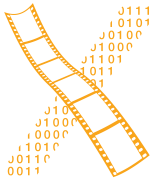

Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

# *Investigating the Use of Large Language Models for DIY Substitution Suggestions*

Master's Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

*by*
*Filiz Guenal*

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 14.02.2024
Submission date: 25.07.2024

Zentrales Prüfungsamt/Central Examination Office

# Eidesstattliche Versicherung
## Statutory Declaration in Lieu of an Oath

Name, Vorname/Last Name, First Name

Matrikelnummer (freiwillige Angabe)
Matriculation No. (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/ Masterarbeit* mit dem Titel
I hereby declare in lieu of an oath that I have completed the present paper/Bachelor thesis/Master thesis* entitled

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Für den Fall, dass die Arbeit zusätzlich auf einem Datenträger eingereicht wird, erkläre ich, dass die schriftliche und die elektronische Form vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
independently and without illegitimate assistance from third parties (such as academic ghostwriters). I have used no other than the specified sources and aids. In case that the thesis is additionally submitted in an electronic format, I declare that the written and electronic versions are fully identical. The thesis has not been submitted to any examination body in this, or similar, form.

Ort, Datum/City, Date

Unterschrift/Signature

*Nichtzutreffendes bitte streichen

*Please delete as appropriate

**Belehrung:**
**Official Notification:**

**§ 156 StGB: Falsche Versicherung an Eides Statt**
Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.
**Para. 156 StGB (German Criminal Code): False Statutory Declarations**
Whoever before a public authority competent to administer statutory declarations falsely makes such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment not exceeding three years or a fine.
**§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt**
(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.
(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.
**Para. 161 StGB (German Criminal Code): False Statutory Declarations Due to Negligence**
(1) If a person commits one of the offences listed in sections 154 through 156 negligently the penalty shall be imprisonment not exceeding one year or a fine.
(2) The offender shall be exempt from liability if he or she corrects their false testimony in time. The provisions of section 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Ort, Datum/City, Date

Unterschrift/Signature

# Contents

**Abstract** xi

**Überblick** xiii

**Acknowledgments** xv

**Conventions** xvii

**1 Introduction** 1

1.1 Motivation and Aim . . . . . 4

1.2 Outline . . . . . 5

**2 Related Work** 7

2.1 DIY Communities and Online Platforms . . . . . 7

2.2 DIY Tutorial Creation and Usage . . . . . 9

2.3 The Potential of Generative AI in DIY Contexts . . . . . 13

**3 LLM-based DIY Substitution Suggestion System** 15

3.1 System Overview . . . . . 15

3.2 Data Extraction from Supported Platforms . . . . . 16
3.2.1 Instructables . . . . . 16
3.2.2 YouTube . . . . . 18
3.3 Large Language Model Selection . . . . . 20
3.4 Considerations for System Output . . . . . 21
3.5 Prompt Engineering . . . . . 23
3.6 System Architecture . . . . . 28
3.6.1 Frontend . . . . . 28
3.6.2 Backend . . . . . 30
3.7 External APIs, Framework, and Libraries . . . . . 31
3.7.1 OpenAI API . . . . . 32
3.7.2 Youtube Data API v3 . . . . . 35
3.7.3 TransnetV2 Framework . . . . . 35
3.7.4 Libraries . . . . . 36

**4 Evaluation** **39**

4.1 Overview of the Evaluation . . . . . 39
4.2 Tutorial Selection . . . . . 40
4.3 Data Collection . . . . . 41
4.4 Statistical Analysis of Substitution Suggestion Identification . . . . . 41
4.4.1 Criteria for Substitution Suggestion . . . . . 42
4.4.2 Dataset Creation . . . . . 43

4.4.3 Metrics for Evaluation . . . . . 43
Accuracy . . . . . 44
Precision . . . . . 45
Recall . . . . . 45
F1-Score . . . . . 46
4.5 Quantitative Evaluation of Suggested Alternatives . . . . . 46
4.5.1 Basic Matching Percentage . . . . . 46
4.5.2 ROUGE Score . . . . . 47
4.5.3 BERT Score . . . . . 49
4.5.4 Data Merging and Data Storage . . . . . 51

**5 Results** **53**

5.1 Results for Substitution Suggestion Identification . . . . . 53
5.1.1 Quantitative Results . . . . . 53
Tutorial 1 . . . . . 54
Tutorial 2 . . . . . 55
Tutorial 3 . . . . . 55
Tutorial 4 . . . . . 56
Tutorial 5 . . . . . 57
5.2 Results for Suggested Alternatives . . . . . 58

**6 Discussion** **63**

6.1 Findings . . . . . 63

6.1.1 Findings and Implications for the Substitution Suggestion Identification . . . . . 63

6.1.2 Findings and Implications for the Suggested Alternatives . . 64

6.2 Limitations of the Current Evaluation . . . . . 65

**7 Summary and Future Work** **67**

7.1 Summary and Contributions . . . . . 67

7.2 Limitations . . . . . 68

7.3 Future Work . . . . . 69

**A ROUGE-L and BERT Score for Each Tutorial** **71**

**Bibliography** **85**

**Index** **89**

# List of Figures and Tables

2.1 Motivations for contributing to DIY communities . . . . . 8

2.2 The reasons why DIY community look at Instructables . . . . . 10

2.3 Influential features of DIY tutorials . . . . . 10

2.4 A taxonomy of the shortcomings identified in online DIY tutorials. . 11

2.5 Taxonomy of Human-Generative AI Interaction . . . . . 13

4.1 The core architecture of BERT score . . . . . 50

5.1 Summary of the Projects Analyzed . . . . . 54

5.2 Evaluation Metrics for All Project Analyses . . . . . 54

5.3 BERT F1-score versus ROUGE-L F1-score . . . . . 59

5.4 Mean values of precision, recall, and F1-metrics calculated with ROUGE-L and BERT scores . . . . . 59

5.5 Bar plots for all the metrics provided by the scores . . . . . 60

5.6 Score Metrics for All Projects . . . . . 61

A.1 Statistics for 2DuJT8-Gn8E . . . . . 71

A.2 Statistics for 80GjcPECN8 . . . . . 72

A.3 Statistics for Aluminium-Can-Roses . . . . . 72
A.4 Statistics for Amazing-3D-Projection-Pyramid-in-10-min-from-Clear 72
A.5 Statistics for Arduino-CNC . . . . . 72
A.6 Statistics for bIJY65guY9A . . . . . 73
A.7 Statistics for Cardboard-Chair . . . . . 73
A.8 Statistics for Chess-Salt-and-Pepper-Mills . . . . . 73
A.9 Statistics for Chess-table-Instruc-table . . . . . 73
A.10 Statistics for d1VcMybY4NQ . . . . . 74
A.11 Statistics for DIY-cat-tent . . . . . 74
A.12 Statistics for DU2R7S0oxx4 . . . . . 74
A.13 Statistics for Enchanted-Forest-Mushroom-Lights . . . . . 74
A.14 Statistics for Fiber-Optic-LED-Lamp . . . . . 75
A.15 Statistics for fjGcvfa4E7c . . . . . 75
A.16 Statistics for Flying-Captain-America-Shield . . . . . 75
A.17 Statistics for Holiday-Light-Tunnel . . . . . 75
A.18 Statistics for How-to-Build-an-Outdoor-Hammock-Stand-25 . . . . . 76
A.19 Statistics for IKEA-HACK-articulating-tablet-mount . . . . . 76
A.20 Statistics for Jewelry-board . . . . . 76
A.21 Statistics for JrG$_I nKQB6g$ . . . . . 76
A.22 Statistics for Laser-Cut-Ambient-Light-With-Kerf-Bends . . . . . 77
A.23 Statistics for Lego-USB-Stick . . . . . 77
A.24 Statistics for Make-a-LEGO-Abrams-Tank . . . . . 77

A.25 Statistics for Mario-Bros-Clock . . . . . 77

A.26 Statistics for mFQS9JySnZ8 . . . . . 78

A.27 Statistics for Pallet-Wine-Rack . . . . . 78

A.28 Statistics for Phenomenal-Augmented-Reality-Allows-Us-to-Watch-Ho . . . . . 78

A.29 Statistics for Pinball-Coffee-Table . . . . . 78

A.30 Statistics for QCGcmzkA12k . . . . . 79

A.31 Statistics for QdnTjmxpBuk . . . . . 79

A.32 Statistics for sDlyksf3ivQ . . . . . 79

A.33 Statistics for Sew-a-Where-the-Wild-Things-Are-hat-pattern . . . . . 79

A.34 Statistics for Shirt-Folding-Board-from-Cardboard-and-Duct-Tape . . 80

A.35 Statistics for Simple-Elegant-Guitar-Stand . . . . . 80

A.36 Statistics for Sous-vide-cooker-for-less-than-40 . . . . . 80

A.37 Statistics for sSgo$_h V - myg$ . . . . . 80

A.38 Statistics for TG1oCqnn7E4 . . . . . 81

A.39 Statistics for Tinker-Bell-Pixie-Dust-Pumpkin-Carving . . . . . 81

A.40 Statistics for turn-signal-biking-jacket . . . . . 81

A.41 Statistics for Turn-Yourself-Into-a-Cartoon . . . . . 81

A.42 Statistics for USB-Volume-Knob . . . . . 82

A.43 Statistics for Valentines-Day-Papercraft-Robot-Cupid . . . . . 82

A.44 Statistics for Weight-Bench-5-positionFlatIncline-doubles-as- . . . . . 82

A.45 Statistics for Wire-Wrapped-Tree-of-Life-Tutorial . . . . . 82

A.46 Statistics for Wooden-Chapati-Maker-at-Home . . . . . 83

A.47 Statistics for yD–59Kq7as . . . . . 83

# Abstract

There are so many online DIY communities which feature a comment section in which people share their knowledge. One of the reasons people get into interaction in the comments section or read them is to find a substitute for a material. However, these valuable comments can be hard to locate among many others, may be overlooked, or are challenging to understand in context. It is suggested to implement improved systems to make it easier for readers to contribute alternative suggestions. This thesis provides a web-based application designed to identify substitution suggestions from the comments section of online DIY tutorials. The system utilizes OpenAI's GPT-4o model to analyze tutorial instructions, accompanying images, and user comments. The performance of the artifact was evaluated through a two-phase process. Firstly, the system's ability to accurately identify substitution suggestions was examined, achieving high accuracy rates exceeding 89% across multiple tutorials. Secondly, the quality of the provided alternatives was assessed using ROUGE and BERT scores, demonstrating high mean values for the BERT scores, indicating alignment between the model's suggestions and the referenced comments. This research contributes to enhancing user experience within online DIY communities by efficiently providing a list of substitution suggestions for a given tutorial. While the initial results are promising, further evaluation is important to thoroughly understand the artifact's performance.

# Überblick

Es gibt viele Online-DIY-Communities, die einen Kommentarbereich haben, in dem Menschen ihr Wissen teilen. Einer der Gründe, warum Menschen in den Kommentarbereich einsteigen oder ihn lesen, ist die Suche nach einem Ersatz für ein Material. Diese wertvollen Kommentare können jedoch unter vielen anderen schwer zu finden sein, werden möglicherweise übersehen oder sind im Kontext schwer zu verstehen. Es wird vorgeschlagen, verbesserte Systeme zu implementieren, die es den Lesern erleichtern, alternative Vorschläge zu machen. In dieser Arbeit wird eine webbasierte Anwendung vorgestellt, die darauf abzielt, Substitutionsvorschläge aus dem Kommentarbereich von Online-Bastelanleitungen zu identifizieren. Das System nutzt das GPT-4o-Modell von OpenAI zur Analyse von Anleitungen, begleitenden Bildern und Nutzerkommentaren. Die Leistung des Artefakts wurde in einem zweistufigen Prozess bewertet. Zunächst wurde die Fähigkeit des Systems untersucht, Substitutionsvorschläge genau zu identifizieren, wobei hohe Genauigkeitsraten von mehr als 89% über mehrere Anleitungen hinweg erreicht wurden. Zweitens wurde die Qualität der angebotenen Alternativen anhand von ROUGE- und BERT-Scores bewertet, wobei sich hohe Mittelwerte für die BERT-Scores ergaben, was auf eine Übereinstimmung zwischen den Vorschlägen des Modells und den referenzierten Kommentaren hinweist. Diese Forschung trägt dazu bei, die Benutzererfahrung in Online-DIY-Communities zu verbessern, indem sie effizient eine Liste von Substitutionsvorschlägen für eine bestimmte Anleitung bereitstellt. Während die ersten Ergebnisse vielversprechend sind, ist eine weitere Auswertung wichtig, um die Leistung des Artefakts gründlich zu verstehen.

# Acknowledgments

I want to thank Prof. Dr. Jan Borchers and Prof. Dr.-Ing. Ulrik Schroeder for examining this thesis.

I would also like to thank my supervisor, Marcel Lahaye, whose support was invaluable to me. I sincerely appreciate his generosity with the time and energy he invested. With his feedback and guidance, writing my thesis became a journey I enjoyed, during which I learned a lot, and ultimately resulted in a thesis I am proud to submit.

Furthermore, I would like to thank i10 for supporting me along the way and providing guidance for my research during the introductory talk. I would especially like to thank Sarah Sahabi, who was always there whenever I had problems.

Last but not least, I would like to thank my family—Hulya Gunal, Aycan Gunal, and Neval Gunal. Your belief in me enabled me to achieve so many things, including completing this thesis. Thank you for always being there for me.

# Conventions

Throughout this thesis we use the following conventions:

- The thesis is written in American English.
- The first person is written in plural form.
- Unidentified third persons are described in female form.

Where appropriate, paragraphs are summarized by one or two sentences that are positioned at the margin of the page.

This is a summary of a paragraph.

# Chapter 1

# Introduction

DIY culture thrives through online communities

DIY is defined as the practice of making, altering, or fixing an item free from the help of paid experts [Kuznetsov and Paulos, 2010]. The low-cost and readily available tools, along with the novel sharing methods, have helped DIY culture expand its reach [Kuznetsov and Paulos, 2010]. Online communities have facilitated the DIY movement as they provide platforms for individuals to share their making processes with a larger audience [Tseng, 2016], argued to be the first step towards the democratization of manufacturing and personal fabrication [Mota, 2011]. There are thousands of online maker communities today varying in size and focus, such as Instructables, Thingiverse, and Youtube [Kuznetsov and Paulos, 2010]. These communities enable people to share their knowledge, get help when they are stuck in the making process, connect, and inspire others [Kuznetsov and Paulos, 2010; Tseng and Resnick, 2014].

DIY's societal impact debated in academia

Buechley et al. [2009] examined DIY, starting from the basics such as methods, communities, and tools. After a panel held at the CHI 2014 conference, the DIY movement became a particular area of interest in the broader Human-Computer Interaction community. In the panel, Fuchsberger et al. [2015] stated that DIY has the potential to be the "third industrial revolution." However, some researchers argue that DIY is a leisure activity that promotes well-being rather than active engagement in societal change [Taylor et al., 2016]. While some researchers

debate whether DIY activity is empowering or not, others have focused on online maker communities and guided research in the Human-Computer Interaction field, examining the challenges within these communities and the considerations needed to address them.

Tutorials crucial for DIY knowledge sharing

Acquisition of new knowledge and skills through self-dependence is the basis of DIY practices [Mota, 2011]. Kuznetsov and Paulos [2010] revealed that one of the main drivers for people to contribute to DIY communities is their core value of sharing information. People tend to learn more when they engage with others by teaching and sharing [Kuznetsov and Paulos, 2010], and novices frequently utilize online communities when they lack in-person assistance [Kwon et al., 2024]. Wang and Noe [2010] defines knowledge sharing as providing task specifics and know-how to aid others and engaging with them to tackle problems, foster creativity, or implement guidelines. Reflecting changes in tools, materials, and techniques within the DIY community, tutorials have gained more significance as a means of knowledge sharing [Tseng and Resnick, 2014].

Creating DIY tutorials has inherent challenges

Even though knowledge sharing is an important aspect of DIY, both creating tutorials and following them present their own challenges. Challenges in creating tutorials hold true for any type and complexity, as makers must go through several steps before putting up a tutorial online. These stages can be grouped into five: planning or designing the making process, getting the necessary materials and tools, measuring or marking out patterns or guides, processing materials with the tools, making the final touches on the parts of the artifact, and finally assembling each component. These stages are performed iteratively, making tutorials more prone to errors. Errors might include mentioning the wrong tools or materials, providing steps in a different order, or completely skipping some steps [Lakier et al., 2018]. One reason for skipping some steps is to write efficiently by only mentioning the necessary steps to remake the projects, failing to offer the challenges faced in the process [Tseng and Resnick, 2014].

