A contextual help browser extension to assist digital illiterate internet users

Student name: Christos Koutsiaris
Student number: 20094922
Supervisor Dr. Brenda Mullally
Date: 20/08/2023

South East Technological University
School of Science and Computing
MSc in Computing (Enterprise Software Systems)

Abstract.

This dissertation examines the implementation of a browser extension that provides contextual help to users when they hover over technological acronyms and abbreviations on websites. The extension uses OpenAI services and a dictionary of technical terms to provide definitions that appear in a tooltip when users hover over a detected acronym or abbreviation. The study investigates the potential use of Artificial Intelligence technologies, particularly Natural Language Processing and Large Language Models, to categorise web pages based on their content as technological or not and for technological acronym identification. The research concludes by evaluating the effects of contextual help assistance on technical text understanding and comprehension and changes in reading rates in users with low levels of digital illiteracy or minimally technology-exposed users.

Keywords:

Contextual Help · Digital Illiteracy · Browser Extension · Artificial Intelligence.

Plagiarism Declaration

I confirm that this master's thesis is entirely my own work, and that all sources and materials used in this thesis are properly referenced and acknowledged. I acknowledge that any form of plagiarism or academic dishonesty is a serious offence and can result in severe consequences, including the revocation of my degree. By signing this declaration, I confirm that I understand the seriousness of these issues and have taken all necessary steps to avoid any act of plagiarism.

Plagiarism Declaration 2
1. Introduction 5
2. Literature Review 6
2.1 Digital Illiteracy 6
2.2 Abbreviations - Acronyms 9
2.3 Artificial Intelligence - Natural Language Processing 10
2.4 Browsers and extensions 11
2.4 Summary 12
3. Working Theory 13
3.1 Research Questions 13
3.2 Research Hypothesis 13
3.3 Contribution to Research Knowledge Anticipated 14
4. Research Design and Methodology 14
4.1 Research Methodology 14
4.2 Research Population and Sampling 16
5. Architecture and Development 18
5.1 Browser extensions 18
5.2 Browser Extension building blocks 18
5.3 Plasmo Framework, a framework for building browser extensions 19
5.4 Development of functionality modules 20
5.4.1 Webpage load detection 20
5.4.2 Content sanitation and parsing 22
5.4.3 Google Cloud Natural Language API(Taxonomy) - OpenAI Detection 23
5.4.4 Detection of acronyms, abbreviations or technical terms. 25
5.4.5 OpenAI - Language Model 26
5.4.6 OpenAI - ChatGPT prompts 27
5.4.7 DOM Manipulation - Appending tooltips 27
5.4.8 Acronyms and abbreviations searchable in a tab page. 28
6. Screenshots of the contextual help browser extension 31
7. Presentation of findings 33
7.1 Introduction to findings 33
7.2 Methodological Checks 33
7.3 Quantitative Data Analysis 34
7.3.1 Demographic Information 34
7.3.2 Technology Usage and Experience 34
7.3.3 Self-assessment of Digital Literacy 35
7.3.4 Frequency of encountering technical terms and acronyms 36
7.3.5 Impact of technical terms and acronyms on comprehension 37
7.3.6 Potential benefits of a contextual help tool 39
7.3.7 Impact of the contextual help browser extension on understanding technical terms and acronyms 40

7.3.8 User experience and clarity of definitions 41
7.3.9 Perceived effects on website performance 42
7.3.10 Inclination to explore additional technical content 43
7.3.11 Time-saving aspect of the contextual help browser extension 44
7.3.12 Additional resources and recommendations 45
7.3.13 Time Taken for Manual Information Search 47
7.3.14 Performance Metrics of the Contextual Help Extension 48
7.4 Qualitative Data Analysis 51
7.4.1 Positive impact on specific target groups 53
7.5 Summary of findings 54
8. Discussion 55
8.1 Answers to research questions 55
8.2 Relationship to findings 56
8.3 Implications 57
8.4 Suggested improvements - Future versions 57
8.5 Questions raised and suggestions for further research 58
8.6 Considerations on the sample size, the statistical significance, and the potential bias 59
8.7 Summary 59
9. Conclusion 60
Video demonstration 60
10. Reference List 61
11. Appendix A - Demographic Information 66
12. Appendix B - Technology Usage and Experience 68

1. Introduction

The exponential growth of technology in the last fifty years has been closely related to the significant achievements in the development of transistors and microchips (Morton and Pietenpol, 1958). Gordon Moore in 1967 (Schaller, 1997) predicted with remarkable accuracy that the number of transistors on a microchip would double every 24 months. Moore's law is still valid today, and the prediction is that the law will not be dead for at least a few more years (Roser and Ritchie, 2013).

By fitting more transistors into smaller components, engineers were able to bring technology closer to every person's life. As a result, we live in a world where devices with unique capabilities, specifications and characteristics surround us. To describe and communicate these characteristics to the consumers, production companies had to invent new words, phrases, definitions, acronyms and abbreviations (Keats, 2010). Naming a product or a service is not an easy task, and very often simplicity and memorability are more important than confusion (Gablerjan, 2015).

Technology and networking allowed an ongoing digital transformation of traditional services, processes and products. This caused an increase in the usage of technical terms, which accelerated significantly with the arrival of the internet and connected systems. Additionally, it created an environment where users are expected to know and understand an extensive glossary of technological terms. Many of these terms appear so often in our daily lives that it would be difficult to avoid them.

The continuous usage of technical terms, or sometimes technical jargon, in websites such as portals, online newspapers and e-commerce websites can block content consumption or decrease text comprehension. That proves to be challenging for people of older generations or for people with minimal knowledge of information technology (Steelman and Wallace, 2017). Individuals who do not utilise technology in their everyday life by consuming, creating and sharing digital content are less likely to have knowledge about new technologies and possibly adopt them. In this group of people digital illiteracy can clearly be observed(Steelman et al., 2017). Furthermore, inequalities between people who understand articles containing technological terms and those who do not might cause personal, social or even financial harm to the latter (Czaja and Urbaniec, 2019).

This dissertation researches and implements a solution to help digitally illiterate or semi-illiterate internet users by providing them with contextual information about technical terms on the spot. Modern browsers allow developers to build extensions that enable parsing and transforming the content before the rendering phase. This dissertation takes advantage of this browser feature and injects helpful content that appears on request.

2. Literature Review

Four major literature areas had to be explored to support the formulation of the research questions. These are Digital Illiteracy, Abbreviations - Acronyms and their use, Artificial Intelligence and, more specifically, the branch Natural Language Processing, and literature about browser extensions, mainly Chrome Browser extensions.

2.1 Digital Illiteracy

Digital illiteracy is an enormous topic that is actively in research primarily because of its impact on societies. Digital illiteracy can be defined as the inability to use technology for reading, writing, communicating and, in general, obtaining information in the digital era. It is essential to highlight that researchers have used related terminology for digital illiteracy in various studies, such as information illiteracy, technological illiteracy, computer illiteracy and media illiteracy.

Leahy and Dolan (Leahy and Dolan, 2009) in their paper provide a definition of digital literacy which focuses mainly on information and communication:

“Digital Literacy involves the confident and critical use of Information Society Technology for work, leisure and communication. It is underpinned by Basic skills in ICT: the use of computers to retrieve, assess, store, produce, present and exchange information, and to communicate and participate in collaborative networks via the internet.”

In a world that uses technology for communication with friends and family, businesses, and government services, not taking the measures to help a significant percentage of the population might lead to exclusion (World Economic Forum, n.d.). The European Union already recognised that digital literacy is a need for its members and included it in the definition of E-inclusion (Europa, n.d.).

Additional studies on digital illiteracy reveal informative patterns and characteristics across populations. For example, countries that have progressed faster than developing countries (UNICEF, n.d.) show lower percentages of digital illiteracy. However, numbers are still evident in developed countries (World Economic Forum, n.d.). Users with low digital literacy skills are not a tiny percentage of the total population even in developed countries such as the Netherlands where 15% of the population does not have advanced digital skills (Digital Government, 2021). Experts in a research published in Pew Research Center share the common view that life in years ahead will be even more technologically driven (Anderson, Rainie and Vogels, 2021). Technology will penetrate even more our lives, and this might cause more inequalities. However new tools will emerge that would help digital illiterate people to adopt more technology and help them live safer, smarter and be more productive. Moreover, many of the proposed solutions to fight digital illiteracy require long-term planning, and the execution process requires approvals from many layers, as well as substantial changes in areas which traditionally are slowly paced, such as the educational system. These solutions also have long validation periods and require agreement from many bodies

to be implemented and successfully executed. Jackman in a recent study emphasises the mismatch between the rapid advancement of digital technology and our adaptation in education, indicating a traditionally slow pace of change in the education system. Even if there have been extensive efforts from countries and organisations in the creation of digital education programs, the weak coordination between them, causes no globally accepted meanings(Jackman et al., 2021).

Paul Gilster, a visionary writer, in his book Digital Literacy which was published when the internet was in its infancy, explains why it is vital for learners, apart from getting the information, also need to have the skill to filter and evaluate it before transforming it into knowledge (Gilster, 1997). Martin and Grudziecki, in their study about Digital literacy (Martin and Grudziecki, 2006), explained a model that consists of three distinct layers. This model can be used as a base for classifying digitally illiterate users.

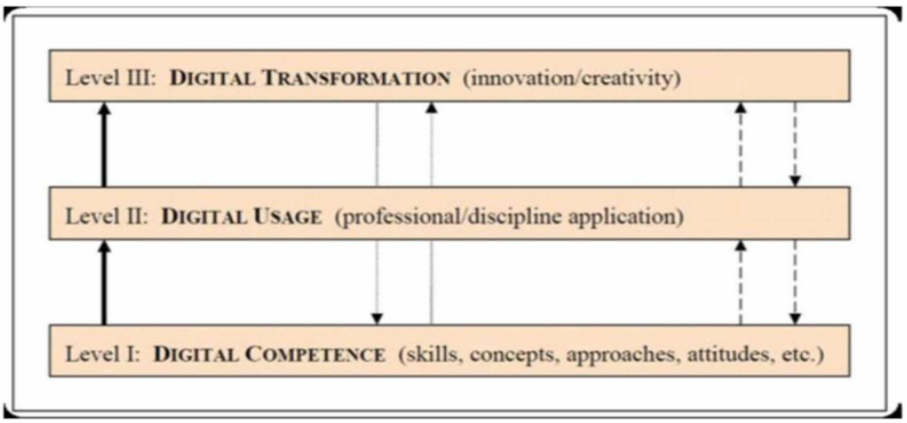


Figure 1: Levels of Digital Literacy (Martin and Grudziecki, 2006)

In this research, the focus will be primarily on users that are on level one, the digital competence level. This layer covers a range of topics such as skills, knowledge, attitude and awareness about digital technology that a person might have. However, for a comprehensive understanding of the potential effectiveness of the contextual help tool, a small subset of participants from level two will also be included. This approach offers a better insight into the tool's applicability to users with more advanced technological skills.

Several factors can be responsible for high levels of digital illiteracy, these include country of origin, gender, education level, profession and age. Niehaves analysed internet adoption among the elderly, and the findings undoubtedly show the impact of socio-demographic variables (Niehaves and Plattfaut, 2014). Schaffer, in his study about the Digital literacy of seniors, lays out several reasons for the cause of high percentage of digital illiteracy in the age groups over 60. These are, different perspectives on approaching new technologies, the lack of interest or the different ways of acquiring knowledge (Schäffer, 2007). Interestingly, older people can cope with technology when needed, especially if there are cognitive visualisation aids such as graphical representations or the user interfaces that follow Gestalt's principles (Gilmore, Tobias and Royer, 1985).

Daily online tasks like transferring funds between bank accounts via the internet can be completed by most people effortlessly nowadays. Unfortunately, many seniors are highly

likely to fail even if they were previously trained. In their research group, Steelman showed significant barriers to continuous improvement in digital skills because of over-reliance on note-taking instead of pattern recognition of the functionality (Steelman et al., 2017). The result was a failure to complete previously taught tasks when there were minor user interface changes.

However, focusing only on groups of people of an older age means excluding significant parts of the total population that might classify as digitally illiterate. Even in European countries where digital illiteracy is less evident, many young people face difficulties.

A revealing report about Digital Divide in Ireland was produced by Accenture, a leading company specialising in information technology services and consulting. The report shows that one in five adults under the age of 34 describes their digital skills as average or below average (Accenture, 2020).

One of the obstacles that people report when it comes to digital competency is access to services. 30% have trouble dealing with the problem because they need to learn how, and another 29% say they need help with where to acquire the knowledge.

Milner, Chief Executive Officer of the Good Things Foundation, an organisation that tries to improve people's lives through digital means, explains that "the digital divide is not binary - you are not just online or offline - it is possible to be an internet user but only have the skills or interest to use digital tools in a limited way" (Accenture, 2020). That confirms the findings in the same report, which shows a spectrum of digital illiteracy.

To fight digital illiteracy, a number of solution programs have been proposed and implemented by organisations such as Unesco. These programs are based on various frameworks, such as the Digital Literacy Global Framework (UNESCO, 2019). All these frameworks are based on data collection for measuring illiteracy, assessments, validations and execution of educational programs designed based on the generated reports. These programs are executed by teachers in the education system or other organisations (nonprofit or for-profit) at a country level.

In the last decade, people can acquire digital skills through ICDL training programs in many countries. The ICDL, a standardised framework designed by European countries, previously known as ECDL, has four programs based on the experience of the person seeking to upskill their knowledge in technology. ICDL has succeeded, with over 16 million people completing the training (ICDL, n.d.). However, as expected not every person with lack of digital skills joins these educational programs.

Multiple times over the years, the members of the G20 forum have expressed the need for a standardised data collection strategy required to measure an internationally accepted Digital Literacy Index. The lack of standardisation affects the implementation of training programs and delays the execution of digital transformation objectives (Chetty et al., 2018).

2.2 Abbreviations - Acronyms

Difficulties may arise when people with different levels of digital literacy skills try to consume online content that contains technical terms such as technological news-related articles, technological product descriptions or even technical specifications and requirements. While there are no studies that specifically address the impact of technological acronyms or abbreviations on individuals with low digital skills, it is well documented that the use of unfamiliar terminology can be a barrier to understanding in any field.

The use of technical terms in everyday life has substantially increased, which can be easily noticed while visiting, for example, news portals. Users often come across texts with abbreviations, acronyms or technical terms in sections about technology news or even general daily news articles that reference something technological. In many cases, content editors' writing style cannot be considered non-technical friendly. Many editors often use technical jargon to show their audience expertise even if they know that the comprehension level will be low (Moldovan, 2022).

That causes readers to partially comprehend the article or spend considerable time deciphering the meaning of acronyms. Shulman showed that it is possible for people who are exposed to jargon to be less interested later in reading similar articles than others who read about the same topics but without the use of specialised terms (Shulman et al., 2020). Appelman suggests that unfamiliar acronyms affect their desire to continue reading (Appelman, 2019). This would possibly have the effect that people with some level of technological illiteracy might not show enough interest or engagement to technological related news and thus they won’t expand their knowledge.

On the other hand, using abbreviations and acronyms is helpful in some cases. It has proven more effective when communicating with peers in closed environments with similar levels of understanding and knowledge. There is frustration when it comes to communicating with people outside of it. Appelmans research showed that readers might not bother when they come across acronyms that they are aware of their meaning in articles. But, they seem to have negative feelings towards acronyms that they do not understand (Appelman, 2021). That means journalists must put more effort into explaining the acronyms and abbreviations. Expecting editors to stop using technical terms or acronyms is unrealistic.

Keeping up with the technical terms is essential as they evolve and are associated with technological achievements. As technology advances, many terms appear less, and others tend to occur more often. For example, internet users who wanted to learn about hard disks in the 2000s would probably come across the technical terms “Megabytes” or “Gigabytes”. That is not the case twenty years later. Hard disks now have higher capacity, so the expectation is to see the term “Terabytes” or “Petabytes”.

Coming up with an assistive solution that would tackle the issue of unknown acronyms, abbreviations, or technical terms that appear in online content is more of a requirement rather than a nice-to-have feature.

2.3 Artificial Intelligence - Natural Language Processing

The year was 1955 when John McCarthy, in a workshop at Dartmouth College, first mentioned the term "Artificial Intelligence" (McCarthy, 2019). With the term Artificial intelligence or the popular, short acronym "AI", we define that with the help of science and technology, machines can have some form of intelligence. Making them intelligent means they could perform tasks that usually require intelligence at a human level. These tasks include pattern recognition, decision-making, computer vision, translation, and more. By having machines able to perform these kinds of tasks, humans can not only automate long, repetitive and tedious tasks but also, with the correct data, can have a view into the future.

Alan Turing, in 1950 published an article where he questioned if one-day machines would have the ability to think (Turing, 1950), and he proposed a validation test now known as the "Turing test". Turing's test tests the capability of a machine to think like a human (Turing, 1950). Multiple times over the last decade, Google demonstrated systems that can pass Turing's test. These systems, in the beginning, had the intelligence of a 13 old child (BBC, 2014). As data collection exponentially grew and scientists developed new and improved algorithms, it made it possible, in specific tasks, for humans to not recognize that they are communicating with a machine.

Natural language processing, a subarea of artificial intelligence, allows machines to understand human speech and text. Humans can program computers to collect data from an input source, clean the redundant parts, and feed that data into a sequence of algorithms. Atkinson and Goot showed that with text mining techniques and automated machine learning in news articles, detection of crucial information about the content, such as news category, is possible (Atkinson and Van der Goot, 2009). Parsing, cleaning, and normalising data from the website using techniques with XPath selectors, as Lammel shows, would allow easier extraction of results and performance (Lämmel, 2007). Features like stemming and lemmatization can help in the preprocessing phase and increase data accuracy (Balakrishnan and Lloyd-Yemoh, 2014). The data can then be split into smaller parts: the tokens, each with a specific weight, a technique known as tokenization. Based on the weights, algorithms can figure out patterns such as sentiment analysis, tags detection and classification.

Classification would be the technique that this study will use for the categorization of the web pages that the user visits. Since the focus is on a group of people with technological illiteracy characteristics, the contextual help extension should have the intelligence to understand when the user is reading an article that is technologically related. That helps avoid false positives. An example of understanding the possible issue would be the acronym “DNS”. In a finance-related article, “DNS” would probably mean “Deferred Net Settlement”. However, in a technology-related article, the definition should be “Domain Name Server”.

In addition, due to the enormous amount of technological acronyms and abbreviations, finding and storing them in a single dictionary would be challenging. For this reason, services from popular online Artificial Intelligence platforms will be used for detection as a supplementary approach to cover the scenarios of abbreviations or acronyms that do not

exist in the dictionary. Services like ChatGPT from OpenAI have the advantage of context awareness which increases the level of assurance in successful detection.

