AN INTELLIGENT MOBILE APPLICATION TO ASSIST IN TAKING MATHEMATICAL NOTES USING SPEECH RECOGNITION AND NATURAL LANGUAGE PROCESSING

Casson Qin[1], Jack Wagner[2]

[1]Diamond Bar High School, 21400 Pathfinder Rd, Diamond Bar, CA 91765
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This study evaluated the accuracy and reliability of Voice Note Taking, a technology designed to transcribe spoken language and support note-taking. The experiment analyzed the transcription accuracy and word definition selectionfeature of Voice Note Taking using a series of audio files featuring individuals speaking in English in different settings.The results showed that Voice Note Taking is reliable and accurate, with an overall transcription accuracy rate of 87.81%. However, the study identified room for improvement, particularly in improving accuracy in noisy environments and developing more sophisticated algorithms for word definition selection. Future research could explore the integration of advanced natural language processing techniques to improve the accuracy of word definition selection, including leveraging machine learning algorithms to recognize the specific context and meaningof words. Several previous studies have shown the potential of mobile note-taking apps to enhance student achievement, satisfaction, and accessibility, suggesting further research in this area. Overall, this study highlights the strengths and limitations of Voice Note Taking and provides insight into potential areas for future development.

KEYWORDS

Natural Language Processing , Speech Recognition, Note Taking, Mathematics

1. INTRODUCTION

Note-taking in class is a common practice among students, yet it can be a challenging task for many [7]. Taking effective notes requires students to capture and comprehend the key points and details of the lecture while simultaneously keeping up with the pace of the instructor. This can prove difficult for students who struggle with attention, retention, or those who have a learning disability. As a result, many students end up with incomplete or inaccurate notes that fail to capture the essence of the lecture, hindering their ability to recall the material later on.

Furthermore, students may experience physical discomfort when taking notes for prolonged periods of time. The act of writing or typing can be tiring and can cause muscle fatigue, cramps, and pain in the hands, wrists, and fingers. This can not only be uncomfortable but can also distract from the lecture and make it difficult to concentrate.

David C. Wyld et al. (Eds): SIPM, ITCA, CMIT, FCST, CoNeCo, SAIM, ICITE, ACSIT, SNLP -2023
pp. 179-189, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.130919

Another issue with traditional note-taking methods is that they can be time-consuming and require a significant amount of effort. Students may spend several hours after class organizing and revising their notes to ensure that they are accurate and comprehensive. This additional workload can be overwhelming for students, particularly those who have other academic or personal responsibilities. several studies have investigated the use of mobile note-taking applications in enhancing student achievement, satisfaction, and accessibility. One study found that the experimental group, which used a mobile note-taking app, had higher scores on a knowledge retention test and reported higher satisfaction levels compared to the control group that used traditional note-taking methods. Another study found that a mobile app utilizing speech recognition and natural language processing techniques improved note-taking efficiency, reduced cognitive load, and enhanced learning outcomes [6]. Additionally, a mobile note-taking application that integrates cloud computing and multiple technologies (voice recognition, handwriting recognition, and image capture) was found to enhance the note-taking experience, improve note quality, and reduce cognitive load. These studies suggest that mobile note-taking apps have the potential to be valuable tools for students and recommend further research in this area.

In recent years, advancements in technology have revolutionized the way we learn and retain knowledge. One such technological innovation is the speech-to-text function, which has transformed the process of note-taking and made it more efficient and convenient. This innovative feature has been incorporated into various software applications, including one that utilizes it to transcribe lectures given by teachers and professors.

The software in question operates by utilizing speech recognition technology to convert the spoken words of instructors into written notes. This functionality can prove especially useful to students with disabilities that may make it difficult for them to take conventional notes, such as those who have hearing or motor impairments. With this technology, such students can concentrate fully on the lecture without the added distraction of having to write everything down. Moreover, the software can be used by anyone who prefers to absorb information by listening rather than reading or writing.

In addition to transcribing lectures, the software also provides links to websites that contain information and resources related to the topics covered in the lecture. This feature allows students to delve deeper into the subject matter and gain a deeper understanding of the concepts. The software automatically scans the notes and identifies key words, then generates a list of word definitions. By making these resources readily available, the software enhances the learning experience and enables students to fully comprehend the lecture material.