Following tutorials presents unique difficulties

Torrey et al. [2009] analyzed how people engaged in craft activities make sense of online documentation. They found

two main problems. First, the information has to be enacted, meaning they have to translate knowledge into practice. Second, readers may not be using the same tools and materials as in the documentation and need to understand how to tailor the information to their unique situation [Torrey et al., 2009; Wakkary et al., 2015]. Skill gaps between the maker and the user when using the same materials or tools add another layer of difficulty to following tutorials [Wakkary et al., 2015; Dalton et al., 2014]. Sometimes, it is even hard to guess the necessary and available tools and competencies of the craftsman [Wakkary et al., 2015].

Material substitution is common in DIY projects

Some of the challenges faced when following DIY tutorials and instructions have led to a necessity for material and tool substitution [Lakier et al., 2018]. Readers often personalize projects [Dix, 2007; Oehlberg et al., 2015; Wolf and Mcquitty, 2011] or seek to improve upon them, requiring different sets of materials and tools [Tseng and Resnick, 2014]. However, even if they aim to replicate the project, they may not have access to the same products in their region [Wakkary et al., 2015; Saakes, 2009]. If they do have access to the materials and tools, they may still prefer to use what they have on hand or lack the necessary knowledge or skills to use them effectively [Dalton et al., 2014; Wakkary et al., 2015]. Furthermore, unclear instructions on how to use a specific tool may lead them to substitute it with another tool [Lakier et al., 2018].

Limitations in current DIY contribution methods

Kuznetsov and Paulos [2010] revealed the most common types of contributions as commenting, asking questions, and providing answers. Readers who use alternative materials, tools, or techniques have limited options to share their processes, typically through the comments section under the instructions. However, this practice may pose challenges in identifying changes, which could go unnoticed, be difficult to filter, or be hard to contextualize [Kuznetsov and Paulos, 2010]. Additionally, Lafreniere et al. [2021] highlighted emerging practices in the comments section, such as decoding tutorial content and verifying technical skills. They argue for new methodologies to be implemented in online tutorials to facilitate better these novel usage scenarios [Lafreniere et al., 2021].

Generative AI as a potential solution

Generative AI has the capacity to create content that is deemed novel and meaningful [Feuerriegel et al., 2023]. OpenAI's GPT-3, short for Generative Pre-Trained Transformer 3, was the pioneering large language model capable of performing a variety of text-processing tasks without requiring fine-tuning [Teubner et al., 2023]. Subsequently, other LLMs like GPT-4[1] and Copilot[2] have emerged and gained widespread use, transforming how we work and interact [Feuerriegel et al., 2023]. Researchers have explored methods to enhance interactions with Generative AI systems [Shi et al., 2024]. Prompt engineering, interactive systems, and visualization serve as robust platforms for collaboration, receiving guidance, accepting recommendations, and refining suggestions from Generative AI systems [Shi et al., 2024].

## 1.1 Motivation and Aim

Thesis explores LLMs for DIY substitutions

In this thesis, we focus on utilizing large language models to enhance the exchange of substitution suggestions. Kuznetsov and Paulos [2010] surveyed over 2600 individuals and found that the most common contributions in online DIY communities are commenting, asking questions, and providing answers. However, a study by Lafreniere et al. [2021] revealed that 51% of comments express praise, encouragement, or gratitude. According to Tseng and Resnick [2014], users share their modifications in the comments section, but these changes can go unnoticed, be difficult to filter, or be hard to contextualize. Another advantage of using large language models is for novice users who may not be familiar with technical terms and are seeking alternatives. With the artifact we provide, they will understand the substituted material and gain insight into why this substitution was made.

[1] `https://openai.com/index/gpt-4/` Accessed 25 June, 2024
[2] `https://copilot.microsoft.com/` Accessed 25 June, 2024

## 1.2 Outline

In Chapter 2 "Related Work", we explore the foundations of DIY communities and online platforms, examining the creation and usage of DIY tutorials. We also investigate the potential of generative AI in DIY contexts.

Chapter 3 "LLM-based DIY Substitution Suggestion System" presents the basis of our work, explains the system overview, data extraction methods from platforms like Instructables[3] and YouTube[4], and our process for selecting an appropriate Large Language Model. We discuss considerations for system output, our approach to prompt engineering along with the steps we followed to improve it, and explain the system architecture, including both frontend and backend. We also discuss the external APIs, frameworks, and libraries utilized in our system.

Chapter 4 "Evaluation" describes our methodology for assessing the effectiveness of our DIY substitution suggestion system. We explain our tutorial selection process, data collection methods, and present a statistical analysis of substitution suggestion identification. We also explain our quantitative evaluation of suggested alternatives, including metrics such as basic matching percentage, ROUGE score, and BERT score.

In Chapter 5 "Results", we present our findings from the evaluation, providing the accuracy, precision, recall and F1-score metrics for the results for substitution suggestion identification for five tutorials. We also share the comments which model falsely referenced as containing substitution suggestion. We also share the analysis of the performance of suggested alternatives with the scores explained in Chapter 4, which are ROUGE score and BERT score.

Chapter 6 "Discussion" provides an interpretation of our results, addressing the limitations of our current evaluation and exploring the implications of our findings.

[3] `https://www.instructables.com/` Accessed July 25, 2024
[4] `https://www.youtube.com/` Accessed July 25, 2024

Finally, in Chapter 7 "Summary and Future Work", we summarize our contributions, addressing the limitations of our current system, and propose suggestions for future research in the intersection of DIY communities and Generative AI.

# Chapter 2

# Related Work

## 2.1 DIY Communities and Online Platforms

Maker culture is a technologically-driven extension of DIY, emphasizing skill development and knowledge sharing.

Maker culture is an extension of the DIY movement, characterized by its versatility and reliance on technology [Landwehr Sydow, 2022]. Makers aim to develop their skills using new materials, tools, and processes. Their activities include remaking artifacts from online tutorials, personalizing existing items, combining multiple artifacts, and sharing their knowledge within the DIY community [Lakier et al., 2018]. According to Wolf and Mcquitty [2011], makers are motivated by their craftsmanship achievements and their desire for uniqueness.

Community engagement driven by inspiration, feedback, and social connections.

When it comes to contributing to communities, the primary motivations of makers include seeking inspiration for future endeavors, acquiring new concepts, and receiving feedback on their work from other community members [Kuznetsov and Paulos, 2010]. However, the motivations to contribute are not solely related to DIY projects. In fact, in a survey conducted by Kuznetsov and Paulos [2010], 80% of participants stated their primary motivation as 'meeting people with similar interests', as can be seen in the Figure 2.1. One participant articulated this as 'to feel connected to like-minded people'. Since these communities

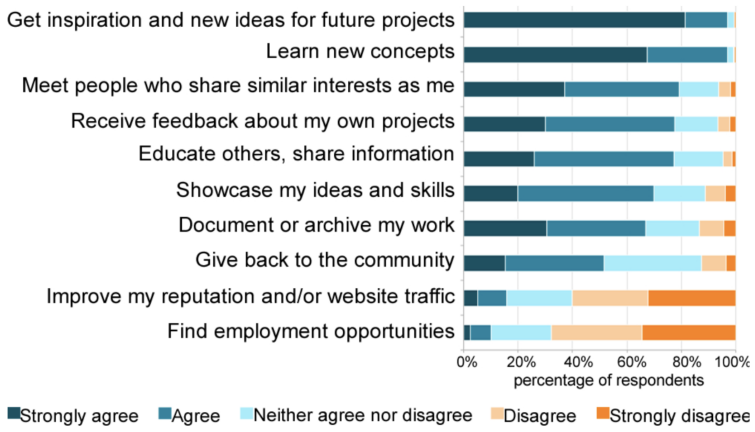


**Figure 2.1:** Motivations to contribute to DIY communities. Figure taken from Kuznetsov and Paulos [2010].

serve as a means to acquire new knowledge and inspiration, improving platforms for knowledge sharing becomes crucial [Kuznetsov and Paulos, 2010].

Online platforms democratize knowledge sharing.

Today, thousands of DIY communities with different sizes, structures, and project formats [Kuznetsov and Paulos, 2010] enable individuals to share their knowledge extensively. Although some of these communities are exclusively online[1] [2], they have significantly enhanced makers' ability to share knowledge on a broader scale. This has led to the democratization of the making process [Tseng and Resnick, 2014].

DIY projects range from woodworking to advanced technologies like 3D printing.

The range of project types makers share in these online DIY communities can vary widely. Some include a variety of project types such as woodworking, laser cutting, electronics, CNC, 3D printing, cooking, and decorating like Instructables, while others focus specifically on areas like 3D printing such as Thingiverse, knitting and crocheting as in Ravelry[3], or hip crafts like Crafter[4]. Additionally, makers utilize certain online social platforms to share their work, even though the primary audience is not necessar-

[1] `https://www.instructables.com/` Accessed July 25, 2024
[2] `https://www.thingiverse.com/` Accessed July 25, 2024
[3] `https://www.ravelry.com/` Accessed July 25, 2024
[4] `https://crafter.com/` Accessed July 25, 2024

ily the DIY community, such as YouTube[5] and TikTok[6]. On these platforms, various DIY tutorials may be shared and discovered, such as those on cosmetic chemistry or photography. Some individuals actively engage across multiple platforms within these communities. Often, individuals participate in several platforms for multiple reasons, one of which is connecting people with different skill sets [Kuznetsov and Paulos, 2010].

## 2.2 DIY Tutorial Creation and Usage

Motivations behind DIY tutorial usage include idea generation and skill acquisition.

The Figure 2.2. shows the answers to the question "Please rank, in order of importance, the reason why you look at Instructable" in a research conducted by Tseng and Resnick [2014]. This shows that the intentions behind utilizing instructions include 'getting ideas for a project' as the primary motivation, followed by 'learning a particular technique' and 'looking for projects I want to create'. These aspects highlight the diverse motivations within DIY communities. When it comes to the most prominent aspects of DIY communities, images of other projects hold the first place, followed by step-by-step instructions of other projects, and feedback and comments about other projects, as depicted in the Figure 2.3 [Kuznetsov and Paulos, 2010].

Challenges in DIY tutorial usage include time constraints and access to necessary tools.

Sharing knowledge through online DIY tutorials and following them pose various challenges [citation needed]. In the same study by Kuznetsov and Paulos [2010], they asked participants the main reasons for not sharing a project they completed. More than half of the participants cited lack of time as the main issue [Kuznetsov and Paulos, 2010]. According to Wakkary et al. [2015], the successful completion of a tutorial is often hindered by issues related to expertise, parts, and tools. Additionally, it is sometimes difficult to fully understand the required tools and materials used in the tutorial.

[5] `https://www.youtube.com/` Accessed July 25, 2024
[6] `https://www.tiktok.com/` Accessed July 25, 2024

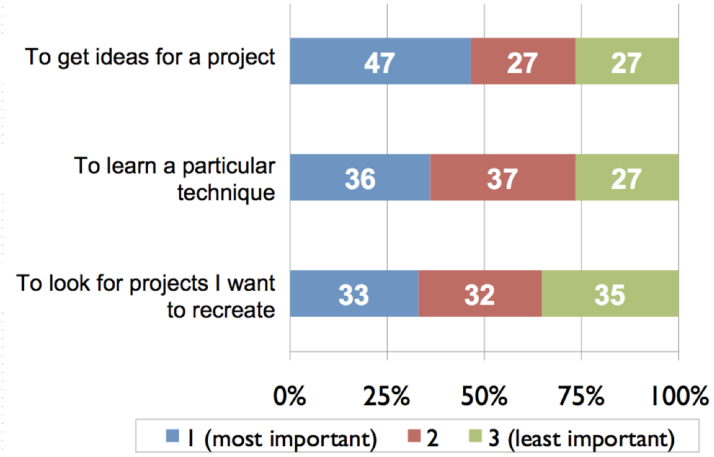


**Figure 2.2:** Answers to the question "Please rank, in order of importance, the reason why you look at Instructable". Figure taken from Tseng and Resnick [2014].

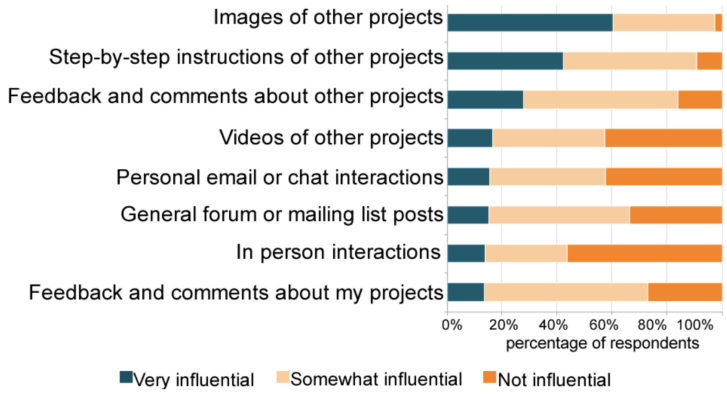


**Figure 2.3:** Features of DIY Instructions that are found influential. Figure taken from Kuznetsov and Paulos [2010].

Inadequacies in DIY tutorials include equipment, process descriptions, and technique explanations.

Lakier et al. [2018] provided a taxonomy of inadequacies within online knowledge resources as depicted in Figure 2.4. The taxonomy categorizes inadequacies into three main groups: equipment, process, and representation, each with various subgroups. Equipment includes materials, tools, parts and techniques. The main issue with materials in instructions is that authors either do not pro-

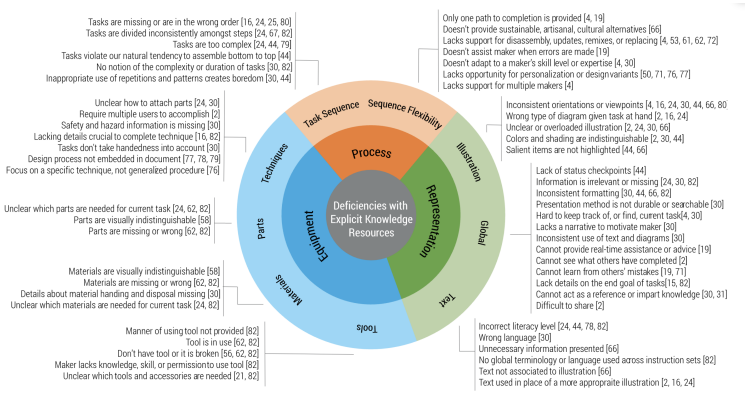


**Figure 2.4:** A taxonomy of the shortcomings identified in online DIY tutorials. Figure taken from Lakier et al. [2018].

vide the given material for a given step or mention incorrect ones. Similar problems exist for the parts.

Uses of online tutorials range from immediate assistance to ongoing skill refinement

Tools present a distinct challenge, as they often require specialized training or authorization, which may not be accessible to all users. The description and explanation of techniques in DIY instructions present particular difficulties. Techniques sometimes remain hidden in the process description, leaving individuals unaware of the required project techniques. In these situations, users are left to infer the appropriate technique, according to Lakier et al. [2018]. When a tutorial fails to provide all the required details to conclude each project phase, readers lose trust in the resource, and their engagement with the project diminishes as they have to look for other resources to acquire more information. Overall, even though each sub-category in the equipment poses unique deficiencies, they are mostly connected to a lack of clarity for the primary audience [Lakier et al., 2018].

Recommendations for improving tutorial effectiveness include better comment organization and feedback integration.

Lafreniere et al. [2021] conducted research on understanding the functions and applications of online tutorials, with a focus on the comments section. They analyzed over 600 comments. They found 'in-task help', 'ongoing skill refinement', and 'shadow and experience an expert's work practices' to be the three main uses of tutorials. For in-task help, individuals usually require urgent assistance with tutorial

content. In the case of ongoing skill refinement, individuals are typically not in immediate need but aim to acquire new skills. Expert shadowing allows novices to mimic the workflow and remake the project, even if they lack the necessary skills.

Generative AI applications in DIY contexts involve media creation from existing content.

When it comes to the use of comments, they have grouped them into five categories. First, readers communicate with authors in the comments section. 51% of the comments had some expression of admiration, support, or appreciation. In some cases, readers commented on the quality of the tutorial. The second use case involves motivation to validate their skill set and ensure they are practicing the best technique possible. However, in this use case, readers ask questions to other readers rather than the author. Third, readers also use the comments section to share their interpretations of the tutorial and help others avert common challenges by providing the errors in the instruction. The fourth group pertains to the refinement of instructions. In some comments, the researchers detected people confirming whether the instructions had worked out for them or not. In 16% of the comments, users provided guidance, recommendations, and alternative methods. Finally, the community also views the platform as an opportunistic assistance resource where they can ask questions roughly related to the tutorial, expecting that people interested in certain types of applications will read their questions and provide help [Lafreniere et al., 2021].

