: The Riemann Hypothesis in computer science

25 results found

The Riemann Hypothesis
Vega - 2020

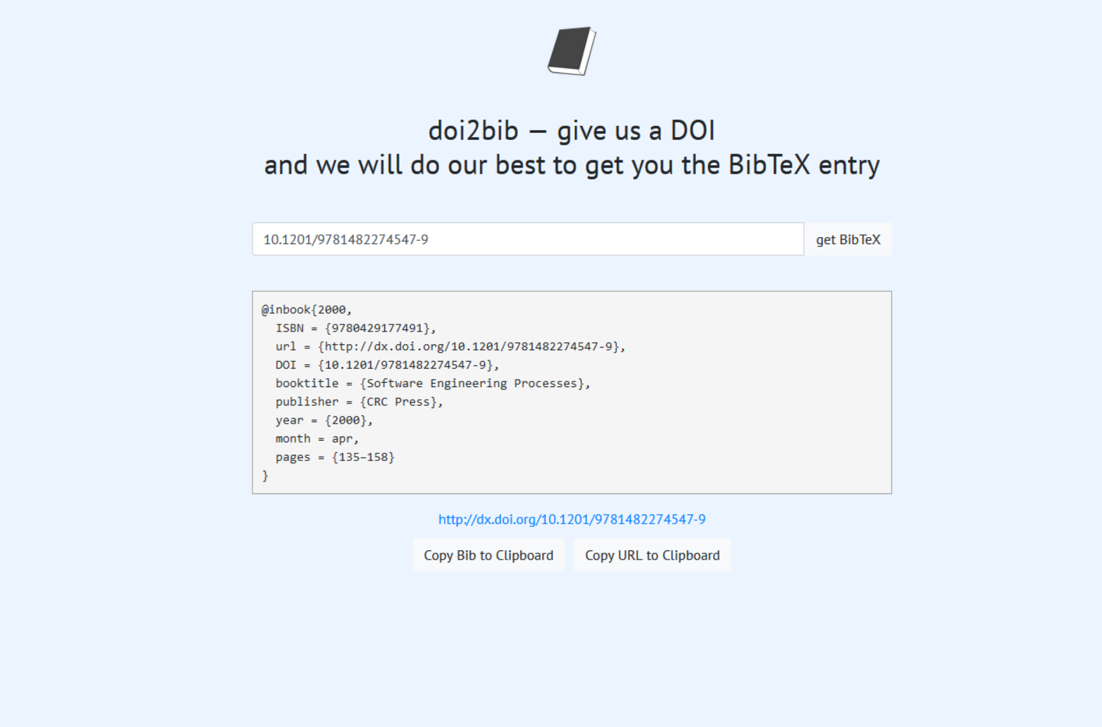
Disproof of the Riemann Hypothesis
Vega - 2021

Disproof of the Riemann Hypothesis
Vega - 2021

Figure 2.1: Example of EasyBib web interface

DOI2Bib

DOI2Bib is a web-based service that allows users to enter a DOI, PMCID, or arXiv ID to instantly retrieve a BiBTeX entry for their reference management needs in LaTeX documents. This tool simplifies the process of converting unique identifiers into formatted BiBTeX references, which is especially useful for users who need to quickly integrate references into their LaTeX projects without manually formatting each entry [13]. DOI2Bib offers a user-friendly interface for converting identifiers into BibTeX entries, making it an efficient tool for those who primarily need quick conversions. While it excels in ease of use, it lacks in citation style variety and does not offer collaboration features or a database for literature search, as it is not the primary function of the service.



doi2bib — give us a DOI
and we will do our best to get you the BibTeX entry

10.1201/9781482274547-9

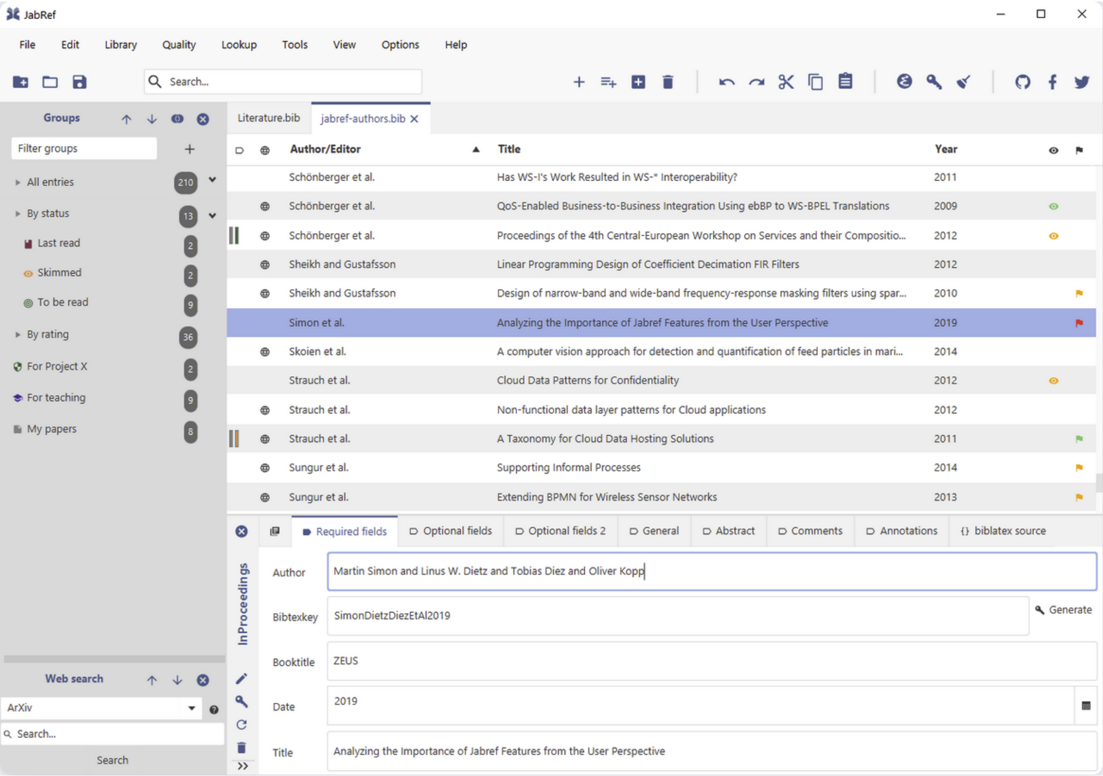
```
@inbook{2000,
  ISBN = {9780429177491},
  url = {http://dx.doi.org/10.1201/9781482274547-9},
  DOI = {10.1201/9781482274547-9},
  booktitle = {Software Engineering Processes},
  publisher = {CRC Press},
  year = {2000},
  month = apr,
  pages = {135–158}
}
```

http://dx.doi.org/10.1201/9781482274547-9

Figure 2.2: Example of DOI2Bib web interface

JabRef

JabRef stands out for its robust BiBTeX management capabilities, directly supporting the creation and management of BiBTeX files used in LaTeX documents. It allows for importing information from various online databases and automatically converts it into BiBTeX format. JabRef's strength lies in its features that include generating BiBTeX entries and also organizing them effectively [14]. Jabref supports a wide range of citation styles and high integration capabilities with scientific databases and LaTeX workflows. Though its collaboration features, while present, are not as developed as some might prefer.



Author/Editor	Title	Year
Schönberger et al.	Has WS-I's Work Resulted in WS-* Interoperability?	2011
Schönberger et al.	QoS-Enabled Business-to-Business Integration Using ebBP to WS-BPEL Translations	2009
Schönberger et al.	Proceedings of the 4th Central-European Workshop on Services and their Compositio...	2012
Sheikh and Gustafsson	Linear Programming Design of Coefficient Decimation FIR Filters	2012
Sheikh and Gustafsson	Design of narrow-band and wide-band frequency-response masking filters using spar...	2010
Simon et al.	Analyzing the Importance of Jabref Features from the User Perspective	2019
Skoien et al.	A computer vision approach for detection and quantification of feed particles in mari...	2014
Strauch et al.	Cloud Data Patterns for Confidentiality	2012
Strauch et al.	Non-functional data layer patterns for Cloud applications	2012
Strauch et al.	A Taxonomy for Cloud Data Hosting Solutions	2011
Sungur et al.	Supporting Informal Processes	2014
Sungur et al.	Extending BPMN for Wireless Sensor Networks	2013