2.4 Browsers and extensions

Browsers have been the medium for accessing internet content over the last thirty years. In the beginning, browsers were text-based programs with limited capabilities; however, with the increase in internet usage, software companies started producing graphical user interface based browsers. The constant addition of features quickly made them one of the most used pieces of software. Today we have a wide variety of browsers that offer different features.

According to several different resources, the most popular browser is Google Chrome. Statcounter, one of the well known platforms for analytics, consistently shows 65+% of users use Chrome (StatCounter, 2019). Similar numbers appear in other analytics platforms such as W3Counter (W3Counter, 2019) and Statista (Statista, n.d.). Chrome was one of the first to offer the ability to users to install browser extensions.

A browser extension is a piece of software that offers additional functionality. For this to be possible, Google had to add an application programming interface (API) for the developers. The functionality offered by this API allows developers to do transformations to the HTML content that the web server sends to the browser before and after the rendering phase (Google, n.d.).

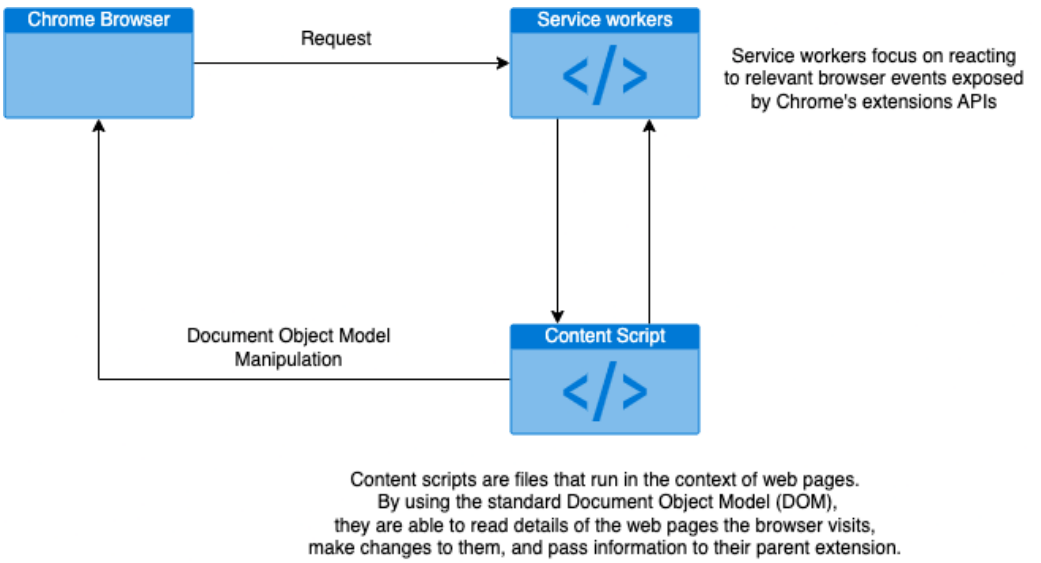


Figure 2: Google Chrome Extension API

Many developers took advantage of this functionality, and now thousands of extensions are available in the Chrome web store.

The type of extensions varies from tools for productivity, accessibility and customization, to ad blockers, to helpful utilities for completing everyday tasks, to tools for developers and even extensions for fun and shopping. Official statistics are not published but based on the number of installations, reports show numbers close to 180.000 extensions available in the Chrome Web Store (DebugBear, 2020).

To start with the development of an extension, plenty of resources are available online and offline. In his book, Mehta sets out the critical pieces for developing an extension, the steps a developer can follow, and what precautions should be considered (Mehta, 2016). In the official documentation website for Chrome extensions, Google offers examples, guides and extensive resources for developers who want to build their extensions. The extension development for this research will follow Google's recommendations for modifying the website's content.

The use of Artificial Intelligence Natural Language Processing services for content detection is one of the possible features that the contextual help extension will try to make use of. The goal would be, with NLP, to predict the website category accurately and, more specifically, identify if the website falls into the technology category.

2.4 Summary

Several studies and projects have shown that having contextual help available when needed improves the execution of a task. User reaches the goal faster and with less frustration. A short list with contextual help tools that were built for users with low digital skills follows:

- Dai Y et al in their study about providing reliable help for web tasks in seniors showed how they implemented the Tipper, a contextual help extension that provides help on icons, links and buttons. Tipper received very positive usability testing results (Dai et al., 2015).
- LemonAid, a selection-based crowdsourced contextual help for web applications is another example of how contextual help can assist on tasks of users with low levels of digital skills. Chilana et al in their study about Lemonaid showed how users can get faster and easier help for everyday tasks with the help of contextual help. The measurements showed that the tool retrieved a result for 90% of help requests (Chilana, Ko and Wobbrock, 2012).
- Creating contextual help tools not only for browsers but also for GUIs using screenshots is something that Tom Yeh et al explored in their paper. Specifically they provided a way to provide contextual help by screen capturing, something that tech support and end users found useful. Results have been validated with 60 real tasks (Yeh et al., 2011).
- Yadav et al, in their study for acronym retrieval through text messaging, showed that help for acronyms even outside of the online environments is possible and can assist semi-illiterate users (Yadav et al., 2015).
- NASA is maintaining a contextual help extension with definitions for space-related content (NASA, 2022). NASA's tool is open sourced so that anybody can contribute to the knowledge base dictionaries. It has more than 7000 code contributions and

over 25000 definitions for the acronyms that NASA uses. The extension tool is available not only in Chrome but in Firefox browser too.

Contextual help tools have a role to play not only in groups of users that have some level of digital illiteracy and try to understand articles with technical terms but also in groups of users who read niche articles. For example, medical articles, law journals, or even space related news. In an environment where terms appear often, tools for contextual help could provide significant help to the users. Contextual help limits the need to switch contexts, and provides assistance on the spot while allowing the user to continue the flow of their task.

3. Working Theory

3.1 Research Questions

The research questions for this research are as follows:

RQ1: Can a contextual help browser extension improve the comprehension of online articles that use technical terms among a group of digitally illiterate internet users?

RQ2: Does the contextual help tool positively impact the reading time for digitally illiterate internet users?

RQ3: How can AI assist a browser extension in the classification of websites?

3.2 Research Hypothesis

By showing contextual information, on-demand (e.g. on mouse hover), to technological terms, users who require additional information while browsing technological webpages can access accurate definitions in real-time without having to do manual research themselves and without leaving the active browser tab.

That can potentially increase the comprehension level of the text since the end-user does not need to have in advance knowledge of technical terms, making it easier to interpret.

A contextual help tool can assist not only users who don't know a technical term but also those looking for validation, references, or even expanding their knowledge. That means that a broader audience could find the contextual help helpful.

3.3 Contribution to Research Knowledge Anticipated

Using acronyms, abbreviations, and technical terms in articles often leads to reader frustration, negative sentiments, and annoyance, particularly among those with limited technological literacy. While individuals closely engaged with technology may effortlessly understand articles that include several technical terms, those with less digital literacy may either need help deciphering the meaning or opt to stop reading (Appelman, 2019).

Existing solutions for enhancing digital skills rely on traditional educational approaches. This research, backed by the literature review, proposes a different solution which addresses the issue more directly and in real time.

Additionally, the ability to keep the definitions of the technical terms constantly updated means more accurate content and broader coverage. The use of artificial intelligence adds a layer of protection, preventing false positive detections and ensuring that the extension's main functionality is enabled only in the appropriate context.

Upon thoroughly investigating this solution's architecture and implementation, and validating its outcomes, this study could serve as a blueprint for addressing illiteracy in other domains, including finance, law, and medicine.

4. Research Design and Methodology

4.1 Research Methodology

This research aims to first develop a contextual help chrome extension and then measure if using the contextual help tool increases the comprehension level of digitally illiterate internet users when reading articles that include technology-related abbreviations, acronyms or terms.

Opting for a mixed method approach to research can be advantageous because it allows the combination of qualitative and quantitative methods to get the most comprehensive and accurate data. Additionally, it will enable the exploration of data from multiple perspectives.

A qualitative method focuses on understanding the experiences and thoughts of individuals or groups. In this research, the people with some level of digital illiteracy. Because qualitative is used to explore complex topics or to gain a deeper understanding of a specific group of people, it can also help to identify solutions to problems that may have been overlooked. A qualitative method also has the advantage that it can be more participatory. The participants have a more significant role in the research process, which can help get a more accurate picture of the data collection. Even if it is more time-consuming because of the type of approaches for collecting the data, such as interviews, surveys, etc., spending more time with participants will give a more detailed understanding of a topic.

This study also requires a quantitative method to investigate whether a difference exists between the time a user takes to look up the definition of an unknown acronym or abbreviation and the duration it takes for the contextual help tool to provide the same information when enabled. Additionally what potentially is the impact of this difference in the time it takes a participant to read a technology related online article.

A mixed method approach which includes elements from qualitative and quantitative methods, gives the best opportunity to explore the effectiveness of the contextual help extension tool. After exploring a number of available mixed methods, such as the Convergent parallel, the Embedded, the Explanatory sequential and the Exploratory sequential, a method that follows the Explanatory sequential design was chosen.

This method focuses mainly on the quantitative phase, which occurs first. The qualitative phase, which comes after, will help to explain the results discovered in the quantitative phase. Toyon, in his study, analyses a set of stages for the explanatory sequential design type research (Toyon, 2021), which consists of an approach to inquiry, data collection, analysis of the quantitative data, data recollection, analysis of the qualitative data, integration and finally, results report. These are the steps that will be followed in this research.

An interactive questionnaire survey will measure the effect of contextual help tool usage. Participants will be asked to answer several multi-choice questions based on an online technological news article. In the beginning, the contextual help tool will be disabled so the participants will have to understand the article based on previous knowledge. Once they finish the reading, the contextual help tool will be enabled and the participant will have to answer the survey questions about the helpfulness, performance and definition accuracy of the tool.

Interviews will be conducted so that a better understanding of the participant's background with the technology will be evaluated. Later, participants will be asked to comment on the contextual help tool's usage and the impact on their article understanding. Participants' feedback about the definitions they received and the tool's user interface will be used to improve later versions. Thematic analysis, a commonly used method in qualitative research, will also allow us to identify and analyse patterns within the dataset. This type of analysis is flexible and can provide rich insights. Moreover, it can help to organise and describe a dataset in detail and interpret various aspects of the research topic.

Organisations for helping digital illiterate people will be approached, and members of these organisations will be asked to participate in the study. The participants that will be taking part in this research will be selected based on the digital skills they believe they have and based on their relationship with technology in general.

The steps planned for the execution of the survey are as follows:

- Participants receive an overview of the research and the issue the contextual help tool tries to address.
- They start by reading an online article about a new laptop release.
- After that, demographic data and technology-related information will be gathered through multiple-choice questions.

- Participants then will enable the contextual help tool, revisit the article, and use the tool.
- Post-use, they complete the survey by assessing the tool's effectiveness, efficiency, and user experience.
- The final task involves participants searching for a specific acronym in a new browser tab. Stopwatch-recorded completion times will be later compared with the tool's time to define all acronyms detected in the initial article.

During the development of the survey design for this research, a formal pilot study was not conducted. However, a considerable amount of informal feedback was collected from various individuals such as colleagues, friends, family, and the supervisor. Their input helped to design and structure the survey, ensuring it was easy to understand and user-friendly.

Moreover, their feedback was essential in refining the questions to be clear and suitable to the research objectives. In addition to this, I oversaw each instance of survey completion. Being present during the survey provided additional support, allowing participants to ask questions or seek clarification. As a result, the survey administration was structured but also flexible and responsive to the needs of the participants.

4.2 Research Population and Sampling

This research's target population was primarily individuals with basic computer skills but lacking extensive digital literacy, particularly in understanding technical terminologies encountered on technology-oriented websites. This group encapsulates various demographics, such as seniors, individuals aged 50 and above, mature students, and professionals whose work does not revolve around daily computer usage.

The rationale behind choosing this population was to gain insights into the experiences of those less familiar with the digital world but engage with it. Understanding how this group interacts with and benefits from a tool that assists comprehension of technical terms is critical when evaluating its impact.

In addition to the primary target group, a number of individuals with advanced technical skills were also included in the study. Their inclusion was to provide a comparative perspective and further assess the broader utility and impact of the contextual help extension. As people with a high level of skills in the digital world, their experience and feedback enrich the understanding of the extension's functionality across a broader spectrum of users.

The individuals for the interviews were selected based on their self-identified proficiency with computer and internet use and their willingness to engage with the study. This selection aimed to provide a diverse and representative sample that would produce a complete understanding of the tool's potential benefits and challenges across various user categories.

Data were collected from three primary pools of users. A total of 25 individuals agreed to participate in the survey, which took place over a period of four weeks, starting on the 6th of April, 2023, and ending on the 4th of May, 2023. The initial ten participants were members of

the River Valley Community Centre located in North Dublin, most of them over the age of 50. The following group, also comprising ten individuals, were from Coláiste Íde, a college of further education located in the same region. The remaining five participants were employees at the restaurant within the SAP company premises in West Dublin. Given the different schedules of the participants, the study was conducted over multiple visits.

These differing backgrounds contributed to the wide range of feedback and perspectives shared about the use and benefits of the contextual help browser extension.

5. Architecture and Development

5.1 Browser extensions

Browser extensions are software programs that add, remove or alternate functionality or extend pre-existing functionality to web browsers, such as Google Chrome, Mozilla Firefox, and Microsoft Edge. Typically extensions are written in HTML, CSS, and JavaScript and can be installed from the respective web stores, such as Chrome Web Store or Firefox Add-On store, however during the development, an extension can still be installed and enabled locally if the user changes the security settings and lift the restrictions that allow unverified extensions to be installed.

Browser extensions interact directly with a web page's Document Object Model (DOM). The DOM is a representation of the objects comprising the structure and content of a document on the web and has a tree-like structure. Extensions can use Javascript to access, manipulate and also interact with DOM.

The main elements of a browser extension are typically a background page, a content script, a component or several components and a manifest file.

To ensure high security and prevent malicious code from executing, Browser vendors made the browser extensions run inside a sandboxed environment with limited permissions. Extensions are granted specific permissions based on their declared functionality, such as the ability to read and modify web pages or to access the browser's storage system.

5.2 Browser Extension building blocks

The building blocks of a browser extension can be categorised in four main components:

- **User Interface** - The user interface is the front-end of the extension that users interact with. It is responsible for displaying the extension's features and options. The user interface can be built using HTML, CSS, and JavaScript, and can be displayed as a toolbar, a popup, a side panel, or a menu. It is important to note that browser extension developers must design and develop a user-friendly interface which is easy to use and understand.

- **Background Scripts** - The background script is the core of the extension that runs in the background and manages the extension's functionality. It is responsible for handling events, executing tasks, and communicating with other components of the extension. It can be written in JavaScript and can use APIs provided by the browser to interact with the browser's internal components.

- **Content Scripts -** Content scripts are responsible for manipulating the content of web pages. They are executed in the context of a web page and can interact with the DOM (Document Object Model) of the page. Content scripts can be used to add or remove elements, modify the appearance of the page, or inject new scripts into the page. Content scripts are also written in JavaScript and can use APIs provided by the browser to interact with the browser's internal components. The contextual help extension main functionality will be part of a content script.

- **Permissions -** Permissions are a security feature of browser extensions that control what the extension is allowed to do. They specify what resources the extension can access, such as tabs, bookmarks, history, or cookies. Permissions are declared in the extension's manifest file and must be approved by the user before the extension can be installed.

One of the most important files in a browser extension is the manifest file. Manifest plays a significant role in defining the extension's characteristics and functionality. The manifest file is written in JSON (JavaScript Object Notation) format and contains metadata about the extension, such as its name, version, description, and author.

The manifest file also declares the permissions that the extension requires. This ensures that the extension only has access to the resources it needs and that the user's privacy and security are protected.

Additionally, the manifest file specifies the location of the extension's files and the scripts that should be loaded when the extension is installed. This allows the browser to load and initialise the extension correctly and ensure that it functions properly.

The manifest file also defines the extension's icons, which are displayed in the browser's toolbar, menus, and other user interface elements. Icons are an essential part of an extension's branding and can help users quickly identify the extension and its purpose.

Finally, the manifest file specifies the browser versions and platforms that the extension is compatible with. This ensures that the extension is installed only on supported browsers and platforms, preventing compatibility issues and errors.

5.3 Plasmo Framework, a framework for building browser extensions

Creating a browser extension, particularly one with multiple features, presents a significant challenge due to the absence of definitive guidelines, established design patterns, or comprehensive architectural plans that have been publicly available and validated for addressing common development issues.

Typically, browser extension developers initiate the development process by going through the browser's extension api documentation and organising a project in a way they believe

can easily scale by utilising tools with which they are familiar and ultimately dedicating time to configuring tasks such as bundling, minification, and formatting.

Although several open-source repositories exist that may serve as a starting point, most of them need to be updated or are highly opinionated. Fortunately, a framework designed specifically for browser extension development has surfaced and gained popularity in recent years. This framework, known as Plasmo, incorporates best practices with regard to file and folder structures, provides numerous out-of-the-box helper functions, uses TypeScript (a JavaScript variant with type safety), maintains up-to-date documentation, and boasts a thriving community of developers who regularly introduce new features. For the development of the contextual help browser extension, Plasmo was the chosen framework.

5.4 Development of functionality modules

For use cases such as the contextual help browser extension, utilising the content scripts block for including the necessary logic appears to be the correct approach. This is because content scripts are specifically designed to enable the modification of web page content. It is important to note that the implementation of the desired functionality will be structured in a non-blocking manner, which entails that the browser will render the HTML Document in its typical fashion while the manipulation operations occur subsequently.

With the help of Typescript’s compiler's module resolution (the process the compiler uses to figure out what an import or export refers to) the functionality will be split further into small logic blocks. Modular programming has a number of benefits. Firstly, simplicity and maintainability. By breaking down code into smaller blocks makes it easier to understand and maintain. Each block can be developed and tested independently, and changes made to one block do not necessarily affect the others. This makes it easier to locate and fix bugs, and to make updates or modifications as needed. Additionally by dividing code into small, reusable modules, developers can save time and reduce redundancy. When a specific functionality is needed, instead of writing new code from scratch, they can simply reuse an existing module. Also modular design allows developers to scale up their applications easily. As new features or requirements are added, developers can create new modules or modify existing ones without affecting the entire application. Finally in large codebases different developers can work on different modules simultaneously, without the fear of disrupting each other's work

The functionality described below gets executed sequentially, with each subsequent step dependent upon the output of the preceding step. This procedure ensures that contextual help tooltips are appended when all the requisite conditions are fulfilled, thus preventing the occurrence of false positives and avoiding the execution of redundant operations.

5.4.1 Webpage load detection

In the content script first an event listener is added for the loading event to the window object. The window.addEventListener("load", ()) method in JavaScript registers an event

listener on the window object to listen for the load event, which is triggered when a web page and all of its resources (such as images, scripts, and stylesheets) have finished loading.