Human-generative AI interaction systems categorized by usage objectives such as enhancing outcomes and automating processes.

Lafreniere et al. [2021] suggested improving the comments section by tagging comments with relevant objectives such as thanks, problem-solving, or providing alternative techniques. They also recommended creating a collection of comments that could help improve tutorial content. Tseng and Resnick [2014] stated that changes readers make to tutorials are often provided in the comments section, which poses some difficulties for the community. These comments might be difficult to find among many others, can be missed, or are difficult to put into context. They also recommended implementing improved systems to facilitate readers in contributing alternatives.

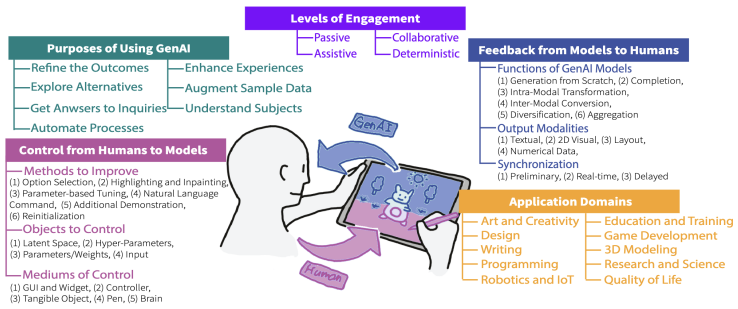


**Figure 2.5:** Taxonomy of Human-Generative AI Interaction. Figure taken from Shi et al. [2024].

## 2.3 The Potential of Generative AI in DIY Contexts

Muller et al. [2022] defined Generative AI as "an AI that uses existing media to create new, plausible media." They conducted a workshop to understand the challenges designers face when using Generative AI techniques. For instance, designers and users struggle to make sense of the internal mechanisms of Generative AI models. These models are ambiguous by nature and feature unpredictable elements. These characteristics of Generative AI methods introduce new challenges for designing human-centered systems, particularly in cases where users interact with AI [Muller et al., 2022].

Generative AI is an AI that can generate new content using the existing ones.

Shi et al. [2024] provided the first thorough literature survey in the field of human-generative AI interaction systems. They made a taxonomy of Human-Generative AI interaction, as shown in the Figure. The taxonomy involves five main sections: "purposes of using GenAI", "feedback from models to humans", "control from humans to models", "levels of engagement", and "application domains".

Taxonomy of Human-Generative AI interactions is defined.

They also defined the first step in analyzing Generative AI systems as identifying the purpose of usage. They classified usage purposes as improving outcomes, searching for alternatives, acquiring answers to questions, comprehending

Identifying the purpose of usage is the first step to analyze Generative AI systems.

a concept, automating procedures, enriching experiences, and elevating sample data [Shi et al., 2024].

# Chapter 3

# LLM-based DIY Substitution Suggestion System

In this thesis, our aim was to make knowledge sharing easier in the DIY community by creating a system that offers readers of a DIY tutorial a complete list of alternative suggestions for the materials, tools, or processes incorporated in the tutorial. Our thesis is centred on developing an application to achieve this objective, and in this section, we will provide a detailed explanation of each step of the design process and implementation. All the code is provided in the GitLab repository[1].

## 3.1 System Overview

Web-based app processes Instructables and YouTube tutorial URLs

We created a web-based application that takes the URL of an online DIY tutorial and provides a list of substitution suggestions as an output. The system only supports URLs for two different platforms: Instructables and YouTube. These websites are inherently very different in terms of

[1] `https://git.rwth-aachen.de/i10/thesis/thesis-filiz-guenal-llm-substitution-comments`

their website structures and the means by which they provide tutorials. These differences in the platforms brought about unique challenges, which were tackled by different approaches. Hence, once the website link is provided, our system first checks whether the given website link is for Instructables or YouTube.

Content extraction focuses on tutorial information and user comments

The next step is to extract content from these online tutorials. We were mainly interested in the information provided in the tutorial and the comments made on it. After fetching this content, we send this data to a large language model through an API call with a message requesting a comprehensive list of substitution suggestions made in the comments section in a specific format. Finally, we display the model's output message directly on the website.

## 3.2 Data Extraction from Supported Platforms

### 3.2.1 Instructables

Instructables: dynamic site needing specialized extraction techniques

Instructables[2] has several DIY tutorials, which are explained in text and sometimes supported by images. The website itself does not support video uploading, but users can embed videos from other platforms into their tutorials. Also, it is a dynamically loading website. In practice, this means that we cannot have the full HTML content of the website once we click on the links. For example, when we click on a link, we will not have the HTML content related to all the comments made in the tutorial. One has to click on the "More Comments" button until all the comments are loaded.

Cheerio and Puppeteer parse Instructables HTML content

The Cheerio[3] library is used for parsing HTML content. However, Cheerio alone may not be able to parse all the HTML content from dynamically loading websites. The Puppeteer[4] library is used to simulate user interactions,

[2] `https://www.instructables.com/` Accessed 25 July, 2024
[3] `https://cheerio.js.org/` Accessed 25 July, 2024
[4] `https://pptr.dev/` Accessed 25 July, 2024

such as clicking on the "More Comments" and "Show replies" buttons.

Data structure includes tutorial steps, comments, and images

After uploading all the HTML content to the webpage using Puppeteer, we extracted the relevant HTML parts to obtain tutorial steps, images used in the tutorial steps, and user comments. To extract these, we used Cheerio to target specific sections of the HTML document using CSS selectors. When it comes to instructions for the tutorial, we collected titles for each step along with a description. As for comments, we extracted the ID, username, and the text, including the same structure for replies to the main comments. Finally, we obtained URLs for images in HTTPS format.

We stored the data in the following JSON data structure:

```
{
  "tutorial": {
    "comments": [
      {
        "commentID": "string",
        "authorName": "string",
        "commentText": "string",
        "replies": [
          {
            "commentID": "string",
            "authorName": "string",
            "commentText": "string"
          }
        ]
      }
    ],
    "instructions": [
      {
        "titleofStep": "string",
        "descriptionofInstruction": "string"
      }
    ]
  },
  "imageUrls": ["string"]
}
```

### 3.2.2 YouTube

YouTube: requires transcript and frame extraction for analysis

Although YouTube[5] is not a website designed for the DIY Community, it offers a wide range of video DIY tutorials and a comment section in text format. Unfortunately, the language model we are using does not currently support video input. As a result, we had to find alternative methods to gather information from the tutorial. We tackled this in two steps: first, we obtained a transcript of the video, and second, we extracted frames from the video. The comments are already in text format, and we did not need to find workarounds to extract them.

There are challenges in extracting relevant transcripts from YouTube tutorials

First, we explored the possibility of obtaining video transcripts directly through an API. To the best of our knowledge, Google API[6] does not offer a video transcript retrieval feature. Afterwards, we searched for an existing library that could facilitate this task and discovered the Python library 'youtube-transcript-api'[7], which includes a function 'get_transcript()' to retrieve the transcript of a video by providing its ID.

There are also some challenges in extracting relevant frames from YouTube tutorials

As for getting the frames for YouTube videos, we had several challenges to get through. On YouTube, DIY tutorial videos may have frames that are loosely related or not related to the actual making process. For example, at the beginning of the video, makers just show themselves or go to the shops to buy the necessary materials and tools. However, the materials and tools that appear in the background may not be used in the making process. So, if we were to extract every frame from the video and provide this to a large language model, it might have resulted in erroneous results. We thought about manually taking screenshots for the frames we find useful, but this would make our system more cumbersome. Hence, we looked for existing frameworks that can extract relevant frames for a given goal from a given YouTube video.

[5] `https://www.youtube.com/` Accessed 25 July, 2024

[6] `https://console.cloud.google.com/apis/library?pli=1` Accessed 25 July, 2024

[7] `https://pypi.org/project/youtube-transcript-api/` Accessed 25 July, 2024

TransNetV2 detects scenes in YouTube videos

Although we could not find any frameworks that exactly do this, we found a framework that extracts frames from a video and can detect scenes. We downloaded the YouTube videos using the pytube library[8] and saved them in the local repository as video files. Then, we used the framework TransNetV2[9] to extract frames for this video and detect the scenes. For a video that was around 10 minutes long, we had around 10000 frames. We selected one frame per scene to reduce this number to one that might not exceed the token limits given by the API we use for the large language model. However, as the number of scenes per video can change and is not strictly correlated to the duration of the video, we kept in mind that we might need to reduce this number further later in the process, depending on the token limits.

Middle frame is selected from each scene for further evaluation

We selected the middle frame for each scene, thinking that it would provide the best representation of the context, and then saved them as PNG files. However, the API for the large language model only accepts images in HTTPs or decoded base64 format. Creating HTTPs would require cloud resources and make the process longer. To meet the API requirements, we decided to convert the PNG files into decoded base64 format using a library called 'base64'. This allowed us to process the data and make it compatible with the API.

YouTube Data API extracts video comments information

To extract comments from YouTube videos, we used the Youtube Data API v3[10]. This API provides direct access to all comments, usernames, and commentIDs without the need for other libraries or frameworks.

This is the JSON data structure that was stored at the end of the data extraction process for Youtube URLs:

```
{
  "tutorial": {
    "comments": [
      {
```

[8] https://pytube.io/en/latest/ Accessed 25 July, 2024

[9] https://github.com/soCzech/TransNetV2 Accessed 25 July, 2024

[10] https://developers.google.com/youtube/v3 Accessed 25 July, 2024

```
        "commentID": "string",
        "authorName": "string",
        "commentText": "string",
        "replies": [
          {
            "commentID": "string",
            "authorName": "string",
            "commentText": "string"
          }
        ]
      }
    ],
    "instructions": [
      {
        "descriptionofInstruction": "string"
      }
    ]
  },
  "imageUrls": "string"
}
```

JSON structure similar, with slight differences noted

This structure is very similar to the one we used for Instructables. The difference in the structure is that the instructions object has two subclasses: the title of the step and the description. This is because Instructables usually provide tutorials in steps. However, we were not able to clearly detect steps from YouTube transcripts. That is why we do not have subclasses for the "instruction" object.

## 3.3 Large Language Model Selection

OpenAI's models chosen for substitution suggestions list

In order to utilize this data to list substitution suggestions for readers of DIY Tutorials, we decided to use a large language model to identify these substitution suggestions from the given data. There are many available large language models, and they may perform better depending on the task they were given. However, testing all the large language models and finding which one performs best was out of the scope of our study. Hence, we opted for the models OpenAI[11] provides, which were the most commonly used generative large language models at the time.

[11] `https://openai.com/` Accessed 25 July, 2024

We started building the system with the model GPT-3.5 Turbo

Throughout the course of this thesis, several new generative large language models were developed by OpenAI and made publicly available. Initially, our web-based application used OpenAI's GPT-3.5 Turbo Model, which was the latest model developed by them at the time. This model can only process and provide text-based data. Hence, we started designing our system with these capabilities in mind.

System improved by utilizing newer models with advanced capabilities

During the development of our application, newer models were released with more advanced capabilities. These models include GPT-4, GPT-4 Turbo, and GPT-4o. GPT-4 can process both text and image data, but it can only generate text outputs. OpenAI claims that this model has better accuracy than their previous releases. Currently, GPT-4o is OpenAI's most advanced model, possessing the same intelligence as GPT-4 but with greater efficiency. It shares the same input and output structure as GPT-4, but will also be capable of accepting video and voice input in the future. With each new model release, we have adapted our system by refining prompts and adding new functionalities that are made possible by the capabilities of the new models.

## 3.4 Considerations for System Output

First, we defined the desired model output

Before feeding the data into our chosen model, we take the time to carefully structure the model's output. This allows us to guide the large language model to produce the desired output by providing the data along with a clear prompt.

Output structure aligns with DIY community values

We took into account the values, motivations, and needs of the DIY community when structuring the model's output. One of the motivations for contributing to DIY communities is the desire to connect with others and establish a sense of identity, so we decided to include the usernames of contributors in the output. To ensure no substitution suggestions are overlooked among many comments, we provided a comprehensive list with all the substitution suggestions provided. Another consideration was to provide context for each alternative. Therefore, we included information

about why a substitution suggestion was made and what it was substituted for. Providing this information by only analyzing tutorial comments failed in some cases. There were comments with substitution suggestions lacking context, as contributors did not mention why they used an alternative or what they substituted. To overcome this, we analyzed both the comments section and the instructions in a DIY tutorial. Through this analysis, we were able to identify the original material, tool, or process and how they were incorporated into the making process.

Output includes original items, alternatives, and contributors

In the end, we decided to have an outcome, which is a list that includes original material, tool or process used in the DIY tutorial, the substitution suggestion, and the username of the contributor. Later, we also included relevant ID for the specific comment to ease the data analysis process and possibly using this to find the comment for future applications.

Comprehensive List of Materials, Tools, and Processes with Substitution Suggestions:

1. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])

2. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])

3. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])

...

## 3.5 Prompt Engineering

We followed a guideline for the prompt engineering

After preparing the data and the desired output format for the model, we thoroughly considered formulating a suitable prompt that would yield an output as close as possible to our desired output. Following the prompt engineering principles and guidelines[12] established by OpenAI and DeepLearning.ai, we were able to formulate a prompt that enables the model to generate an output that aligns with the desired outcome.

Prompt engineering ensures clear, specific instructions

The first prompt engineering principle is to write clear and specific instructions. This can be achieved by using delimiters, asking for a structured output, checking whether requirements are met, and using few-shot prompting. The second principle is to give the model time to think. This principle can be applied by specifying steps to complete a task or instructing the model to develop its own solution before jumping to a conclusion. While considering these principles, we had to iterate our prompt many times based on the results we obtained.

We sent the data in a specific format to ensure clarity

To provide a clear definition and structure, we sent our data in JSON format, as shown previously, and asked OpenAI to identify substitution suggestions. We requested that these suggestions be provided in a list with a predefined structure. Our initial prompt was as follows:

[12] `https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/` Accessed 25 July, 2024

> Given a text in JSON format containing a series of comments, I am seeking substitution suggestions that are explicitly mentioned in the comments from contributors along with their usernames and ID of the comment. Please focus on offering alternatives or substitutions for the materials or tools explicitly mentioned in the JSON text. Be sure about the accuracy of the output, and do not make hallucinations. If there is no explicit substitution suggestion, do not list anything. Output Format:
>
> 'Comprehensive List of Materials and Tools with Substitution Suggestions:'
>
> Material/Tool (Alternative: Substitution suggestion | contributor ~ commentID)
>
> Material/Tool (Alternative: Substitution suggested | contributor ~ commentID)
>
> Material/Tool (Alternative: Substitution suggested | contributor ~ commentID)

Initial prompt had issues with irrelevant suggestions

The first problem we encountered was that some comments, while potentially perceived as substitution suggestions in a general context, were not relevant to DIY applications. We defined a substitution suggestion in the maker domain as a suggestion to replace or add a material, tool, or process in a project to change or improve at least one aspect of the project while keeping the overall result equal or similar. After incorporating this definition into the prompt, our results improved in detecting substitution suggestions as we had defined them.

Refined format to address naming and structural issues

The second issue arose from inconsistencies in naming concepts between the JSON text we provided, the instructions given to the model, and the output format we requested. For instance, in the JSON text, we labeled the username information as 'authorName', instructed the model to provide 'usernames', and used the placeholder 'contributor' to describe the desired output format. Another inconsistency

was our inclusion of processes in the substitution suggestion definition without reflecting this in the provided output format. Moreover, the defined output format did not clearly distinguish between placeholders and fixed text. We defined the desired output to minimize these problems as follows:

> Comprehensive List of Materials and Tools with Substitution Suggestions:
>
> 1. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])
>
> 2. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])
>
> 3. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])
>
> ...

Adjusted prompt for nuanced substitution suggestions

At times, the model identified questions that asked for substitutions as substitution suggestions themselves. To address this, we initially instructed it to provide only explicit substitution suggestions. However, this approach did not perform well, resulting in a significant decrease in the number of identified substitution suggestions. As a result, we added the following sentence to our prompt:

> Consider both explicit suggestions and nuanced or implied substitutions that can be reasonably inferred from the comments.