Author: Martin Simon and Linus W. Dietz and Tobias Diez and Oliver Kopp

Bibtexkey: SimonDietzDiezEtAl2019

Booktitle: ZEUS

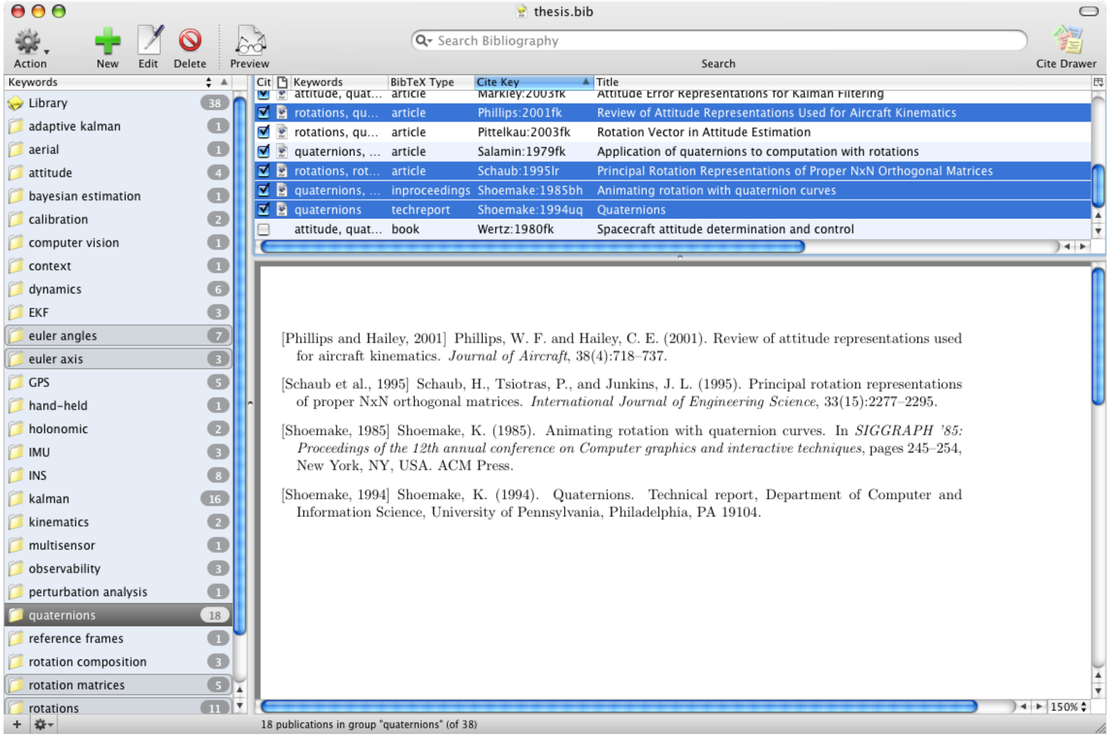
Date: 2019

Title: Analyzing the Importance of Jabref Features from the User Perspective

Figure 2.3: Example of JabRef interface

BibDesk

BibDesk is made for managing bibliographies on macOS and integrates with LaTeX workflows, emphasizing its utility for managing and creating BiBTeX entries. It automates the process of generating citation keys and links PDFs to BiBTeX entries, enhancing the management of academic references within a LaTeX environment. BibDesk's platform-specific availability limits its wider use but does not detract from its effectiveness in handling BiBTeX formats [15]. While it offers an excellent user experience, it is only available on macOS which greatly hurts it's availability for the average user. Its database and citation style support are strong, but it falls slightly behind JabRef in terms of the wideness of its database and collaboration features, which are more limited.



Keywords	
Library	38
adaptive kalman	1
aerial	1
attitude	4
bayesian estimation	1
calibration	2
computer vision	1
context	1
dynamics	6
EKF	3
euler angles	7
euler axis	3
GPS	5
hand-held	1
holonomic	2
IMU	3
INS	8
kalman	16
kinematics	2
multisensor	1
observability	3
perturbation analysis	1
quaternions	18
reference frames	1
rotation composition	3
rotation matrices	5
rotations	11

Cit	Keywords	BibTeX Type	Cite Key	Title
☑	attitude, quat...	article	Markley:2003fk	Attitude Error Representations for Kalman Filtering
☑	rotations, qu...	article	Phillips:2001fk	Review of Attitude Representations Used for Aircraft Kinematics
☑	rotations, qu...	article	Pittelkau:2003fk	Rotation Vector in Attitude Estimation
☑	quaternions, ...	article	Salamin:1979fk	Application of quaternions to computation with rotations
☑	rotations, rot...	article	Schaub:1995lr	Principal Rotation Representations of Proper NxN Orthogonal Matrices
☑	quaternions, ...	inproceedings	Shoemake:1985bh	Animating rotation with quaternion curves
☑	quaternions	techreport	Shoemake:1994uq	Quaternions
□	attitude, quat...	book	Wertz:1980fk	Spacecraft attitude determination and control

[Phillips and Hailey, 2001] Phillips, W. F. and Hailey, C. E. (2001). Review of attitude representations used for aircraft kinematics. *Journal of Aircraft*, 38(4):718–737.

[Schaub et al., 1995] Schaub, H., Tsiotras, P., and Junkins, J. L. (1995). Principal rotation representations of proper NxN orthogonal matrices. *International Journal of Engineering Science*, 33(15):2277–2295.

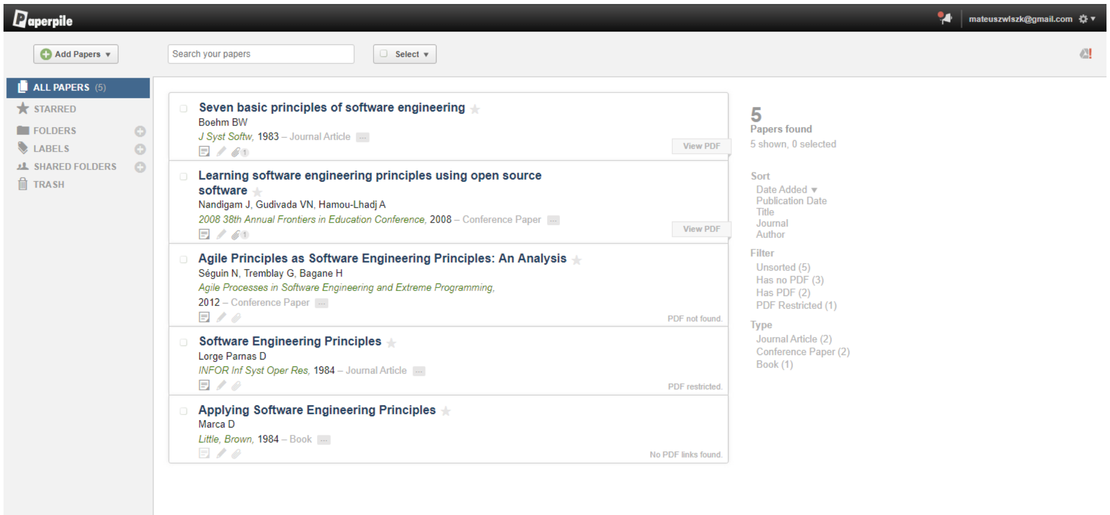
[Shoemake, 1985] Shoemake, K. (1985). Animating rotation with quaternion curves. In *SIGGRAPH '85: Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, New York, NY, USA. ACM Press.

[Shoemake, 1994] Shoemake, K. (1994). Quaternions. Technical report, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104.

Figure 2.4: Example of BibDesk interface

Paperpile

Paperpile excels in integrating reference management with Google Docs but also supports standard citation formats like BiBTeX, crucial for LaTeX users. It streamlines the addition of citations into documents and automatically formats bibliographies in BiBTeX, facilitating seamless academic writing workflows. Paperpile's strength is its integration with cloud-based platforms, which enhances collaboration among researchers, although its direct manipulation of BiBTeX is not as detailed as in some dedicated desktop applications [16]. Paperpile represents a modern approach to reference management. It scores highly across all criteria, especially in collaboration, where its cloud-based nature allows for sharing and annotation among researchers.



□ **Seven basic principles of software engineering** ☆
Boehm BW
J Syst Softw, 1983 – Journal Article

□ **Learning software engineering principles using open source software** ☆
Nandigam J, Gudivada VN, Hamou-Lhadj A
2008 38th Annual Frontiers in Education Conference, 2008 – Conference Paper

□ **Agile Principles as Software Engineering Principles: An Analysis** ☆
Séguin N, Tremblay G, Bagane H
Agile Processes in Software Engineering and Extreme Programming,
2012 – Conference Paper
PDF not found.

□ **Software Engineering Principles** ☆
Lorge Parnas D
INFOR Inf Syst Oper Res, 1984 – Journal Article
PDF restricted.

□ **Applying Software Engineering Principles** ☆
Marca D
Little, Brown, 1984 – Book
No PDF links found.

5
Papers found
5 shown, 0 selected

Sort
Date Added ▾
Publication Date
Title
Journal
Author

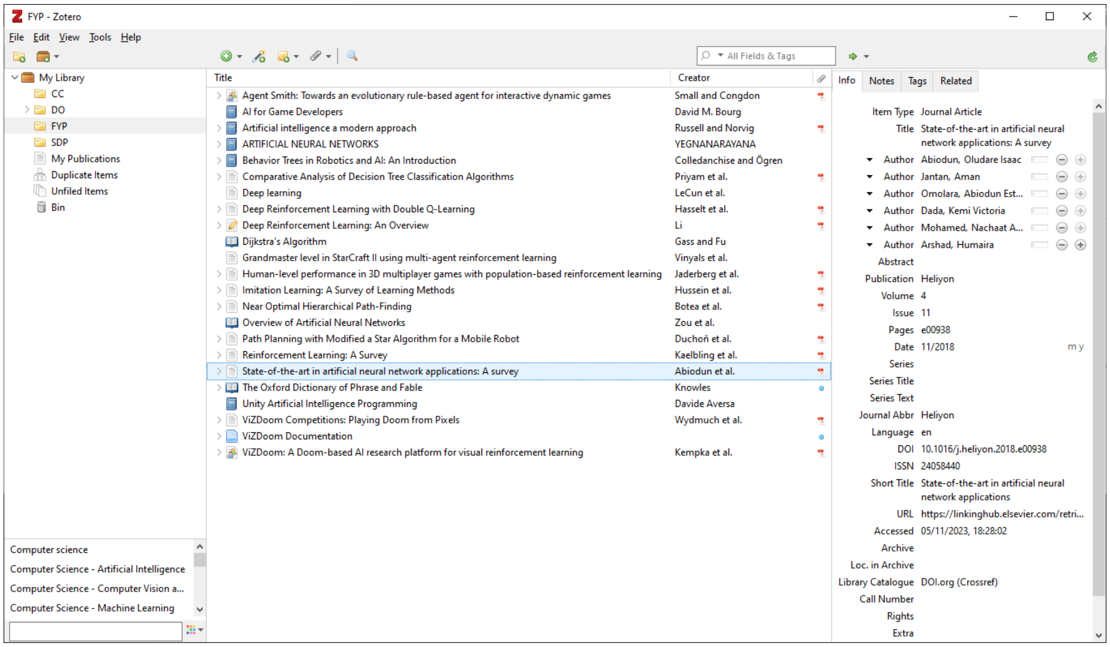
Filter
Unsorted (5)
Has no PDF (3)
Has PDF (2)
PDF Restricted (1)

Type
Journal Article (2)
Conference Paper (2)
Book (1)

Figure 2.5: Example of Paperpile web interface

Zotero

Zotero manages references directly from web browsers, making it especially suitable for researchers who gather sources online. It offers support for BiBTeX, allowing for direct export of references in BiBTeX format. Zotero's integration with a wide array of web browsers and word processors makes it highly adaptable to various workflows, particularly those involving LaTeX documents. Its user interface is intuitive, facilitating easy management of references and their incorporation into academic writing. Zotero also offers collaboration features, including shared libraries and group management. Zotero's ability to integrate with word processors and web browsers makes it highly versatile for academic writing, especially for those working in LaTeX environments [17].



