RATIONAL MOBILE APPLICATION TO DETECTLANGUAGE AND COMPOSE ANNOTATIONS: NOTESPEAK APP

Yingzhi Ma[1] and Yu Sun[2]

[1]Crean Lutheran High School, Irvine, CA 92618
[2]California State Polytechnic University, Pomona, CA, 91768

ABSTRACT

Students in international classroom settings face difficulties comprehending and writing down data shared with them, which causes unnecessary frustration and misunderstanding. However, utilizing digital aids to record and store data can alleviate these issues and ensure comprehension by providing other means of studying/reinforcement. This paper presents an application to actively listen and write down notes for students as teachers instruct class. We applied our application to multiple class settings and company meetings, and conducted a qualitative evaluation of the approach.

KEYWORDS

Digital learning aids, digital note-taking, note-taking mobile applications.

1. INTRODUCTION

The term "language barrier" refers to the limitations imposed between two or more people when trying to communicate using different accents, wording, or language. This is a common problem that applies to everyday situations, and also school or educational settings. In the modern world, people move or travel across continents every day, and with more than 6,500 languages worldwide, there is bound to be miscommunication [1]. Resolving these issues with an application capable of language detection and translation is an effective and popular solution.

Some of the benefits of utilizing such an application include: having indefinite access to the written information, being able to translate the content to other languages, and being able to store the audio/video for future analysis. This technology is being improved over time, but is not yet fully developed. Unfortunately, the solutions previously mentioned still have barriers to overcome before they can reach maximum efficiency. For instance, the transcribing and translation of audio may not be accurate. In other words, the devices or applications used are not yet perfected, so there exists the possibility that information or context may be lost. Nonetheless, the ultimate goal of this application is to facilitate learning for students or other interested parties by allowing them to digest information at their own pace.

Some of the app technologies that have been developed so far that are available on either Google Play or the Apple Store include: One Note, Nono Notes, Microsoft Notes, Ulysses, and Noted. However, these apps assume that the user is only focused on making and sharing the notes they create. The Notespeak App, on the other hand, is capable of doing this while also providing instant word recognition and image services as well.

David C. Wyld et al. (Eds): ITCSE, ICDIPV, NC, CBIoT, CAIML, CRYPIS, ICAIT, NLCA - 2021
pp. 229-237, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.110918

Many of the aforesaid apps are actually less efficient than people may assume. In some cases, the cost of the application is greater than the standard of the service it provides. For example, Evernote does not utilize the full potential of dictation, which is usually the case for note apps. Their implementations are also limited in scale, and keyword detection with image examples are not offered either [10]. Other techniques, such as instant sharing through the platform, are not offered by apps such as Ulysses. This simplified approach does not satisfy the need of students and businesspeople who need to take notes in the moment. The methods and algorithms such companies employ are not equipped for such fast-paced environments, which is one of the major reasons the Notespeak app was developed. A second practical problem is that note app services are often not user friendly or intuitive. They have lots of information or features that complicate navigation. For example, Bearn Note looks aesthetically pleasing, but has various visual distractions.

In this paper, we trace the development of our own mobile application—Notespeak—that offers a specific set of services to attain better results than those already mentioned. Our main goal is to provide users with an audio transcription service with multiple storing and editing options, which may be expanded upon in future updates. This was inspired by a team member's desire toperform better within classes taught in languages other than his native language.

The first and most prominent feature of our app is the audio detector. It actively listens for recognizable speech patterns, identifies the language spoken, and initiates note-taking. This feature is provided through Google Audio services and has an error rate of approximately 5% [2]. The second feature compiles and organizes the data written by Google Audio service and adds images to the data collected for identified keywords. These keywords are detected using natural language processing techniques and the images are provided by Bing. The images provide context and visual aid to make the notes easier for users to comprehend. The third and last feature offered is saving or sending the recorded information to other electronic devices for storage.