Change the prompt to make the model list multiple alternatives separately when applicable

There were instances where users suggested different alternatives for the same material, tool, or process. In these cases, the model's output was inconsistent in presenting multiple alternatives. Furthermore, when the model did provide multiple alternatives for the same item, the output format varied inconsistently. To address this issue, we added the following sentence:

> If there is more than one alternative for the same material, tool, or process, list them separately.

Analysis of instructions enhances context understanding

At this stage, we were solely analyzing comments to compile this list. However, we soon integrated instructions from the DIY tutorial to include the original materials, tools, or processes used, as we discovered that some substitution suggestions lacked this context. Upon incorporating this additional data, we observed that the model began identifying substitutions mentioned in the instructional section. To minimize these occurrences, we added the following sentence to the prompt:

> Disregard any substitution suggestions found in the project instructions.

Final prompt includes image analysis

Finally, we integrated images and frames from the instructions, leveraging the capabilities of the new large language models provided by OpenAI, which can process image inputs. To ensure the model accurately identifies the materials, tools, and processes depicted in these frames, we also instructed it to list them.

Some prompt engineering tactics were not helpful in our case

There were also cases where we experimented with tactics recommended by OpenAI but later decided to omit them as they did not yield satisfactory results in our specific case. One tactic we tested was to define everything in steps to allow the model more time to process information, but this approach did not perform well for us. It's possible

that the issue lies not just in defining steps but also in how we structured them or the specific wording we used. It's important to note that these tactics may still be effective in other contexts. Further iterations might have improved results, but due to time constraints in our thesis, we chose to end further experimentation. Our final prompt is given below.

We define substitution suggestions in the maker domain as recommendations to replace or add materials, tools, or processes in a project to alter or enhance at least one aspect of the project while maintaining the overall outcome.

Analyze the provided JSON text containing comments to identify substitution suggestions made by contributors regarding materials, tools, or processes used in the project. Consider both explicit suggestions and nuanced or implied substitutions that can be reasonably inferred from the comments. Disregard any substitution suggestions found in the project instructions.

If there is more than one alternative for the same material, tool, or process, list them separately.

Format the output as follows:

Comprehensive List of Materials and Tools with Substitution Suggestions:
1. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])
2. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])
3. [Original Material/Tool/Process] (Alternative: [Substitution Suggestion] | [authorName] ~ [commentID])
...

Include all relevant substitution suggestions from the comments, whether explicit or reasonably inferred, while maintaining transparency about the source and nature of each suggestion.

When analyzing the images, identify original materials, tools, or processes used in the project. Include these in your list, even if there are no substitution suggestions for them.

## 3.6 System Architecture

After finalizing the key decisions discussed in the previous chapters, we will now focus on designing the LLM-based DIY Substitution Suggestion System architecture in detail. This chapter will explain the reasoning behind our architectural choices, highlighting important components and how they interact with each other.

Client-server architecture chosen for performance and scalability

The system adopts a client-server architecture comprising a frontend for user interactions on the client-side and a backend for data processing and API integration on the server-side. The decision to implement a client-server architecture for the LLM-based DIY Substitution Suggestion System was based on optimizing performance, scalability, and user experience. By separating frontend and backend functionalities, we achieved better maintainability and the ability to update components without disrupting the entire system. Additionally, the client-server model supports scalability by efficiently managing varying loads and accommodating a growing user base without sacrificing performance. This architecture aligns well with modern development practices and enables the use of robust technologies for both frontend and backend, facilitating development efficiency and promoting compatibility with existing APIs and libraries.

In the upcoming part, we will conduct an in-depth analysis of the elements comprising our system architecture. Our focus will include the frontend design, structured to ensure a streamlined user experience, the backend infrastructure enhanced for efficient data processing and smooth integration, and the selection and integration of external APIs and frameworks for the integrity of system operations.

### 3.6.1 Frontend

HTML, CSS and JavaScript used for front-end

The front-end interface of the DIY Substitution Suggestion System comprises three primary components intended to enhance user interaction and efficiently present system out-

puts. These components are integrated using HTML for structural purposes, CSS for styling, and JavaScript for dynamic behavior.

Frontend includes URL input, submit button, and result display area

The URL input field allows users to enter the URL of the DIY project they want to analyze. It is an HTML <input> element styled with CSS. The placeholder text "Enter URL here" provides guidance for users when entering the project URL. Below the URL input field, there is a Submit Button with the ID "myButton" which triggers the analysis process when activated. The Result Display Area is an HTML <textarea> element with the ID "resultText". It is set to read-only to prevent user edits. This area displays the outcome of the analysis and contains the placeholder text "Result will be displayed here" to inform users of its purpose.

CSS flexbox used for centralized alignment and appearance

The front-end interface has been designed to utilize CSS flexbox, ensuring the centralized alignment of content both vertically and horizontally within the webpage. All interface elements are enclosed within a container  to maintain consistent alignment and a cohesive appearance. In order to guarantee readability across diverse devices and browsers, we have opted for the Arial font.

JavaScript handles user interactions and result updates

JavaScript enriches the user interface through handling user interactions. An event listener has been incorporated into the Submit Button to retrieve the URL provided by the user and initiate a POST request to the '/clicked' endpoint for subsequent backend processing. Upon receipt of a response, JavaScript proceeds to update the Result Display Area with the analysis results. Moreover, comprehensive error-handling mechanisms have been put in place to address and record any encountered issues, ensuring a high level of reliability.

### 3.6.2 Backend

Backend uses Node.js and Express.js for server setup

The DIY Substitution Suggestion System's backend infrastructure is built with Node.js, utilizing the Express.js framework for efficient server setup and route management. Node.js serves as a JavaScript runtime environment for executing server-side code, while Express.js provides a structured framework that simplifies backend development tasks.

Server initialization and route management are key components of the backend architecture

The backend architecture consists of several key components, each contributing to the system's functionality and flexibility. The server is initialized at its core using 'const app = express()' and configured to listen on port 3000 to handle incoming HTTP requests. Static files, important for rendering the frontend interface, are served from the 'public' directory using 'app.use(express.static('public'))', ensuring fast content delivery to users. Route management is handled through Express.js route handlers. For example, a GET route ('app.get('/')') is set up to serve the main HTML file, providing an uninterruoted navigation experience for users visiting the system's home page.

POST route handles URL submission and data processing

To enable dynamic interaction and data processing, the backend includes a POST route ('app.post('/clicked')') designed to handle URLs submitted by users. This route initiates essential operations for extracting and analyzing data from sources, which are Instructables and Thingiverse. Upon receiving a URL, the backend identifies its source and triggers corresponding data fetching functions. These functions utilize advanced techniques, including web scraping with Puppeteer and data extraction with Cheerio, to gather comprehensive project details, comments, and multimedia assets associated with the submitted URLs.

OpenAI and YouTube Data APIs integrated for analysis

The backend's capabilities are further enhanced through integration with external APIs. The OpenAI API is incorporated into the backend using const openai = new OpenAI( apiKey: '...'). This integration enables analysis of the retrieved data through the large language model, entrusting the system to identify substitution suggestions based on the collected tutorial data. Second used external API

is the YouTube Data API v3. It is utilized in the backend to retrieve data related to video comments, namely the ID, username, and text of the comment.

Error handling ensures robust system performance

The implementation of error handling mechanisms, through the utilization of try-catch blocks throughout the backend, serves to mitigate potential risks associated with API calls, web scraping operations, and data processing tasks, ensuring robust system performance and reliability. Moreover, errors encountered during these processes are recorded in the console, facilitating prompt debugging and maintenance. Additionally, utility functions such as URL validation (isYoutubeUrl, isInstructablesUrl, isThingiverseUrl), YouTube video ID extraction (extractVideoId), and Thingiverse URL modification (appendCommentsToThingiverseUrl) contribute to the backend's efficiency in processing and validating user-submitted URLs.

In conclusion, the backend structure of the DIY Substitution Suggestion System exemplifies an integration of Node.js, Express.js, external APIs, web scraping technologies, and error handling mechanisms. This architecture enables flexible handling of diverse DIY project URLs, comprehensive data gathering, and insightful analysis through advanced AI models. By combining these elements, the system delivers a robust platform for users to explore and discover alternative solutions tailored to their DIY needs.

## 3.7 External APIs, Framework, and Libraries

Our system architecture includes two external APIs, one framework, and several libraries. In this section, I will provide an overview of how these APIs and the framework operate, as well as discuss two libraries used in the project. Although we used other libraries as well, we believe they do not require a detailed explanation of why they were chosen or how they work.

### 3.7.1 OpenAI API

OpenAI API provides access to large language models

The OpenAI API[13] provides access to OpenAI's large language models. This access is necessary for obtaining a comprehensive list of substitution suggestions by identifying potential substitutions in the comments section, understanding the reasoning behind the suggested substitution, and determining the substituted material, tool, or process. To send an API request, we need to define a message and select which model to use. Message parts require the selection of a role and content. The role can be defined as 'user' or 'system', while the content can be text- or image-based depending on the model. There are several model options available, such as GPT-3, GPT-4, and GPT-4o. This API is integrated into the backend of the system and is used with an API key for authentication requirements.

We use the model GPT-4o

In our implementation, we utilize the OpenAI API through the requestChatGpt function. This function is designed to handle requests for both YouTube and Instructables by using different message structures. Separate message structure was needed to handle different image data formats we have for these websites. In image data we provide to OpenAI API, we use HTTPs links for Instructables, and decoded base64 format for selected YouTube frame. The message structure has been refined to handle these two different formats. Then, the API call is made using the openai.chat.completions.create() method, which takes two main parameters: messages and model[14]. The model we use is specified as 'GPT-4o'. This model selection can be easily adjusted as newer models become available. This is the JavaScript code for the API call:

```
1    const data = await openai.chat.completions.create
         ({
2    messages: messages,
3    model: 'gpt-4o',
4 });
```

Different message structures for YouTube and Instructables data

The messages array is constructed separately based on the

[13] https://openai.com/index/openai-api/ Accessed 25 July, 2024
[14] https://platform.openai.com/docs/models Accessed July 25, 2024

source of the data, either Instructables or YouTube. It uses the same prompt which provided in Chapter 3.5 before. This prompt provides context and instructions for the AI model.

For YouTube sources, the messages array includes:

- A system message with instructions.
- A user message containing:
  - A text introduction for project images.
  - Base64-encoded images from the project instructions (up to 3 images).
  - The project instructions and comments as text.

For Instructables sources, the messages array includes:

- The same system message as for YouTube.
- A user message containing:
  - A text introduction for project images.
  - Up to 3 image URLs from the project.
  - The project instructions and comments as text.

This is the JavaScript code we have formulated for the messages:

```
let messages;
if (source === "YouTube") {
    messages = [
        {
            role: "system",
            content: systemMessageContent
        },
        {
            role: 'user',
            content: [
                { type: "text", text: "Images from
                    Project Instructions:" },
                ...(base64Images ? base64Images.map(
                    base64Image => ({
```

```
                    type: "image_url",
                    image_url: { url: base64Image }
                })) : []),
                { type: "text", text: "\nProject
                    Instructions and Comments:\n" +
                    tutorial }
            ]
        }
    ];
} else if (source === "Instructables") {
    let imageUrlsArray = [];
    if (typeof imageUrls === 'string') {
        imageUrlsArray = imageUrls.split(',').map(url
            => url.trim());
    } else if (Array.isArray(imageUrls)) {
        imageUrlsArray = imageUrls;
    }

    imageUrlsArray.slice(0, 3);

    messages = [
        {
            role: "system",
            content: systemMessageContent
        },
        {
            role: 'user',
            content: [
                { type: "text", text: "Images from
                    Project Instructions" },
                ...limitedImageUrls.map(url => ({
                    type: "image_url",
                    image_url: { url: url }
                }))
            ]
        }
    ];
}
```

Error handling for API request process implemented

Error handling is implemented to catch and log any issues that occur during the API request process. This includes errors in data fetching, message construction, or the API call itself. By formatting our API requests in this manner, we enable the selected large language model to analyze textual and visual content from DIY tutorials. As a result, it can identify substitution suggestions made in the comments

section and provide them in a list, along with relevant context and credits in a format described in the prompt.

Token limitations restrict number of images sent

GPT-4 can process up to 128,000 tokens at once[15]. Due to these token limitations, we had to limit the number of images sent to the model to three. We chose to use the first three images for both Instructables and YouTube since materials and tools are typically displayed at the beginning of the instructions. While sending more than three images might be possible for some tutorials analyzed, we maintained this consistent limit to ensure uniform data processing.

### 3.7.2 Youtube Data API v3

YouTube Data API fetches comment data

Youtube Data API v3 is used to get data about comments from YouTube videos with an authentication key. This API is used to fetch the username, ID number, and text of the comment from Youtube videos[16]. As OpenAI models currently accept either image or text-based content as input, we had to process YouTube videos to provide tutorial information. In order to get visual information, we utilized the framework TransNetV2. This framework can extract key frames for a given video and provide the start and end timestamps for each scene. We saved middle frame for each scene for a given video to later provide these as an input to OpenAI API. For the voice part,we get transcript of the video using the library YouTubeTranscriptApi[17].

### 3.7.3 TransnetV2 Framework

TransNetV2 identifies shot transitions

The TransNetV2 Framework is a system designed to accurately identify shot transitions in videos. It uses a 3D convolutional neural network architecture that is optimized for video processing operations [Soucek and Lokoc, 2020].

[15] `https://platform.openai.com/docs/models` Accessed July 25, 2024

[16] `https://developers.google.com/youtube/v3/docs` Accessed July 25, 2024

[17] `https://pypi.org/project/youtube-transcript-api/` Accessed 25 July, 2024

The important features of TransNetV2 are as follows as given in the related paper by Soucek and Lokoc [2020] :

- **Architecture:** TransNetV2 utilizes a 3D CNN architecture with Dilated DCNN cells to detect subtle shot transitions in videos.
- **Batch Normalization and Skip Connections:** These are included to improve training stability and network performance.
- **Convolution Kernel Factorization:** TransNetV2 applies convolution to successfully extract image features and temporal relationships separately.
- **Frame Similarities as Features:** TransNetV2 applies learned features and similarity estimations to identify visual changes, which are assumed to be indicative of shot transitions.
- **Multiple Classification Heads:** TransNetV2 uses a dual-headed classification strategy to predict single-frame and multi-frame transitions within videos.

TransNetV2 used to capture video shot transitions

TransNetV2 processes YouTube videos to capture shot transition frames and timestamps in our system architecture. With this information, we were able to detect how many scenes are in a video. We selected the middle frame for each scene and used this as input for the OpenAI API. By integrating TransNetV2, our system gains an understanding of video tutorials, helping precise and relevant substitution suggestions for materials, tools, and processes used in DIY videos.

### 3.7.4 Libraries

Throughout the project, we utilized multiple libraries. In this section, we will give detail on two of them, which were

used to extract data from the Instructables website. By understanding how these libraries are utilized in the project, one can improve the system by supporting more DIY platforms.

Puppeteer and Cheerio extract data from Instructables

In order to extract data from the Instructables website, Puppeteer and Cheerio are used. Puppeteer, a Node.js library, is capable of loading data from websites with dynamic content, like Instructables. Meanwhile, Cheerio is a library that can parse and manipulate HTML content. Although some websites may require additional technologies for parsing, these two are generally sufficient for extracting data from many online DIY tutorial community websites.

Puppeteer mimics user interactions, Cheerio parses HTML

Puppeteer mimics user interactions such as clicking buttons and scrolling through pages. Cheerio works together with Puppeteer by parsing HTML content during web scraping operations, extracting specific elements such as comments, author details, and step-by-step instructions with given CSS selectors.

Effective use of Puppeteer requires understanding website structure

To effectively use Puppeteer, one must identify the website content that is not initially loaded and determine which buttons need to be clicked to load the HTML content they want to extract. For the Instructables platform, the buttons in question are "More Comments" and "Show Replies" for our system. When utilizing Cheerio, it is necessary to understand the HTML structure of the page and find unique selectors to precisely query and extract the desired data.

# Chapter 4

# Evaluation

This evaluation aims to validate the system's underlying concept. This concept can serve as a foundation for further research to explore the potential of large language models to identify substitution suggestions within the comments section of online DIY tutorials. We followed two distinct evaluation methodologies to gain more perspectives on the system's performance.

In this section, we will describe the methodologies used to evaluate the system's underlying concept. We will specify the metrics used for the evaluation, discuss the preliminary results obtained, and suggest directions for future work to extend the evaluation process.

## 4.1 Overview of the Evaluation

Two methodologies used to evaluate system conceptually

To assess the concept, we followed a systematic approach. First, we defined the criteria for selecting tutorials and identified the required data for each tutorial. Next, we collected all comments from these tutorials. Then, we obtained the model output for these tutorials and saved them in CSV files. To analyze the data, we followed two distinct methods.