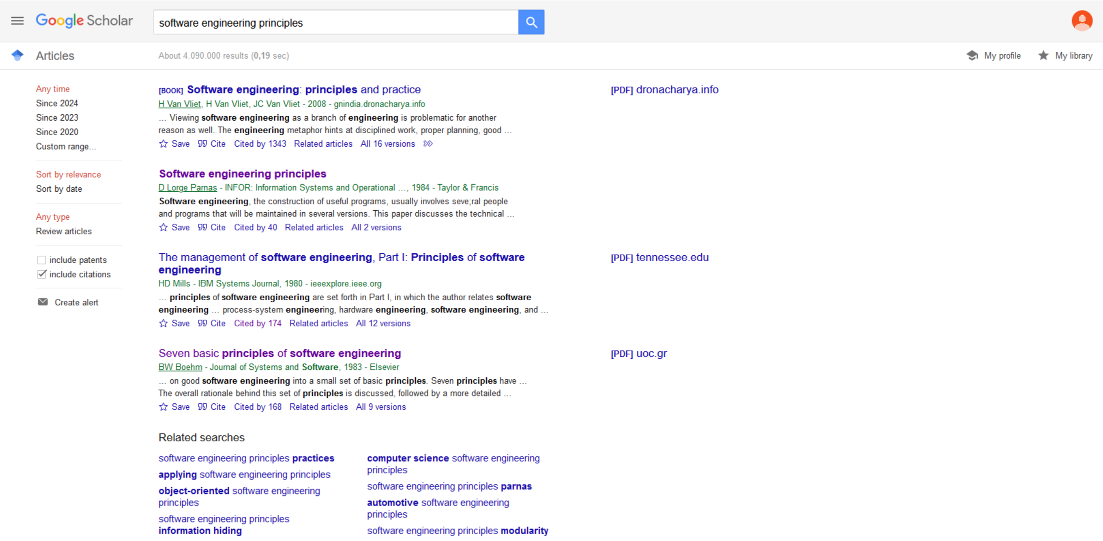
Title	Creator
Agent Smith: Towards an evolutionary rule-based agent for interactive dynamic games	Small and Congdon
AI for Game Developers	David M. Bourg
Artificial intelligence a modern approach	Russell and Norvig
ARTIFICIAL NEURAL NETWORKS	YEGNANARAYANA
Behavior Trees in Robotics and AI: An Introduction	Colledanchise and Ögren
Comparative Analysis of Decision Tree Classification Algorithms	Priyam et al.
Deep learning	LeCun et al.
Deep Reinforcement Learning with Double Q-Learning	Hasselt et al.
Deep Reinforcement Learning: An Overview	Li
Dijkstra's Algorithm	Gass and Fu
Grandmaster level in StarCraft II using multi-agent reinforcement learning	Vinyals et al.
Human-level performance in 3D multiplayer games with population-based reinforcement learning	Jaderberg et al.
Imitation Learning: A Survey of Learning Methods	Hussein et al.
Near Optimal Hierarchical Path-Finding	Botea et al.
Overview of Artificial Neural Networks	Zou et al.
Path Planning with Modified a Star Algorithm for a Mobile Robot	Duchoň et al.
Reinforcement Learning: A Survey	Kaelbling et al.
State-of-the-art in artificial neural network applications: A survey	Abiodun et al.
The Oxford Dictionary of Phrase and Fable	Knowles
Unity Artificial Intelligence Programming	Davide Aversa
ViZDoom Competitions: Playing Doom from Pixels	Wydmuch et al.
ViZDoom Documentation	
ViZDoom: A Doom-based AI research platform for visual reinforcement learning	Kempka et al.

Item Type	Journal Article
Title	State-of-the-art in artificial neural network applications: A survey
Author	Abiodun, Oludare Isaac
Author	Jantan, Aman
Author	Omolara, Abiodun Est...
Author	Dada, Kemi Victoria
Author	Mohamed, Nachaat A...
Author	Arshad, Humaira
Abstract	
Publication	Heliyon
Volume	4
Issue	11
Pages	e00938
Date	11/2018 m y
Series	
Series Title	
Series Text	
Journal Abbr	Heliyon
Language	en
DOI	10.1016/j.heliyon.2018.e00938
ISSN	24058440
Short Title	State-of-the-art in artificial neural network applications
URL	https://linkinghub.elsevier.com/retri...
Accessed	05/11/2023, 18:28:02
Archive	
Loc. in Archive	
Library Catalogue	DOI.org (Crossref)
Call Number	
Rights	
Extra	

Figure 2.6: Example of Zotero interface

Google Scholar

Google Scholar, is a search engine used for finding academic papers, it offers features that assist in reference management. The relevant feature is the ”Cite” function, which allows users to obtain citations in several formats, including BibTeX. By clicking the quotation mark icon below a search result, users can access and copy the citation in BibTeX format, simplifying the task of citing sources in LaTeX documents. While Google Scholar has an extensive database and easy access to scholarly works, it lacks direct integration with reference management tools and does not offer features for collaboration or advanced citation management. However, its ability to quickly search for and provide BibTeX entries makes it a valuable resource for researchers needing to incorporate references into LaTeX projects efficiently [18].



software engineering principles

About 4.090.000 results (0,19 sec)

[BOOK] **Software engineering**: **principles** and practice
H Van Vliet, H Van Vliet, JC Van Vliet - 2008 - gnindia.dronacharya.info
... Viewing **software engineering** as a branch of **engineering** is problematic for another reason as well. The **engineering** metaphor hints at disciplined work, proper planning, good ...
☆ Save 99 Cite Cited by 1343 Related articles All 16 versions
[PDF] dronacharya.info

Software engineering principles
D Lorge Parnas - INFOR: Information Systems and Operational ..., 1984 - Taylor & Francis
Software engineering, the construction of useful programs, usually involves seve;ral people and programs that will be maintained in several versions. This paper discusses the technical ...
☆ Save 99 Cite Cited by 40 Related articles All 2 versions

The management of **software engineering**, Part I: **Principles** of **software engineering**
HD Mills - IBM Systems Journal, 1980 - ieeexplore.ieee.org
... **principles** of **software engineering** are set forth in Part I, in which the author relates **software engineering** ... process-system **engineer**ing, hardware **engineering**, **software engineering**, and ...
☆ Save 99 Cite Cited by 174 Related articles All 12 versions
[PDF] tennessee.edu

Seven basic **principles** of **software engineering**
BW Boehm - Journal of Systems and **Software**, 1983 - Elsevier
... on good **software engineering** into a small set of basic **principles**. Seven **principles** have ... The overall rationale behind this set of **principles** is discussed, followed by a more detailed ...
☆ Save 99 Cite Cited by 168 Related articles All 9 versions
[PDF] uoc.gr

Related searches

software engineering principles **practices**
applying software engineering principles
object-oriented software engineering principles
software engineering principles **information hiding**
computer science software engineering principles
software engineering principles **parnas**
automotive software engineering principles
software engineering principles **modularity**

Figure 2.7: Example Google Scholar web interface

In the comparison table below 2.1, is a summary of how each tool performed based on the evaluated criteria.

Table 2.1: Comparison of Existing Reference Management Tools

Tool	User-Friendly	Citation Styles	Integration	Database	Collaboration
EasyBib	Moderate	High	Moderate	Low	Low
DOI2Bib	High	Low	Moderate	N/A	N/A
JabRef	High	High	High	High	Moderate
BibDesk	Moderate	High	High	Moderate	Low
Paperpile	High	High	High	High	High
Zotero	High	High	High	High	High
Google Scholar	High	Low	Low	High	N/A

2.4 Limitations of Current Systems

While the existing tools and services offer robust solutions for managing bibliographic references, they are unable to process and convert free-form references into BibTeX format. The usability of existing systems varies, with some offering more user-friendly interfaces than others. However, a common challenge across all tools is the process of converting large volumes of free-form references to BiBTeX. The task often involves multiple steps, from importing or inputting the references to correcting format discrepancies manually, this process is time consuming and prone to errors, which can significantly reduce the overall quality and reliability of reference management.

The accuracy of converting free-form references to BibTeX entries is crucial for maintaining the integrity of academic work. However, current systems may generate errors in metadata extraction, such as incorrect author names, publication years, or journal titles. These inaccuracies necessitate manual review and correction, undermining the efficiency of automated conversion processes.

Through the analysis of current solutions, these limitations highlighted the need for a more nuanced solution capable of handling the complexities of free-form reference management on a large scale. The Reference Wizard application will attempt at offering a more intuitive approach to overcoming the current challenges in academic reference management.

Chapter 3

Solution Design

3.1 Introduction

The ”Reference Wizard” project, in theory, should be able to convert free-form academic references to the structured LaTeX formatted bibliographies. This chapter describes the design and the planned implementation of the Reference Wizard, focusing on the use of LLMs to parse, interpret, and convert references with accuracy. By mapping out requirements, system setup, and LLMs unique functionalities, Reference Wizard aims to manage and convert bibliographies per academic authors' preferences swiftly and efficiently.

LLMs such as GPT and BERT (Bidirectional Encoder Representations from Transformers) have changed the field of NLP with their deep understanding of language semantics and syntax [19]. These models were trained on extensive datasets with diverse textual content, they can also comprehend and generate text with coherence and relevance to input.