One of the major benefits of using this software is the reduction in the use of pencil lead. As a result of the speech-to-text function, students no longer have to take extensive written notes. Instead, they can simply listen to the lecture and allow the software to transcribe it for them. This not only saves pencil lead, but also reduces the amount of time and effort required to take notes, allowing students to focus on the lecture and retain more information.

The aim of Voice Note Taking is to provide accurate transcriptions of spoken language to support users in achieving their note-taking goals. To evaluate transcription accuracy, we conducted an experiment on a series of audio files featuring individuals speaking in English in different settings. We enlisted a team of two people who transcribed three five-minute speeches, including a business meeting, a mathematics lecture, and a political debate. We then compared the human-generated transcriptions to those produced by Voice Note Taking and identified any missing, additional, or incorrect words.

The experiment showed that the overall transcription accuracy of Voice Note Taking was 87.81%, indicating that Google's speech recognition services are highly accurate [4]. However, we found some discrepancy when compared to a study by Emil Protalinski, who reported a 4.9% error rate as of 2017. This suggests that our experiment may not have been conducted under the most optimal conditions, but it still approximates the expected result. The study found that the speech recognition services provided better results in environments with low background/white noise, which is especially useful for classroom and business settings.

In addition to transcription accuracy, we also evaluated Voice Note Taking's ability to define technical terms within their given context. The accuracy of the algorithm's word definition feature was evaluated in a separate experiment, which showed an overall correctness rate of 85.72%. The word definition selection algorithm uses the select NLTK tool to extract keywords in the notes to identify the correct definition.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Developing the Voice Note Taking App

Developing the Voice Note Taking App was a daunting task due to the use of the Flutter coding language. Although the language was initially selected for its cross-platform compatibility with both Android and IOS devices, it proved to be complex and presented several challenges during development. Compared to other coding languages, there was a dearth of information available online about Flutter, which made it difficult to progress consistently. In addition, numerous bugs needed to be addressed to ensure the smooth functioning of the various services provided by the Voice Note Taking App. To overcome these obstacles, we had to invest a significant amount of time and effort into researching various files, utilizing APIs, and optimizing the code. Despite the difficulties faced, the team persevered and after several months of dedicated work, most of the issues were successfully resolved.

2.2. Selecting the Right Tools and Technologies

In addition to navigating website security measures, beginners in web scraping often face the challenge of selecting the right tools and technologies for building an effective scraping script. This requires acquiring knowledge of web scraping libraries and frameworks such as Beautiful Soup and Scrapy, which can be a time-consuming and challenging process. Furthermore, it can be difficult to determine the best approach for scraping specific websites or types of data, which requires an understanding of the website's structure and content. Overcoming these challenges requires a willingness to learn and experiment with different tools and techniques, as well as a determination to stay up-to-date with the latest trends and developments in the field of web scraping.

2.3. The Keyword Detection Services

Another obstacle we encountered was related to the keyword detection services. To accomplish this, we began by creating a stop word list, which is essentially a comprehensive list of common and filler words that was alphabetized for ease of use. This list was then compared to the transcribed text, and any overlapping words were deleted. While the stop word list method provedeffective, the team also explored the use of APIs to achieve the same result. However, this requiredtesting and comparing several APIs to determine which one was the most reliable and useful for their needs. Ultimately, Amazon Comprehend emerged as the most effective option due to its machine learning capabilities that enable it to identify and extract key phrases from the context ofthe English transcription.

3. SOLUTION

The app's user-friendly interface and innovative approach to note-taking provide a unique opportunity for students to improve their understanding of mathematical topics. By using speech recognition and natural language processing, the app simplifies the note-taking process and allows students to focus on learning the material. The ability to edit notes and access links and definitions also makes it a valuable study tool, particularly for students who require additional support to excel academically.

The Voice Note Taking application includes a user interface that incorporates a login page. Upon successful login, users can utilize the app's speech recognition feature to take notes on various mathematical topics. The app then transfers these notes to the server's back end for further processing. The back end employs natural language processing (NLP) to extract mathematical terms from the notes. After extracting the relevant terms, the server sends a list of links and definitions corresponding to the mathematical terms back to the app for the user's reference. This process ensures that users can better understand the subject matter and improve their note-taking efficiency.

Furthermore, the app allows users to edit their notes by adding or deleting content providing an opportunity to further enhance their learning experience. This feature is particularly useful when revisiting older notes that need updating or clarifying. The app also assists students in retaining information more effectively by enabling them to focus on actively engaging with the material rather than spending a significant amount of time transcribing notes.