The code for the listener in my GitHub repository:
https://github.com/unseen1980/acro-helper/blob/main/contents/loader.ts#L64

When the load event is fired, any functions or code included in the second parameter of the addEventListener method will be executed. This can be useful for performing actions that require all page resources to be loaded, such as initialising scripts or manipulating the DOM. Once the page is loaded in the callback function of the event listener I start the execution of the logic blocks.

The execution of the logic blocks in my GitHub repository:
https://github.com/unseen1980/acro-helper/blob/main/contents/loader.ts#L68-L122

These blocks, grouped into four phases (A,B,C,D) can be seen in the diagram below:

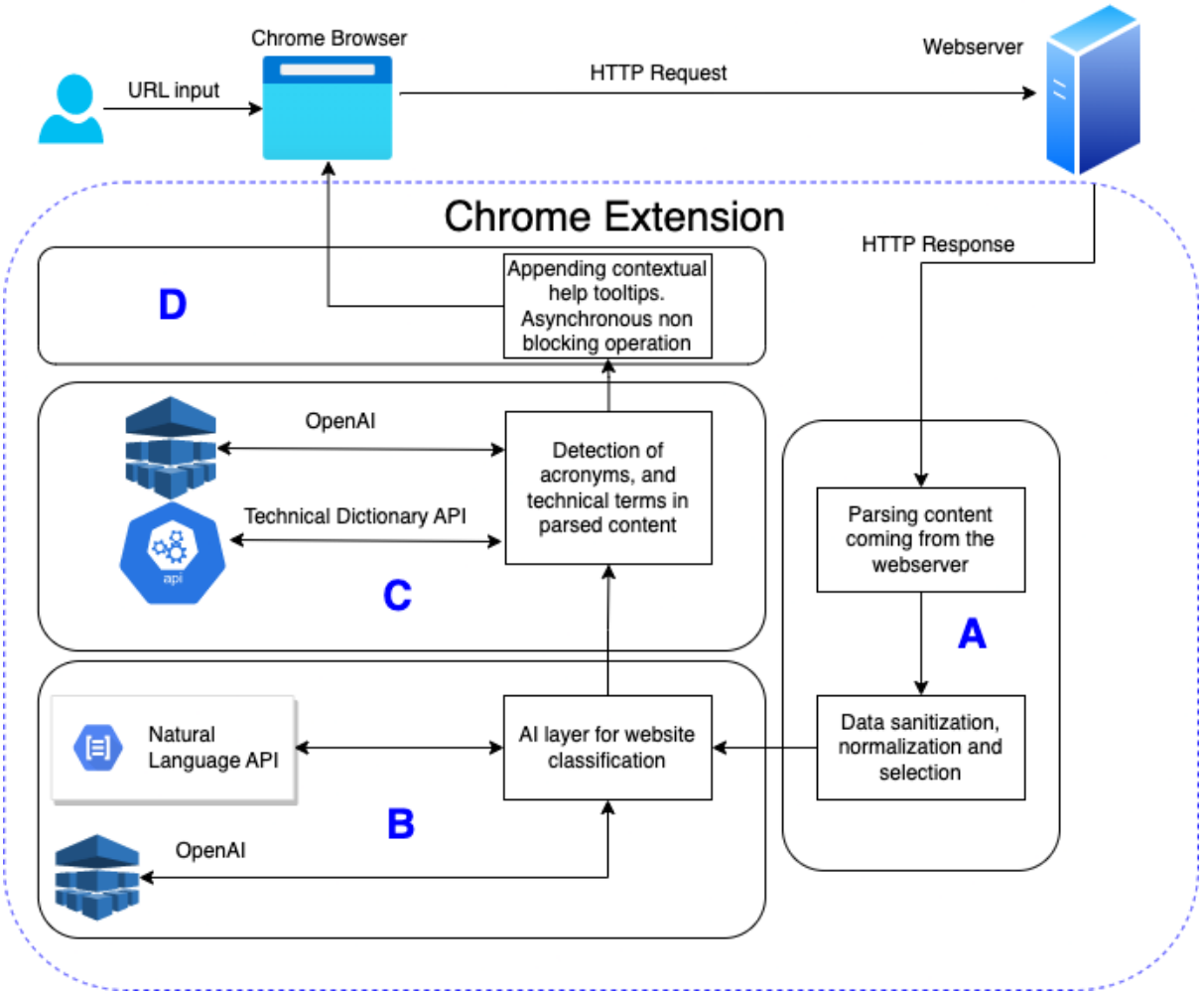


Figure 3: Chrome Extension Architecture

5.4.2 Content sanitation and parsing

Before manipulating the DOM, in this case by appending the contextual help tooltips, I need to identify if the website the user is browsing is technology related. To do so, I need to execute a number of functions.

Starting with the sanitization which has the purpose of stripping out areas of the webpage that have a high possibility of not including acronyms or abbreviations related to technology. Our target is to capture only the text of the main article. For example, it is common these days for most websites to have popups about privacy or cookies. The content of these pop ups is usually irrelevant to the content of the main article, so they can be excluded. Other areas that, with some level of confidence, are safe to exclude are the header, the navigation sections, the footer, photo galleries and multimedia sections. Finally, tag elements that include references to javascript code, such as script tags, can also be excluded. In phase A, the execution of functions for parsing and sanitisation takes place.

The sanitiser function, which takes an argument expected to be a document object model (DOM) element. The function performs a sanitisation operation on the given document object to remove certain unwanted elements.

The function first defines a variable that contains a comma-separated list of CSS selectors targeting specific elements. These selectors target elements such as headers, footers, scripts, certain types of links and buttons associated with cookies or consent requests, and certain types of ads and sidebars.

Next, the function uses the "querySelectorAll()" method of the document object to find all the elements that match the given selectors and stores the resulting NodeList in a variable called "elements".

Finally, the function iterates over each element in the NodeList type variable “elements” using a loop. For each element, the function logs a message to the console indicating that it is being removed (so that it is easier for developers to debug) and then calls the remove method to remove the element from the DOM. By targeting specific semantic HTML elements, such as <header>, <script>, or <footer> and using targeted CSS selectors, the cleaning process can be executed with an acceptable level of confidence. Once these elements are removed, the sanitiser function returns the cleaned document for further processing when calling the sanitiser function. It is important to highlight that a document clone is passed as an argument when calling the function and not the original because, at this stage, manipulating the DOM is not required.

The next step is to clean even further the remaining document and, more specifically, to extract only the text from the main article. For this, a popular library from the Mozilla foundation called Readability is used. The @mozilla/readability library is a JavaScript library designed to help developers create applications to present web content in a simplified, more readable format.

The library uses a series of heuristics to determine which parts of the page are the main content, such as looking for the largest block of text, analysing the HTML structure, and checking for standard article metadata like the title, author, and publish date. It also applies some natural language processing techniques to help identify and extract content from within the page.

Once the main content has been identified and extracted, the @mozilla/readability library returns it in a standardised format, which developers can use. The library also provides options to customise the extraction process, such as setting a minimum content length or specifying a maximum number of images to include; however, for the contextual help extension, there is no need to change the default settings. Overall, the @mozilla/readability library provides a convenient and reliable way to extract the main content from a webpage, and the result is structured in a developer-friendly format.

The final step in the cleaning process is a white space trimmer function. This function takes an argument article and returns a trimmed version of its text content with all whitespace characters (including newlines and carriage returns) removed.

The function first checks if the article argument is defined and if it has a textContent property. If both conditions are met, the function trims the text content using the trim() method to remove whitespace characters from the beginning and end of the string and then replaces all remaining whitespace characters (including newlines and carriage returns) with a single space character using the replace() method and regular expressions.

The code for these functions in my GitHub repository:

- https://github.com/unseen1980/acro-helper/blob/main/lib/sanitizer.ts
- https://github.com/unseen1980/acro-helper/blob/main/lib/readability.ts
- https://github.com/unseen1980/acro-helper/blob/main/lib/text-cleaner.ts

5.4.3 Google Cloud Natural Language API(Taxonomy) - OpenAI Detection

Once the main article has been detected and the text has been extracted it is time to proceed with the next phase (Phase B) which is the classification of the webpage. The role of this step is to detect if the webpage falls into the category of technology related. Google Cloud platform offers a service that can be used for this and it is part of the Natural Language Processing API. This API provides natural language understanding technologies, such as sentiment analysis, entity recognition, entity sentiment analysis, and other text annotations, to developers. More specifically the service that the extension will use is the Content Classification which analyses a document and returns a list of content categories that apply to the text found in the document. The categories that are returned are part of the Google Taxonomy. Google taxonomy refers to the hierarchical classification system used by Google to organise and categorise information on the web. This taxonomy is used by Google's search algorithms to understand the content of web pages and deliver relevant search results to users.

The Google taxonomy includes a broad range of categories and subcategories, covering topics such as arts & entertainment, business, health, science, and more. Each category is further divided into subcategories, allowing for more granular classification of web content. For the contextual help extension categories related to technology such as “computers”, “electronics”, “software”, “internet”, “engineering” etc. should be present in the service response.

For the categorisation, two functions are required. The first is the request to Google NLP API, and the second is the translation of the results that the first function returned. The name of the first function is just “categorisation”, and it tries to send a POST request to a Google API endpoint to classify text based on its content. The function sends the request with a JSON payload that contains the text to be analysed. The response from the API is then sent back to the client as a JSON object through the response object. A 400 error is sent back to the client if there is any error during the process. The second function takes the array of categories that Google NLP API returns as input. Then it checks if any of these categories match a pre-defined list of allowed technology-related categories. If at least one of the input categories matches an allowed category, the function returns true, indicating that the webpage is tech-related. Otherwise, it returns false. The list of categories contains the following: "Computers", "Electronics, "Software", "Internet", "Computer Science", "Computer Education", "Computer & Video Games", "Computers & Electronics", "Technology News", "Engineering & Technology", "Technology".

The code for Google taxonomy function in my GitHub repository:
https://github.com/unseen1980/acro-helper/blob/main/lib/taxonomy.ts

The taxonomy-based web page category classification approach provided by Google's NLP API works well in most cases. However, there are cases where there is a misclassification. Especially when technology terms are present within the context of another topic, such as politics or finance. For example, a page about cryptocurrencies and blockchain laws within a political context would be classified as 'Politics', even though it contains many technology-related terms.

An additional layer of analysis using OpenAI's ChatGPT was introduced to address these misclassifications. After the initial classification by Google's NLP API, the page's content is passed to ChatGPT with a specific prompt asking it to check for the presence of technology-related terms.

The implementation involves a function that sends the webpage content to ChatGPT with a request to return a boolean value indicating whether the text includes technology-related terms. The ChatGPT's response is returned to the client as a JSON object. The function chatgptIsTechContent is the implementation of this interaction on the client side.

The code for ChatGPT classification functions in my GitHub repository:

- https://github.com/unseen1980/acro-helper/blob/main/lib/key-finder.ts#L79
- https://github.com/unseen1980/acro-helper-dictionary/blob/main/application.js#L86

By applying this dual-layered approach, there is an improvement in the webpage category classification. It is also essential to mention that Google Taxonomy classification always

executes first, and only if the result comes back as a non-technology webpage does the logic for contextual analysis by ChatGPT get executed.

This combination allows us to more accurately determine whether a webpage is technology-related, even when tech terms are used in non-technology contexts.

5.4.4 Detection of acronyms, abbreviations or technical terms.

At this stage, we have verified that the user is browsing a technology-related web page, and there is a possibility that this webpage includes one or more acronyms or abbreviations. To effectively identify them, several methods are available. Firstly, a lookup operation in a technological acronym/abbreviation dictionary is performed. Such a dictionary is stored on a remote web server and is developed explicitly for the contextual help extension. To minimise the initial payload that needs to be transmitted, the contextual help extension initially requests form the remote server to retrieve all the acronyms or abbreviations without their corresponding definitions, thus preventing a downgrade in performance of the extension.

After successfully retrieving the dictionary acronyms, the subsequent function can receive them as a parameter. The acronym finder function searches for the acronyms returned from the remote server within the webpage's body. Once an acronym is detected, an additional request is made to the server to retrieve its definition. To accomplish this task, the function uses a regular expression constructed using the RegExp constructor to search within the webpage's body for instances of the acronym or abbreviation. The function appends them to an array if it finds any.

Finally, at this stage, an array of JavaScript objects contains all the detected acronyms/abbreviations in key-value pairs format. The crucial element in this function, which holds the detection logic, is the regex expression that matches the acronyms from the dictionary with the words found in the webpage document.

More specifically, a new RegExp(\\b${key}\\b, "gi") is used, which creates a regular expression object. The key variable represents an acronym from the dictionary. The regular expression pattern being created matches a complete word, identified by the word boundary anchor \\b, and the key variable dynamically determines this boundary. The "g" and "i" flags make the pattern global and case-insensitive, respectively.

To provide more detail, new RegExp() is a constructor that creates a new regular expression object. As mentioned earlier, ${key} is a template literal that inserts the value of the key variable into the regular expression pattern. The \\b is a word boundary anchor that matches the position between a word character and a non-word character. Another \\b follows, matching the position between a word character and a non-word character. Finally, "gi" is the flag passed to the constructor to make the pattern global and case-insensitive. The "g" flag indicates that the pattern should be applied globally, i.e., to all occurrences of the pattern in the text. The "i" flag indicates that the pattern should be case-insensitive, matching the word regardless of whether it is in uppercase or lowercase letters.

The code for acronym detection functions in my GitHub repository:

- https://github.com/unseen1980/acro-helper/blob/main/lib/dictionary.ts
- https://github.com/unseen1980/acro-helper/blob/main/lib/key-finder.ts#L13

5.4.5 OpenAI - Language Model

Remarkable advancements in artificial intelligence have recently allowed developers to create solutions addressing the complexities of textual comprehension. Simply relying on dictionary lookups and web-based entries for accurate definitions might not be enough for all scenarios; such methods need more context awareness. An innovative service called ChatGPT from the OpenAI foundation was released into the market in 2022 and is proficient at scanning text for acronyms or abbreviations while providing contextually applicable definitions. Harnessing the power of a language model, ChatGPT gets understanding from the extensive training data it has been exposed to. As it searches vast textual sources, it uncovers complex patterns and relationships between words and phrases within their contexts. Thus, the model grasps the meaning of language, unravelling the meanings of words and phrases from their contextual surroundings.

How does ChatGPT determine context? The method is by using contextual word embeddings. These embeddings are vector representations of words learned based on the surrounding words in a sentence or paragraph. Using contextual embeddings, the model captures the nuances of word meaning that depends on the context in which they appear. This allows the model better to understand the meaning of words and phrases in context.

Furthermore, attention mechanisms play a vital role in ChatGPT's contextual comprehension. The model promotes awareness of a word or phrase's particular context by concentrating on selecting portions of input text relevant to a task or query. Over the years the neural network models that are specialising in natural language understanding and in content generation have evolved. ChatGPT has so far four versions. The GPT, GPT-2, GPT-3, and GPT-4. The main differences between them are the number of model parameters and the size of the dataset that they were trained on. For reference in GPT-2 there were 1.5 billion parameters and in GPT-3 this number grew to 175 billion. In the latest version, the GPT-4 the number of parameters is not disclosed however considering the improved accuracy it can be assumed that the number of parameters is way higher than the previous versions.

ChatGPT uses various language modelling algorithms to unravel the text context. These algorithms predict the likelihood of word sequences based on preceding text, allowing the model to figure out the most plausible meaning of a word or phrase in its unique context. Collectively, these sophisticated techniques empower the model to navigate textual context and accurately predict word and phrase meanings in diverse scenarios.

A function which makes an API request to the OpenAI REST service was created. The function receives the cleaned and trimmed text of the webpage as a parameter. The text is part of a question which asks the service to detect acronyms or abbreviations and return a javascript array of objects. Each object should include the detected acronym/abbreviation and its definition. By utilising OpenAI's service, the contextual help extension can access

data beyond dictionary entries, benefiting from the wealth of information contained within ChatGPT's language model.

The code for acronym detection function with OpenAI in my GitHub repository:
https://github.com/unseen1980/acro-helper/blob/main/lib/key-finder.ts#L41

5.4.6 OpenAI - ChatGPT prompts

The communication between the contextual help extension and the AI language model, when required, is happening via prompts. Prompts are inputs that the user is giving to the model. These inputs can be a question, a statement, a request or even an example that would help the AI generate the appropriate responses in the desired format.

A number of crucial aspects need to be considered when designing and implementing the prompts. Well-designed prompts will lead to well-formed and high-confidence responses.

First of all clarity, it is crucial for the prompt to be specific and clear so that the AI can understand the context and the intention of the specific request. The prompt should not be written in a way that can be translated in numerous ways. The next element that needs to be considered is the amount of detail that can be provided. The prompt should contain sufficient information to help the AI generate an accurate and comprehensive response without using probabilistic methods. To guide the AI model, specific requirements or background information or even more updated data than those that the AI already has can be passed in the prompt.

Format of the response is also essential, primarily when the response of the AI will be used in application programming interfaces. For example, in the contextual help extension, the expected response from AI should be formed as a Javascript object that can be parsed without errors. In order to achieve this type of response in all requests, an example of a javascript object with key-value properties can be provided to the prompt.

Finally, experimenting and tuning the prompts will most likely give better results. Adjustments and optimizations are critical phases of prompt development. Once the mentioned factors are considered, the AI responses will be significantly close to the desired outcome.

The code for OpenAI prompts in my GitHub repository:
https://github.com/unseen1980/acro-helper-dictionary/blob/main/application.js#L65-L105

5.4.7 DOM Manipulation - Appending tooltips

The final step is to append the useful data that was collected from the execution of the previous functions into the DOM of the web page. To do so I create a "replacer" function to identify particular words within a web page section and provide them with definitions.

The “replacer” takes three parameters, a "selector" to specify the web page section, a "word" representing the word needing definition, and the "definition" itself. The "selector" locates the web page section and collects its contents. The function uses a pattern to find the acronym or abbreviation without interfering with HTML tags. It then swaps each occurrence of the acronym or abbreviation with unique HTML code, which inserts the "definition" into a tooltip.

The HTML that will be used are the elements <dfn> and <abbr>. These elements provide additional information into specific terms or abbreviations, enhancing web pages with valuable context. The <dfn> tag, short for "definition, flags the enclosed text as a term's defining instance. Upon introducing a term, I wrap it within <dfn> tags. Browsers often style <dfn> text distinctively, such as with italics, signalling its defining nature.
Next the <abbr> tag which represents "abbreviation" and marks abbreviations or acronyms. Its main purpose lies in revealing the full meaning of abbreviations as users hover or focus on them. The "title" attribute needs to be added inside the <abbr> tag to achieve this.