3.1.1 Training and Capabilities

LLMs go through training processes on large amounts of data, which enables the models to learn the intricacies or nuances of a given language, including syntax, grammar, and context. Utilization of GPT models allows for the production of text sequences based on input prompts. While BERT's bidirectional training architecture excels in understanding the context of words in sentences. Making

LLMs suitable at tasks like text summarization, translation, question-answering and text generation.

3.1.2 Application to Reference Management

Converting free-form references into BiBTeX entries requires the ability to discern and categorize textual components, this needs to be done accurately and consistently. In order to achieve this, there needs to be a certain understanding of the language, LLMs have shown that they are proficient for this task as they have the functionality of understanding context in which terms are used, distinguishing between author's names, publication titles, and other citation/reference elements. LLMs perform well in recognizing and adapting to different citation formats, this can be due to the fact that they are trained on large amounts of text, often varying in format and style.

3.1.3 LLMs in the Reference Wizard

In the Reference Wizard project, the LLM will serve as the core component for parsing and structuring references. The selected model will receive input in the form of free-form reference text. It needs to analyze and extract key components such as author(s), title, publication year, and more, using its deep language understanding. The model has to structure these components into a BibTeX entry, adhering to the specified citation style, and, generate a preview for the user to utilize.

3.2 Project Requirements

With the help of the analysis of current solutions and essential features identified in Chapter 2, the system must accurately parse and convert free-form references into the BibTeX format, minimizing errors in citation data. It must be capable of accommodating a range of citation styles and be adaptable to less common formats. The tool should handle references efficiently providing a suitable approach for users with a need for converting their bibliographies. User friendliness is key, with an interface that simplifies the process of converting and managing references.

3.3 Design Considerations

Several factors influenced the design of the Reference Wizard, including the selection of the appropriate LLM, the need for computational efficiency, and the user experience. Computational demands, access, and ease of use of the application was considered with the project's goals in mind. Additionally, the interface design focused on simplicity and intuitiveness, enabling users to easily input free-form references and receive BibTeX outputs with minimal interaction.

3.4 System Architecture

The architecture of the Reference Wizard is designed to leverage the strengths of LLMs [20] while providing an ideal user experience. It consists of the following components:

1. **Input Interface:** A simple GUI interface where users can input manually or upload their list of free-form references via file upload.

2. **Pre-processing:** This component will parse the input data, clean it thoroughly by removing any common textual discrepancies before sending it to the LLM for processing.

3. **LLM Processing:** The application will use a selected LLM to parse and interpret the free-form references, extracting relevant information such as authors, titles, publication years, etc.

4. **Output Interface:** Users can review, edit, and download their converted references in BibTeX format through the UI.

3.5 Selection of Large Language Model

Determining the LLM, due to the project's requirements, was based on it's accuracy, adaptability to various reference styles, and speed of conversion. BERT or newer iterations of GPT may be considered based on access, computational cost, and specific capabilities relevant to parsing academic references [21]. The LLM will be chosen for the end-product after comparative analysis of current models, focusing on their performance in NLP, conversion speed and their ability to strictly adhere to instructions.

3.6 Implementation Plan

The Reference Wizard development was rolled out in stages, beginning with creating a prototype to assess the functions of reference conversion. The first steps involved integrating the selected LLM with an user interface to confirm the idea and collect input. Later phases enhanced features and upgraded the user interface. This phased approach allowed for iterative improvements [22], ensuring that the final product effectively met the needs of its users while leveraging the latest advancements in LLM technology.

Chapter 4

Implementation

4.1 Introduction

This section explores how the Reference Wizard project was created starting from the idea, to an operational application. It discusses the selection of technologies the incorporation of Language Models (LLMs), the design of a user interface for producing BibTeX entries from free-form references and on the obstacles faced during the development process and the strategies employed to address them.

4.2 Technology Selection

4.2.1 Programming Language

Python's association with AI models, particularly in natural language processing and machine learning [23] is one of the features that make it a suitable candidate for the Reference Wizard application. Python has a broad variety of libraries and frameworks that are well-suited for NLP, and GUI development, which greatly reduce the time required to develop an application like the ”Reference Wizard”. Libraries such as TensorFlow, PyTorch, and Hugging Face's Transformers simplify the process of fine-tuning and deploying machine learning models, including LLMs like GPT. Libraries such as GPT4All [24] provide high-level APIs for interacting with pre-trained models, this abstraction layer can enable developers to utilize LLMs with minimal coding, making it more accessible to integrate advanced natural language capabilities into applications. Python has compatibility with cloud-based services which could potentially enable the scaling of AI-powered applications, allowing for projects to handle increased workloads or complex computations on large datasets without compromising performance purely based on automatic scaling based on the size of the input [25]. Python has a large community which actively helps with contributing to the development of these new tools, libraries and new AI technologies, and since python has a clear and straight-forward syntax, it's a suitable choice for developing a prototype quite quickly [26].

4.2.2 GPT4All

GPT4All [24] is a platform in the domain of NLP and artificial intelligence (AI), designed to utilize the capabilities of Generative Pre-trained Transformer models. It is on the forefront of machine learning technology by offering LLMs proficient in text generation and comprehension. GPT4All enables developers to utilize pre-trained models for various applications, from text completion and summarization to semantic parsing and language translation.

Features and Capabilities

GPT4All models are best used for understanding the nuances of context and adjusting their responses accordingly, making them highly suitable for projects that require sophisticated language models. These models are trained on a broad and varied range of data, which gives them the ability to understand different contexts, infer meanings, and generate responses that are both coherent and relevant to the given situation. Moreover, GPT4All models are designed to be adaptable, allowing for the fine-tuning of models on specific datasets to optimize performance for particular tasks. It also has scalability, supporting various model sizes to balance between computational efficiency and language comprehension. Additionally, GPT4All is built on Python, simplifying the integration of advanced NLP features into applications by offering straightforward APIs for interacting with the models. This combination of features makes GPT4All a versatile and powerful tool for incorporating advanced linguistic capabilities into a wide array of projects.

Usage in the Reference Wizard Project

As "Reference Wizard" uses a GPT4All LLM, it plays a significant role by enabling the conversion of of free-form references into structured BibTeX entries. It allows for analysing references to identify components such as authors, titles, and publication details, among varied formats. From its analysis, the model generates BibTeX entries that adhere to academic citation standards. The project can fine-tune the model by modifying its paramaters or utilize its adaptability to ensure compliance with various citation styles, this can help automate bibliography management, which is especially beneficial for large datasets. This minimizes parsing and formatting errors through training on extensive data. GPT4All's role in the Reference Wizard is an example of the application of LLM technology to streamline and improve traditionally manual tasks.

4.2.3 Model Selection for GPT4All Integration

The selection of the optimal GPT4All model for the Reference Wizard project involved evaluating various models based on their capabilities, performance, and suitability for the task of converting academic references into structured BibTeX entries. The following models were considered:

1. **mistral-7b-openorca**

   Pros: This model was trained on the OpenOrca dataset and is known for its quick response times and its ability to handle high-quality conversational exchanges effectively. It's really good at adapting to different kinds of input, which makes it versatile.

   Cons: When it comes to tasks that need very specific instructions to be followed, like creating perfectly formatted BibTeX entries from different academic references, it might not always perform as precisely as needed.

2. **mistral-7b-instruct**

   Pros: Specifically designed for following instructions, this model promises fast responses and is expected to excel at tasks requiring adherence to explicit guidelines.

Cons: The model's focused training on instruction following might limit its conversational fluency, potentially affecting its ability to deal with ambiguously formatted references.

3. **falcon-newbpe-q4**

Pros: This model is decent at both conversational tasks and following specific instructions. It responds incredibly quickly, which can be a huge plus. Its flexibility seems like it would be really useful for dealing with a variety of reference formats.

Cons: Since it's trained on a very wide range of data, there's a bit of unpredictability in the quality of its responses. This is especially true when it comes to generating structured outputs, like BibTeX entries, where precision is everything.

4. **13b-snoozy**

Pros: Known for high-quality responses based on comprehensive training data, this model could potentially offer depth in understanding and processing academic references.

Cons: The slower response times could impact the user experience, particularly when processing large batches of references.

5. **mpt-7b-chat-newbpe**

Pros: Its novel architecture and fast, chat-oriented interactions make this model appealing for applications requiring quick, conversational responses.

Cons: The lack of specific optimization for instruction following may hinder its performance in accurately generating structured BiBTeX entries from unstructured references.

After careful consideration of the above models, "mistral-7b-openorca" was selected for the Reference Wizard project. This model offers a balanced mix of fast response times, conversational and contextual understanding capabilities, and suitability for commercial use. Its training on the OpenOrca dataset provides a solid foundation for generating accurate and contextually relevant BibTeX entries from academic references. While it may not be as specialized in instruction following as some other models, its overall performance and flexibility make it the most suitable option.

4.2.4 Tkinter

Tkinter was selected as the GUI framework for its straightforwardness and because it comes bundled with Python [27], which removes the need for extra dependencies. This choice was motivated by several factors that made Tkinter particularly appealing for the project. Firstly, its simplicity and ease of use make it possible to quickly develop interfaces that are simple and functional, fitting the project's goal of creating a user-friendly platform for rapid prototyping. Additionally, Tkinter's compatibility across multiple platforms, including Windows, macOS, and Linux, means that the Reference Wizard can be used by a wide range of users without needing significant adjustments for different operating systems. Although Tkinter may not have as many advanced features as some other GUI frameworks, its available functionalities perfectly match the project's requirements. The primary aim is to provide a straightforward interface for entering references and displaying the resulting BiBTeX entries, which Tkinter can provide.

4.2.5 Regular Expressions

The Python 're' library, which allows for the use of regular expressions in python, plays a critical role in preprocessing reference inputs before they are sent to the GPT-4All model. It is used to clean and format the references, such as removing unnecessary whitespace, hyphens, and other formatting issues that could affect the model's ability to generate accurate BibTeX entries [28].