First method: statistics on correctly identified substitutions

We calculated some statistics to assess the system's performance in identifying substitution suggestions given a set of comments. We calculated how many substitution suggestions were correctly identified and how many accurately mentioned the substituted material, tool or process. This evaluation was only performed for five tutorials, but it gave us valuable insight as we also tried to shed light on patterns where the model either failed to recognize valid substitution suggestions or provided incorrect substitution suggestions.

Second method: automated metrics on larger dataset

In the second phase of the evaluation, we expanded our analysis to a larger dataset of 47 projects. For this phase, we computed some scores by comparing the referenced comments for a particular substitution suggestion with the suggested alternatives by the model. While these automated metrics do not fully capture the system's ability to identify substitution suggestions, as, for example, these scores cannot differentiate a question from a statement, they provide valuable insights into aspects like the presence of hallucinations in the large language model. These scores help us understand how well the suggested comments align with the referenced comments in terms of content and meaning, though they have limitations in assessing the precise appropriateness of each substitution.

Common processes: tutorial selection and data collection

In the upcoming sections, we will first provide the initial process of the evaluation which were common for both evaluations. These processes include tutorial selection and data collection. Then, we will give more detail on how we performed these two evaluation methods.

## 4.2 Tutorial Selection

Criteria set for tutorial selection from platforms

We considered the system requirements and the characteristics of the chosen online DIY platforms to set clear criteria for tutorial selection. For tutorials from Instructables, we initially selected tutorials with a maximum of 30 images to not exceed the token limits of the selected large language model. However, we later realized that this criteria was not restrictive enough, leading us to select only the first three

images in the tutorial. Even so, some tutorials exceeded the maximum token limit and were automatically excluded from the analysis as we did not get the model output data for those tutorials. Regarding YouTube tutorials, we established two criteria. Firstly, each video should contain only one DIY tutorial, as we found videos containing multiple tutorials. Secondly, we aimed to select tutorials with informative transcripts, ensuring the audio of the video has information about the DIY tutorial and there is not only background music.

## 4.3 Data Collection

Data collected includes comments, model output, project details

For the data collection, we first considered which data we might need for the data analysis. For the dataset obtained from the tutorial, we got all the comments along with their relevant comment IDs to match the comments in this dataset and the model output dataset later. Also, for the model output dataset, we saved the contributor's username and the overall model output. Later, we saved the provided alternative and model output output separately.

Some other information about tutorials also saved for possible further research

Along with these, we saved the number of used materials, number of used tools, list of materials used, list of tools used, number of comments in the tutorial, and type of the project as machinery, no machinery or digital machinery for possible further research to understand the patterns.

## 4.4 Statistical Analysis of Substitution Suggestion Identification

Labelled comments for the analysis

To evaluate the performance of the system in terms of identifying substitution suggestions, we first labelled each comment based on whether they included substitution suggestions in the tutorial dataset. We did a similar labeling for the model output dataset. We labelled whether the substitution suggestion is correct and, if so, whether the mentioned original item is correct. For both labeling, we fol-

lowed the criteria that we set for substitution suggestions. Then, using these two datasets, we evaluated accuracy, precision, recall and F1-score for 5 tutorials. Although the dataset were pretty limited, it gave us some valuable insights about some patterns.

### 4.4.1 Criteria for Substitution Suggestion

Criteria defined for identifying substitution suggestions

Criteria were needed to determine what could be considered a substitution suggestion to label the data. We already defined substitution suggestions and put this into the prompt we use when we send data to the model. However, there were cases in which we had to think more about whether something could be accepted as a substitution suggestion during the evaluation.

Questions without confirmation not considered as substitution suggestions

For example, there were questions asking whether a certain material, tool, or process can be substituted with another one. If there were no reply confirming it can be substituted, these comments categorized as not containing substitution suggestion.

Improvement suggestions classified as substitution suggestions

Another case involved suggestions to improve the project by adding more materials, tools, processes, or combinations of them. Even though no substituted items were involved, these suggestions aligned with our definition of substitution suggestions. Hence, we classified them as containing substitution suggestions.

Source information for materials and tools not considered substitution suggestions

Finally, some comments requested sources for specific materials and tools, or reported where they found them. While this information is valuable for sharing knowledge, it does not align with our definition of substitution suggestions. Therefore, these types of comments were not considered as substitution suggestions.

### 4.4.2 Dataset Creation

Ground truth data created by labelling

In order to create the ground truth data for this method, we utilized the tutorial dataset. For that, we labeled each comment as whether it has a substitution suggestion or not based on the criteria we defined. At the end, we had the following columns in the ground truth data:

- Comment ID parsed from the Website
- Comment parsed from the Website
- Username of the contributor parsed from the Website
- Label showing how many substitution suggestions the comment contains, manually identified using the defined criteria

Model output dataset created for comparison

For the model output, we used the model output dataset. We checked whether the identified substitution suggestion aligned with the defined criteria. Finally, we had the data with the following columns:

- Comment ID identified by the model
- Suggested alternative identified my the model
- Complete model output which is shown in the website
- Label showing if the substitution suggestion is correct

### 4.4.3 Metrics for Evaluation

Metrics: accuracy, precision, recall, F1-score for evaluation

In this method, we aimed to evaluate the model's performance in identifying substitution suggestions within software project comments. To achieve this, we computed statistical metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into various aspects

of the model's performance, including its ability to correctly identify substitution opportunities and avoid unnecessary suggestions. Although this analysis was conducted on only five projects to validate the concept, we also performed a qualitative analysis to identify patterns of success and failure.

We used the following definitions to calculate the accuracy, precision, recall, and F1-score:

- **True Positive (TP):** Substitution suggestions that were correctly identified by the model.
- **False Positive (FP):** Instances where the model incorrectly identified a substitution suggestion.
- **False Negative (FN):** Actual substitution suggestions that were not identified by the model.
- **True Negative (TN):** Comments that did not have substitution suggestions and were correctly identified as such.

#### Accuracy

Accuracy measures the proportion of correctly identified instances (both true positives and true negatives) out of all instances. It provides an overall measure of how well the model performs across all comments.

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

In our case, this formulation would look like this:

$$\text{Accuracy} = \frac{\text{Correctly Categorized Comments}}{\text{Total Number of Comments}}$$

**Precision**

Precision is the ratio of correctly identified substitution suggestions (true positives) to the total number of substitution suggestions made by the model (both true positives and false positives). It indicates how reliable the model's suggestions are when it identifies a substitution opportunity.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

In our case, this formulation would look like this:

$$\text{Precision} = \frac{\text{Correctly Identified Substitution Suggestions}}{\text{Total Substitution Suggestions Identified by Model}}$$

**Recall**

Recall measures the proportion of actual substitution suggestions that the model correctly identified. It reflects the model's ability to capture all possible substitution opportunities.

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

In our case, this formulation would look like this:

$$\text{Recall} = \frac{\text{Correctly Identified Substitution Suggestions}}{\text{Total Substitution Suggestions}}$$

**F1-Score**

The F1-score is the harmonic mean of precision and recall. It provides a balance between the two metrics, offering a single score that considers both false positives and false negatives, which is particularly useful when dealing with imbalanced datasets.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

By calculating these metrics, we gain comprehensive insights into the model's strengths and weaknesses in identifying valid substitution suggestions and avoiding incorrect ones. This analysis helps us understand the model's effectiveness and areas for improvement in processing software project comments.

## 4.5 Quantitative Evaluation of Suggested Alternatives

Second phase of the evaluation: linguistic similarity

During the second phase of our evaluation, we extended our analysis to a larger dataset comprising 47 projects. This phase was intended to evaluate the linguistic and semantic similarity between the model's suggested alternatives and the referenced comments for each substitution suggestion. To achieve this, we used three complementary metrics to evaluate these aspects: Basic Matching Percentage, ROUGE-L Score, and BERT Score.

### 4.5.1 Basic Matching Percentage

Basic matching percentage measures word overlap

The basic matching percentage is a method to measure how many words in the alternative provided by the model's output also exist in the words in the referenced comment.

This metric is independent of the length of the comment

This metric calculates the proportion of identical words

shared between the two texts divided by the words in the alternative. This makes the metric independent from the length of the comment.

For example, if the original comment is "I used a metal component instead of rubber to make the product stronger." and the alternative is provided as "metal" by the model, the percentage would be 100%. If the alternative would be "metal component to enhance strength", the percentage would be 60%, as the common number of words is three, which are "metal", "component", and "to", and the total number of words in the alternative is given.

Limitations of basic matching percentage

Although this metric might be helpful to spot hallucinations made by the model when the score is zero, for example, it provides limited information. Even a score of 0 may not reflect the model's performance, as there might be some synonyms used in the alternative section. Similarly, a score might be high due to some words like "to", "a". If we use the previous example comment, which is "I used a metal component instead of rubber to make the product stronger", and if the alternative this time is the "a wood component to insulate". This time, the common words would be "a", "component" "to", and we will still get 60% as a result, even though the provided alternative is wrong.

### 4.5.2 ROUGE Score

ROUGE score has five types

The ROUGE score is being used to evaluate the performance of summarization tasks done by large language models by comparing them to reference summaries that were created by humans. This score has five different types, and their suitability may vary based on the specific use case, as stated by Lin [2004].

- **ROUGE-N:** Computes the overlap of n-grams between the prediction and reference summaries.
- **ROUGE-L:** Calculates the longest common subsequence between the prediction and reference summaries.

- **ROUGE-W:** A version of ROUGE-L that gives more importance to consecutive matches.
- **ROUGE-S:** Measures skip-bigram co-occurrence statistics between summaries.
- **ROUGE-SU:** An version ROUGE-S that also includes unigram matching.

ROUGE-L chosen for flexibility and suitability

We used the ROUGE-L score because it is more suitable to evaluate the overlapping between the original comment and the provided alternative. First reason why we used ROUGE-L is that DIY comments often feature different phrasings and word orders, and this is something ROUGE-L handles effectively by evaluating the longest common subsequence rather than relying on exact n-gram matches. This flexibility makes it more adaptable to our text types. Additionally, the alternatives provided by the model are usually short compared to the original comment, and ROUGE-L has demonstrated strong performance in evaluating short summaries, as shown in the DUC 2003 data for headline-like summaries. Finally, ROUGE-L is a model that is a more straightforward approach that is easier to interpret than the other models. These reasons make ROUGE-L a better choice to evaluate how the alternative is presented to the users.

Five main parameters of ROUGE-L

In order to apply ROUGE-L score, we utilized the "rouge-score" library[1] in Python. This metric takes three mandatory arguments[2]:

- **Predictions:** The provided text for the alternative is given as predictions.
- **References:** The original comment corresponding to the given comment ID by the model for the alternative is given as reference.
- **Rouge_types:** "rougeL" is selected as this is for the model ROUGE-L we preferred.

[1] `https://pypi.org/project/rouge-score/`
[2] `https://huggingface.co/spaces/evaluate-metric/rouge`

- **Use_aggregator:** This parameter is kept true as in the default case. This means that ROUGE calculates an overall score by aggregating the scores of each text.
- **Use_stemmer:** We enable word stemming by setting this parameter to true, which strips word suffixes and reduces words to their root. This facilitates a more accurate comparison between predicted and reference texts.

ROUGE-L performance indicators: precision, recall, F1-score

After setting parameters and applying the ROUGE-L score, we will have three performance indicators: precision, recall, and F1-score. In this project, these indicators reflect the following:

- **Precision:** How much of the text for the alternative consists of sequences that are also found in the original comment.
- **Recall:** How well the text for the alternative overlaps the sequences from the original comment.
- **F1-Score:** This score is the harmonic mean of precision and recall. A higher score means that both the precision and recall are high. In other words, a high score indicates a more accurate and comprehensive text for the provided alternative, which is created by the model.

### 4.5.3 BERT Score

BERT Score evaluates contextual overlapping

This metric evaluates the overlapping contextually. The other two scores could not understand the context and could only provide surface-level lexical matching. The BERT score allows us to determine whether the provided alternatives retain the intent and meaning of the original comments, even if the wording is different.

The Figure 4.1 illustrates the core architecture of the BERT score, which is as follows [Zhang et al., 2019]:

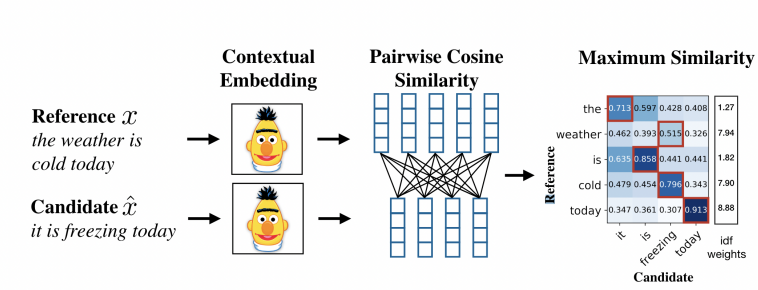


**Figure 4.1:** The core architecture of BERT score. Figure taken from Zhang et al. [2019]

- **Contextual Embedding:** Each sentence in the reference and candidate text is tokenized by using contextual embeddings.
- **Pairwise Cosine Similarity:** Reference and candidate sentences get tokenized, and the tokens are represented as contextual embeddings.
- **Maximum Similarity:** Then, each token from the reference text matches with the most similar token in the candidate text, or vice versa.

In order to apply BERT score, we utilized the "bert-score" library[3] in Python. This function takes three mandatory arguments:

- **Predictions:** The provided text for the alternative is given as predictions.
- **References:** The original comment corresponding to the given comment ID by the model for the alternative is given as reference.
- **Lang:** This parameter is to define the language. We selected it as English, as the majority of the analysis of our comments were written in English.

[3] `https://pypi.org/project/bert-score/` Accessed 25 July, 2024

After setting parameters and applying the BERT score, we will have the precision, recall, and F1-score. These measures will provide the same information as those provided for the ROUGE-L score.

### 4.5.4 Data Merging and Data Storage

Parsed data to smooth analysis

To make the analysis process smoother, we gathered some data from the ground truth dataset and model output data by using the comment IDs. For each comment identified as containing substitution suggestion, we get the comment from the ground truth data using commentID, and the identified alternative from the model output data. Eventually, we applied the previously mentioned scores (basic matching score, ROUGE-L score, and BERT-score) and saved all the obtained metrics in different columns.

Finally, we had a CSV file with the following columns:

- Comment ID
- Original Comment
- Suggested Alternative
- GPT Output
- Matching Percentage
- ROUGE-L Precision
- ROUGE-L Recall
- ROUGE-L F1-Score
- BERT Precision
- BERT Recall
- BERT F1-Score

Later, the mean value, standard deviation value, minimum value, and maximum value of the metrics provided by the

ROUGE-L and BERT score for each tutorial are calculated and saved to provide a better overview of the results. The basic matching percentage is excluded from this as it did not provide much insight.

# Chapter 5

# Results

In this section, we provided the results of both evaluations methods employed. For the first evaluation method, which is the statistical analysis of substitution suggestion identification, we will provide the statistical analysis and also analyze the results qualitatively to shed light on the patterns we encountered. For the second evaluation method, which is the quantitative evaluation of suggested alternatives, we will provide the metrics for an overall matching score, ROUGE-L score, and BERT score.

## 5.1 Results for Substitution Suggestion Identification

### 5.1.1 Quantitative Results

In this section, we provide precision, recall, accuracy, and F1-Score metrics for the model's performance to identify the substitution suggestions.

5 tutorials were analyzed for this evaluation

As mentioned in the evaluation part, we did the analysis for five different tutorials that were randomly selected from our tutorial dataset.

These tutorials had different properties in terms of the project type, number of comments, number of materials used, number of tutorials used, and platform. In the Table 5.1, we are providing the properties for these 5 tutorials.

| DIY Tutorial | Number of Comments | Number of Materials and Tools | Platform |
|---|---|---|---|
| 1 | 218 | 4 | Instructables |
| 2 | 424 | 6 | Instructables |
| 3 | 459 | 12 | Instructables |
| 4 | 81 | 7 | Instructables |
| 5 | 87 | 7 | YouTube |

**Table 5.1:** Summary of the Projects Analyzed

The Table 5.1 shows four tutorials from the DIY community website Instructables and one from YouTube. The number of comments ranges between 81 and 459, and the number of materials and tools hovers over 4 to 12.