The "title" attribute is an HTML attribute which provides extra information about an element, like a brief explanation or description. When paired with the <abbr> tag, the "title" attribute expands the abbreviation's form. When users hover over the abbreviation, most browsers reveal a tooltip containing the "title" attribute's information. In the "replacer" function, <dfn> and <abbr> tags, along with the "title" attribute, are used for embedding definitions or explanations in specified terms as tooltips. This seamless integration empowers readers to grasp term meanings without resorting to external sources.

Finally, the original content is getting removed from the selected web page section and the updated content—now decorated with definitions—returns to the same web page section. With this function, I can effortlessly add definitions throughout the web page articles and clarify any acronyms, abbreviations or technical terms for our readers' benefit.

The code for appending the tooltips in my GitHub repository:
https://github.com/unseen1980/acro-helper/blob/main/lib/replacer.ts

5.4.8 Acronyms and abbreviations searchable in a tab page.

One special feature of the Plasmo framework is the Tab Pages. Tab Pages are regular web pages that come with the extension bundle. This allows extensions to redirect to these pages programmatically. In the contextual help tool, the Tab Pages feature has been used to develop search functionality as an addon for the contextual help browser extension. Users are redirected to the Dictionary search page by clicking the relevant button when the extension menu is visible.

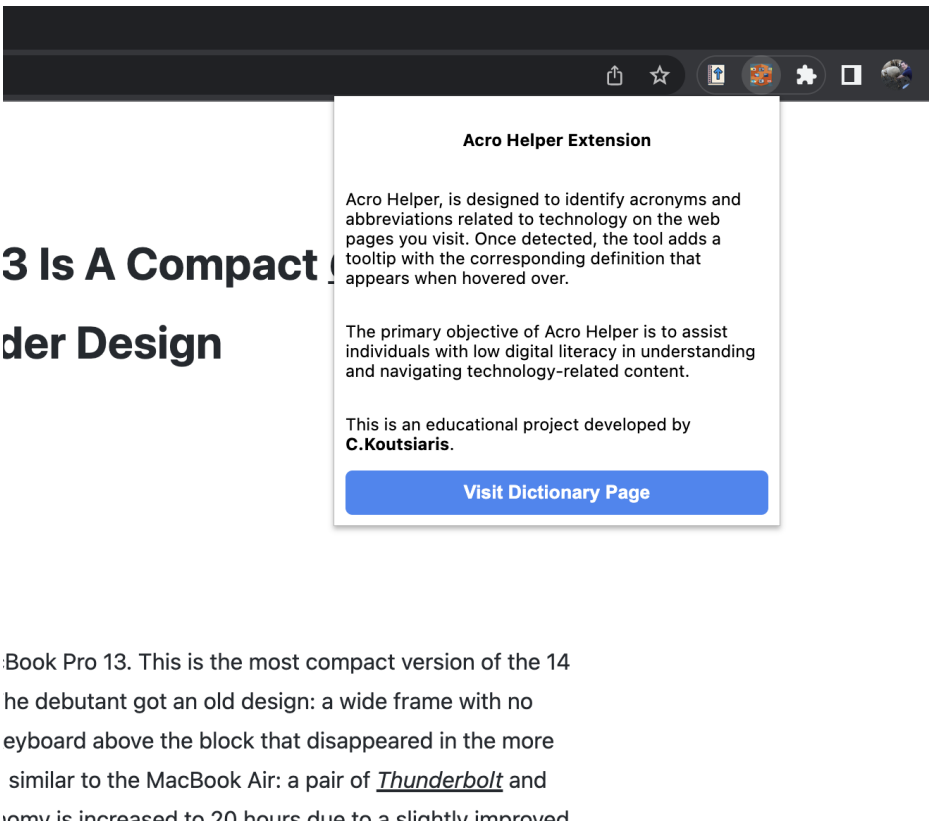


Figure 4: Contextual help extension popup appears when user clicks on the menu icon. Button for visiting the dictionary page is included.

The implementation of the Tab Page is in the ReactJS component DictionaryPage. DictionaryPage is designed to be an interactive page where users can search through a dictionary of acronyms and abbreviations. The function fetches the dictionary data asynchronously from a remote server and stores it in the component's local state. The data is then filtered based on the user's input to a search field, allowing real-time search functionality. The results are rendered as a list on the page. That gives the users that came across a technical acronym outside of the browser environment the ability to retrieve a definition.

Overall, the Tab Pages feature of the Plasmo framework has been a helpful feature for adding extra functionality to the contextual help browser extension. The enablement of creating a dedicated, interactive page for dictionary search enhances the user experience and the overall functionality of the extension.

The code for the dictionary search tab page in my Github repository:

- https://github.com/unseen1980/acro-helper/blob/main/tabs/dictionary.tsx

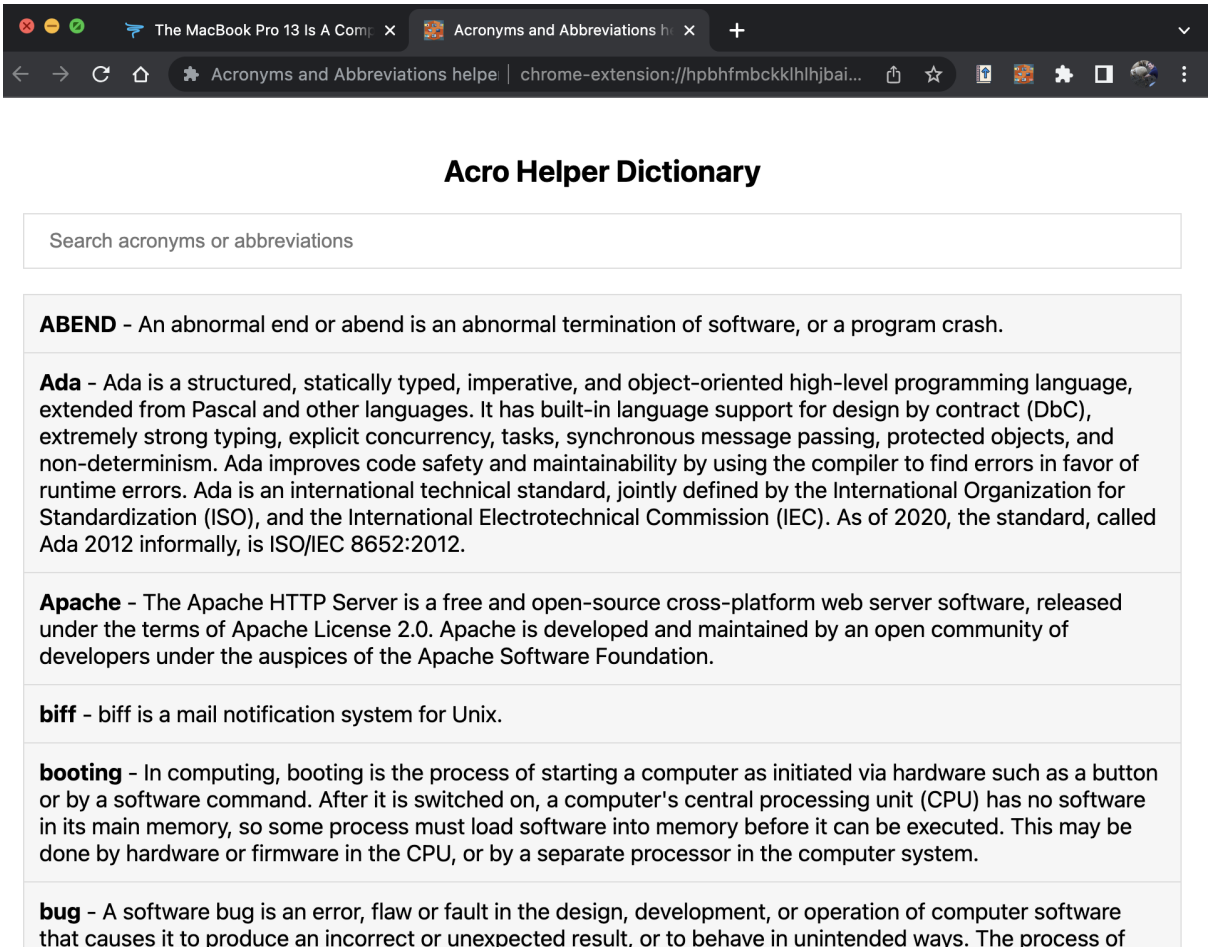


Figure 5: Dictionary page default view

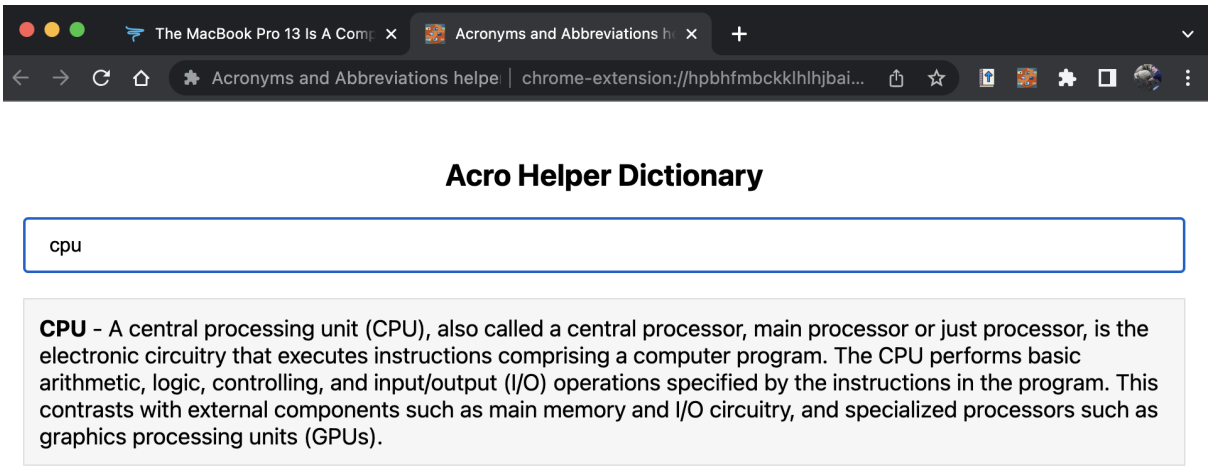


Figure 6: Dictionary page result based on word typed in search box.

6. Screenshots of the contextual help browser extension

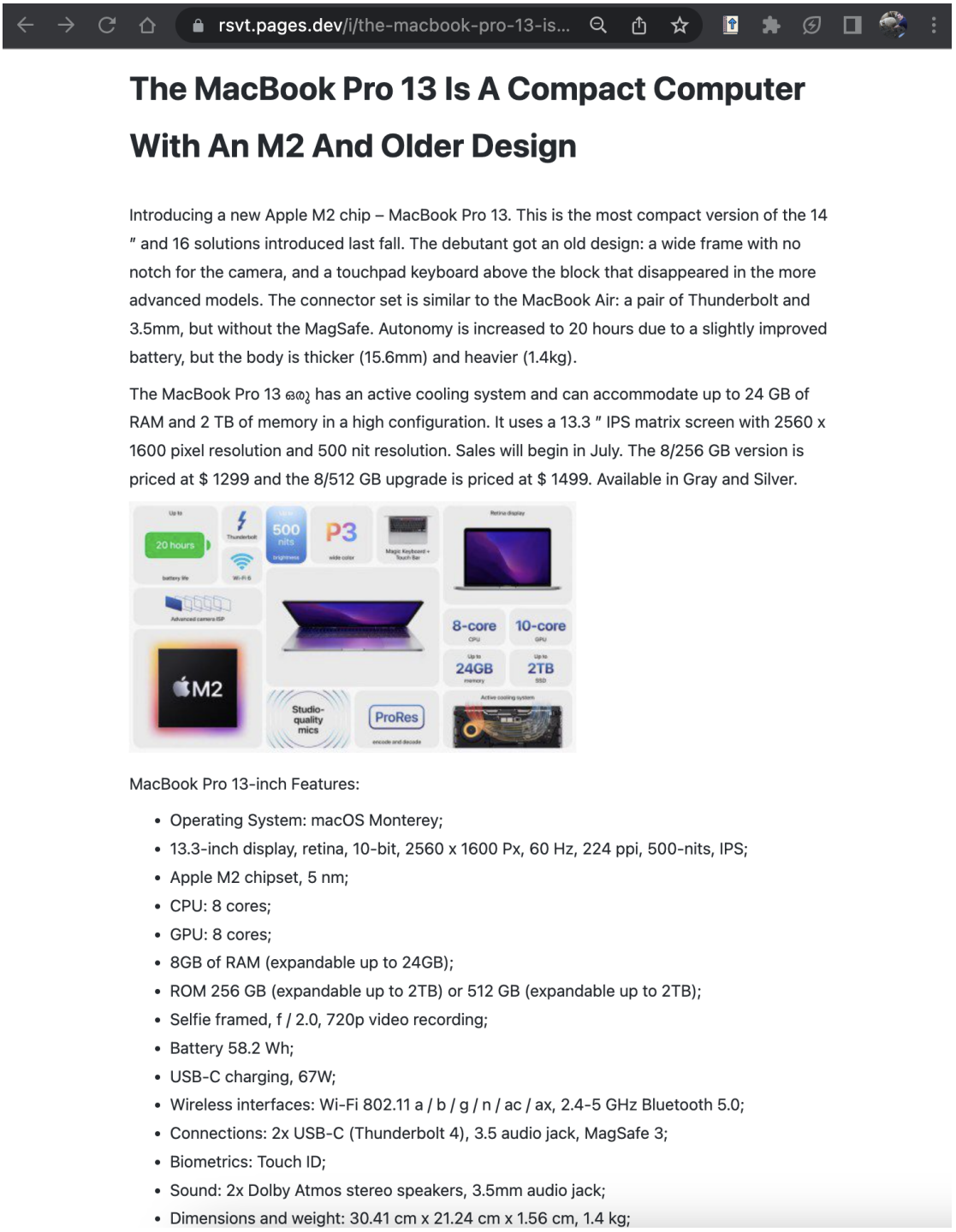


*Figure 7: A technology related online article **WITHOUT** the extension enabled*

*Figure 8: A technology related online article **WITH** the extension enabled. User hovers over the acronym “CPU” and the definition appears*

7. Presentation of findings

7.1 Introduction to findings

The study aims to understand the impact of a contextual help browser extension in improving the comprehension of technological-related online articles that contain technical acronyms or abbreviations among internet users with various levels of digital illiteracy. For this type of study the mixed-method approach was chosen and the research findings will be presented in the following sections.The research questions focus on the effects of a contextual help browser extension usage, the influence on the reading time that the users spend on an article and how Artificial Intelligence can assist with website classification and technical acronyms detection.

Following an Explanatory Sequential design, a quantitative analysis with the help of an interactive questionnaire was conducted. The participants additionally had to use the first stable version of the contextual help browser extension by visiting an online article and reading definitions of acronyms they did not understand by hovering over them. Furthermore, benchmarks were conducted to measure the performance of the contextual help extension at various stages. The subsequent phase was qualitative, where each participant went through an interview.

7.2 Methodological Checks

After exploring a number of mixed methods, the Explanatory Sequential was chosen for this study because it can provide a comprehensive understanding of the contextual help browser extension. As mentioned in the Research Methodology section, this method has two distinct phases. It starts with quantitative data collection and follows qualitative data collection. This research design tries to explain the findings of the quantitative phase with the help of the data collected in the qualitative phase.

7.3 Quantitative Data Analysis

The first part of the interactive questionnaire survey was designed to gather information on participants' demographic information, technology usage, and self-assessment of their digital literacy. The free edition of Qualtrics Survey software was used to build the questionnaire and to generate the reports. The first part of the survey included the following group of questions:

7.3.1 Demographic Information

The demographic distribution of the study's participants show a diverse age, gender, and educational backgrounds. Most of the participants (60%) were aged between 31 to 59 years, with 36% falling within the 18 to 30 age range, and 4% aged 60 and above. In terms of gender representation, 52% identified as female, followed by 44% identifying as male. None of the participants identified as Non-binary or third gender, and 4% preferred not to disclose their gender.

Evaluating the educational backgrounds, over half of the respondents (52%) had completed only high school education. Meanwhile, 36% held a Bachelor's Degree or its equivalent, and the minority (12%) had attained a Master's Degree or PhD or its equivalent. Visual representations of this data can be found in the Appendix A - Demographic Information.

7.3.2 Technology Usage and Experience

Most participants read technology-related articles online 2-3 times a week (40%). 24% read once weekly, and the same percentage read less than once a week, while 12% engage 4-6 times weekly.

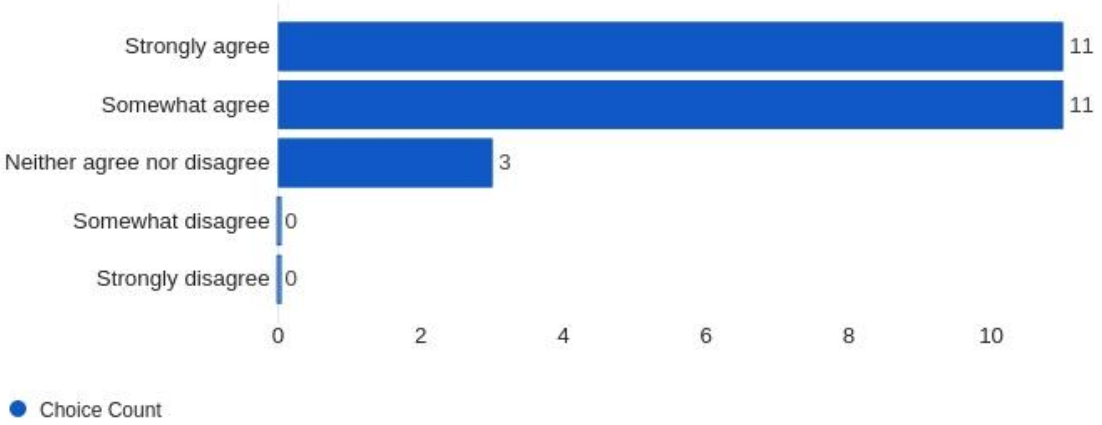
72% of participants rated their technological expertise as intermediate, comfortable with some advanced skills. Beginners make up 20%, with a basic understanding and limited experience. Only 4% each identified as advanced or experts, indicating proficiency in various technologies. Remarkably, none identified as beginners.

Most participants (68%) use a computer daily. 16% use it 4-6 times a week, 12% 1-3 times a week, and a small 4% never do, even in this tech-oriented sample.

Internet purposes ranged: 30.30% for professional activities, 27.27% for social networking, and 21.21% each for entertainment and educational resources. For a detailed breakdown, see the Appendix B - Technology Usage and Experience.