4.2.6 Version Control with Git and GitHub

Git [29], along with GitHub [30], is used for version control of the Reference Wizard. This toolset makes sure that code changes are tracked over time, allowing for the safe storage of the project's codebase in a remote repository on GitHub. Github [30], and git [29] in general enables continuous integration and deployment workflows, issue tracking, and feature branching, which are integral to the agile development process.

4.3 System Architecture

The architecture of the Reference Wizard is designed for efficiency and simplicity, facilitating the conversion of free-form academic references into BiBTeX format. This section outlines the system's modular design, highlighting the interaction between its primary components.

4.3.1 Component Descriptions

As mentioned in Chapter 3 - the system comprises several key components, each responsible for a distinct aspect of the reference conversion process as seen in figure 4.1:

- **User Interface:** Developed with Tkinter, it provides an intuitive platform for users to input and manage references.

- **Preprocessing Module:** Utilizes Python's 're' library to clean and format input references, preparing them for processing.

- **GPT4All Integration:** Core to the application, this module interacts with the GPT-4All model to generate BiBTeX entries from the processed references.

- **Output Formatting Module:** Post-processes the GPT-4All output to ensure the generated BiBTeX entries adhere to citation standards.

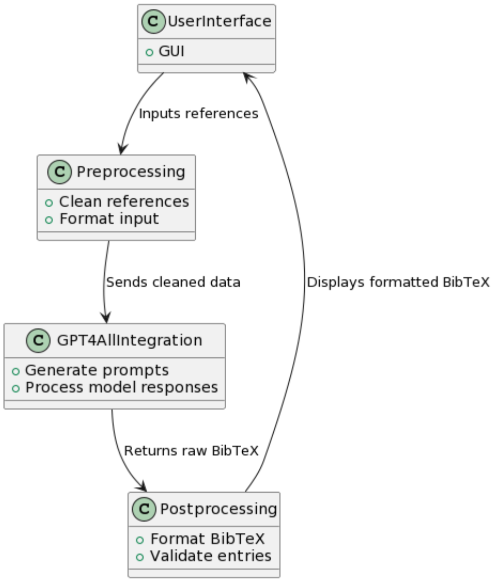


Figure 4.1: System Architecture

4.4 Detailed Implementation

4.4.1 User Interface Design and Implementation

By the ”Reference Wizard” utilizing Tkinter, a widely used GUI library in Python, the design ensures that users can navigate the application, from entering references to receiving formatted BibTeX entries. Tkinter, while not a very sophisticated library, allows for very a simplistic UI, which, is an integral part of the application.

Design Principles

- The UI is structured to reduce cognitive load, presenting only essential elements for task completion without overwhelming the user.

- Controls are logically placed and labeled clearly, guiding users naturally through the process of inputting, converting, and exporting references, ”Reference Wizard” follows a very simple UI template, where buttons like ”Settings” are placed on the top-left of the window.

- The application provides immediate visual feedback for user actions, such as successful import/export operations and the conversion process.

Tkinter Implementation

Tkinter's widget-based architecture facilitates the modular design of the Reference Wizard's UI. For the input and output areas, `ScrolledText` widgets create a straightforward interface for users to paste references and view the generated BibTeX entries. Buttons like ”Import”, ”Export”, ”Generate”, and ”Copy to Clipboard” are implemented using Tkinter's `Button` widget. To maintain UI responsiveness, particularly during the conversion process, Python's `threading` module is used which allows the application for data processing and converting references in the background, preventing UI freezes. Threading is also used to disable the user's ability to invoke the model again, while it's in the middle of generating, preventing potential crashes.

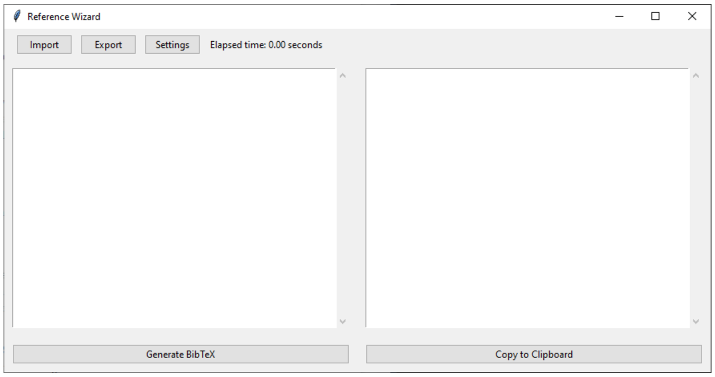


Figure 4.2: Reference Wizard User Interface

As seen in figure 4.2 The design of the interface is arranged to match the step-by-step flow of converting references. The ”Import” and ”Export” buttons are located at the top for easy access since they are used at the beginning and end of the process, respectively. This placement underlines their importance in the workflow. The middle section is reserved for the areas where users input and view the converted references, highlighting the main purpose of reference conversion. Additionally, the ”Generate” button, which starts the conversion, is given a stand-out position to motivate users to initiate the process after entering their references.

4.4.2 Processing Pipeline

The Reference Wizard transforms free-form references into structured BiBTeX entries through a carefully designed processing pipeline. This pipeline consists of preprocessing of input, interaction with the GPT4All model, and postprocessing of the output to ensure high-quality BiBTeX entries.

Preprocessing

In an effort to clean the input before directing it to the LLM, the "Reference Wizard" utilizes Regular Expressions (regex) which removes any excess or inconsistent whitespace within references. Regex also helped with joining words that got split up at line breaks, which often happens because of hyphenation. It's essential that this input is preprocessed as to keep titles and names intact. For example, consider a reference from a paper that, when copied, might seem broken up due to line break or hyphenation, such as:

```
Smith, J. (2020). Understanding the complexities of
   ↪ climate change. Journal of Environ-
mental Science, 15(4), 123-134.
```

This example illustrates how hyphenation can break words at the end of lines, which regular expressions can join to maintain the flow of the text. By executing regex-based operations, the preprocessing ensures that references are in a clean, consistent format, used for subsequent processing by the LLM.

Reference Segmentation and Queue Processing

Before the segmentation and queue processing strategy was introduced, the "Reference Wizard" struggled handling large volumes of references within a single prompt. One issue was that, the model's maximum number of tokens it could generate in one response to a single prompt was too low. When a user tried to input a large amounts of references, the model often reached it's max token limit before completing the conversion of all references, effectively, stopping halfway through it's generation which would lead to incomplete outputs. For users with large bibliographies that wanted a quick and easy conversion, this was a problem, as the system could not reliably process all entries within a single operation. In addressing this challenge, the Reference Wizard incorporates a segmentation and queue processing strategy. This approach is needed for handling batches of references efficiently, ensuring each is accurately converted. The application starts by breaking down the collection of inputs (references) into segmented parts. This division is determined by recognizing patterns that distinguish one citation, from another. The idea is that each citation is distinguished by a sequence like a line break, which helps in organizing the text into distinct sections. This method creates a list or queue of references, each used for individual processing. Following segmentation, the application processes each reference sequentially. This queued approach is important for maintaining the integrity of each conversion, as it allows for individual attention to the nuances of each reference by the LLM. By implementing a segmentation and queue processing system, the Reference addressed the limitations of not being able to convert in bulk by breaking down the input into manageable units, each processed independently. This approach bypassed the max token limitation of the model by distributing the workload across multiple prompts but also enhanced the system's overall reliability as it ensures that each reference is processed within the model's operational constraints.

GPT4All Integration

GPT4All's advanced NLP capabilities are key to converting cleaned references into structured BiBTeX entries. This stage involves careful and dynamic prompt crafting in order to utilize the model effectively.

Model Invocation and Prompt Engineering Invoking the GPT4All model *mistral-7b-openorca* is what converts the raw user input of free-form reference into the structured BiBTeX output. This process is highly dependent on the prompt engineering, which involves the careful design of input prompts to guide the model's generation capabilities. The system template and prompt template play extremely important roles in this context.

- **System Template:** The system template is a parameter that creates context for the model to utilize when responding to user queries. In this specific scenario, it instructs the model to act as a ”reference wizard” tasked with transforming free-form references into formatted BiBTeX entries. To reduce computation overhead and reduce the chance of ”hallucinations” occurring, or irrelevant outputs from the model, the system template should be clear and concise. Through trial and error, it was discovered that shorter, rule-focused templates lead to more efficient and accurate model performance. ”Reference Wizard” uses a system template that emphasizes the generation of BiBTeX entries with specific instructions on cite keys and the exclusion of fields not explicitly mentioned in the input.

- **Prompt Template:** The prompt template dynamically incorporates the user’s input reference into the system’s context, creating a tailored prompt for each conversion task. This customization ensures that the model’s response is directly relevant to the provided reference, adhering to the instructions laid out in the system template. By stating ”USER:” followed by the input reference, and expecting the model’s output to follow ”ASSISTANT:”, a structured dialogue is simulated, focusing the model’s attention on the conversion task at hand.

Concise and clear templates ensure that the model’s computational resources are directed towards understanding and executing the task, rather than parsing unnecessary context. By controlling the prompt’s structure and content strictly, the likelihood of the model generating irrelevant or incorrect information (hallucinations) is significantly reduced.

An example of a system prompt used for the model is shown below:

```
Imagine you are a reference wizard,
when a user presents you with a reference,
your task is to reformat any given reference
into its corresponding BibTeX.
Use the author’s name and year to form cite keys.
Your response should ONLY include the BiBTeX.
```

```
If the ISSN, DOI,
or any other fields are not explicitly
provided in the reference,
do not add them to the BiBTeX.
```

This prompt specifies how the model should handle input references, focusing on creating to the point, rule-based BiBTeX entries without adding unnecessary information.