For each tutorial, we got the following results given in Table 5.2:

| Tutorial | Precision | Recall | Accuracy | F1-Score |
|---|---|---|---|---|
| 1 | 0.8750 | 0.3500 | 0.9083 | 0.5000 |
| 2 | 0.9412 | 0.2667 | 0.8939 | 0.4195 |
| 3 | 0.8667 | 0.3421 | 0.9412 | 0.4906 |
| 4 | 0.7143 | 0.7143 | 0.9506 | 0.7143 |
| 5 | 0.6667 | 0.2500 | 0.9195 | 0.3636 |

**Table 5.2:** Evaluation Metrics for All Project Analyses

Table 5.2 presents the evaluation metrics for each project, including precision, recall, accuracy, and F1-Score. The results are summarized in the following along with some comments to later discuss the potential patterns of the model.

#### Tutorial 1

Tutorial 1: High precision and accuracy, low recall

Tutorial 1 achieved a precision of 0.8750 and an accuracy

of 0.9083. However, its recall was relatively low at 0.3500, leading to an F1 Score of 0.5000. This suggests that while the model was accurate in its predictions, it missed a significant portion of relevant items.

The only substitution suggestion that was provided but was categorized as false truth was the following comment:

> "You get it normal stationary shops here ask for OHP Sheets"

This comment suggests a method to find a specific product, but it does not align with our definition of suggested alternatives, as the mentioned materials were only referenced in the tutorial.

**Tutorial 2**

Tutorial 2: Highest precision, lower recall

Tutorial 2 demonstrated the highest precision of 1.0000 but had a lower recall of 0.2667, resulting in an F1 Score of 0.4195. This indicates that the model was less effective in capturing all relevant instances in this case than it was for Tutorial 1.

**Tutorial 3**

Tutorial 3: Balanced precision and accuracy, moderate recall

Tutorial 3 showed a precision of 0.8667 and an accuracy of 0.9412, similar to Tutorial 2 in terms of accuracy but with a slightly better precision. Its recall was 0.3421, leading to an F1 Score of 0.4906, reflecting a balance between precision and recall.

There were two false positives in the analysis. First one was the following:

> "Does anyone know if using E6000 would work? Because me and a friend want to test it out."

In this comment, "E6000" is provided as an alternative. However, the comment just asks whether E6000 could be used or not, and there were no confirming replies. Hence, this is categorized as a false negative.

The other false positive result was the following:

> "wow, i tried it and it works! Thx"

Here, the suggested alternative was "Smaller 2X2 LEGO brick for flash drives." Although the provided alternative was made in some other comments, the provided comment ID and the author were incorrect. Hence, this result also counted as a false positive.

**Tutorial 4**

Tutorial 4 had notably high values for both recall (0.7143) and accuracy (0.9506), with a precision of 0.7143 and an F1 Score of 0.7143. This project performed well across all metrics, indicating a balanced performance with high effectiveness in both identifying and correctly predicting relevant items.

There were two false positives in the results for this tutorial. One is the following:

> "Do you think using a real pumpkin you wouldn't get the same effect because the candle isn't as bright?"

Two substitution suggestions were provided by referencing this comment. The first one was "Candle", and the second one was "Battery Candle Light". This comment gets a reply from the author of the author:

> "I used the artificial pumpkin cause I could take it out every year and a real pumpkin probably

> be hard to get the fine detail cuts, Candle lights. now I use the battery candle light in our kids room for a night light but in this picture I used a Mini light bulb gave it a brighter look on tinkerbell"

In the reply, the author suggests battery candle lights but not Candles. Hence, the provided alternative, "Candle," is categorized as a false positive.

**Tutorial 5**

Tutorial 5 (YouTube): Lowest precision and recall

Tutorial 5, the only project selected from YouTube, had the lowest precision (0.6667) and recall (0.2500), resulting in an F1 Score of 0.3636. Despite its relatively high accuracy of 0.9195, the tutorial struggled with lower precision and recall, reflecting difficulties in both capturing relevant instances and minimizing false positives.

In this tutorial, there were 8 substitution suggestions and 2 of them correctly identified by the model. There were also two false positives.

One false positive in the results for Tutorial 5 were the following:

> "Awesome mirror are you using sandpaper made for glass?"

This comment involves a question regarding the type of sandpaper used in the tutorial. This comment was identified as containing a substitution suggestion, with the suggested substitution given as "Specifically made for glass sandpaper". There was one reply to this command, stating they used a normal sandpaper.

The other false positive result stems from the format of the question. 4 substitution suggestions referred to the same issue and suggested the same thing. The model provided the output in the following style:

Safety Measures (Implied suggestion: Use of safety glasses, gloves, and mask recommended by multiple comments for protection | Various Authors M̃ultiple CommentIDs)

Although the identified substitutions are all correct, no specific comment ID or username was provided. Hence, this is categorized as a false positive. If we were to assume that 7 out of 8 substitution suggestions were correctly identified, we would get a better result for all four metrics.

## 5.2 Results for Suggested Alternatives

In this section, we will provide the results to understand how well the suggested substitution is given by the model, considering the referenced comment.

We calculated a basic matching score, ROUGE-L score, and BERT score for 47 different tutorials from YouTube and Instructables, with varying numbers of materials and tools and volumes of comments. The number of comments ranged between 18 and 459. There were 14 tutorials from YouTube and 33 from Instructables.

We included statistics for each tutorial in Appendix A, showing the mean, standard deviation, minimum, and maximum values for metrics calculated by ROUGE-L and BERT scores. The Figure 5.3 shows the relation between ROUGE-L F1-Score and BERT F1-Score. Even though there is a correlation, there are multiple data points, which is 0 for the ROUGE-L F1-Score but higher than 0.75 for the BERT F1-Score.

Figure 5.4 presents a bar plot comparing the mean values of ROUGE-L Score and BERT Score metrics for our analysis, while Table 5.6 provides the exact values for these metrics, along with standard deviation, minimum, and maximum values. The Figure 5.4 shows that the mean precision calculated using the BERT score (0.867) is higher than the precision obtained with the ROUGE-L score (0.633).

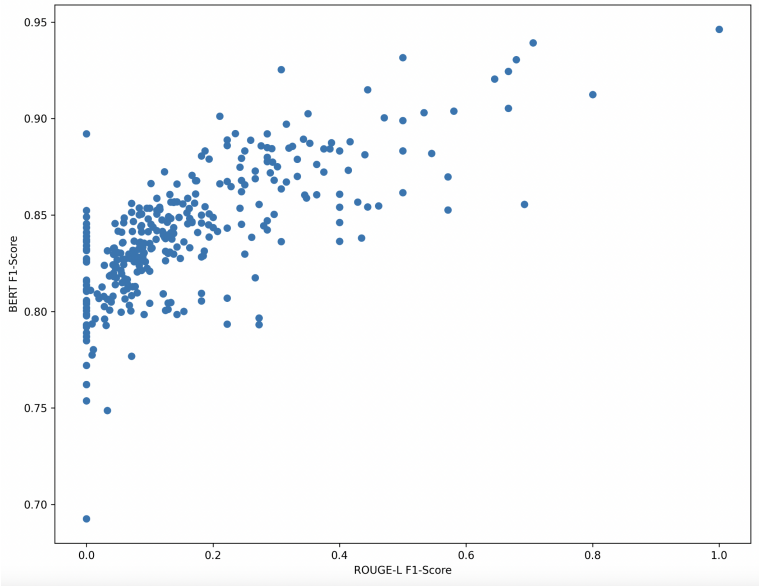


**Figure 5.3:** BERT F1-score versus ROUGE-L F1-score

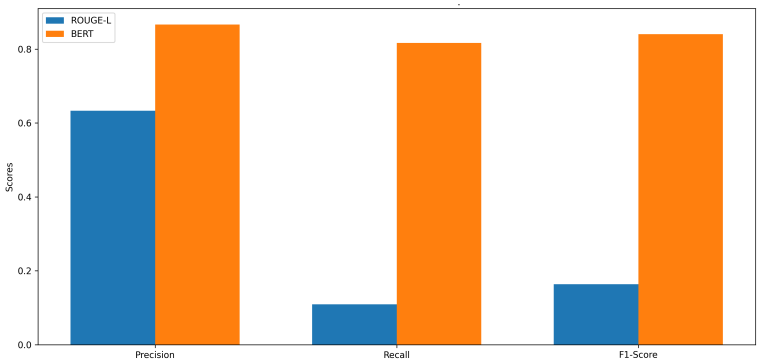


**Figure 5.4:** Mean values of precision, recall, and F1-metrics calculated with ROUGE-L and BERT scores

Additionally, there is a difference between the recall and F1-Score metrics of the two methods. Both recall and F1-Score values are higher for the BERT score (0.817 and 0.840, respectively) compared to the ROUGE-L score (0.109 and 0.164, respectively). These findings suggest that the BERT score provides more favourable results in terms of precision, recall, and F1-Score for our specific dataset.

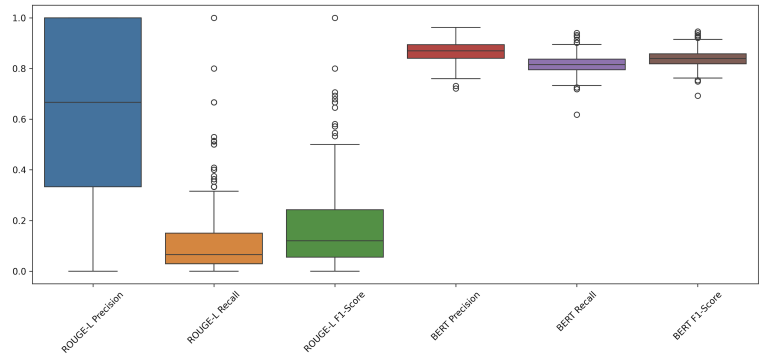


**Figure 5.5:** Bar plots for all the metrics provided by the scores

Metrics calculated by BERT score show lower standard deviation

Table 5.6 also shows that the standard deviation is higher across all the metrics for the ROUGE-L score compared to the BERT score.

Additionally, the BERT score provides metrics with mean values that are all higher than 0.80. The best metric is BERT Precision with 0.867, followed by BERT F1-Score with 0.840, and BERT Recall with 0.807.

Lastly, in the Figure 5.5, we observe outlier values for the metrics provided by the scores, except the one ROUGE-L Precision. Also, the outlier values for the other ROUGE-L metrics are at a greater distance compared to the metrics calculated with the BERT score.

Another observation is that the difference between the first and third quartiles of the metrics provided by BERT is narrower than those provided by ROUGE-L.

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.633 | 0.365 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.109 | 0.130 | 0.000 | 1.000 |
| ROUGE-L F1-Score | 0.164 | 0.161 | 0.000 | 1.000 |
| BERT Precision | 0.867 | 0.040 | 0.721 | 0.962 |
| BERT Recall | 0.817 | 0.037 | 0.617 | 0.940 |
| BERT F1-Score | 0.840 | 0.034 | 0.693 | 0.946 |

**Table 5.6:** Score Metrics for All Projects

# Chapter 6

# Discussion

In this section, we will discuss the results for both the performance of identifying substitution suggestions and how well the alternatives are explained to the user. However, it's important to note that the dataset we have is very limited, and this section only serves as a preliminary discussion of the initial findings.

## 6.1 Findings

### 6.1.1 Findings and Implications for the Substitution Suggestion Identification

During prompt engineering, we iterated the prompt with the aim of correctly identifying each comment. From Table 5.2, we that the accuracy is higher than 0.89 for all the projects. This aligns with the prompt we provided.

The high precision indicates that when the model identifies a comment as containing a substitution suggestion, it is usually correct. This is a positive attribute, as it minimizes false positives and reduces the likelihood of presenting irrelevant or incorrect information to users. The consistently high accuracy across all tutorials (>89%) further supports the model's overall reliability in classifying comments.

The high accuracy (>89%) supports the model's overall reliability in classifying comments.

By providing less restrictive comments, recall might be increased

Recall is the worst metric for all the tutorials. This means that the model did not perform well to capture all possible substitution suggestions. Phrases such as "Consider both explicit suggestions and nuanced or implied substitutions that can be reasonably inferred from the comments.", which is in the prompt we provide, make it more restrictive to identify a substitution suggestion. By providing less restrictive comments, recall might be increased.

To reduce false positives, addressing certain patterns may be helpful.

We have seen some patterns in comments which were classified as false positives. For example, questions asking whether a material or tool could be substituted with something else are classified as substitution suggestions even though there is no confirmation. Another pattern was that comments that suggest places to find a certain material or tool were sometimes classified as a substitution suggestion. Moreover, in cases where the same substitution is made by several users, the model may not list all of them, which results in a lower recall. By instructing the model to exclude these cases, we could reduce the number of false positives.

### 6.1.2 Findings and Implications for the Suggested Alternatives

The BERT score consistently outperformed the ROUGE-L score across all metrics (precision, recall, and F1 Score). This aligns with our inspection of provided alternatives that are concise yet meaningful. This discrepancy suggests that the BERT score may be more suitable for evaluating the quality of suggested alternatives in this context, as it can also capture similar words.

Model can provide a meaningful alternative from the comment

The high mean BERT scores (precision: 0.867, recall: 0.817, F1-score: 0.840) indicate that the suggestions of the model align closely with the original comment in terms of semantic meaning. This is a promising result, suggesting that the model can effectively capture and provide the essential meaning from the comments.

The lower performance of the ROUGE-L score, particularly in terms of recall and F1-score, may be attributed to

its focus on lexical overlap rather than semantic similarity. This limitation becomes apparent in cases where the model rephrases the suggestion using different words but maintains the same meaning.

The higher standard deviation observed in ROUGE-L scores compared to BERT scores suggests that ROUGE-L is more sensitive to variations in phrasing and lexical choice. This sensitivity may not be desirable when evaluating suggested alternatives, as it could penalize valid suggestions that use different vocabulary to express the same idea.

For further evaluations, scores like BERT, which can understand the contextual meaning, could be preferred.

## 6.2 Limitations of the Current Evaluation

The dataset is too limited to effectively evaluate this system. Extending the dataset would allow us to better understand how the model behaves and whether the values for the provided metrics follow a similar pattern.

Our current evaluation assumes the model provides the output in the desired format we defined in the prompt. When it does not, we are unable to parse different components of the output, such as the original item, suggested alternative, username of the contributor, and the comment ID of the referenced comment.

Our current evaluation assumes the model provides the output in the desired format.

Several factors might influence the results, such as the type of projects, the number of comments in a tutorial, the number of substitution suggestions provided, and the number of materials and tools used in the project. Our analysis currently does not provide these insights.

Effect of some other factors should be evaluated

# Chapter 7

# Summary and Future Work

While additional evaluation is required, our system shows promise in enhancing the user experience within DIY communities by offering a list of substitution suggestions for given tutorials.

Future research could further assess and refine the system, potentially leading to improvements. Such advancements could significantly aid the DIY community by facilitating more effective knowledge sharing through the use of large language models to identify substitution suggestions.

## 7.1 Summary and Contributions

In this thesis, we aimed to facilitate the process of finding substitution suggestions in the comments section of online DIY tutorials. A web-based application is designed to smooth this process by employing a large language model. OpenAI's GPT-4o model is used to identify these substitution suggestions, and we provided the tutorial instructions with images, along with the comments, with a prompt that describes what the model should do with the provided data. Several iterations were made for the prompt to in-

crease the reliability of the system. We evaluated this concept in terms of the identification of substitution suggestions and providing the alternatives in a meaningful matter. The system is reliable in terms of identifying substitution suggestions, as we get high accuracy (>89%) for all the tutorials, by considering the limited data we have. Also, the model can provide an alternative aligned with the referenced comment, as the mean values of BERT scores were high. Even though the results are promising, further evaluation is needed to understand the performance of the artifact.

With this thesis, we provided an artifact which could be used to identify substitution suggestions from the comments section of a tutorial by employing a large language model.

## 7.2 Limitations

The concept is only tested out for two online DIY platforms, namely Instructables and Youtube. However, there are many other of them which may pose difficult challenges.

Only one large language model is tested.

We utilized the model GPT-4o for our system. However, the models are rapidly evolving. Even GPT-4o is not fully available with all of its features, for example, they will make video input available soon, but we had to find out some work around to process video tutorials. Also, token limits were a limiting factor. Even only with 3 image data, we could not analyze certain.

We couldn't find an automated technique to check if the suggested substitutions were correct, so we had to do it manually. Doing this manually introduced human errors and made it difficult to analyze large amounts of data within a specific timeframe. Due to these limitations, we only collected a small number of tutorials and couldn't properly evaluate the system. We only managed to create a framework for evaluating the system and perform some of them to validate the concept.