7.3.3 Self-assessment of Digital Literacy

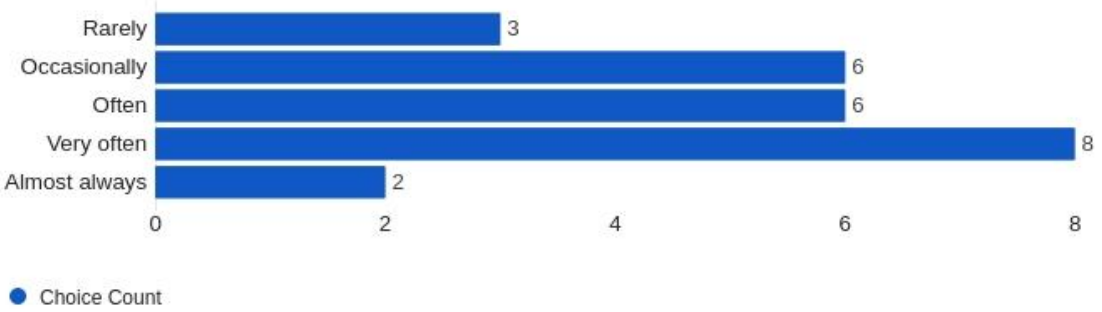
1. I am technologically literate, which means I have the ability to effectively use digital devices, software, and the internet for various tasks, including communication, information seeking, and problem-solving.



Answer	%
Strongly agree	44.00%
Somewhat agree	44.00%
Neither agree nor disagree	12.00%
Somewhat disagree	0.00%
Strongly disagree	0.00%
Total	100%

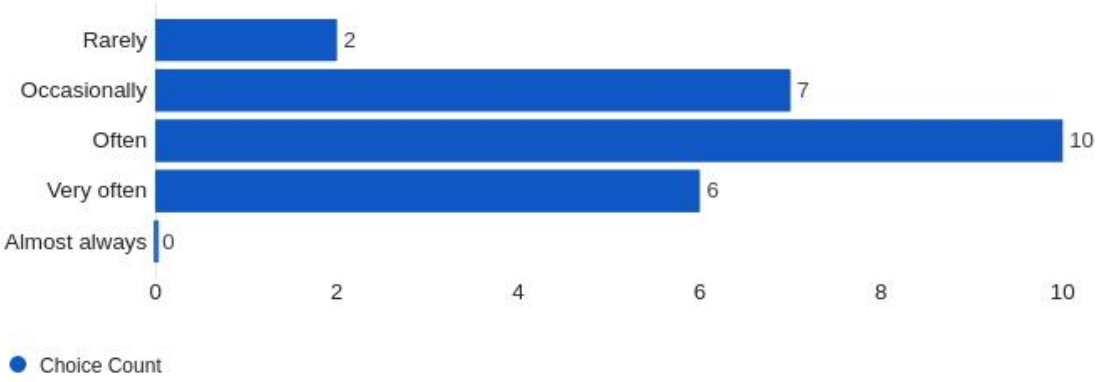
7.3.4 Frequency of encountering technical terms and acronyms

2. How often do you come across technical terms or acronyms when browsing the internet?



Answer	%
Rarely	12.00%
Occasionally	24.00%
Often	24.00%
Very often	32.00%
Almost always	8.00%
Total	100%

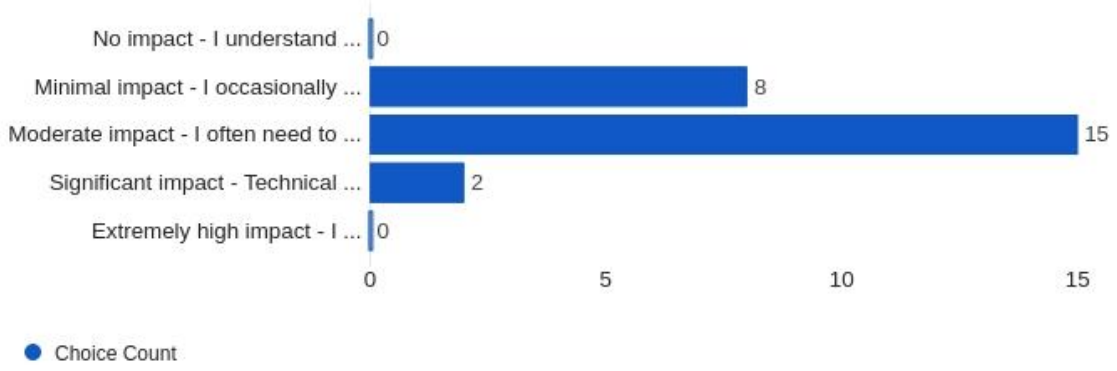
3. How often do you encounter technical terms or acronyms that you don't understand while browsing the internet?



Answer	%
Rarely	8.00%
Occasionally	28.00%
Often	40.00%
Very often	24.00%
Almost always	0.00%
Total	100%

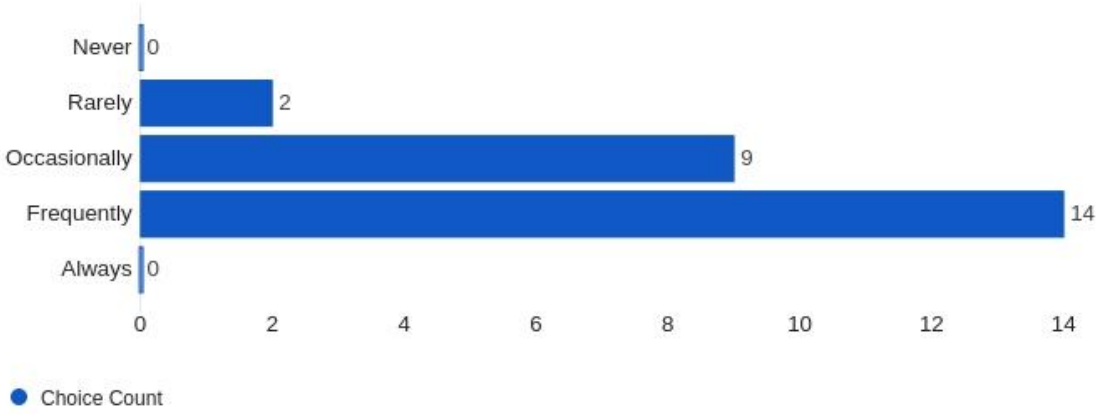
7.3.5 Impact of technical terms and acronyms on comprehension

4. To what extent do technical terms or acronyms impact your comprehension of online articles?



Answer	%
No impact - I understand technical terms and acronyms	0.00%
Minimal impact - I occasionally need to look up terms or acronyms, but it doesn't significantly slow my understanding	32.00%
Moderate impact - I often need to look up terms or acronyms, which somewhat slows my understanding	60.00%
Significant impact - Technical terms or acronyms frequently hinder my comprehension and require me to spend more time understanding the content	8.00%
Extremely high impact - I struggle to understand articles with technical terms or acronyms and often abandon reading them	0.00%
Total	100%

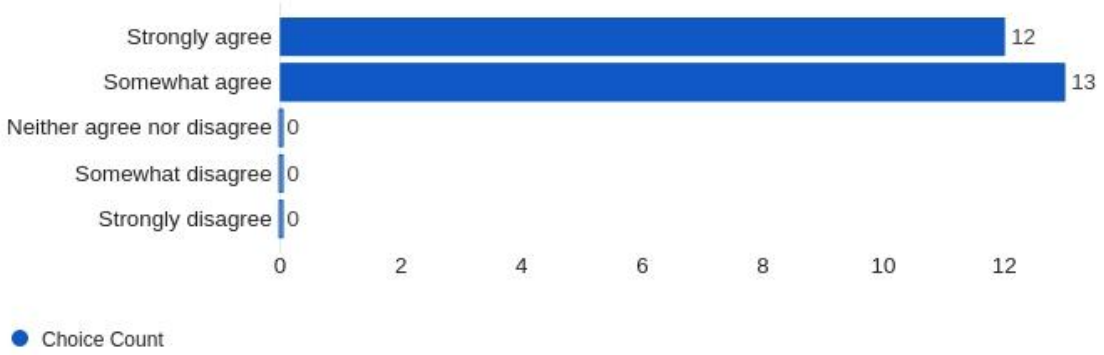
5. How often do you seek external help (e.g., through a search engine) to clarify unfamiliar technical acronyms or abbreviations you encounter online?



Answer	%
Never	0.00%
Rarely	8.00%
Occasionally	36.00%
Frequently	56.00%
Always	0.00%
Total	100%

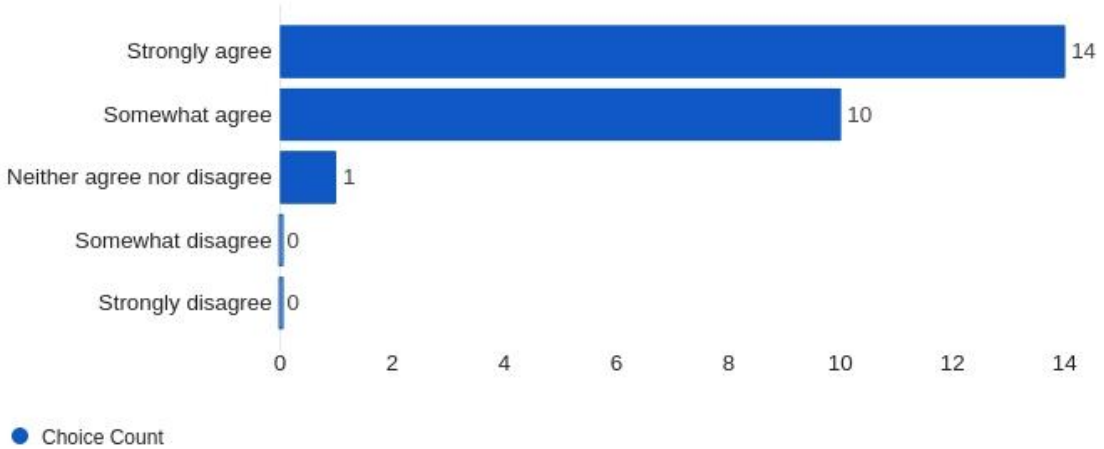
7.3.6 Potential benefits of a contextual help tool

6. Would a browser extension that offers explanations for technical terms and acronyms within online articles enhance your comprehension of the content?



Answer	%
Strongly agree	48.00%
Somewhat agree	52.00%
Neither agree nor disagree	0.00%
Somewhat disagree	0.00%
Strongly disagree	0.00%
Total	100%

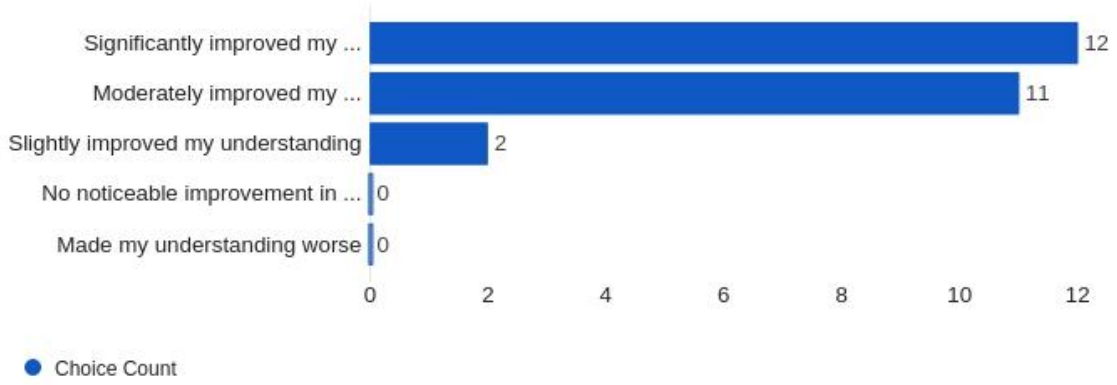
7. Do you believe that a contextual help tool, which provides definitions and explanations for technical terms and acronyms, would improve your reading efficiency for online articles containing such language?



Answer	%
Strongly agree	56.00%
Somewhat agree	40.00%
Neither agree nor disagree	4.00%
Somewhat disagree	0.00%
Strongly disagree	0.00%
Total	100%

7.3.7 Impact of the contextual help browser extension on understanding technical terms and acronyms

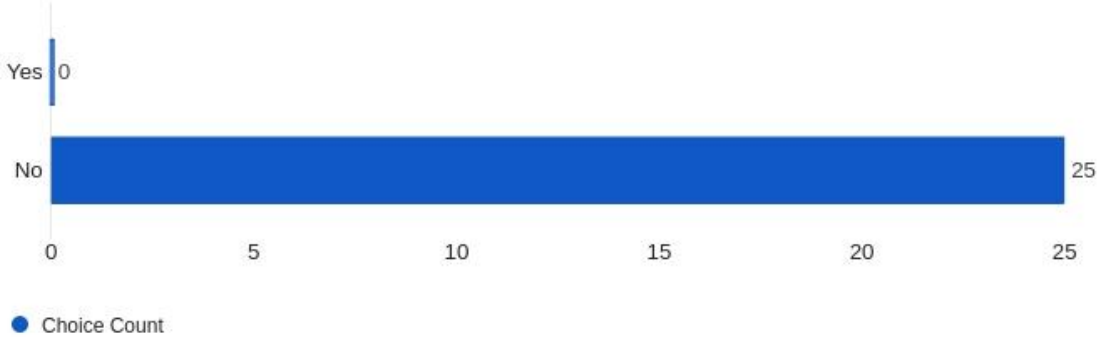
8. To what extent did the contextual help browser extension improve your understanding of technical terms and acronyms while reading the provided online article?



Answer	**%**
Significantly improved my understanding	48.00%
Moderately improved my understanding	44.00%
Slightly improved my understanding	8.00%
No noticeable improvement in understanding	0.00%
Made my understanding worse	0.00%
Total	100%

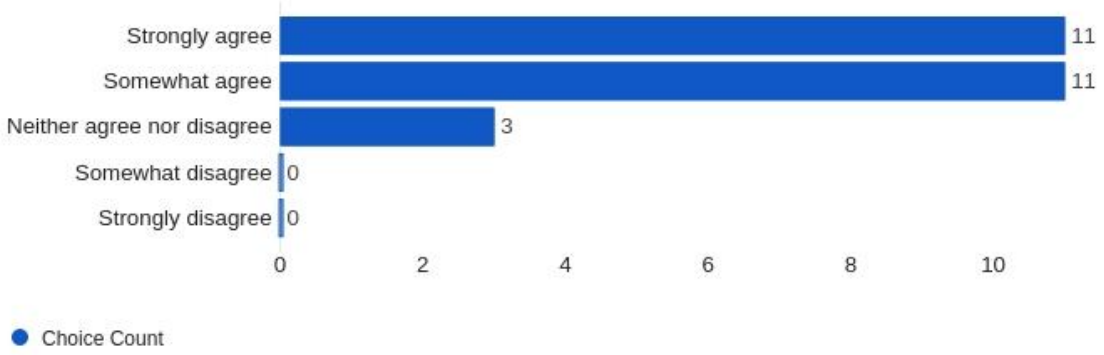
7.3.8 User experience and clarity of definitions

9. Did the tooltips provided by the contextual help browser extension distract you or negatively impact your reading experience in any way?



Answer	%
Yes	0.00%
No	100.00%
Total	100%

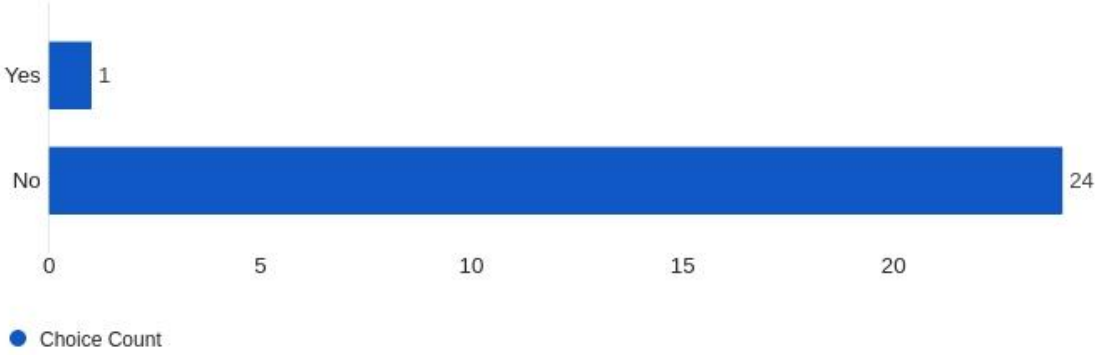
10. Were the definitions and explanations provided by the tooltips clear and easy to understand?



Answer	%
Strongly agree	44.00%
Somewhat agree	44.00%
Neither agree nor disagree	12.00%
Somewhat disagree	0.00%
Strongly disagree	0.00%
Total	100%

7.3.9 Perceived effects on website performance

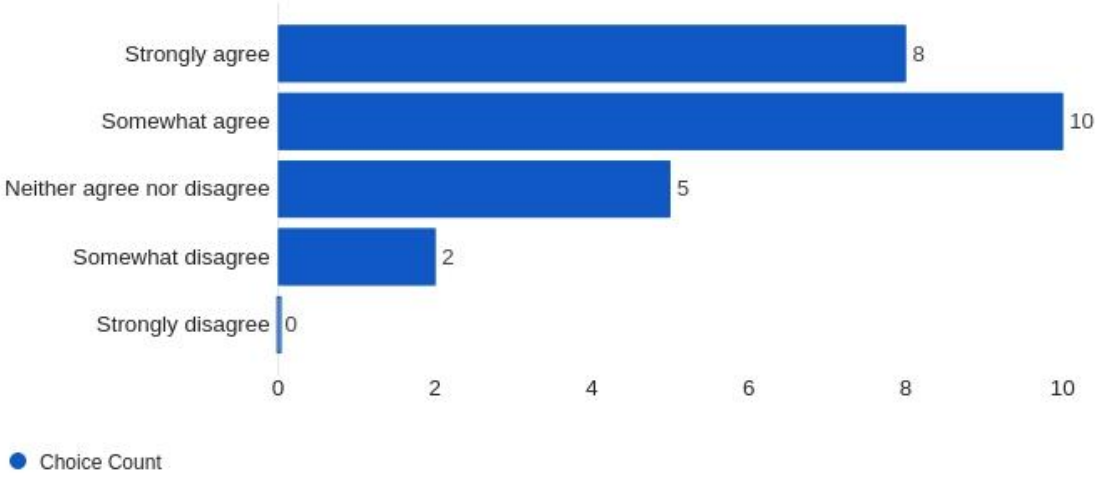
11. Did you feel the browser extension impacted the loading speed or performance of the websites you visited?