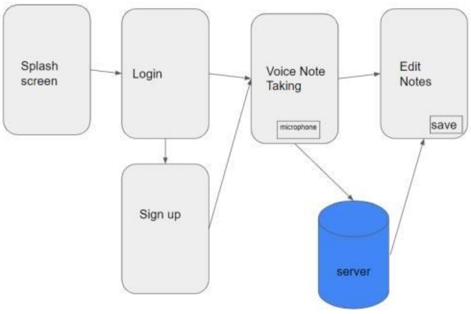
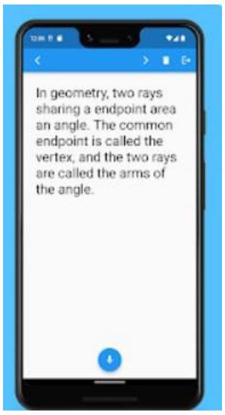


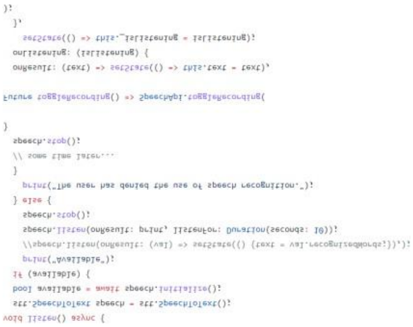
Figure 1. Overview of the solution

The development of Voice Note Taking was accomplished by utilizing Flutter. Flutter is a mobile application development framework that is open-source and was created by Google [14]. With Flutter, developers can build high-quality mobile applications for both iOS and Android platforms using a single codebase. Dart programming language is used by Flutter, which comes with a range of pre-built widgets and tools to make building custom user interfaces, handling user input, accessing device APIs, and other tasks easier. Additionally, Flutter provides developers with several features, such as hot reload, which allows them to make changes to the code and see the results in real-time [12]. Flutter also enables developers to access platform-specific features and APIs with platform-specific integrations.The components involved are illustrated in the figures below. Figure 1 shows the app's speech recognition function, which employs the dart speech recognition library to handle audio input [8]. This function listens to the microphone continuously after being initiated and sends the recorded data to Google's speech recognition function [5]. The data is then transformed into text and displayed on the user's screen, as shown in Figure 2.



In geometry, two rays sharing a endpoint area an angle. The common endpoint is called the vertex, and the two rays are called the arms of the angle.

Figure 2. Transformed text



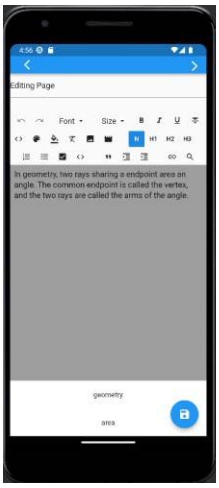
```
void listen() async {
  stt.SpeechToText speech = stt.SpeechToText();
  bool available = await speech.initialize();
  if (available) {
    print("Available");
    //speech.listen(onResult: (val) => setState(() {text = val.recognizedWords;}));
    speech.listen(onResult: print, listenFor: Duration(seconds: 10));
    speech.stop();
  } else {
    print("The user has denied the use of speech recognition.");
  }
  // some time later...
  speech.stop();
}

Future toggleRecording() => SpeechApi.toggleRecording(
  onResult: (text) => setState(() => this.text = text),
  onListening: (isListening) {
    setState(() => this._isListening = isListening);
  },
);
```

Figure 3. Screenshot of code 1

The "listen" function is an asynchronous function that uses the device's microphone to listen to the user's speech. It begins by creating an instance of the "SpeechToText" class from the "stt" library and then checks whether speech recognition is available. Once speech recognition is confirmed to be available, the function listens to the user's speech for a duration of 10 seconds before stopping.

The second function is called "toggle Recording" and it uses the "Speech Api" class to toggle speechrecognition on and off. This function takes two parameters: "onResult" and "onListening". The "onResult" parameter is a callback function that is called when speech recognition is successful and it receives the recognized speech as a text string. The "onListening" parameter is another callback function that is called when speech recognition is started or stopped, and it receives a boolean value indicating whether the device is currently listening to the user's speech. This function updates the UI to show whether the device is currently listening to the user's speech [11].