There are certain limitations regarding the prompt. As we changed the selected large language model during the research, the model output changed along with it and brought out some challenges. At first, the model was able to provide an output in a format that we defined in a consistent manner. However, with new models emerging and with the additions to the data, the output format becomes inconsistent. As our data analysis also requires a certain output format to process everything smoothly, this caused problems. Also, the output format we defined was limited to listing substitution suggestions that were actually not substituting anything but improving the project. This has escalated the inconsistent formatting as sometimes the model was giving the reason for the substitution in the part where it should list the original item, and sometimes in the part for the alternative. Offering the reason rather than the replaced item was another inconsistency we faced during the analysis.

## 7.3 Future Work

As this system is built to provide a foundation for further research, in this section, we will comprehensively provide the ways in which this system can be further improved or used in further research.

System performance should be tested with a larger dataset

First of all, the system might be tested out with a larger dataset. We think that the complexity of the project, the number of comments in a tutorial, and the type of the project might be influencing factors for the performance. With a larger dataset, these factors could be tested.

Another suggestion would be to perform a qualitative analysis of the system. We find that the model acts similarly for certain comment structures. For example, if a person asks if they can substitute an item with something else, the model considers this a substitution.

# Appendix A

# ROUGE-L and BERT Score for Each Tutorial

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.310 | 0.366 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.095 | 0.104 | 0.000 | 0.250 |
| ROUGE-L F1-Score | 0.136 | 0.143 | 0.000 | 0.316 |
| BERT Precision | 0.854 | 0.042 | 0.784 | 0.904 |
| BERT Recall | 0.828 | 0.051 | 0.725 | 0.887 |
| BERT F1-Score | 0.840 | 0.043 | 0.754 | 0.892 |

**Table A.1:** Statistics for 2DuJT8-Gn8E

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.656 | 0.442 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.074 | 0.060 | 0.000 | 0.150 |
| ROUGE-L F1-Score | 0.131 | 0.103 | 0.000 | 0.250 |
| BERT Precision | 0.847 | 0.031 | 0.811 | 0.902 |
| BERT Recall | 0.786 | 0.046 | 0.719 | 0.836 |
| BERT F1-Score | 0.815 | 0.038 | 0.762 | 0.867 |

**Table A.2:** Statistics for 80GjcPECN8

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.500 | NaN | 0.500 | 0.500 |
| ROUGE-L Recall | 0.125 | NaN | 0.125 | 0.125 |
| ROUGE-L F1-Score | 0.200 | NaN | 0.200 | 0.200 |
| BERT Precision | 0.863 | NaN | 0.863 | 0.863 |
| BERT Recall | 0.825 | NaN | 0.825 | 0.825 |
| BERT F1-Score | 0.844 | NaN | 0.844 | 0.844 |

**Table A.3:** Statistics for Aluminium-Can-Roses

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.766 | 0.269 | 0.333 | 1.000 |
| ROUGE-L Recall | 0.132 | 0.094 | 0.037 | 0.312 |
| ROUGE-L F1-Score | 0.213 | 0.126 | 0.067 | 0.417 |
| BERT Precision | 0.873 | 0.032 | 0.829 | 0.929 |
| BERT Recall | 0.817 | 0.031 | 0.783 | 0.875 |
| BERT F1-Score | 0.844 | 0.029 | 0.806 | 0.889 |

**Table A.4:** Statistics for Amazing-3D-Projection-Pyramid-in-10-min-from-Clear

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.616 | 0.325 | 0.167 | 1.000 |
| ROUGE-L Recall | 0.107 | 0.094 | 0.025 | 0.316 |
| ROUGE-L F1-Score | 0.172 | 0.135 | 0.043 | 0.444 |
| BERT Precision | 0.880 | 0.026 | 0.841 | 0.945 |
| BERT Recall | 0.826 | 0.028 | 0.789 | 0.887 |
| BERT F1-Score | 0.852 | 0.024 | 0.829 | 0.915 |

**Table A.5:** Statistics for Arduino-CNC

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.485 | 0.502 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.060 | 0.078 | 0.000 | 0.235 |
| ROUGE-L F1-Score | 0.095 | 0.122 | 0.000 | 0.364 |
| BERT Precision | 0.830 | 0.030 | 0.800 | 0.885 |
| BERT Recall | 0.803 | 0.022 | 0.781 | 0.851 |
| BERT F1-Score | 0.816 | 0.023 | 0.799 | 0.861 |

**Table A.6:** Statistics for bIJY65guY9A

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.833 | 0.289 | 0.500 | 1.000 |
| ROUGE-L Recall | 0.130 | 0.112 | 0.028 | 0.250 |
| ROUGE-L F1-Score | 0.212 | 0.175 | 0.054 | 0.400 |
| BERT Precision | 0.866 | 0.027 | 0.835 | 0.887 |
| BERT Recall | 0.814 | 0.024 | 0.790 | 0.838 |
| BERT F1-Score | 0.839 | 0.010 | 0.830 | 0.850 |

**Table A.7:** Statistics for Cardboard-Chair

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.778 | 0.385 | 0.333 | 1.000 |
| ROUGE-L Recall | 0.033 | 0.000 | 0.033 | 0.033 |
| ROUGE-L F1-Score | 0.063 | 0.002 | 0.061 | 0.065 |
| BERT Precision | 0.850 | 0.027 | 0.820 | 0.870 |
| BERT Recall | 0.779 | 0.014 | 0.766 | 0.794 |
| BERT F1-Score | 0.813 | 0.005 | 0.806 | 0.817 |

**Table A.8:** Statistics for Chess-Salt-and-Pepper-Mills

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.778 | 0.192 | 0.667 | 1.000 |
| ROUGE-L Recall | 0.247 | 0.084 | 0.167 | 0.333 |
| ROUGE-L F1-Score | 0.366 | 0.091 | 0.267 | 0.444 |
| BERT Precision | 0.869 | 0.058 | 0.810 | 0.926 |
| BERT Recall | 0.838 | 0.014 | 0.825 | 0.852 |
| BERT F1-Score | 0.853 | 0.035 | 0.817 | 0.887 |

**Table A.9:** Statistics for Chess-table-Instruc-table

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.778 | 0.385 | 0.333 | 1.000 |
| ROUGE-L Recall | 0.190 | 0.103 | 0.077 | 0.278 |
| ROUGE-L F1-Score | 0.304 | 0.161 | 0.125 | 0.435 |
| BERT Precision | 0.868 | 0.056 | 0.819 | 0.930 |
| BERT Recall | 0.818 | 0.033 | 0.783 | 0.848 |
| BERT F1-Score | 0.842 | 0.043 | 0.801 | 0.887 |

**Table A.10:** Statistics for d1VcMybY4NQ

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.666 | 0.364 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.105 | 0.156 | 0.000 | 0.667 |
| ROUGE-L F1-Score | 0.150 | 0.185 | 0.000 | 0.800 |
| BERT Precision | 0.873 | 0.035 | 0.817 | 0.931 |
| BERT Recall | 0.817 | 0.038 | 0.760 | 0.894 |
| BERT F1-Score | 0.843 | 0.033 | 0.793 | 0.912 |

**Table A.11:** Statistics for DIY-cat-tent

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.226 | 0.245 | 0.000 | 0.667 |
| ROUGE-L Recall | 0.074 | 0.093 | 0.000 | 0.316 |
| ROUGE-L F1-Score | 0.106 | 0.124 | 0.000 | 0.414 |
| BERT Precision | 0.849 | 0.033 | 0.788 | 0.913 |
| BERT Recall | 0.829 | 0.016 | 0.810 | 0.852 |
| BERT F1-Score | 0.838 | 0.021 | 0.800 | 0.873 |

**Table A.12:** Statistics for DU2R7S0oxx4

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.589 | 0.220 | 0.200 | 1.000 |
| ROUGE-L Recall | 0.068 | 0.068 | 0.003 | 0.273 |
| ROUGE-L F1-Score | 0.110 | 0.094 | 0.007 | 0.375 |
| BERT Precision | 0.870 | 0.026 | 0.819 | 0.903 |
| BERT Recall | 0.809 | 0.038 | 0.745 | 0.866 |
| BERT F1-Score | 0.838 | 0.026 | 0.780 | 0.884 |

**Table A.13:** Statistics for Enchanted-Forest-Mushroom-Lights

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.671 | 0.435 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.036 | 0.045 | 0.000 | 0.129 |
| ROUGE-L F1-Score | 0.066 | 0.080 | 0.000 | 0.229 |
| BERT Precision | 0.850 | 0.033 | 0.803 | 0.896 |
| BERT Recall | 0.797 | 0.029 | 0.753 | 0.835 |
| BERT F1-Score | 0.822 | 0.028 | 0.793 | 0.865 |

**Table A.14:** Statistics for Fiber-Optic-LED-Lamp

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.195 | 0.166 | 0.000 | 0.400 |
| ROUGE-L Recall | 0.126 | 0.120 | 0.000 | 0.250 |
| ROUGE-L F1-Score | 0.149 | 0.135 | 0.000 | 0.273 |
| BERT Precision | 0.839 | 0.036 | 0.788 | 0.887 |
| BERT Recall | 0.775 | 0.095 | 0.617 | 0.869 |
| BERT F1-Score | 0.805 | 0.068 | 0.693 | 0.873 |

**Table A.15:** Statistics for fjGcvfa4E7c

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.798 | 0.371 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.205 | 0.267 | 0.000 | 1.000 |
| ROUGE-L F1-Score | 0.283 | 0.285 | 0.000 | 1.000 |
| BERT Precision | 0.884 | 0.037 | 0.838 | 0.961 |
| BERT Recall | 0.835 | 0.040 | 0.776 | 0.932 |
| BERT F1-Score | 0.859 | 0.035 | 0.824 | 0.946 |

**Table A.16:** Statistics for Flying-Captain-America-Shield

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.641 | 0.237 | 0.417 | 1.000 |
| ROUGE-L Recall | 0.242 | 0.201 | 0.029 | 0.514 |
| ROUGE-L F1-Score | 0.330 | 0.250 | 0.054 | 0.679 |
| BERT Precision | 0.909 | 0.039 | 0.857 | 0.962 |
| BERT Recall | 0.852 | 0.045 | 0.788 | 0.901 |
| BERT F1-Score | 0.879 | 0.041 | 0.821 | 0.931 |

**Table A.17:** Statistics for Holiday-Light-Tunnel

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.692 | 0.474 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.231 | 0.208 | 0.000 | 0.500 |
| ROUGE-L F1-Score | 0.338 | 0.276 | 0.000 | 0.667 |
| BERT Precision | 0.914 | 0.033 | 0.867 | 0.938 |
| BERT Recall | 0.860 | 0.022 | 0.837 | 0.891 |
| BERT F1-Score | 0.886 | 0.019 | 0.861 | 0.905 |

**Table A.18:** Statistics for How-to-Build-an-Outdoor-Hammock-Stand-25

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.899 | 0.152 | 0.667 | 1.000 |
| ROUGE-L Recall | 0.181 | 0.155 | 0.014 | 0.409 |
| ROUGE-L F1-Score | 0.272 | 0.209 | 0.028 | 0.581 |
| BERT Precision | 0.881 | 0.039 | 0.834 | 0.931 |
| BERT Recall | 0.834 | 0.044 | 0.773 | 0.902 |
| BERT F1-Score | 0.857 | 0.040 | 0.803 | 0.904 |

**Table A.19:** Statistics for IKEA-HACK-articulating-tablet-mount

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.820 | 0.296 | 0.100 | 1.000 |
| ROUGE-L Recall | 0.094 | 0.054 | 0.024 | 0.167 |
| ROUGE-L F1-Score | 0.165 | 0.091 | 0.039 | 0.286 |
| BERT Precision | 0.884 | 0.033 | 0.816 | 0.936 |
| BERT Recall | 0.807 | 0.016 | 0.783 | 0.830 |
| BERT F1-Score | 0.844 | 0.020 | 0.805 | 0.879 |

**Table A.20:** Statistics for Jewelry-board

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.337 | 0.325 | 0.000 | 0.857 |
| ROUGE-L Recall | 0.046 | 0.062 | 0.000 | 0.214 |
| ROUGE-L F1-Score | 0.078 | 0.097 | 0.000 | 0.333 |
| BERT Precision | 0.830 | 0.035 | 0.778 | 0.887 |
| BERT Recall | 0.820 | 0.022 | 0.791 | 0.857 |
| BERT F1-Score | 0.825 | 0.023 | 0.798 | 0.870 |

**Table A.21:** Statistics for JrG$_I$$nKQB6g$

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.000 | NaN | 0.000 | 0.000 |
| ROUGE-L Recall | 0.000 | NaN | 0.000 | 0.000 |
| ROUGE-L F1-Score | 0.000 | NaN | 0.000 | 0.000 |
| BERT Precision | 0.824 | NaN | 0.824 | 0.824 |
| BERT Recall | 0.798 | NaN | 0.798 | 0.798 |
| BERT F1-Score | 0.810 | NaN | 0.810 | 0.810 |

**Table A.22:** Statistics for Laser-Cut-Ambient-Light-With-Kerf-Bends

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.547 | 0.300 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.076 | 0.069 | 0.000 | 0.250 |
| ROUGE-L F1-Score | 0.128 | 0.107 | 0.000 | 0.400 |
| BERT Precision | 0.869 | 0.032 | 0.813 | 0.930 |
| BERT Recall | 0.811 | 0.028 | 0.763 | 0.880 |
| BERT F1-Score | 0.839 | 0.024 | 0.806 | 0.886 |

**Table A.23:** Statistics for Lego-USB-Stick

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.035 | NaN | 0.035 | 0.035 |
| ROUGE-L Recall | 0.800 | NaN | 0.800 | 0.800 |
| ROUGE-L F1-Score | 0.068 | NaN | 0.068 | 0.068 |
| BERT Precision | 0.731 | NaN | 0.731 | 0.731 |
| BERT Recall | 0.891 | NaN | 0.891 | 0.891 |
| BERT F1-Score | 0.803 | NaN | 0.803 | 0.803 |

**Table A.24:** Statistics for Make-a-LEGO-Abrams-Tank

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.362 | 0.398 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.036 | 0.052 | 0.000 | 0.167 |
| ROUGE-L F1-Score | 0.063 | 0.089 | 0.000 | 0.286 |
| BERT Precision | 0.851 | 0.046 | 0.777 | 0.916 |
| BERT Recall | 0.803 | 0.020 | 0.775 | 0.846 |
| BERT F1-Score | 0.826 | 0.029 | 0.789 | 0.880 |

**Table A.25:** Statistics for Mario-Bros-Clock

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.629 | 0.369 | 0.143 | 1.000 |
| ROUGE-L Recall | 0.142 | 0.148 | 0.038 | 0.400 |
| ROUGE-L F1-Score | 0.222 | 0.205 | 0.061 | 0.571 |
| BERT Precision | 0.835 | 0.033 | 0.790 | 0.877 |
| BERT Recall | 0.811 | 0.023 | 0.788 | 0.848 |
| BERT F1-Score | 0.822 | 0.022 | 0.800 | 0.853 |

**Table A.26:** Statistics for mFQS9JySnZ8

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 1.000 | 0.000 | 1.000 | 1.000 |
| ROUGE-L Recall | 0.156 | 0.076 | 0.087 | 0.238 |
| ROUGE-L F1-Score | 0.265 | 0.113 | 0.160 | 0.385 |
| BERT Precision | 0.910 | 0.014 | 0.899 | 0.925 |
| BERT Recall | 0.833 | 0.013 | 0.822 | 0.847 |
| BERT F1-Score | 0.870 | 0.013 | 0.859 | 0.884 |

**Table A.27:** Statistics for Pallet-Wine-Rack

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.714 | 0.488 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.016 | 0.014 | 0.000 | 0.039 |
| ROUGE-L F1-Score | 0.030 | 0.027 | 0.000 | 0.075 |
| BERT Precision | 0.822 | 0.051 | 0.721 | 0.872 |
| BERT Recall | 0.786 | 0.026 | 0.763 | 0.842 |
| BERT F1-Score | 0.803 | 0.027 | 0.749 | 0.826 |

**Table A.28:** Statistics for Phenomenal-Augmented-Reality-Allows-Us-to-Watch-Ho

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.867 | 0.208 | 0.400 | 1.000 |
| ROUGE-L Recall | 0.125 | 0.085 | 0.030 | 0.297 |
| ROUGE-L F1-Score | 0.204 | 0.119 | 0.059 | 0.440 |
| BERT Precision | 0.900 | 0.033 | 0.826 | 0.931 |
| BERT Recall | 0.825 | 0.028 | 0.778 | 0.862 |
| BERT F1-Score | 0.861 | 0.023 | 0.811 | 0.885 |