Answer	%
Yes	4.00%
No	96.00%
Total	100%

7.3.10 Inclination to explore additional technical content

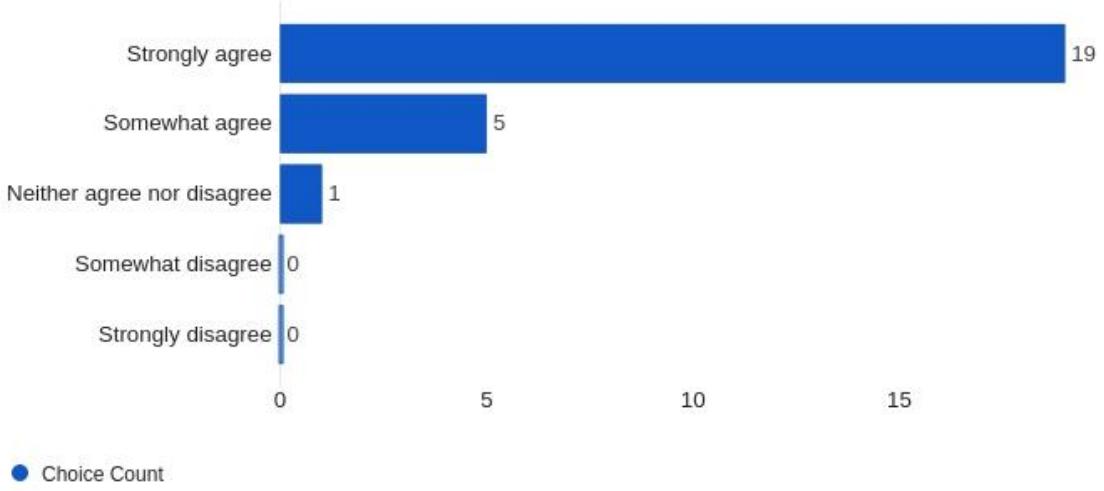
12. After using the contextual help browser extension, do you feel more inclined to explore additional technical content online, knowing that you have a tool to assist you with understanding?



Answer	%
Strongly agree	32.00%
Somewhat agree	40.00%
Neither agree nor disagree	20.00%
Somewhat disagree	8.00%
Strongly disagree	0.00%
Total	100%

7.3.11 Time-saving aspect of the contextual help browser extension

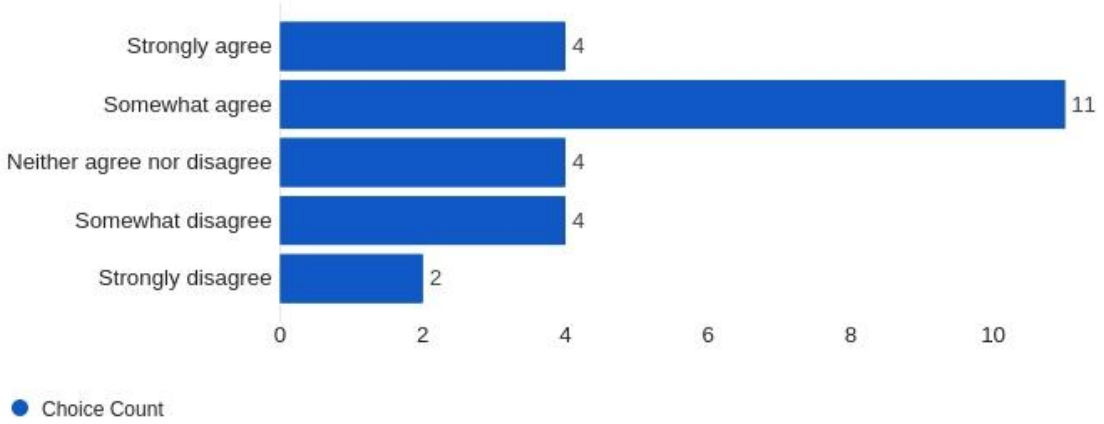
13. Did you find that the contextual help browser extension saved you time when trying to understand technical terms or acronyms, as compared to searching for their meanings manually?



Answer	%
Strongly agree	76.00%
Somewhat agree	20.00%
Neither agree nor disagree	4.00%
Somewhat disagree	0.00%
Strongly disagree	0.00%
Total	100%

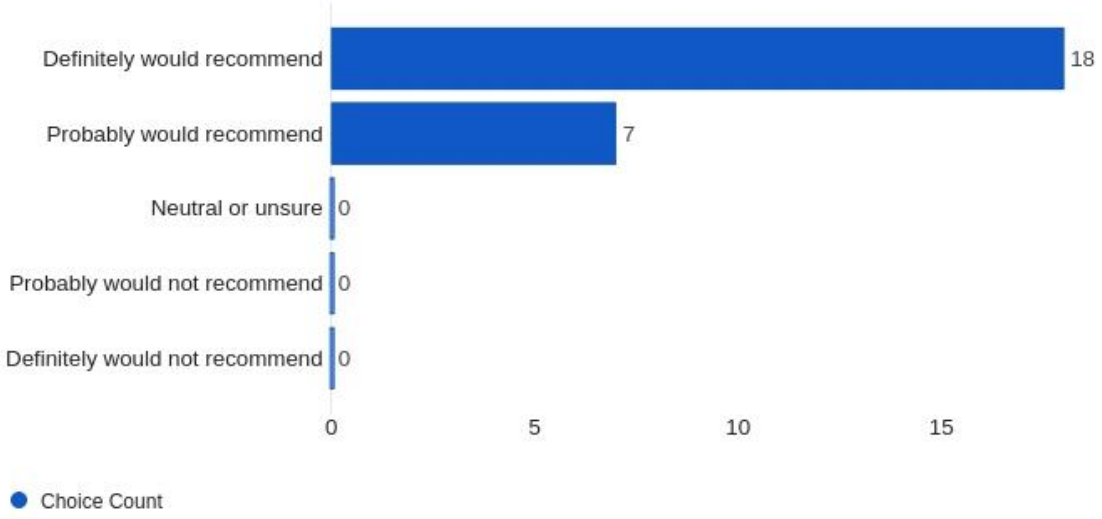
7.3.12 Additional resources and recommendations

14. If the browser extension were to offer additional resources, such as links to related articles or tutorials for further learning, would you find that helpful?



Answer	%
Strongly agree	16.00%
Somewhat agree	44.00%
Neither agree nor disagree	16.00%
Somewhat disagree	16.00%
Strongly disagree	8.00%
Total	100%

15. Would you recommend a contextual help browser extension to others who may have difficulty understanding technical terms and acronyms?

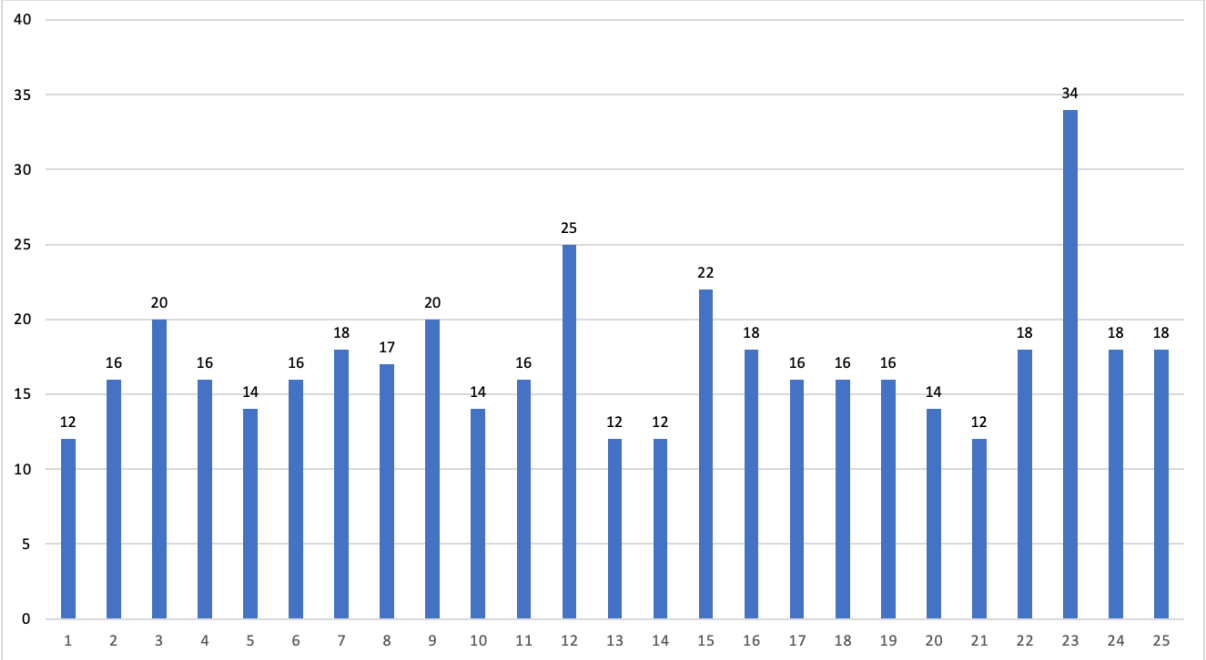


Answer	**%**
Definitely would recommend	72.00%
Probably would recommend	28.00%
Neutral or unsure	0.00%
Probably would not recommend	0.00%
Definitely would not recommend	0.00%
Total	100%

7.3.13 Time Taken for Manual Information Search

A simple approach was adopted using a stopwatch to measure the time it took each participant to search for the definition of the "**CPU**" acronym using Google. The timing began when participants started their search process and stopped when they indicated they had found the definition. This measurement was recorded in seconds. This exercise aimed to understand the time it typically takes an individual to locate such information without the help of the contextual help browser extension. This measurement provides valuable insight into the effectiveness and efficiency of the tool when compared to traditional search methods.

16. How long (in seconds) did it take each participant to search for the definition of the “**CPU”** acronym using Google?



N	**Valid**	25
	Missing	0
Mean		17.2000
Median		16.0000
Std. Deviation		4.71699
Range		22.00
Minimum		12.00
Maximum		34.00

7.3.14 Performance Metrics of the Contextual Help Extension

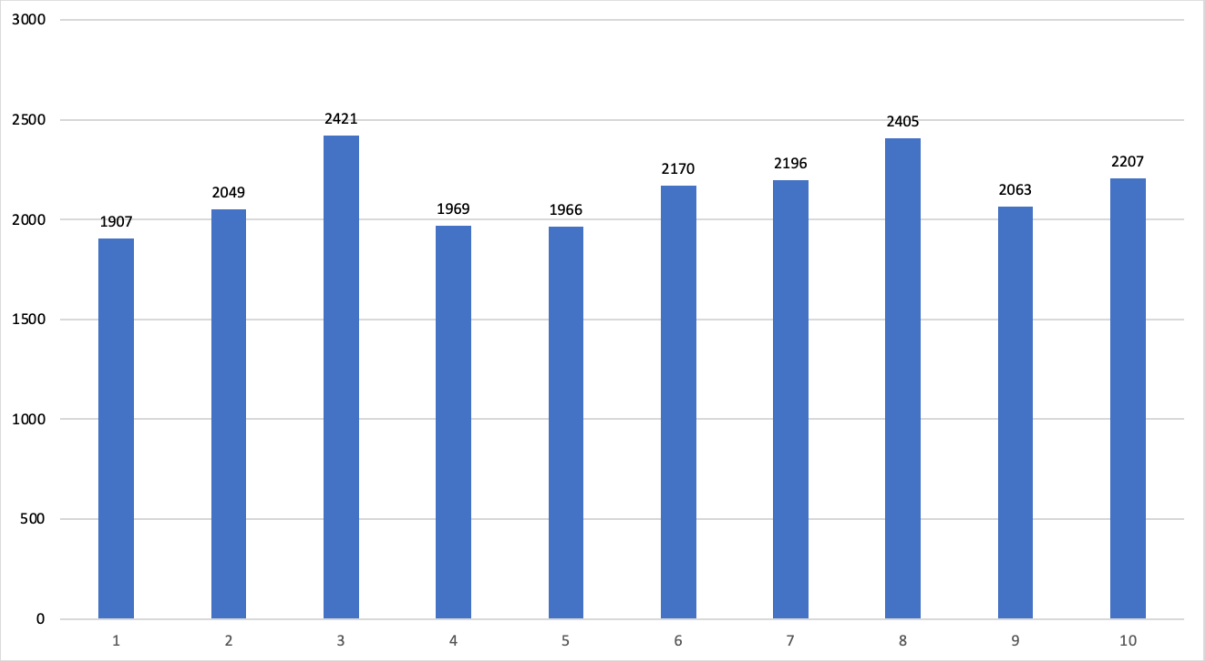
In order to evaluate the performance of the execution of the contextual help tool the performance.now() method was used. This method is a part of the Performance Web API that provides precise timestamps, enabling developers to measure the time taken for different application parts to execute.

The performance.now() method returns a timestamp, measured in milliseconds, accurate to one-thousandth of a millisecond. It represents the time passed since the time of origin, typically when the function was loaded. In the contextual help tool, the timer starts when the "load" event is emitted from the webpage. This is a signal that the webpage's contents have been successfully loaded in the user's browser and that the contextual help tool can start the execution of the sequence of steps for the tooltip appending functionality. Exactly after the load event, there is a constant variable "const start = performance.now()". At the end of the main function execution, the timer was stopped with const endDictionary = performance.now(). The difference between these two timestamps, const durationDictionary = endDictionary - start, gives the time taken for the execution of the function in milliseconds. This value was then logged to the console with console.log('Execution time dictionary: ${durationDictionary} milliseconds') and then saved in a spreadsheet. In the same way a timestamp was created for the definitions retrieval from ChatGPT with a variable const endGPT = performance.now(). The timer then stopped const durationGPT = endGPT - start and the findings were recorded.

Using performance.now() in this way allows a precise measurement of the function's execution time and provides valuable data for assessing performance. It is an essential step in performance optimization, as it offers insights into how the function behaves in different environments and can help identify potential areas for improvement.

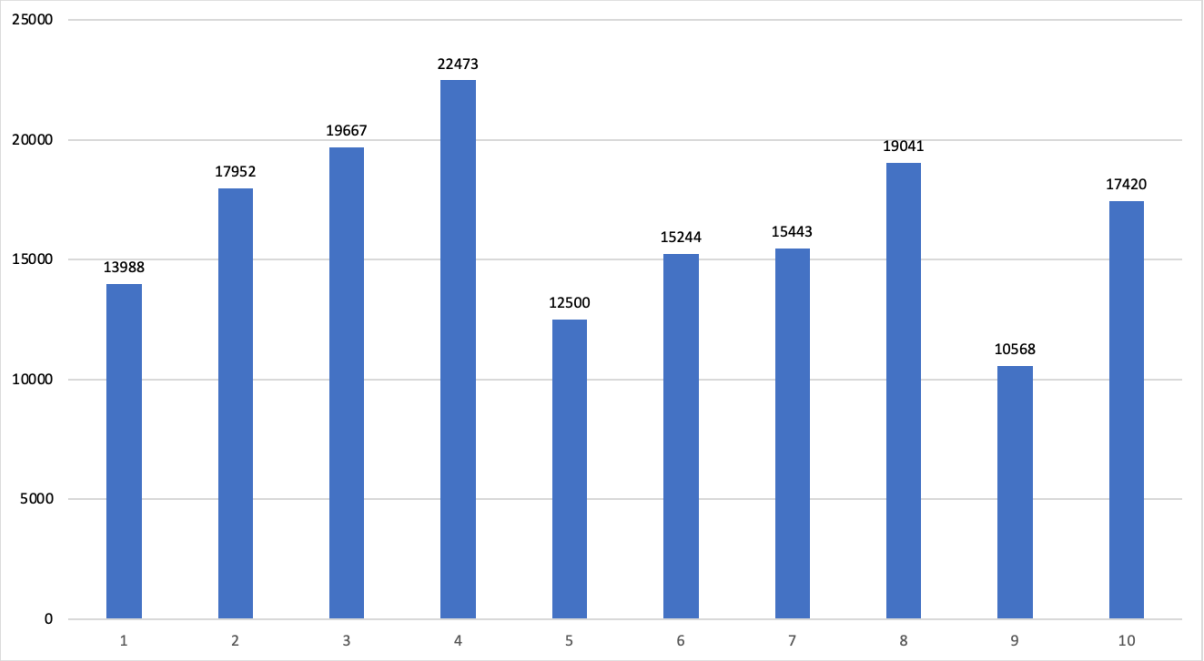
For the calculation of descriptive statistics (Mean, Median, Std deviation, Min, Max) the IBM SPSS software was used.

17. Time taken (in milliseconds) for the contextual help extension to append tooltips with definitions from the dictionary [10 executions].



N	**Valid**	10
	Missing	0
Mean		2135
Median		2116
Std. Deviation		178
Range		514
Minimum		1907
Maximum		2421

18. Time taken (in milliseconds) for the contextual help extension to append tooltips with definitions from the OpenAI - ChatGPT [10 executions].



N	**Valid**	10
	Missing	0
Mean		16429
Median		16431
Std. Deviation		3578
Range		11905
Minimum		10568
Maximum		22473

7.4 Qualitative Data Analysis

As mentioned in the research methodology section, thematic analysis has been used in this study to identify common patterns within the feedback provided by users on the contextual help browser extension. Given the type of this study, the use of thematic analysis allows us to translate qualitative data into meaningful conclusions, providing valuable insights into user experiences. In total, 25 interviews were conducted and based on the feedback gathered, it grouped into the identified themes below.

- **Clarity and depth of definitions**
  Participants suggested that an option for different levels of definitions would be helpful. Having the option to set different levels of definitions would allow them to configure it precisely for their needs and also it will allow them to suggest the tool to more people. Either to people with limited digital skills or those more technologically advanced. Additionally, a common comment was that they prefer simpler, shorter and more meaningful definitions. After asking them which type of definitions they preferred between the dictionary and the AI definitions, around 70% preferred the definitions from AI mostly because the definitions were more straight to the point. On the other hand, definitions from the dictionary provide more details that are not always required.

- **User interface**
  Comments around the presentation of the definition and the styling, were also recorded. Most participants found the tooltip good enough for getting the definition, and the underline that appears under an acronym is enough to make them aware of the functionality. When hovering over it, the cursor changing to a question mark is helpful because it distinguishes the contextual help functionality from other features, for example hyperlinking. However, once again, users prefer to have options. There were comments for more control over the styling options, such as the ability to increase the font size or the colour of the text. Finally, one interesting comment was that the definition in the tooltip could have two parts, a title which should be the acronym or abbreviation expanded and the actual definition separated by a new line below. As the user explained, this has the benefit that the user in many cases will not have to read the full definition because the disambiguated acronym would be enough to get the meaning.

- **Additional information or resources**
  When participants were asked to comment if they wanted to see something more in the tooltip, most of them thought that just the definition was enough; however, a few of them suggested that the definitions could include the source or a link for further reading. Another interesting comment was that, wherever possible, images could be appended in the tooltip to aid comprehension.