GPT4All Parameters

The performance and output quality of the GPT4All model can be fine-tuned through various parameters. Here is a detailed exploration of key parameters and their effects:

- **max_tokens:** Specifies the maximum number of tokens (words or pieces of words) the model generates in response to the prompt. A higher value allows for longer outputs, while a lower value limits the response length. In the context of BiBTeX generation, adjusting this parameter helps control the detail level of each entry. Higher values for low-character inputs can induce hallucinations within the model's responses.

- **temp (Temperature):** Influences the randomness in the model's responses. A higher temperature encourages more varied and creative outputs, potentially increasing diversity but risking factual accuracy. Conversely, a lower temperature produces more conservative and predictable outputs, which is preferred for generating accurate BiBTeX entries.

- **top_k:** Limits the model's choices to the top-k most likely next tokens at each step. A lower top-k value increases the likelihood of the model selecting high-probability tokens, reducing randomness and improving the predictability of the output.

- **top_p (Nucleus Sampling):** Allows the model to consider a dynamic number of tokens at each step, choosing from the smallest set whose cumulative probability exceeds the top-p threshold. This parameter provides a balance between diversity and accuracy in the generated text.

- **min_p:** Ensures that the model only considers tokens with a probability above this threshold, effectively filtering out highly improbable tokens from the generation process. This can help increase the overall quality of the output by eliminating less relevant suggestions.

- **n_batch:** Determines the number of prompt tokens processed in parallel. Increasing this value can reduce response time at the cost of higher computational demand.

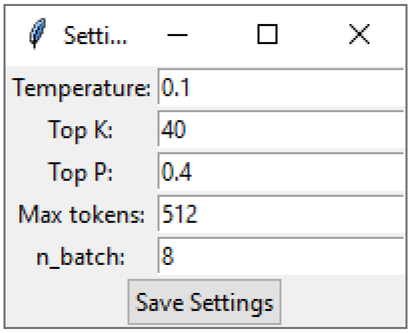
- **streaming:** When set to True, the model generates tokens incrementally, allowing for real-time interaction with the generation process. This can be particularly useful for applications requiring immediate feedback or iterative development of outputs. The default value is set to False, as Reference Wizard has no need to stream the model's response.

Understanding and carefully adjusting these parameters is essential for tailoring the GPT4All model's behavior to the specific needs of the Reference Wizard project, ensuring that the generated BiBTeX entries meet the desired standards of accuracy and format. This careful prompt engineering and parameter tuning optimize the model's output for the specific task of generating BiBTeX entries from academic references.

4.4.3 Settings

A crucial aspect of the Reference Wizard application is its flexibility, allowing users to fine-tune the behavior of the GPT4All model via a settings page. This feature allows users to adapt the model's performance to their specific requirements.

As seen from figure 4.3, the settings page is implemented using Tkinter, providing a user-friendly interface where users can adjust key parameters of the GPT4All model. Each parameter is accompanied by an input field, the input field is already filled out with the default values, this is because it allows users unfamiliar with the underlying technology to make informed adjustments. Users can adjust the maximum number of tokens generated, balancing between detail and conciseness in the output. The default value is set to 512 tokens per reference, but users can increase or decrease this based on their needs. By adjusting the temperature, users



Temperature: 0.1
Top K: 40
Top P: 0.4
Max tokens: 512
n_batch: 8

Figure 4.3: Reference Wizard's Settings Page

can make the model's responses more deterministic (lower temperature) or more creative (higher temperature), with a default setting of 0.1. Top-k and Top-p parameters control the selection process of the next tokens. Users can modify these values to influence the randomness and predictability of the text generation, with default values set to 40 (Top-k) and 0.4 (Top-p), respectively. Upon adjusting the parameters, users can save their settings, which are immediately applied to the model. This dynamic adjustment capability ensures that the Reference Wizard can be effectively used across a wide range of reference conversion tasks, from straightforward citation formatting to more complex entries requiring nuanced understanding.

By enabling parameter adjustments, the application becomes more versatile and also educates users on the complexities of AI-driven text generation. This offers a level of customization that allows both novice and experienced users to customize their output.

4.4.4 Help Page

In order to provide an overview of the role of different parameters found in the settings page, the "Reference Wizard" uses a help page.

The help page guides the user on whether increasing/decreasing the value of a parameter might help output the correct formatting or improve the efficiency of the conversion process.

4.4.5 Model Processing and Output Generation

After preprocessing the user's input, the Reference Wizard feeds the preprocessed data into the GPT4All model, along with the parameters such as *top_p*, *temp*, and *max_tokens*. This section describes how the model utilizes these inputs to generate BiBTeX entries, which are then presented to the user through the GUI.

The interaction between the preprocessed input and the GPT4All model is a phase where the conversion from free-form references to structured BiBTeX entries occurs.

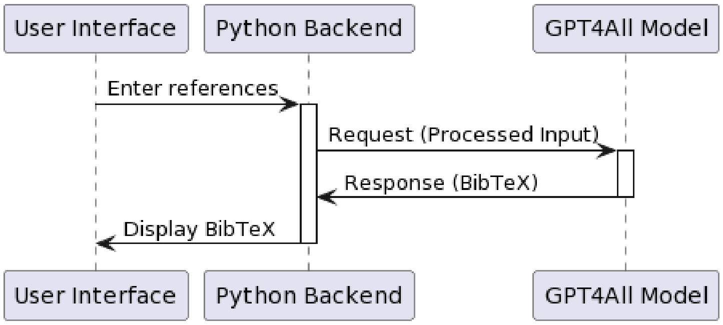


Figure 4.4: Sequence Diagram

The sequence diagram (Figure 4.4) describes the start of the interaction from the user input through the GUI, the invocation of the GPT4All model with the specified parameters, to the final presentation of generated BiBTeX entries in the GUI. The GUI plays a key role in presenting the model-generated BiBTeX entries to the user. Once the model completes its processing, the entries are:

1. Displayed in the output section of the GUI, allowing users to review and verify the correctness and completeness of the citations.
2. Made available for export, or 'copy-to-clipboard' enabling users integrate these entries into their LaTeX documents or reference management software.

4.5 Input and Output Examples

To illustrate the Reference Wizard's processing pipeline and its effectiveness, this section presents a specific example of an input free-form reference and the corresponding output BiBTeX entry generated by the system.

4.5.1 Input Reference

Consider the following free-form reference as input to the Reference Wizard:

```
Doe, John. "An Analysis of Machine Learning." Journal of AI Research,
vol. 15, no. 1, pp. 123-135, Jan. 2020.
```

This reference contains all necessary components such as author names, article title, journal name, volume, issue, page numbers, and publication date, in a free-form reference format.

4.5.2 Processed and Generated BiBTeX Entry

After undergoing the preprocessing, and processing with the LLM, the Reference Wizard produces the following structured BiBTeX entry:

```
@article{john_Doe2020,
    author = {John Doe},
    title = "An Analysis of Machine Learning",
    journal = "Journal of AI Research",
    volume = "15",
    number = "1",
    pages = "123-135",
    month = "Jan.",
    year = "2020"
}
```

The BiBTeX entry contains all the necessary fields that were provided and extracted from the free-form reference, including the author name, title, journal and its specific volume, number and pages, and the year alongside the month of release. This result shows that the tool is effective at converting free-form references to a BiBTeX format.

Chapter 5

Evaluation

This chapter will be the evaluation of the Reference Wizard, focusing on its performance, accuracy, and user experience. The evaluation process was designed to assess the tool's capability to accurately convert a diverse set of academic references into BiBTeX format and to measure the usability and user satisfaction. Two primary methods were employed: an analysis of the tool's accuracy across a dataset of 1000 references and a user study involving a questionnaire to collect qualitative feedback on the user experience.

5.1 Approach

5.1.1 Dataset Preparation and Extraction Process

To evaluate the Reference Wizard's ability to accurately convert academic references into BiBTeX format, a comprehensive dataset of 1000 references was compiled. The dataset aimed to represent a wide variety of disciplines within the scientific community, therefore ensuring a thorough assessment of the tool's versatility and accuracy across different citation formats and fields of study.

The dataset was gathered through a search and selection process on Google Scholar, utilizing keywords associated with various scientific disciplines, which allowed for identification of academic papers with references from different citation styles and formats.

Example of Field-Specific Searches:

1. **Medicine:** The search keyword ”medicine” led to the selection of the paper ”Occupational, Duke. ’Environmental Medicine.’ Environmental Health Perspectives 107.5 (1999): 417.”

2. **Engineering:** The keyword ”engineering” brought up ”Jensen, Klavs F. ’Microreaction engineering—is small better?.’ Chemical Engineering Science 56.2 (2001): 293-303.”

3. **Software Engineering:** The search for ”software engineering” brought up ”Ziv, Hadar, Debra Richardson, and René Klösch. ’The uncertainty principle in software engineering.’ submitted to Proceedings of the 19th International Conference on Software Engineering (ICSE’97). 1997.”

4. **Business:** The search for ”business” came up with ”Drucker, Peter F. ”The theory of the business.” Alfred P. Sloan: Critical evaluations in business and management 2.2 (1994): 258-282.”

Throughout the extraction process, it was made sure to preserve the integrity of the references despite facing some challenges with formatting when directly copying and pasting from the source papers. For instance, there were spacing errors that had to be corrected in names e.g., changing ”Mark Ste k” to ”Mark Stefik” to ensure accuracy. Additionally, there was a character inserted between each reference in the dataset to align with the Reference Wizards input format allowing for processing of each entry.

The references were sourced from the bibliographies of academic papers, chosen for their varied subject matter to ensure a comprehensive test of the Reference Wizard’s versatility. The chosen sources were kept in their original state as seen in the publications with only essential adjustments made for correct formatting. This method aimed to mirror real life situations where users might enter references containing mistakes or some discrepancies.

5.1.2 Accuracy Assessment

Out of 1000 references, 50 references were randomly chosen to be manually inspected, the accuracy was evaluated through a detailed comparison between the BiBTeX entries generated by the tool and a gold standard set of BiBTeX entries obtained from Google Scholar for the same references. This comparison focused on the correctness of key bibliographic details such as author names, titles, publication years, and publishers. Fields not present in the original free-form references, such as DOI, ISBN, and URLs, were excluded from the evaluation criteria, as these could not be generated by the Reference Wizard without the corresponding input data.