The application was tested multiple times to gather concrete results on whether the included features worked properly or not. In three application scenarios, we examined how the three features mentioned previously work with different volume levels and speech patterns. First, we tested the audio detector on a recorded lecture given by a Harvard professor. It detected over 95% of the audio and provided multiple pictures to let people understand the context of the lecture much better. This is especially helpful for users whose understanding of English is not strong, since it offers the opportunity to learn from images as well. In the second scenario, the focus was on keyword detection to decide which words would have images linked to them. This required testing on diverse lectures using complex words, such as engineering and science. Adding a mode where users can select important words for consideration to improve the service in future updates is also under consideration. In the third scenario, the saving of data was tested using a couple of scenarios: saving data after users are done, and also while transcribing. As further updates are made, we would like to be able to offer temporary data storage in case of disconnection, as well as audio recording while lectures are transcribed.

The rest of this paper is organized into different sections. First, Section 2 will list three challenges that we faced while implementing our ideas. The section following will explain the core features and implementation details.

2. CHALLENGES

In order to build an application to actively listen and record notes for students' lectures, a few challenges have been identified as follows.

2.1. Challenge 1: Flutter Coding Language

One of the major challenges in creating the Notespeak App was using the Flutter coding language. Chosen originally for its compatibility with both Android and IOS devices, it was complex to advance consistently, since it does not have as much information online as other coding languages do. There were also more than a handful of bugs that had to be dealt with to get the different services Notespeak App offers up and running. After looking up documentation on various files, employing API's, optimizing code, and a couple of months of dedicated work, most of the issues were solved. Going through this experience should diminish further challenges and make future updates and development changes easier.

2.2. Challenge 2: Acquiring a Good Transcription Service

Acquiring a good transcription service was another challenge. Various services had to be researched and tested, and many unfortunately had issues. The majority had high error rates, higher than 10% in most cases, and some of them could not detect words at all. There were payment barriers too, but fortunately many of them had free testing for a limited number of words per day. Some of the API's were not fast enough and the process of evaluating API's took longer than other aspects because each test required looking up the documentation for every API along with the JSON response format. Implementing it correctly into the code took multiple tries, and on some occasions the API's were outdated or even completely dysfunctional. The last step was considerably time consuming, but Google translation services was ultimately chosen for its good audio detection rate, easy to read documentation, and fast and free transcription access.

2.3. Challenge 3: Keyword Detection Services

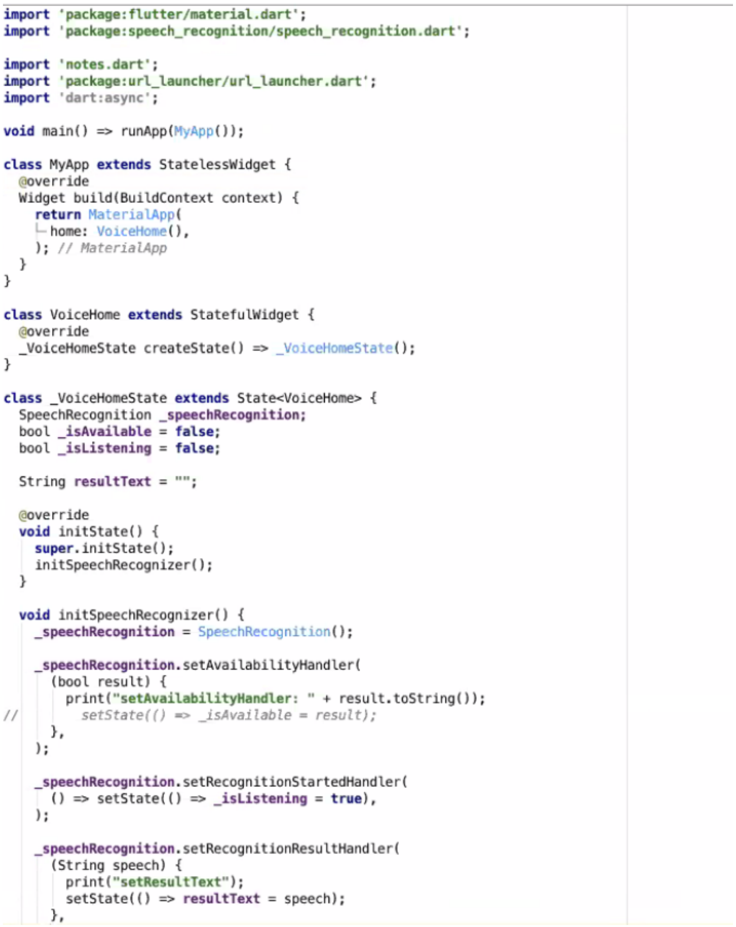
The last challenge was the keyword detection services. The keywords were sent to an image searcher. The first test for selecting keywords consisted of making a stopword list, which is a document including an alphabetized list of common and filler words. The stopword list was then used to compare it to the transcribed text and delete any words both shared. The use of API's was also considered to achieve the same effect, and much like with the transcription service, a few had to be tested and compared along with the stopword list method. In the end, the most reliable and useful was Amazon Comprehend. It uses machine learning to identify and extract key phrases from the given context of the English transcription. Since we plan to expand the language transcription features and have the app become more useful in international situations, we selected this service [8].