The application utilizes Natural Language Processing (NLP) techniques to extract mathematical terms from the notes taken by the users. This approach can provide significant benefits to the users in terms of improving their study habits. The backend uses the Spacy library to identify the key mathematical terms and provide corresponding definitions and resources that can help the users understand the concepts they are studying better. SpaCy is a Python-based open-source library designed to perform advanced natural language processing (NLP) tasks with speed and efficiency. It provides pre-trained models that can process text in multiple languages such as English, German, French, and Spanish, and includes a range of features for text processing, including part-of-speech tagging, named entity recognition, dependency parsing, and more. Moreover, it has a robust matching engine that can identify words and phrases in text using statistical or rule-based methods. SpaCy is used for many different applications, including information extraction, text classification, chatbots, and more. It is a widely adopted library in both academic and industrial settings for various NLP tasks [15].

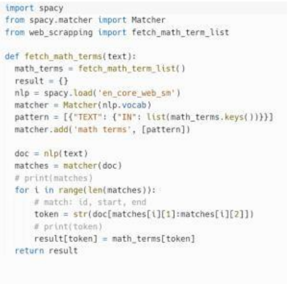


In geometry, two rays sharing a endpoint area an angle. The common endpoint is called the vertex, and the two rays are called the arms of the angle.

geometry

area

Figure 4. Editing page 1



```
import spacy
from spacy.matcher import Matcher
from web_scrapping import fetch_math_term_list

def fetch_math_terms(text):
  math_terms = fetch_math_term_list()
  result = {}
  nlp = spacy.load('en_core_web_sm')
  matcher = Matcher(nlp.vocab)
  pattern = [{"TEXT": {"IN": list(math_terms.keys())}}]
  matcher.add('math terms', [pattern])

  doc = nlp(text)
  matches = matcher(doc)
  # print(matches)
  for i in range(len(matches)):
      # match: id, start, end
      token = str(doc[matches[i][1]:matches[i][2]])
      # print(token)
      result[token] = math_terms[token]
  return result
```

Figure 5. Screenshot of code 2

This code is a Python function called "fetch_math_terms" that uses the Natural Language Processing (NLP) library, spaCy, to extract mathematical terms from a given input text. It first calls a separate function called "fetch_math_term_list" from a module named "web_scrapping" to retrieve a list of mathematical terms.

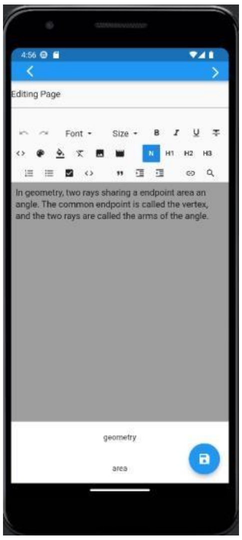
The function then loads a pre-trained spaCy model called "en_core_web_sm" for processing text, and creates a "matcher" object from the spaCy "Matcher" class [10]. The "matcher" object is used to match tokens in the input text that match the mathematical terms retrieved earlier.

The "matcher" is given a pattern of the form [{"TEXT": {"IN": list(math_terms.keys())}}], which specifies that the pattern should match any token that appears in the list of mathematical terms keys.

The input text is then processed by the spaCy model, and the "matcher" is applied to the resulting "doc" object. The matches returned by the "matcher" are then iterated over, and for each match, the corresponding token and its definition are added to the "result" dictionary.

Finally, the function returns the "result" dictionary containing the extracted mathematical terms and their definitions.

The app includes a feature that identifies mathematical terms within the notes entered by the user. Once identified, the terms are highlighted, and the user can click on them to access a separate page that provides the definition of the term, along with additional resources such as formulas, examples, and practice problems. This feature helps users to better understand the mathematical concepts they are studying and promotes active learning by allowing them to explore the material in more depth. Additionally, by providing quick access to definitions and resources, this feature can save users time and effort in searching for information online or in textbooks.