During the interviews, all the participants expressed the helpfulness of the contextual help browser extension in improving the understanding of technical terms and acronyms while browsing online articles. Their comments can be further summarised into:

- **Improved comprehension**
  Participants found the tool made understanding unknown acronyms or abbreviations in the articles they read easier. Others noted that even if they knew what a specific acronym was about, the tool allowed them to clarify the meaning and get a better understanding quickly.

- **Motivation to explore more online technical articles**
  Several participants, especially those older, shared that the tool encouraged them to explore more technical content online, knowing they have an aid to assist them in understanding technical terms and acronyms.

- **Improved reading efficiency**
  Most participants commented on how helpful it is that the tool, which provides simple definitions on the spot, improves their reading efficiency. The time they save by using the tool was greatly appreciated, especially by users who read technological-related articles more often and have to search for the meanings of acronyms more frequently. Opening a new browser tab and searching in a search engine for a definition, one acronym at a time, is something not needed anymore.

The below selection of comments shows the appreciation of the tool and also includes suggestion comments by study participants for future versions.

- **Comments about the improved comprehension and reading efficiency**
  - "This is a great idea, I always ask my techy friend on WhatsApp."
  - "We have lots of acronyms in my work, and I need to search often. I don't know how you are doing it, but it is great. Can you do it for codes of car parts?"
  - "I feel like I can consume online content more effectively with this tool. The in-line explanations keep my understanding high. No constant back-and-forth with Google to translate technical terms."
  - "My understanding of technical articles has improved, and it feels like I'm reading more smoothly."
  - "I actually like that I don't have to pause and look up terms; the info is right there."
  - "I can see how helpful this tool is because I get the full picture without needing external help."
  - "Maintaining focus is much easier when explanations are right there. The reading flow isn't disrupted"
  - "It's like having an instant translator for tech language."
  - "The time saved on researching tech terms was surprising. It's a big boost to my reading efficiency."
  - "I am not good with computers and I always ask my husband for things I don't know, this tool is helpful for me."

- **Comments about the willingness to read more technology related articles if they have a tool like this**

    - "I don't know much about tech, but this tool would make me want to read more about it."
    - "Of course I will click on more links of tech articles, I will be the smartest among my friends!"
    - "This tool would encourage me to dive in more often in tech articles."
    - "The extension would help me to explore tech articles I'd usually skip."
    - "Now that I've used this tool, I feel more motivated to click on tech news links and blogs."

- **Comments about the provided definitions and the UX:**
    - "An option for different levels for definitions."
    - "I don't know English very well, simple definitions would be better for me"
    - "Styling changes, make it a little bit more obvious for attention. Underlining an acronym is not enough."

Additionally during the interview process, an interesting pattern was observed: a difference between the participants' perceived and actual digital skills. Many participants claimed to have a proficient understanding of technology and classified themselves as intermediate users. However, their responses to basic queries about everyday tech devices, such as computers and mobiles, were incomplete or incorrect. That could suggest overestimating one's technical skills and digital competence.

While these individuals may be competent at using technology for their daily needs, they appear not to have a good understanding of widely used technical terminology and concepts. Being digitally competent or literate, in the sense of being able to operate technological devices, does not always correlate to a good level of comprehension or ability to understand underlying technical terms and principles.

This gap between self-perception and reality might have implications for digital education, as it shows the need to investigate different approaches to effectively improve digital literacy.

7.4.1 Positive impact on specific target groups

The narrative of one particular participant offers a case study that illustrates the potential value and impact of the contextual help browser extension in supporting unique needs. This participant is a young male in his twenties studying cybersecurity in college for further education, self-identified as being on the autism spectrum and experiencing short-term memory difficulties. These conditions pose distinct challenges to his learning experience, particularly when dealing with online educational materials containing technical terms.

The participant described his current approach to unfamiliar technical terms, including writing them down or taking screenshots. He then searches in a new browser tab to get their meaning. His short-term memory condition necessitates this ineffective method. If he directly opens a new tab to search for the term, he often forgets what he intended to search for. Therefore, this adds additional effort and complexity to his learning process.

Upon learning about the capabilities of the contextual help browser extension, the participant expressed enthusiasm and a strong desire for early access. He believed this tool could mitigate his challenges during his studies. The extension would assist his learning process by eliminating the need to note down and manually search for each unfamiliar term he encounters in his reading. Instead, immediate, in-context definitions would be available, facilitating seamless comprehension of the material and aiding memory retention.

The participant's anticipation about the tool's potential benefits extends to its potential impact on others with similar conditions, such as his sister. This case illustrates the real-world implications of this research and how the contextual help browser extension might enhance digital literacy and learning experiences. It highlights the significance of creating digital tools that cater the user needs and empower online learning environments.

7.5 Summary of findings

The findings of this research project reveal that the contextual help tool can potentially improve the comprehension and reading efficiency of users with low levels of digital skills when visiting online articles with technology acronyms and abbreviations.

Data from the quantitative research reveal significant reduction in users' time seeking definitions through external searching, for example, search engines, and improvement in reading efficiency is observable. Moreover, the tool's performance when appending the tooltips with the definitions in DOM is satisfactory, as the measurements show.

Qualitative findings also confirmed the tool's usefulness after the vast majority of the participants found the tool helpful. Most users appreciated the convenience of having the definitions of the acronyms readily available without having to search manually. Despite a small percentage of requests for different levels of definitions, simpler language, or styling changes, most of the users found that the definitions provided were clear, and in general, the overall user experience was positive.

In conclusion, the findings show the potential of the contextual help browser extension to assist users of various levels of digital illiteracy. However, further exploration is required regarding the presentation and content of the definitions.

8. Discussion

8.1 Answers to research questions

Research Question 1

Can a contextual help browser extension improve the comprehension of online articles that use technical terms among a group of digitally illiterate internet users?

Based on the mixed-methods approach of the study, a contextual help browser extension can improve the comprehension of online articles that use technical terms among a group of digitally illiterate internet users. Both the quantitative and qualitative data provided evidence that the browser extension helped these users understand technical terms and acronyms they encountered while browsing the internet. This point was also confirmed during the interviews, following the feedback from the participants, where a significant number found the tool beneficial for their comprehension of technical content. An overwhelming majority of participants found that the contextual help browser extension significantly (48%) or moderately (44%) improved their understanding of technical terms and acronyms within online articles. Additionally, all participants agreed, to some extent, that the extension would enhance their comprehension of content (48% strongly agreeing and 52% somewhat agreeing). The correlation between contextual help browser extension usage and improved comprehension is evident, establishing the utility of such a tool among digitally illiterate users.

Research Question 2

Does the contextual help tool positively impact the reading time for digitally illiterate internet users?

The data collected suggested that the contextual help tool can impact the reading time for digitally illiterate internet users, but not in the way one might expect. The tool successfully reduced users' time to understand unfamiliar technical terms or acronyms, as they did not need to conduct manual searches for definitions. Most users (76% strongly agree and 20% somewhat agree) found that the contextual help browser extension saved them time when trying to understand technical terms or acronyms, compared to manually searching for their meanings. This data indicates a strong correlation between the extension's use and the reading time's efficiency. Additionally, the quantitative data support this inference, with the time taken by the extension to provide definitions (in milliseconds) being significantly lower than the average time participants took to manually search for the definition of the CPU acronym on Google. However, it did not significantly speed up the overall reading time, likely because users spent time interacting with the tool's features. The tool, therefore, made the reading process more efficient rather than faster.

Research Question 3

How can AI assist a browser extension in the classification of websites?

AI can significantly assist a browser extension in classifying websites by enabling the extension to recognize and interpret technical terms and acronyms accurately and in real-time. AI's ability to process and analyse the amounts of text content that an average online technological article has in a short period also means that it can handle almost real-time detection and interpretation of technical language across various websites, making the contextual help tool highly efficient. Using online machine learning services such as Google NLP and OpenAI ChatGPT, shows that the existing AI algorithms can successfully identify patterns and context to determine which terms require tooltips with definitions. While the time taken by AI is greater than simply retrieving definitions from a dictionary, it is still significantly less than the time users take to search for definitions manually. That suggests that AI can play a crucial role in enhancing the efficiency of a browser extension and, by extension, the user's browsing experience. However, the potential of AI in further learning from user interactions and improving the accuracy and relevance of the tooltips over time is something that could be explored in future research.

8.2 Relationship to findings

The study data, supported by a 92% majority of participants, indicates a link between the contextual help browser extension with an improved understanding of technical terms among digitally illiterate users. Most participants reported that the tool significantly (48%) or moderately(44%) increased their comprehension of technical language in online articles.

In addition, the data indicates that the browser extension boosts reading efficiency. A substantial 96% of participants confirmed that the tool saved time when deciphering technical terms compared to the time they manually searched for definitions. That aligns with the quantitative data, demonstrating that the extension (with an average tooltip append time of 2135 milliseconds for dictionary definitions and 16429 milliseconds for AI-generated definitions) provided definitions significantly faster than users manually searching for them.

As for incorporating AI into the browser extension, the study reveals that AI can augment the functionality of a browser extension. Despite AI taking longer to append tooltips with definitions than dictionary retrieval, it remained significantly quicker than users manually searching for definitions. However, a tiny proportion of users (1 out of 25) noticed an impact on the loading speed or performance of the websites visited due to the browser extension. That could not be possible due to the asynchronous operation of the contextual help tool.

In conclusion, the study's findings support the initial hypotheses that a contextual help browser extension can improve comprehension and reading efficiency among digitally illiterate users. Moreover, AI can significantly impact this process.

8.3 Implications

The implications of this study are primarily how contextual help tools can improve digital literacy skills and enhance the online experience. Secondly the study demonstrates how to integrate AI technology with a browser extension..

The study highlights the importance of contextual help browser extensions in bridging the gap between digital literacy levels among internet users. With 92% of participants reporting an improved understanding of technical abbreviations and acronyms, the tool can aid users who often encounter and struggle with technical language online.

Since 96% of users said the extension saved time in understanding technical terms, such tools can enhance reading efficiency and overall productivity. That is particularly relevant in the current era, where digital content consumption is high.

The integration of AI into the browser extension, despite being slower than the dictionary based tooltips (16429 milliseconds vs 2135 milliseconds), was still significantly faster than manual searches (average of 17.2 seconds for a single acronym). That highlights the potential of AI in improving the performance and utility of digital tools.
Additionally, the fact that all participants reported no negative impact on their reading experience due to tooltips and 88% of users were more inclined to explore additional technical content online post-use underscores the potential of such extensions to improve the user experience and encourage learning.

Finally, the response to the potential offer of additional resources, such as links to related articles or tutorials (60% agreed or strongly agreed), indicates an appetite for supplementary learning resources. That suggests that extensions could be further developed to provide comprehensive learning support.

These findings can have implications for broader digital literacy initiatives. They emphasise the potential of browser extensions with the help of Artificial Intelligence to assist users in navigating the increasingly complex digital landscape.

8.4 Suggested improvements - Future versions

During the interviews, participants shared invaluable feedback, and benchmarks provided crucial metrics, both of which outline the path for future improvements.

The performance subsection in the quantitative section highlights that appending tooltips with definitions from OpenAI - ChatGPT in the contextual help extension consumes significantly more time (in milliseconds) than using definitions from the dictionary. This increase in time is a predictable outcome due to the complex nature of the technology. To optimise performance, future iterations of the contextual help tool can store acronyms or abbreviations along with their definitions in the dictionary. This way, when a user revisits a site with previously encountered acronyms, the tool retrieves the definitions from the dictionary rather than OpenAI services, a process that metrics prove is multiple times faster.

Improvements in the UI/UX form is the next significant area of focus. User feedback suggests a preference for customizable tooltip styles, such as text size, colour, borders, and shadows. For instance, many older users would appreciate an option to increase the font size. Introducing such simple customization options should enhance user satisfaction.

Expanding the number of acronyms and definitions in the dictionary could improve the usefulness of the contextual help tool. This expansion could occur through AI assistance, as previously suggested, or by incorporating definitions from a variety of sources. While the current version of the tool sources its dictionary definitions from Wikipedia, many other datasets are freely available online. Adding these datasets to the dictionary should improve the detection of technological acronyms and abbreviations.

Despite the potential for a big list of improvements, one impactful upgrade would involve offering tooltips in other languages to serve a broader audience. This could be achieved through online translation services or by creating dictionaries in different languages. The browser can easily detect localization, and upon successful detection, the tool can utilise the appropriate dictionary.

Lastly, extending the tool's availability to domains outside technology based on user feedback would make it valuable to a wider range of users. Many users reported that a tool like this could prove useful in finance, law, medicine, car manufacturing and more. Minimal modifications to website classification and the addition of the right dictionary datasets could allow the contextual help tool to serve a broader audience across various domains.

8.5 Questions raised and suggestions for further research

The study's findings show the impact of the contextual help browser extension on enhancing digital literacy and the potential areas of improvement and future exploration. The promising results also open up a series of questions that subsequent studies could investigate to optimise the advantages of such tools.

The contextual understanding topic needs to be explored more deeply in future studies. During the testing phase of the contextual help tool some false positive detections, primarily from the dictionary definitions were identified . The current logic for using definitions from the dictionary, unfortunately, overlooks context. For instance, the word "post" can carry multiple meanings in a sentence, such as "I will send it by post." However, the tool inaccurately attaches a tooltip with the definition "Power On Self Test". This issue doesn't occur with the detection mechanism of OpenAI - ChatGPT. Consequently, an AI algorithm should be implemented directly in the contextual help tool to have tooltip logic enhanced without relying on remote services.

We should also consider extending the tool to other platforms. This research concentrates on delivering contextual help via a browser extension and assisting users when they browse technological online articles. What challenges would I encounter if I were to transform this project into a tool that uses the same database and exact detection mechanisms but

operates on other platforms? One possibility is a plugin for Microsoft Word or Outlook. Many users spend significant time reading offline documents, especially in corporate environments. How might I employ a contextual help tool in this kind of software?

The final question worth investigating concerns tool enhancement and user contribution. The current version of the tool doesn't have a mechanism to gather user feedback. If I were to develop a feature that allows users to suggest changes about the tool and contribute definitions for new acronyms or abbreviations, it would encourage communication and an open-source culture. While designing and implementing a strategy for this could present challenges due to factors like licensing, it could yield significant benefits for the tool's future if done well.

By studying these questions and finding answers, an even more effective tool for promoting digital literacy could be developed.

8.6 Considerations on the sample size, the statistical significance, and the potential bias

The sample size of this study, consisting of 25 interviews, might be a limiting factor for the ability to generalise the results to a broader population. The reasons behind the small sample size are practical constraints, including participants' unavailability, unwillingness to engage in a lengthy interview. Furthermore, the participant selection criteria, such as basic digital skills and interest or need in reading online technology articles, further constrained the accessible sample pool. As such, the statistical significance of the results needs to be considered, and the interpretation should be indicative rather than definitive.

Additionally, bias may have been introduced into the sample selection due to self-selection bias. Those with a perceived lack of engagement with technology may have been less likely to participate, potentially skewing the sample towards those with a higher interest in or need to understand technical content.

Despite these limitations, the data gathered provide valuable insights into the experiences of the target demographic and the potential benefits of the tool being investigated.

8.7 Summary

The study demonstrates the effectiveness of a contextual help browser extension, enhanced by AI, in improving digitally illiterate users' comprehension of technical terms and reading efficiency. It also shows areas for improvement, including definition retrieval performance optimization, definitions dictionary expansion, UI/UX enhancements, and support for languages other than English.

Finally, its potential use across domains such as finance, law, and medicine indicates the tool's broad applicability. The study's findings highlight the value of such digital literacy tools and pave the way for further exploration and development.

9. Conclusion

This research explored the potential impact of a contextual help browser extension on comprehending technical acronyms and abbreviations within online technology articles among digitally illiterate users. It also aimed to understand how AI could help in providing contextual help in an efficient and meaningful way.

The study's findings demonstrated the tool's usefulness in enhancing comprehension of technical content and improving reading efficiency. The tool was well-received by the participants, who found the immediate, in-context definitions beneficial for their reading experience. The data also showed a reduction in the time taken to find definitions, indicating an improved efficiency in reading.

The study also identified areas for further improvement and customization. The feedback received is insightful for future iterations of the tool and underlines the importance of a user-centric approach in designing and developing digital aids.

One remarkable finding was the potential value and impact of the contextual help browser extension in supporting unique needs. The case study of a participant with short-term memory difficulties demonstrated how such a tool could address learning challenges and reinforce the significance of creating inclusive digital tools that empower online environments.

Exploring AI's role in this research highlighted current AI technologies' potential and limitations. Despite the longer times for retrieving AI definitions, users generally preferred them, revealing a possibility for AI to augment the browser extension's functionality.

To conclude, the research provided insights into the potential of a contextual help browser extension assisted by AI to support users with varying levels of digital literacy. In an increasingly digitised world, developing user-focused tools is critical in ensuring all users can navigate and make sense of the vast array of online content. This study marks a step in that direction.

Video demonstration

A video demonstration is available in this link: https://youtu.be/FiYI-itulUQ

10. Reference List

Accenture (2020). *Bridging the gap – Ireland's digital divide*. [online] Accenture.com. Available at: https://www.accenture.com/ie-en/insights/local/digital-divide [Accessed 20 Aug. 2023].

Anderson, J., Rainie, L. and Vogels, E. (2021). *Experts Say the 'New Normal' in 2025 Will Be Far More Tech-Driven, Presenting More Big Challenges*. [online] Pew Research Center: Internet, Science & Tech. Available at: https://www.pewresearch.org/internet/2021/02/18/experts-say-the-new-normal-in-2025-will-be-far-more-tech-driven-presenting-more-big-challenges/ [Accessed 20 Aug. 2023].

Appelman, A. (2019). 'ALPHABET SOUP': Examining Acronym and Abbreviation Style in Headlines. *Journalism Practice*, 14(7), pp.1–16. doi:https://doi.org/10.1080/17512786.2019.1642125.

Appelman, A. (2021). Written in Code: Exploring the Negative Effects of Acronyms in News Headlines. *Journalism Practice*, 16(7), pp.1–17. doi:https://doi.org/10.1080/17512786.2020.1867622.

Atkinson, M. and Van der Goot, E. (2009). Near real time information mining in multilingual news. *Proceedings of the 18th international conference on World wide web - WWW '09*, [online] pp.1153–1154. doi:https://doi.org/10.1145/1526709.1526903.

Balakrishnan, V. and Lloyd-Yemoh, E. (2014). *Stemming and Lemmatization: A Comparison of Retrieval Performances - Volume 2 Number 3 (Aug. 2014) - LNSE*. [online] www.lnse.org. Available at: http://www.lnse.org/show-34-165-1.html [Accessed 20 Aug. 2023].