Comparison Process:

For each reference in the dataset, the output BiBTeX entry from the Reference Wizard was compared side-by-side with its counterpart from Google Scholar. This process involved a detailed examination of the entries to assess the accuracy of the conversion.

5.1.3 Example of a Successful Conversion:

Free-form reference from dataset:

```
"Feldman RG: Occupational and Environmental Neurotoxicology.
Hagerstown, MD:Lippincott-Raven, 1999"
```

Correct BiBTeX according to Google Scholar:

```
@book{feldman1999occupational,
title={Occupational and Environmental Neurotoxicology},
author={Feldman, R.G.},
isbn={9780781717397},
lccn={98013524},
url={https://books.google.dk/books?id=ZcNrAAAAMAAJ},
year={1999},
publisher={Lippincott-Raven}
}
```

Outputted BiBTeX made by Reference Wizard:

```
@Book{Feldman99,
author = "Feldman, RG",
title = "Occupational and Environmental Neurotoxicology",
publisher = "Lippincott-Raven",
address = "Hagerstown, MD",
year = "1999"
}
```

In this example, the Reference Wizard successfully captured the bibliographic details from the free-form reference. The absence of fields such as DOI, ISBN, or URL in the Reference Wizard's output is consistent with the evaluation criteria, as these details were not included in the original input and therefore not expected in the generated BiBTeX entry.

5.1.4 Example of Failed Conversion

When evaluating the effectiveness of the Reference Wizard, it was important to not only showcase its achievements but to also investigate times where it didn't meet expectations. An instance is discussed below demonstrating a scenario in which the Reference Wizard failed to produce a BiBTeX citation when compared to Google Scholars reference standard.

Original Free-Form Reference:

```
Ellis J, Mulligan I, Rowe J, Sackett DL.
Inpatient general medicine is evidence based. Lancet
1995;346:407-10.
```

Reference Wizard Output:

```
@article{Ellis_Mulligan_Rowe_Sackett_1995,
author = {Ellis, J and Mulligan, I and Rowe, J and Sackett, DL},
title = {Inpatient general medicine is evidence based.
Lancet 1995;346:407-10},
year = {1995}
}
```

Correct BiBTeX According to Google Scholar:

```
@article{sackett1995inpatient,
title={Inpatient general medicine is evidence based},
author={Sackett, DL and Ellis, J and Mulligan, I and Rowe, J},
journal={The Lancet},
volume={346},
number={8972},
pages={407--410},
year={1995},
publisher={Elsevier}
}
```

There are several differences between the output from the Reference Wizard and the accurate BiBTeX entry from Google Scholar in this example. The Reference Wizard missed out on details like the journal name, volume, number and page range opting to include some of this data in the title section. Also, the order of authors and the exclusion of the publisher point to areas where the Reference Wizard's processing could be improved. This specific failure showcases the challenges the Reference Wizard has with certain reference formats, particularly those that do not conform to the most common patterns.

5.2 Runtime Performance

In a test involving 1000 academic references, the total runtime was 32,065 seconds, averaging approximately 32 seconds per reference which is a metric that highlights the current processing speed and sets a benchmark for future performance optimizations.

5.2.1 User Study Design

To better understand the Reference Wizard's user experience, a user study was conducted using a questionnaire. The main goal of this study was to collect input on aspects of the tool, such as its ease of use, interface friendliness and the accuracy of its reference conversion. This input is crucial, for pinpointing strengths and areas where enhancements can be made to ensure that the tool meets user requirements effectively.

The questionnaire received input from a range of people including those who are well versed in reference organization tools - or computer science in general, and newcomers who may be using such a tool for the first time. This diverse group of participants was selected to gather a range of perspectives and experiences that mirror the user base of the Reference Wizard.

The questionnaire consisted of seven main questions, structured to capture both quantitative and qualitative data. Participants were asked to rate their experience on a scale (e.g., Very Easy to Very Difficult, Very Satisfied to Very Dissatisfied) in areas such as ease of use, intuitiveness of the user interface, accuracy of reference conversion, and speed of conversion. The questionnaire also sought to gauge the overall satisfaction with the Reference Wizard and the likelihood of participants recommending the tool to others. This part of the survey aimed to capture the tool's perceived value from the user's perspective. At the end of the questionnaire, each person was asked to provide detailed feedback and suggestions for improvement.

5.3 Results

5.3.1 Accuracy Assessment Results

Out of the 50 randomly chosen references, 46 (92%) were found to be completely correct, aligning well with the manually verified BiBTeX entries. 3 (6%) references had minor formatting issues, such as the publication year appearing in the title field or other ”odd” formatting errors that did not impact the usability of the BiBTeX entries. 1 (2%) reference was incorrectly converted, with a hallucinated publication year that did not match the original source. This degree of accuracy demonstrates the Reference Wizard's effectiveness in converting a wide variety of references into the BiBTeX format.

5.3.2 Runtime Evaluation

When compared to other reference management tools, the processing time of the Reference Wizard might is a massive drawback. Many contemporary tools aim for quicker reference handling, though this may sometimes come at the cost of accuracy. The Reference Wizard tries to balance its speed with its comprehensive functionality, like handling diverse formats and the computational power requirement that comes along with it.

5.3.3 User Study Results

The findings of the user study offer insights into how users view the Reference Wizard's usability, design, accuracy, conversion speed and satisfaction. Feedback collected from six participants helps identify both the tool's strengths and areas that could be improved. Here are the detailed results below:

Ease of Use: The majority of participants found the Reference Wizard very easy to use (66.7%), with the rest (33.3%) rating it as somewhat easy which suggests that the tool is accessible to users, offering a user-friendly experience that does not demand a lot of knowledge beforehand.

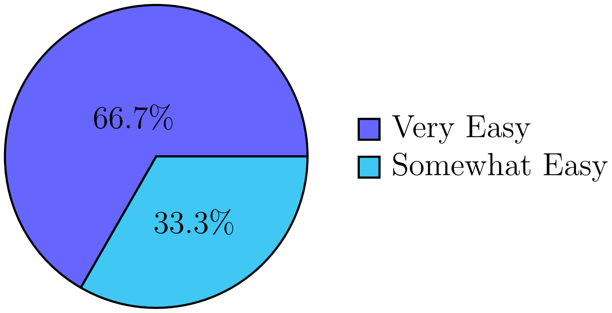


Figure 5.1: Pie chart showing user responses for Ease of Use

Intuitiveness of the User Interface: The user interface was well received, with most participants (83.3%) finding it very intuitive and one participant (16.7%) rating it as somewhat intuitive. This indicates that the design and layout of the interface allows for navigation and operation of the tool.

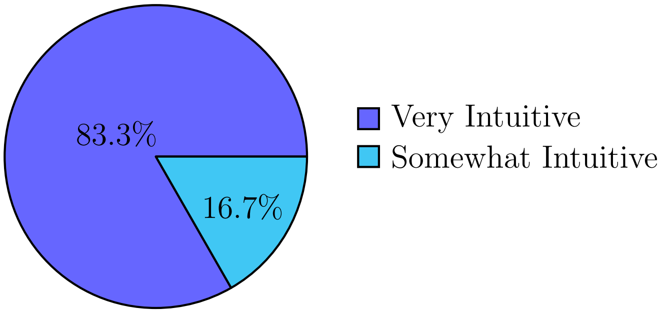


Figure 5.2: Pie chart showing user responses for Intuitiveness of the User Interface

Accuracy of Reference Conversion: All participants (100%) rated the accuracy of reference conversion as very accurate, underscoring the tool's capability to generate BiBTeX entries that closely match the expected output.

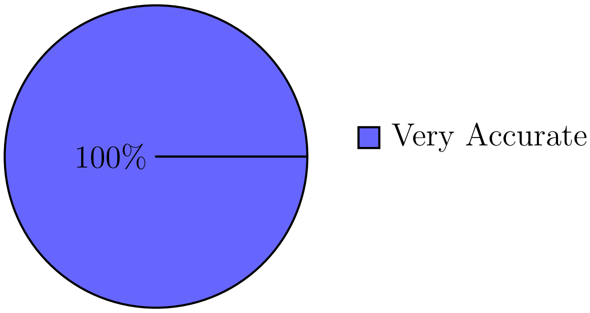


Figure 5.3: Pie chart showing user responses for Accuracy of Reference Conversion

Speed of Conversion: The speed of conversion was the primary area of concern, with the majority of participants expressing dissatisfaction (66.7%) somewhat dissatisfied, 16.7% very dissatisfied) and one participant (16.7%) remaining neutral. This feedback highlights the need for optimization to enhance the tool's efficiency.

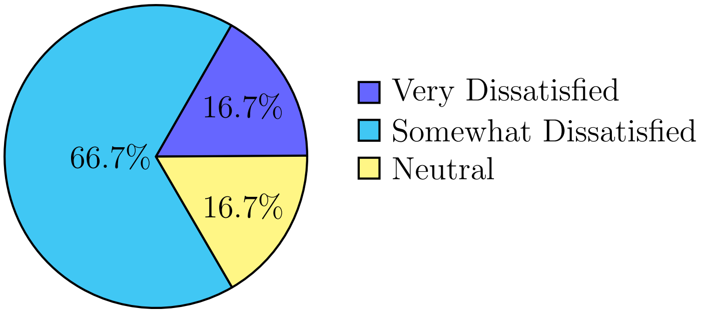


Figure 5.4: Pie chart showing user responses for Speed of Conversion

Overall Satisfaction: Overall satisfaction levels were generally positive, with most participants (83.3%) somewhat satisfied and one participant (16.7%) very satisfied. This suggests that despite concerns regarding conversion speed, the tool meets users' needs to an extent.