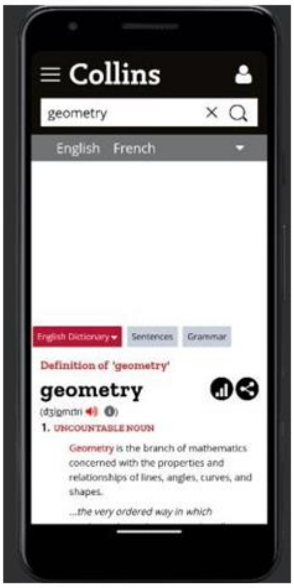


In geometry, two rays sharing a endpoint area an angle. The common endpoint is called the vertex, and the two rays are called the arms of the angle.

geometry

area

Figure 6. Editing page 2



Definition of 'geometry'

geometry

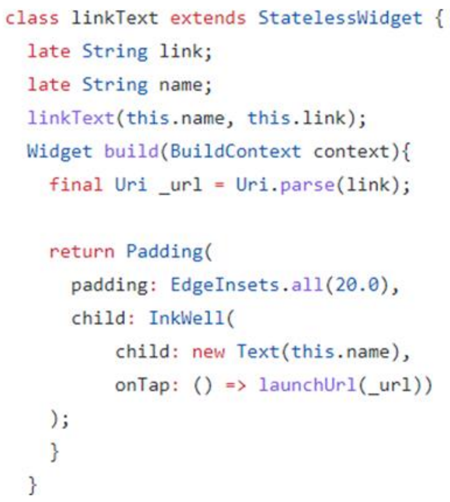
(dʒɪɒmɪtri)

1. UNCOUNTABLE NOUN

Geometry is the branch of mathematics concerned with the properties and relationships of lines, angles, curves, and shapes.

...the very ordered way in which

Figure 7. The meaning of geometry



```
class linkText extends StatelessWidget {
  late String link;
  late String name;
  linkText(this.name, this.link);
  Widget build(BuildContext context){
    final Uri _url = Uri.parse(link);

    return Padding(
      padding: EdgeInsets.all(20.0),
      child: InkWell(
          child: new Text(this.name),
          onTap: () => launchUrl(_url))
    );
    }
  }
```

Figure 8. Screenshot of code 3

The code defines a stateless widget class called "linkText". The widget takes two parameters, "name" and "link", which are used to create a link with the given name. The "link" parameter is the actual URL that the link should direct to.

In the build method of the widget, it creates a new Uri object using the "link" parameter, which can be used to launch the URL when the link is clicked [9]. The widget is then wrapped in an InkWell, which provides a touch response when the link is tapped.

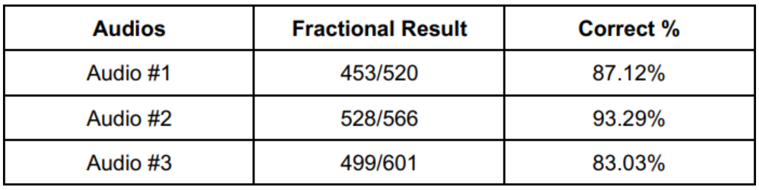
When the link is tapped, the "onTap" function is called, which launches the URL using the "launchUrl" function, passing in the Uri object created earlier. The "launchUrl" function is not defined in the code provided, but it is likely a custom function that launches the URL using the appropriate platform-specific method (e.g. "launch" method in Android or "open" method in iOS).

The widget is also wrapped in a Padding widget to provide some padding around the link for better visual appearance.

4. EXPERIMENT

4.1. Experiment 1

To ensure that Voice Note Taking can effectively support users in achieving their note-taking goals, we need to ensure that it provides accurate transcription results. To evaluate transcription accuracy, we conducted a test to detect technical terms spoken in real-time and verify that they were transcribed appropriately within their given context. The experiment was conducted on a series ofaudio files featuring individuals speaking in English in different settings. To transcribe the files, we enlisted a team of two people who listened to and transcribed three five-minute speeches, including a business meeting, a mathematics lecture, and a political debate. We then compared the human-generated transcriptions to those produced by Voice Note Taking, identifying any missing, additional, or incorrect words. By conducting this test, we can ensure that Voice Note Taking meets the high standards of accuracy required for effective note-taking.



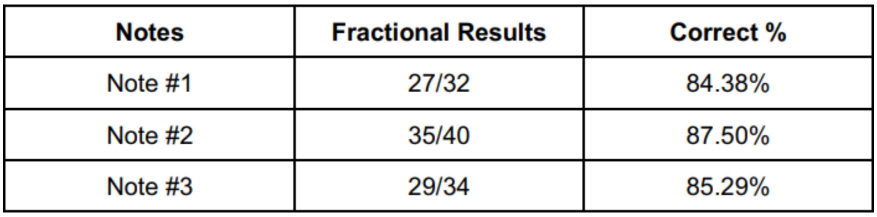
Audios	Fractional Result	Correct %
Audio #1	453/520	87.12%
Audio #2	528/566	93.29%
Audio #3	499/601	83.03%

Figure 9. Table of experiment 1

Table 1 displays a high success rate of 87.81% overall for all three videos, indicating that Google's speech recognition services are highly accurate [13][14]. However, our findings show some discrepancy when compared to a study by Emil Protalinski, who reported a 4.9% error rate as of 2017 [2]. This suggests that our experiment may not have been conducted under the most optimal conditions, but it still approximates the expected result.