**Table A.29:** Statistics for Pinball-Coffee-Table

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.524 | 0.360 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.160 | 0.117 | 0.000 | 0.300 |
| ROUGE-L F1-Score | 0.233 | 0.174 | 0.000 | 0.462 |
| BERT Precision | 0.865 | 0.039 | 0.811 | 0.900 |
| BERT Recall | 0.820 | 0.040 | 0.760 | 0.872 |
| BERT F1-Score | 0.842 | 0.034 | 0.785 | 0.886 |

**Table A.30:** Statistics for QCGcmzkA12k

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.886 | 0.250 | 0.286 | 1.000 |
| ROUGE-L Recall | 0.174 | 0.146 | 0.047 | 0.529 |
| ROUGE-L F1-Score | 0.266 | 0.186 | 0.083 | 0.692 |
| BERT Precision | 0.856 | 0.036 | 0.798 | 0.920 |
| BERT Recall | 0.822 | 0.036 | 0.789 | 0.886 |
| BERT F1-Score | 0.838 | 0.035 | 0.797 | 0.899 |

**Table A.31:** Statistics for QdnTjmxpBuk

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.700 | 0.265 | 0.500 | 1.000 |
| ROUGE-L Recall | 0.177 | 0.138 | 0.071 | 0.333 |
| ROUGE-L F1-Score | 0.259 | 0.155 | 0.125 | 0.429 |
| BERT Precision | 0.903 | 0.026 | 0.875 | 0.925 |
| BERT Recall | 0.823 | 0.028 | 0.804 | 0.855 |
| BERT F1-Score | 0.861 | 0.026 | 0.838 | 0.889 |

**Table A.32:** Statistics for sDlyksf3ivQ

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.533 | 0.510 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.053 | 0.068 | 0.000 | 0.200 |
| ROUGE-L F1-Score | 0.093 | 0.114 | 0.000 | 0.333 |
| BERT Precision | 0.854 | 0.040 | 0.789 | 0.936 |
| BERT Recall | 0.808 | 0.020 | 0.784 | 0.845 |
| BERT F1-Score | 0.830 | 0.022 | 0.804 | 0.879 |

**Table A.33:** Statistics for Sew-a-Where-the-Wild-Things-Are-hat-pattern

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.929 | 0.189 | 0.500 | 1.000 |
| ROUGE-L Recall | 0.072 | 0.045 | 0.020 | 0.150 |
| ROUGE-L F1-Score | 0.131 | 0.077 | 0.038 | 0.261 |
| BERT Precision | 0.881 | 0.037 | 0.841 | 0.951 |
| BERT Recall | 0.807 | 0.023 | 0.782 | 0.837 |
| BERT F1-Score | 0.842 | 0.023 | 0.819 | 0.881 |

**Table A.34:** Statistics for Shirt-Folding-Board-from-Cardboard-and-Duct-Tape

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.671 | 0.338 | 0.143 | 1.000 |
| ROUGE-L Recall | 0.117 | 0.097 | 0.048 | 0.333 |
| ROUGE-L F1-Score | 0.185 | 0.138 | 0.071 | 0.500 |
| BERT Precision | 0.884 | 0.030 | 0.854 | 0.937 |
| BERT Recall | 0.836 | 0.048 | 0.761 | 0.926 |
| BERT F1-Score | 0.859 | 0.039 | 0.805 | 0.932 |

**Table A.35:** Statistics for Simple-Elegant-Guitar-Stand

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.762 | 0.321 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.049 | 0.047 | 0.000 | 0.158 |
| ROUGE-L F1-Score | 0.089 | 0.082 | 0.000 | 0.273 |
| BERT Precision | 0.858 | 0.037 | 0.759 | 0.891 |
| BERT Recall | 0.798 | 0.025 | 0.757 | 0.823 |
| BERT F1-Score | 0.826 | 0.024 | 0.777 | 0.856 |

**Table A.36:** Statistics for Sous-vide-cooker-for-less-than-40

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.500 | NaN | 0.500 | 0.500 |
| ROUGE-L Recall | 0.083 | NaN | 0.083 | 0.083 |
| ROUGE-L F1-Score | 0.143 | NaN | 0.143 | 0.143 |
| BERT Precision | 0.896 | NaN | 0.896 | 0.896 |
| BERT Recall | 0.821 | NaN | 0.821 | 0.821 |
| BERT F1-Score | 0.857 | NaN | 0.857 | 0.857 |

**Table A.37:** Statistics for $\mathrm{sSgo}_h V - myg$

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.000 | NaN | 0.000 | 0.000 |
| ROUGE-L Recall | 0.000 | NaN | 0.000 | 0.000 |
| ROUGE-L F1-Score | 0.000 | NaN | 0.000 | 0.000 |
| BERT Precision | 0.822 | NaN | 0.822 | 0.822 |
| BERT Recall | 0.801 | NaN | 0.801 | 0.801 |
| BERT F1-Score | 0.811 | NaN | 0.811 | 0.811 |

**Table A.38:** Statistics for TG1oCqnn7E4

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.639 | 0.427 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.039 | 0.024 | 0.000 | 0.074 |
| ROUGE-L F1-Score | 0.073 | 0.045 | 0.000 | 0.138 |
| BERT Precision | 0.851 | 0.026 | 0.816 | 0.881 |
| BERT Recall | 0.806 | 0.038 | 0.768 | 0.878 |
| BERT F1-Score | 0.828 | 0.021 | 0.798 | 0.852 |

**Table A.39:** Statistics for Tinker-Bell-Pixie-Dust-Pumpkin-Carving

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.659 | 0.302 | 0.286 | 1.000 |
| ROUGE-L Recall | 0.097 | 0.075 | 0.023 | 0.211 |
| ROUGE-L F1-Score | 0.157 | 0.111 | 0.044 | 0.348 |
| BERT Precision | 0.883 | 0.013 | 0.855 | 0.895 |
| BERT Recall | 0.814 | 0.027 | 0.775 | 0.872 |
| BERT F1-Score | 0.847 | 0.018 | 0.827 | 0.883 |

**Table A.40:** Statistics for turn-signal-biking-jacket

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.727 | 0.396 | 0.133 | 1.000 |
| ROUGE-L Recall | 0.054 | 0.044 | 0.004 | 0.125 |
| ROUGE-L F1-Score | 0.095 | 0.079 | 0.009 | 0.222 |
| BERT Precision | 0.816 | 0.020 | 0.795 | 0.838 |
| BERT Recall | 0.791 | 0.021 | 0.760 | 0.817 |
| BERT F1-Score | 0.803 | 0.019 | 0.778 | 0.828 |

**Table A.41:** Statistics for Turn-Yourself-Into-a-Cartoon

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.860 | 0.165 | 0.600 | 1.000 |
| ROUGE-L Recall | 0.123 | 0.123 | 0.043 | 0.375 |
| ROUGE-L F1-Score | 0.198 | 0.170 | 0.083 | 0.545 |
| BERT Precision | 0.903 | 0.016 | 0.881 | 0.928 |
| BERT Recall | 0.822 | 0.024 | 0.785 | 0.857 |
| BERT F1-Score | 0.861 | 0.017 | 0.836 | 0.882 |

**Table A.42:** Statistics for USB-Volume-Knob

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.000 | NaN | 0.000 | 0.000 |
| ROUGE-L Recall | 0.000 | NaN | 0.000 | 0.000 |
| ROUGE-L F1-Score | 0.000 | NaN | 0.000 | 0.000 |
| BERT Precision | 0.826 | NaN | 0.826 | 0.826 |
| BERT Recall | 0.847 | NaN | 0.847 | 0.847 |
| BERT F1-Score | 0.836 | NaN | 0.836 | 0.836 |

**Table A.43:** Statistics for Valentines-Day-Papercraft-Robot-Cupid

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.690 | 0.267 | 0.250 | 1.000 |
| ROUGE-L Recall | 0.028 | 0.017 | 0.014 | 0.062 |
| ROUGE-L F1-Score | 0.053 | 0.031 | 0.027 | 0.115 |
| BERT Precision | 0.865 | 0.042 | 0.794 | 0.909 |
| BERT Recall | 0.796 | 0.016 | 0.777 | 0.819 |
| BERT F1-Score | 0.828 | 0.020 | 0.806 | 0.856 |

**Table A.44:** Statistics for Weight-Bench-5-positionFlatIncline-doubles-as-

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.566 | 0.437 | 0.000 | 1.000 |
| ROUGE-L Recall | 0.199 | 0.190 | 0.000 | 0.512 |
| ROUGE-L F1-Score | 0.283 | 0.245 | 0.000 | 0.667 |
| BERT Precision | 0.888 | 0.063 | 0.785 | 0.947 |
| BERT Recall | 0.849 | 0.042 | 0.789 | 0.903 |
| BERT F1-Score | 0.867 | 0.050 | 0.787 | 0.925 |

**Table A.45:** Statistics for Wire-Wrapped-Tree-of-Life-Tutorial

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.577 | 0.243 | 0.267 | 0.789 |
| ROUGE-L Recall | 0.300 | 0.253 | 0.088 | 0.667 |
| ROUGE-L F1-Score | 0.354 | 0.243 | 0.159 | 0.706 |
| BERT Precision | 0.921 | 0.020 | 0.902 | 0.939 |
| BERT Recall | 0.876 | 0.062 | 0.804 | 0.940 |
| BERT F1-Score | 0.898 | 0.042 | 0.851 | 0.939 |

**Table A.46:** Statistics for Wooden-Chapati-Maker-at-Home

| | Mean | Standard Deviation | Min | Max |
|---|---|---|---|---|
| ROUGE-L Precision | 0.259 | 0.181 | 0.000 | 0.400 |
| ROUGE-L Recall | 0.144 | 0.111 | 0.000 | 0.250 |
| ROUGE-L F1-Score | 0.184 | 0.136 | 0.000 | 0.296 |
| BERT Precision | 0.869 | 0.021 | 0.844 | 0.896 |
| BERT Recall | 0.842 | 0.024 | 0.821 | 0.876 |
| BERT F1-Score | 0.856 | 0.022 | 0.832 | 0.886 |

**Table A.47:** Statistics for yD–59Kq7as

# Bibliography

[1] Leah Buechley, Daniela K. Rosner, Eric Paulos, and Amanda Williams. DIY for CHI: methods, communities, and values of reuse and customization. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, page 4823–4826, New York, NY, USA, 2009. Association for Computing Machinery. doi.org/10.1145/1520340.1520750.

[2] Matthew A. Dalton, Audrey Desjardins, and Ron Wakkary. From DIY tutorials to DIY recipes. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, page 1405–1410, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2559206.2581238.

[3] Alan Dix. Designing for appropriation. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but Not as We Know It - Volume 2*, BCS-HCI '07, page 27–30, Swindon, GBR, 2007. BCS Learning & Development Ltd.

[4] Stefan Feuerriegel, Jochen Hartmann, Christian Janiesch, and Patrick Zschech. Generative AI. *Business amp; Information Systems Engineering*, 66(1):111–126, September 2023. doi.org/10.1007/s12599-023-00834-7.

[5] Verena Fuchsberger, Martin Murer, Manfred Tscheligi, Silvia Lindtner, Andreas Reiter, Shaowen Bardzell, Jeffrey Bardzell, and Pernille Bjørn. The Future of Making: Where Industrial and Personal Fabrication Meet. *Aarhus Series on Human Centered Computing*, 1:4, 10 2015. doi.org/10.7146/aahcc.v1i1.21394.

[6] Stacey Kuznetsov and Eric Paulos. Rise of the expert amateur: DIY projects, communities, and cultures. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, page 295–304, New York, NY, USA, 2010. Association for Computing Machinery. doi.org/10.1145/1868914.1868950.

[7] Nahyun Kwon, Tong Steven Sun, Yuyang Gao, Liang Zhao, Xu Wang, Jeeeun Kim, and Sungsoo Ray Hong. 3DPFIX: Improving Remote Novices' 3D Print-

ing Troubleshooting through Human-AI Collaboration Design. *Proc. ACM Hum.-Comput. Interact.*, 8(CSCW1), apr 2024. doi.org/10.1145/3637288.

[8] Ben Lafreniere, Andrea Bunt, Matthew Lount, and Michael Terry. Understanding the Roles and Uses of Web Tutorials. *Proceedings of the International AAAI Conference on Web and Social Media*, 7(1):303–310, Aug. 2021. doi.org/10.1609/icwsm.v7i1.14413.

[9] Matthew Lakier, Michelle Annett, and Daniel Wigdor. Automatics: Dynamically Generating Fabrication Tasks to Adapt to Varying Contexts. *ACM Trans. Comput.-Hum. Interact.*, 25(4), aug 2018. doi.org/10.1145/3185065.

[10] Sophie Landwehr Sydow. *Makers, Materials and Machines : Understanding Experience and Situated Embodied Practice in the Makerspace*. PhD thesis, Stockholm University, Department of Computer and Systems Sciences, 2022.

[11] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004.

[12] Catarina Mota. The rise of personal fabrication. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, CC '11, page 279–288, New York, NY, USA, 2011. Association for Computing Machinery. doi.org/10.1145/2069618.2069665.

[13] Michael Muller, Lydia B Chilton, Anna Kantosalo, Charles Patrick Martin, and Greg Walsh. GenAICHI: Generative AI and HCI. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery. doi.org/10.1145/3491101.3503719.

[14] Lora Oehlberg, Wesley Willett, and Wendy E. Mackay. Patterns of Physical Design Remixing in Online Maker Communities. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 639–648, New York, NY, USA, 2015. Association for Computing Machinery. doi.org/10.1145/2702123.2702175.

[15] Daniel Saakes. Big lampan lamps: designing for DIY. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, CC '09, page 403–404, New York, NY, USA, 2009. Association for Computing Machinery. doi.org/10.1145/1640233.1640322.

[16] Jingyu Shi, Rahul Jain, Hyungjun Doh, Ryo Suzuki, and Karthik Ramani. An HCI-Centric Survey and Taxonomy of Human-Generative-AI Interactions, 2024. URL https://arxiv.org/abs/2310.07127.

[17] Tomás Soucek and Jakub Lokoc. TransNet V2: An effective deep network architecture for fast shot transition detection. *CoRR*, abs/2008.04838, 2020.

[18] Nick Taylor, Ursula Hurley, and Philip Connolly. Making Community: The Wider Role of Makerspaces in Public Life. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 1415–1425, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/2858036.2858073.

[19] Timm Teubner, Christoph Flath, Christof Weinhardt, Wil Aalst, and Oliver Hinz. Welcome to the Era of ChatGPT et al.: The Prospects of Large Language Models. *Business Information Systems Engineering*, 65, 03 2023. doi.org/10.1007/s12599-023-00795-x.

[20] Cristen Torrey, Elizabeth F. Churchill, and David W. McDonald. Learning how: the search for craft knowledge on the internet. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 1371–1380, New York, NY, USA, 2009. Association for Computing Machinery. doi.org/10.1145/1518701.1518908.

[21] Tiffany Tseng. Making make-throughs : documentation as stories of design process. 2016.

[22] Tiffany Tseng and Mitchel Resnick. Product versus process: representing and appropriating DIY projects online. In *Proceedings of the 2014 Conference on Designing Interactive Systems*, DIS '14, page 425–428, New York, NY, USA, 2014. Association for Computing Machinery. doi.org/10.1145/2598510.2598540.

[23] Ron Wakkary, Markus Lorenz Schilling, Matthew A. Dalton, Sabrina Hauser, Audrey Desjardins, Xiao Zhang, and Henry W.J. Lin. Tutorial Authorship and Hybrid Designers: The Joy (and Frustration) of DIY Tutorials. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 609–618, New York, NY, USA, 2015. Association for Computing Machinery. doi.org/10.1145/2702123.2702550.

[24] Sheng Wang and Raymond A. Noe. Knowledge sharing: A review and directions for future research. *Human Resource Management Review*, 20(2):115–131, 2010. doi.org/10.1016/j.hrmr.2009.10.001.

[25] Marco Wolf and Shaun Mcquitty. Understanding the do-it-yourself consumer: DIY motivations and outcomes. *AMS Review*, 1, 12 2011. doi.org/10.1007/s13162-011-0021-2.

[26] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

# Index

abbrv . . . . . *see* abbreviation

discussion . . . . . 63–65

evaluation . . . . . 39–52

future work . . . . . 69

Generative AI . . . . . 13, 14

Prompt engineering . . . . . 23, 27

results . . . . . 53

Typeset July 25, 2024