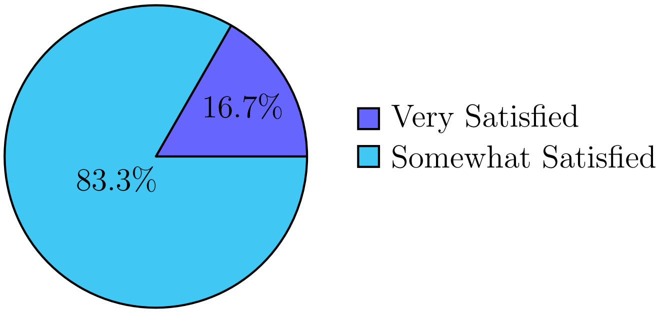


Figure 5.5: Pie chart showing user responses for Overall Satisfaction

Likelihood to Recommend: The likelihood of recommending the tool to others was mixed, with equal numbers of participants somewhat likely, neutral, and very likely to recommend. The range of reactions suggests that different users see the tools worth depending on their needs and experiences.

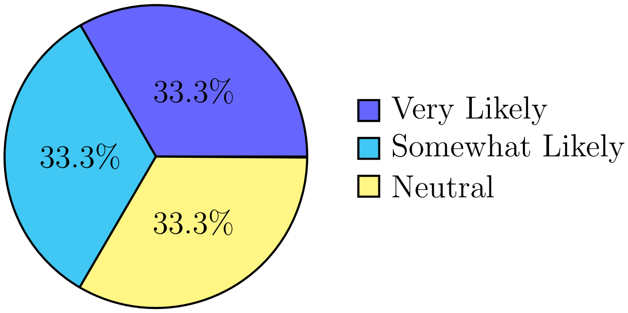


Figure 5.6: Pie chart showing user responses for Likelihood to Recommend

Additional Feedback: Participants also provided constructive suggestions for improvement, highlighting areas such as UI clarity, conversion speed, and the potential benefits of a web application format. Specific recommendations include adding tooltips for button functionalities and addressing the conversion time to enhance user experience.

5.4 Discussion

5.4.1 Interpretation of Results

The user study and accuracy assessment collectively highlight the Reference Wizard's proficiency in converting academic references into BiBTeX format, with high marks for accuracy and user interface intuitiveness. The overwhelmingly positive feedback on the accuracy of reference conversion validates the tool's core functionality, confirming its effectiveness in automating a crucial aspect of academic writing.

However, the primary concern among users relates to the speed of conversion, indicating a need for optimization to improve processing times. This aspect is critical for enhancing user satisfaction and the overall utility of the tool, especially for users handling large volumes of references.

The mixed responses regarding the likelihood to recommend the tool suggest that while the Reference Wizard is valued for its specific functionalities, its adoption may be hindered by the current desktop application format and the conversion speed issue. The suggestion to develop a web application version points to an opportunity to increase accessibility and user engagement.

5.4.2 Comparison with Existing Tools

5.5 Limitations

The evaluation faced limitations, including the scope of the dataset, which, while diverse, was derived from a limited number of references. Additionally, the accuracy assessment relied on manual comparison, which could introduce subjective judgment in identifying formatting issues. The user study received feedback from only 6 participants, which could mean that the results from the questionnaire might not accurately predict and reflect the preferences of other potential users.

Chapter 6

Conclusion

This project introduced the Reference Wizard, a specialized tool designed to automate the conversion of free-form academic references into their respective structured BiBTeX format. The primary motivation behind this development was to address a gap in existing solutions, which largely focused on manual entry or required references to be in a specific format for conversion e.g., DOIs, PMCIDs, or arXiv IDs . Unlike these solutions, Reference Wizard caters specifically to the needs of users who handle large volumes of academic references, providing an efficient and accurate mechanism for mass conversion.

The Reference Wizard stands out by offering a seamless conversion process that significantly reduces the manual labor involved in preparing bibliographies for academic writing, especially when utilizing LaTeX. Through an intuitive user interface and a robust backend powered by the GPT4All model *mistral-7b-openorca*, the tool demonstrates high accuracy in generating BiBTeX entries from a wide variety of reference formats. After testing the tool for accuracy and conducting a user study, it is evident that the tool performs across various reference formats. User feedback also confirms that the tool is user friendly and useful, although, there are areas for enhancement in terms of conversion speed and interface clarity.

6.1 Future Work

Based on findings and user feedback, four areas were found to be most helpful in future development of the Reference Wizard:

Web Application Development: Moving from a desktop program to a web-based platform shows potential for improving ease of access and user friendliness. Shifting to a web based application may make it easier for users to get started and encourage people to use it by integrating directly into the workflow of academic writing.

Optimization of Conversion Speed: In response to the participants' main concern, the focus should be on enhancing the speed of conversion. One approach could be to investigate more sophisticated queuing methods that enable batch processing of citations thus cutting down the conversion time while maintaining accuracy. Another potential enhancement might be to boost resources leading to a speed up, in the conversion process.

Expansion of Model Capabilities: To improve the Reference Wizard's performance, training an LLM that is specifically designed for the application's requirements could prove to be worthwhile. Adjusting a model to identify and interpret the patterns and structures to academic references could significantly boost precision and effectiveness. This personalized method could help address the instances of formatting issues as the model would become more proficient, at managing the various formats and citation styles prevalent in scholarly works. This LLM could also adjust flexibly to shifts in citation conventions ensuring that the Reference Wizard remains a dependable tool, for academic writing and research.

Enhanced User Interface: Incorporating user suggestions for interface improvements, such as adding tooltips and clarifying button functionalities, can significantly improve the user experience. Enhancements aimed at making the tool more intuitive and user-friendly will likely increase user satisfaction.

6.2 Closing Remarks

Reference Wizard shows great promise in terms of the automation of bibliography management for academic writing. It streamlines the process of converting free-form references to structured BiBTeX entries, therefore offering value to both researchers and students, even academic professionals. This tool, may be an invaluable asset in the field of academic writing and in preparing publications. This project described the development process of the Reference Wizard and provided some potential avenues of future work to improve its efficiency and user-satisfaction.

Bibliography

[1] Oren Patashnik. “BibTeXing”. In: *Documentation for general BibTeX users* 3.84 (1988), p. 1.

[2] Feng Jiang, Kuang Wang, and Haizhou Li. “Bridging Research and Readers: A Multi-Modal Automated Academic Papers Interpretation System”. In: *arXiv preprint arXiv:2401.09150* (2024).

[3] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[4] Alec Radford et al. “Improving Language Understanding by Generative Pre-training”. In: (2018).

[5] Andrew Drozdov et al. “Compositional semantic parsing with large language models”. In: *The Eleventh International Conference on Learning Representations*. 2022.

[6] Tom B Brown et al. “Language Models are Few-Shot Learners”. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[7] Min-Yuh Day et al. “A knowledge-based approach to citation extraction”. In: vol. 2005. Sept. 2005, pp. 50–55. ISBN: 0-7803-9093-8. DOI: `10.1109/IRI-05.2005.1506448`.

[8] Ron Gilmour and Laura Cobus-Kuo. “Reference Management Software: a Comparative Analysis of Four Products.” In: *Issues in Science and Technology Librarianship* 66 (Sept. 2011). ISSN: 1092-1206. DOI: `10.29173/istl1521`. URL: `http://dx.doi.org/10.29173/istl1521`.

[9] Arman Cohan et al. "SPECTER: Document-level Representation Learning using Citation-informed Transformers". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), pp. 2270–2282.

[10] M. Kathleen Kern and Merinda Kaye Hensley. "Citation Management Software". In: *Reference &amp; User Services Quarterly* 50.3 (Mar. 2011), pp. 204–208. ISSN: 1094-9054. DOI: `10.5860/rusq.50n3.204`. URL: `http://dx.doi.org/10.5860/rusq.50n3.204`.

[11] Catriona Macaulay et al. "Usability and User-Centered Design in Scientific Software Development". In: *IEEE Software* 26 (Jan. 2009), pp. 96–102. DOI: `10.1109/MS.2009.27`.

[12] *EasyBib: Free Bibliography Generator*. `https://www.easybib.com`. Accessed: 2023-10-05.

[13] *bib2doi: A Tool for Converting BibTeX Entries to Digital Object Identifiers*. `https://example.com/bib2doi`.

[14] *JabRef Reference Manager*. `https://www.jabref.org/`.

[15] *BibDesk Bibliographic Reference Manager*. `https://bibdesk.sourceforge.io/`.

[16] *Paperpile: Modern reference and PDF management*. `https://www.paperpile.com/`.

[17] *Zotero — Your personal research assistant*. `https://www.zotero.org`.

[18] *Google Scholar*. `https://scholar.google.com`.

[19] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[20] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *arXiv preprint arXiv:1908.10084* (2019).

[21] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. "A Primer in BERTology: What we know about how BERT works". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 842–866.

[22] Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change (2nd Edition)*. Addison-Wesley Professional, 2004.

[23] François Chollet. *Deep Learning with Python*. Manning, Nov. 2017.

[24] Nomic AI. *GPT4All: Accessible Pre-trained Models*. `https://gpt4all.io/`. 2022.

[25] Armando Fox. "Cloud Computing & What's in It for Me as a Scientist?" In: *Science* 331.6016 (2011), pp. 406–407. DOI: `10.1126/science.1198981`.

[26] Fernando Pérez, Brian E. Granger, and John D. Hunter. "Python: An Ecosystem for Scientific Computing". In: *Computing in Science & Engineering* 13.2 (2011), pp. 13–21. DOI: `10.1109/MCSE.2010.119`.

[27] Mark Lutz. *Programming Python*. O'Reilly Media, Inc., 2013.

[28] Jan Goyvaerts and Steven Levithan. *Regular Expressions Cookbook*. O'Reilly Media, Inc., 2012.

[29] Scott Chacon and Ben Straub. *Pro Git*. Apress, 2014.

[30] *Github*. `https://github.com/`.