4.2. Experiment 2

In some cases, the word definitions for the notes did not align with the given context. For instance, the word "bat" can refer to either a flying mammal or a piece of baseball equipment. We determined the number of correct definitions provided based on the context of the notes. To achieve this, we conducted another test using the same three audio files from our initial experiment. The definitions shown were meticulously examined by two testers, who categorized them as either correct or incorrect within their respective contexts.



Notes	Fractional Results	Correct %
Note #1	27/32	84.38%
Note #2	35/40	87.50%
Note #3	29/34	85.29%

Figure 10. Table of experiment 2

The accuracy of Voice Note Taking's transcription and word definition features were evaluated inan experiment. The study showed that the overall correctness rate of word definition produced bythe algorithm was 85.72%, indicating that Voice Note Taking can reliably define words that matchthe context of the notes. The word definition selection algorithm uses the select NLTK tool to extract keywords in the notes to identify the correct definition. The study found that the speech recognition services provided better results in environments with low background/white noise, which is especially useful for classroom and business settings. However, there is still room for improvement. The transcription accuracy achieved was within the 87.81% range, while correct image selection based on keywords averaged 85.72%, which was better than the expected 70%.

5. Related Work

Demir and Aşıksoy investigated the effects of using note-taking applications on mobile devices on student achievement and satisfaction [1]. The study involved 73 university students who were divided into two groups: a control group that used traditional pen and paper note-taking methods, and an experimental group that used a mobile note-taking application. The results showed that the experimental group had higher scores on a test of knowledge retention, indicating that mobile note-taking apps can have a positive effect on student achievement. Additionally, the experimental group reported higher levels of satisfaction with their note-taking experience than the control group. The paper concludes that mobile note-taking apps can be a valuable tool for students and recommends further research in this area. In our study, we designed a note-taking software that uses speech recognition technology to convert spoken words into written notes. This feature is particularly helpful for students with disabilities that may hinder their ability to take traditional notes. The software also provides links to websites related to lecture topics, generates lists of word definitions, and enhances the learning experience. Demir and Aşıksoy study investigated the effects of using mobile note-taking applications on student achievement and satisfaction. Both studies highlight the benefits of using technology to enhance the note-taking experience for students, particularly in terms of accessibility, knowledge retention, and satisfaction.

Yang et al. present the design and evaluation of a mobile app for note-taking in classroom contexts [2]. The app utilizes speech recognition and natural language processing techniques to transcribe lectures and generate note summaries. The study involved 60 participants who used the app during a semester-long course. The results showed that the app was effective in improving note-taking efficiency, reducing cognitive load, and enhancing learning outcomes. The paper also discusses the design implications and limitations of the app, and suggests future research directions for mobile note-taking technology. Our study focuses on a note-taking app that uses speech recognition technology to transcribe lectures and generate notes, while Yang et al. study evaluates a mobile app designed for note-taking in classroom contextsBoth studies discuss the use of technology for note-taking, specifically in the form of mobile apps. Both studies recognize the potential benefits of mobile note-taking apps, including increased efficiency and improved learning outcomes. However, the first study emphasizes the accessibility benefits for students with disabilities, while the second study emphasizes the potential reduction of cognitive load. Both studies suggest further research in this area to explore the full potential of mobile note-taking technology.

Kim and Lee presented the design and development of a mobile note-taking application that uses cloud computing to provide users with a variety of services [3]. The app integrate voice recognition, handwriting recognition, and image capture technologies to allow users to take notes in various formats. The paper discusses the technical details of the app's design and implementation, including the architecture, algorithms, and user interface. It also reports on the results of a user study that was conducted to evaluate the app's usability and effectiveness. The results of the study indicated that the app was effective in enhancing the note-taking experience, improving the quality of notes, and reducing the cognitive load of note-taking. Both studies focus on the development and evaluation of mobile note-taking applications that utilize speech recognition technology to transcribe lectures. The first study emphasizes the benefits of using such an application for students with disabilities, as well as for those who prefer to learn by listening rather than writing. The second study highlights the integration of cloud computing and multiple technologies (voice recognition, handwriting recognition, and image capture) to allow for a more flexible and versatile note-taking experience. Both studies report positive outcomes in terms of improving note-taking efficiency, reducing cognitive load, and enhancing learning outcomes. The studies suggest that mobile note-taking applications have the potential to be valuable tools for students and recommend further research in this area.