BBC (2014). Computer AI passes Turing test in 'world first'. *BBC News*. [online] 9 Jun. Available at: https://www.bbc.com/news/technology-27762088 [Accessed 20 Aug. 2023].

Chetty, K., Qigui, L., Gcora, N., Josie, J., Wenwei, L. and Fang, C. (2018). Bridging the digital divide: measuring digital literacy. *Economics: The Open-Access, Open-Assessment E-Journal*, 12(23). doi:https://doi.org/10.5018/economics-ejournal.ja.2018-23.

Chilana, P.K., Ko, A.J. and Wobbrock, J.O. (2012). LemonAid. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [online] doi:https://doi.org/10.1145/2207676.2208620.

Czaja, I. and Urbaniec, M. (2019). Digital Exclusion in the Labour Market in European Countries: Causes and Consequences. *European Journal of Sustainable Development*, 8(5), p.324. doi:https://doi.org/10.14207/ejsd.2019.v8n5p324.

Dai, Y., Karalis, G., Kawas, S. and Olsen, C. (2015). Tipper. *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, [online] pp.1773–1778. doi:https://doi.org/10.1145/2702613.2732796.

DebugBear (2020). *Counting Chrome Extensions – Chrome Web Store Statistics | DebugBear*. [online] www.debugbear.com. Available at: https://www.debugbear.com/blog/counting-chrome-extensions [Accessed 20 Aug. 2023].

Digital Government. (2021). *Digital inclusion*. [online] Available at: https://www.nldigitalgovernment.nl/overview/accessibility/digital-inclusion-everyone-must-be-able-to-participate/ [Accessed 20 Aug. 2023].

Europa (n.d.). *Digital inclusion | Shaping Europe's digital future*. [online] digital-strategy.ec.europa.eu. Available at: https://digital-strategy.ec.europa.eu/en/policies/digital-inclusion [Accessed 20 Aug. 2023].

Gablerjan, N. (2015). The Weird Science of Naming New Products. *New York Times*. [online] Available at: https://www.mendeley.com/catalogue/42e2c45f-4c85-3b45-bdf0-40ec9871db47 [Accessed 20 Aug. 2023].

Gilmore, G.C., Tobias, T.R. and Royer, F.L. (1985). Aging and similarity grouping in visual search. *Journal of Gerontology*, [online] 40(5), pp.586–592. doi:https://doi.org/10.1093/geronj/40.5.586.

Gilster, P. (1997). *Digital literacy.* [online] New York ; Chichester: Wiley Computer. Available at: https://openlibrary.org/works/OL2627594W/Digital_literacy.

Google (n.d.). *API Reference*. [online] Chrome Developers. Available at: https://developer.chrome.com/docs/extensions/reference/ [Accessed 20 Aug. 2023].

ICDL (n.d.). *ICDL*. [online] ICDL Europe. Available at: https://icdleurope.org/ [Accessed 20 Aug. 2023].

Jackman, J.A., Gentile, D.A., Cho, N.-J. and Park, Y. (2021). Addressing the digital skills gap for future education. *Nature Human Behaviour*, 5(5). doi:https://doi.org/10.1038/s41562-021-01074-z.

Keats, J. (2010). *Virtual Words*. [online] Oxford University Press. Available at: https://academic.oup.com/book/40770.

Lämmel, R. (2007). Scrap your boilerplate with XPath-like combinators. *Proceedings of the 34th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages - POPL '07*, [online] pp.137–142. doi:https://doi.org/10.1145/1190216.1190240.

Leahy, D. and Dolan, D. (2009). Digital Literacy – Is It Necessary for eInclusion? *HCI and Usability for e-Inclusion*, [online] pp.149–158. doi:https://doi.org/10.1007/978-3-642-10308-7_10.

Martin, A. and Grudziecki, J. (2006). DigEuLit: Concepts and Tools for Digital Literacy Development. *Innovation in Teaching and Learning in Information and Computer Sciences*, [online] 5(4), pp.249–267. doi:https://doi.org/10.11120/ital.2006.05040249.

McCarthy, J. (2019). *Professor John McCarthy - Contributions and Impact*. [online] Stanford.edu. Available at: http://jmc.stanford.edu/contributions/index.html [Accessed 20 Aug. 2023].

Mehta, P. (2016). *Creating Google Chrome Extensions*. [online] Berkeley, CA: Apress. doi:https://doi.org/10.1007/978-1-4842-1775-7.

Moldovan, A. (2022). Technical Language as Evidence of Expertise. *Languages*, [online] 7(1), p.41. doi:https://doi.org/10.3390/languages7010041.

Morton, J.A. and Pietenpol, W.J. (1958). The Technological Impact of Transistors. *Proceedings of the IRE*, [online] 46(6), pp.955–959. doi:https://doi.org/10.1109/JRPROC.1958.286834.

NASA (2022). *nasa/NASA-Acronyms*. [online] GitHub. Available at: https://github.com/nasa/NASA-Acronyms [Accessed 20 Aug. 2023].

Niehaves, B. and Plattfaut, R. (2014). Internet adoption by the elderly: employing IS technology acceptance theories for understanding the age-related digital divide. *European Journal of Information Systems*, [online] 23(6), pp.708–726. doi:https://doi.org/10.1057/ejis.2013.19.

Roser, M. and Ritchie, H. (2013). Technological Change. *Our World in Data*. [online] Available at: https://ourworldindata.org/technological-change [Accessed 20 Aug. 2023].

Schäffer, B. (2007). The Digital Literacy of Seniors. *Research in Comparative and International Education*, [online] 2(1), pp.29–42. doi:https://doi.org/10.2304/rcie.2007.2.1.29.

Schaller, R.R. (1997). Moore's law: past, present and future. *IEEE Spectrum*, [online] 34(6), pp.52–59. doi:https://doi.org/10.1109/6.591665.

Shulman, H.C., Dixon, G.N., Bullock, O.M. and Colón Amill, D. (2020). The Effects of Jargon on Processing Fluency, Self-Perceptions, and Scientific Engagement. *Journal of Language and Social Psychology*, [online] 39(5-6), p.0261927X2090217. doi:https://doi.org/10.1177/0261927x20902177.

StatCounter (2019). *StatCounter Global Stats - Browser, OS, Search Engine Including Mobile Usage Share*. [online] StatCounter Global Stats. Available at: https://gs.statcounter.com/ [Accessed 20 Aug. 2023].

Statista (n.d.). *UK: internet browser market share 2010-2022*. [online] Statista. Available at: https://www.statista.com/statistics/489408/internet-browser-market-share-in-the-by-month/ [Accessed 20 Aug. 2023].

Steelman, K. and Wallace, C. (2017). Breaking barriers, building understanding. *ACM SIGACCESS Accessibility and Computing*, [online] (118), pp.9–15. doi:https://doi.org/10.1145/3124144.3124146.

Steelman, K.S., Tislar, K.L., Ureel, L.C. and Wallace, C. (2017). Eliciting Best Practices in Digital Literacy Tutoring: A Cognitive Task Analysis Approach. *Human Aspects of IT for the Aged Population. Aging, Design and User Experience*, [online] pp.447–460. doi:https://doi.org/10.1007/978-3-319-58530-7_34.

Toyon, M.A.S. (2021). Explanatory sequential design of mixed methods research: Phases and challenges. *International Journal of Research in Business and Social Science (2147-4478)*, 10(5), pp.253–260. doi:https://doi.org/10.20525/ijrbs.v10i5.1262.

Turing, A.M. (1950). Computing Machinery and Intelligence. *Mind*, [online] LIX(236), pp.433–460. doi:https://doi.org/10.1093/mind/lix.236.433.

UNESCO (2019). *Tools to Help Countries Measure Digital Literacy*. [online] uis.unesco.org. Available at: https://uis.unesco.org/en/news/tools-help-countries-measure-digital-literacy [Accessed 20 Aug. 2023].

UNICEF (n.d.). *Digital Divide and Digital Literacy*. [online] UNICEF Global Development Commons. Available at: https://gdc.unicef.org/resource/digital-divide-and-digital-literacy [Accessed 20 Aug. 2023].

W3Counter (2019). *W3Counter: Global Web Stats*. [online] W3counter.com. Available at: https://www.w3counter.com/globalstats.php [Accessed 20 Aug. 2023].

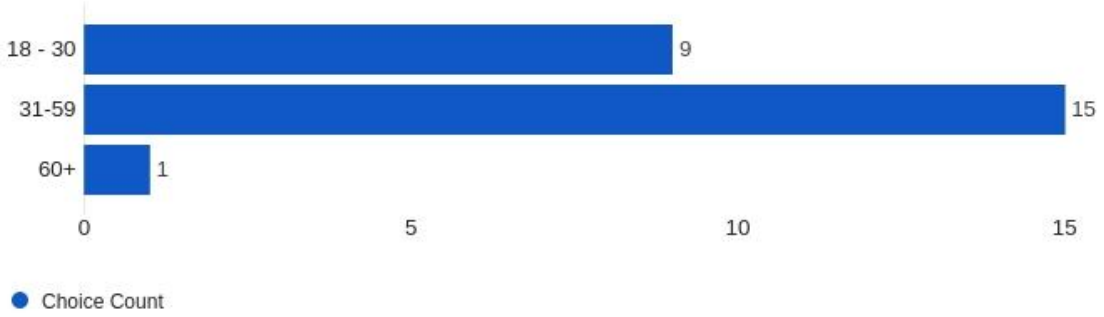
World Economic Forum. (n.d.). *Which European countries have the most digital skills?* [online] Available at: https://www.weforum.org/agenda/2022/04/europe-basic-digital-skills/ [Accessed 20 Aug. 2023].

Yadav, P., Pal, S., Kumar, R., Singh, S. and Singh, H. (2015). Popular Acronym Retrieval through Text Messaging. *Proceedings of the Forum for Information Retrieval Evaluation on - FIRE '14*. [online] doi:https://doi.org/10.1145/2824864.2824889.

Yeh, T., Chang, T.-H., Xie, B., Walsh, G., Watkins, I., Wongsuphasawat, K., Huang, M., Davis, L.S. and Bederson, B.B. (2011). Creating contextual help for GUIs using screenshots. *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. [online] doi:https://doi.org/10.1145/2047196.2047214.

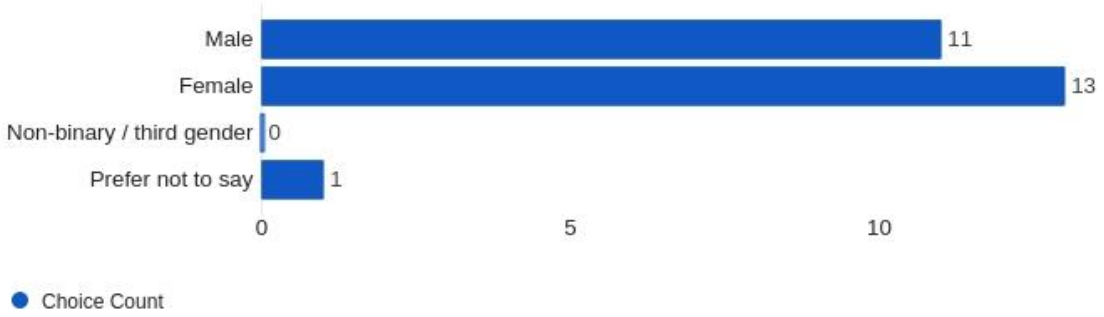
11. Appendix A - Demographic Information

What is your age?



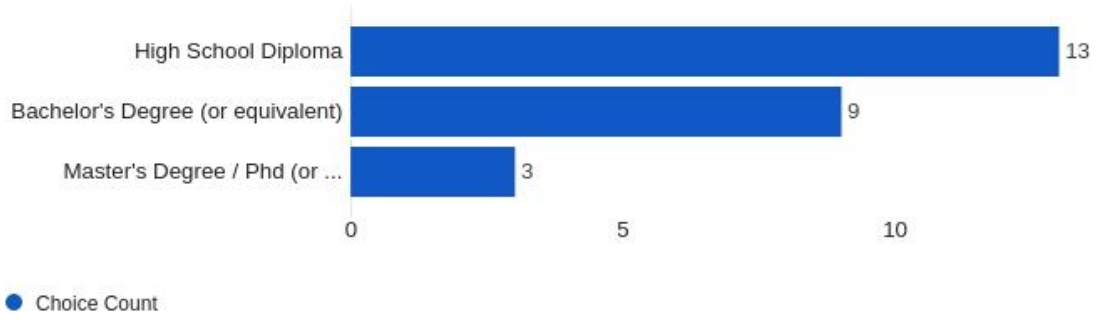
Answer	%
18 - 30	36.00%
31-59	60.00%
60+	4.00%
Total	100%

What is your gender?



Answer	**%**
Male	44.00%
Female	52.00%
Non-binary / third gender	0.00%
Prefer not to say	4.00%
Total	100%

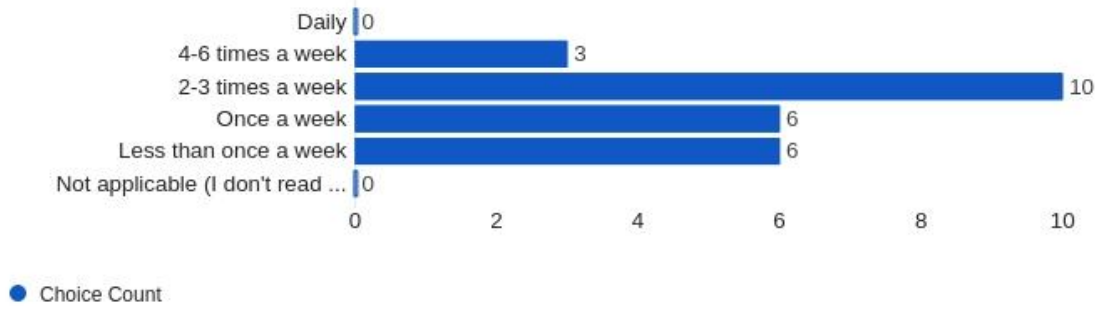
What is your highest level of education completed?



Answer	%
High School Diploma	52.00%
Bachelor's Degree (or equivalent)	36.00%
Master's Degree / Phd (or equivalent)	12.00%
Total	100%

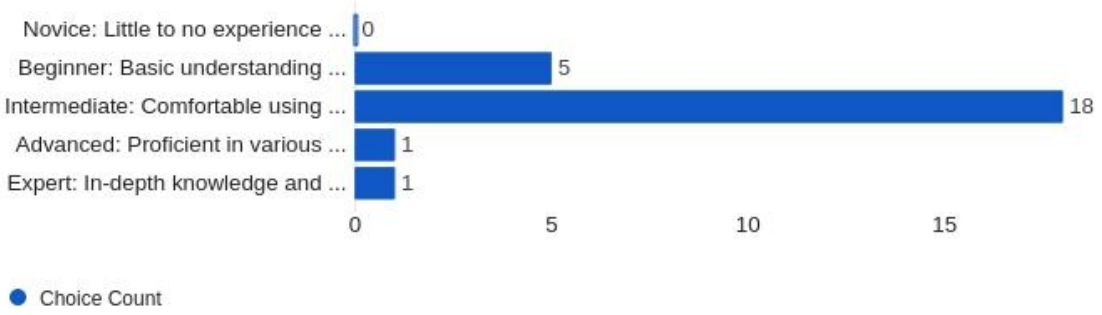
12. Appendix B - Technology Usage and Experience

How often do you read articles related to technology on the internet per week?



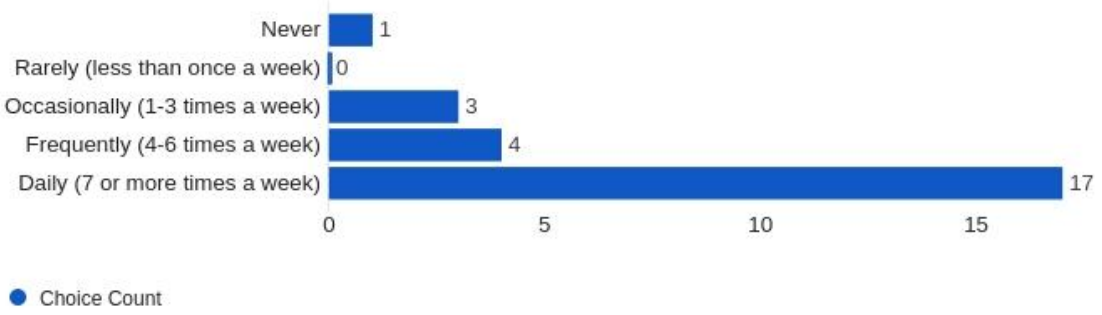
Answer	%
Daily	0.00%
4-6 times a week	12.00%
2-3 times a week	40.00%
Once a week	24.00%
Less than once a week	24.00%
Not applicable (I don't read technology-related articles)	0.00%
Total	100%

How would you assess your level of expertise in using technology?



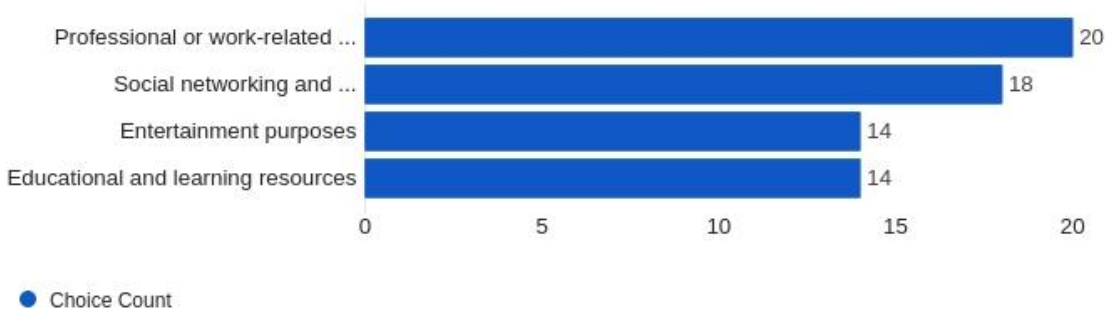
Answer	%
Novice: Little to no experience with technology	0.00%
Beginner: Basic understanding and limited experience	20.00%
Intermediate: Comfortable using technology with some advanced skills	72.00%
Advanced: Proficient in various technologies and able to troubleshoot issues	4.00%
Expert: In-depth knowledge and experience across a wide range of technologies	4.00%
Total	100%

How frequently do you utilise a computer for professional or personal activities?



Answer	%
Never	4.00%
Rarely (less than once a week)	0.00%
Occasionally (1-3 times a week)	12.00%
Frequently (4-6 times a week)	16.00%
Daily (7 or more times a week)	68.00%
Total	100%

What is the reason for using the internet [Multiple answers allowed]?



Answer	%
Professional or work-related activities	30.30%
Social networking and communication	27.27%
Entertainment purposes	21.21%
Educational and learning resources	21.21%
Total	100%