6. CONCLUSIONS

Future research could focus on exploring the integration of more advanced natural language processing techniques to improve the accuracy of word definition selection. This could include leveraging machine learning algorithms to better recognize the specific context and meaning of words, and using this understanding to select the most appropriate definition. Such advancements could greatly enhance the performance of Voice Note Taking and make it an even more valuable tool for individuals in a range of settings, including students, professionals, and researchers.

While the study found that Voice Note Taking is reliable and accurate, there is still room for improvement in the areas of accuracy in noisy environments and more sophisticated algorithms for word definition selection. The results showed that the speech recognition services provided better transcription results in environments with low background/white noise, making it especially useful for classroom and business settings. However, this highlights the need for further development to improve accuracy in noisy environments, such as busy cafes or public spaces. Additionally, the algorithm for word definition selection showed a high success rate of 85.72% in providing definitions that match the context of the notes. Nevertheless, further advancements could be made to develop more sophisticated algorithms that can accurately differentiate between words with multiple meanings and select the most relevant definition.

REFERENCES

[1] Wurst, Christian, Claudia Smarkola, and Mary Anne Gaffney. "Ubiquitous laptop usage in higher education: Effects on student achievement, student satisfaction, and constructivist measures in honors and traditional classrooms." Computers & Education 51.4 (2008): 1766-1783.

[2] Sharples, Mike, et al. "Mobile learning: Small devices, big issues." Technology-enhanced learning: Principles and products (2009): 233-249.

[3] Stacy, Elizabeth Moore, and Jeff Cain. "Note-taking and handouts in the digital age." American journal of pharmaceutical education 79.7 (2015).

[4] Protalinski, Emil. "ProBeat: Has Google’s word error rate progress stalled." The Machine: Making Sense of AI (2019).

[5] Fu, Yu, et al. "Comparison of perceptual differences between users and designers in mobile shopping app interface design: Implications for evaluation practice." IEEE Access 7 (2019): 23459-23470.

[6] Ranchal, Rohit, et al. "Using speech recognition for real-time captioning and lecture transcription in the classroom." IEEE Transactions on Learning Technologies 6.4 (2013): 299-311.

[7] Van Arnhem, Jolanda-Pieta. "Unpacking evernote: Apps for Note-Taking and a repository for Note-Keeping." The Charleston Advisor 15.1 (2013): 55-57.

[8] Bokhove, Christian, and Christopher Downey. "Automated generation of ‘good enough’ transcripts as a first step to transcription of audio-recorded data." Methodological innovations 11.2 (2018): 2059799118790743.

[9] Gajski, Daniel D., Frank Vahid, and Sanjiv Narayan. "A system-design methodology: Executable-specification refinement." Proceedings of European Design and Test Conference EDAC-ETC-EUROASIC. IEEE, 1994.

[10] Walsh, Emily, and Ilseung Cho. "Using Evernote as an electronic lab notebook in a translational science laboratory." SLAS Technology 18.3 (2013): 229-234.

[11] Këpuska, Veton, and Gamal Bohouta. "Comparing speech recognition systems (Microsoft API, Google API and CMU Sphinx)." Int. J. Eng. Res. Appl 7.03 (2017): 20-24.

[12] Assefi, Mehdi, et al. "An experimental evaluation of apple siri and google speech recognition." Proccedings of the 2015 ISCA SEDE 118 (2015).

[13] Van Arnhem, Jolanda-Pieta. "Unpacking evernote: Apps for Note-Taking and a repository for Note-Keeping." The Charleston Advisor 15.1 (2013): 55-57.

[14] Kuzmin, Nikita, Konstantin Ignatiev, and Denis Grafov. "Experience of developing a mobile application using flutter." Information Science and Applications: ICISA 2019. Springer Singapore, 2020.

[15] Srinivasa-Desikan, Bhargav. Natural Language Processing and Computational Linguistics: A practical guide to text analysis with Python, Gensim, spaCy, and Keras. Packt Publishing Ltd, 2018.