3. SOLUTION

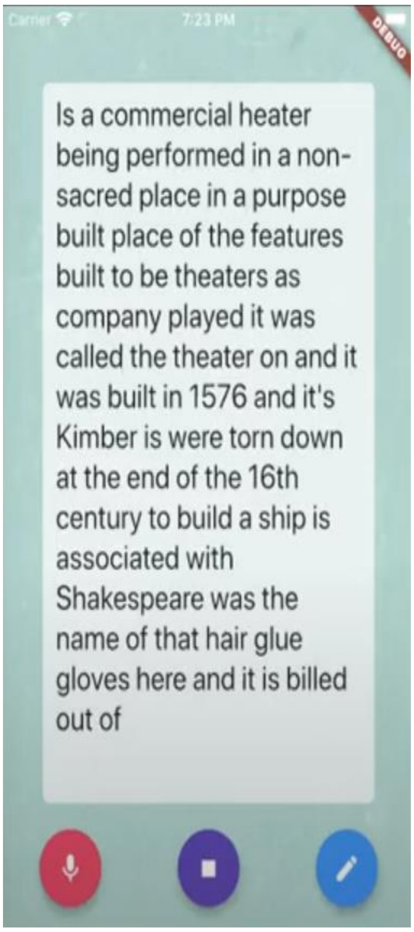
Notespeak is a mobile application focused on transcription and note-taking services primarily for educational environments. The application starts when the user prompts it by clicking on a button. It then actively listens for surrounding voice input. The system then transcribes all detectable language spoken in real-time until the user indicates it should stop. The system then proceeds to analyse, detect keywords, and organize data into an organized note file that may be stored and shared. The main functions of our application include the use of proper language detection, transcription services, recording information into a text document, and identifying keywords within the recorded material that may connect to and find relevant images. The notes can be titled and stored within the user's phone with the intention of studying or referencing the material at a later time. Since Notespeak integrates the use of multiple API's offering high-quality services, it can be utilized in similar, functional contexts. Besides a classroom setting, Notespeak is perfect for conference meetings, speeches, and day-to-day conversations [9].

The main technical challenges of the system are proper information organization and display readability. We look forward to improving the current services and implementing more features in the future. Within the application scope, we look forward to adding translation, manual editing, and audio storage components.

Notespeak, as previously mentioned, was coded using the Flutter language. The figures below detail the components utilized to make it work. Figure 1 depicts the speech recognition function.It uses the dart speech recognition library to process audio input. When initialized, it actively listens through the use of the device's microphone. The data captured is then sent to a speech recognition function provided by Google [2]. It then converts this data into a string of text,which is displayed on the user’s screen as seen in Figure 2.



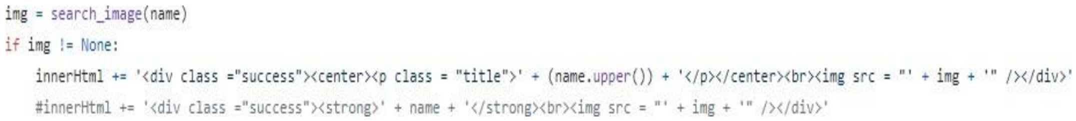
```
import 'package:flutter/material.dart';
import 'package:speech_recognition/speech_recognition.dart';

import 'notes.dart';
import 'package:url_launcher/url_launcher.dart';
import 'dart:async';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: VoiceHome(),
    ); // MaterialApp
  }
}

class VoiceHome extends StatefulWidget {
  @override
  _VoiceHomeState createState() => _VoiceHomeState();
}

class _VoiceHomeState extends State<VoiceHome> {
  SpeechRecognition _speechRecognition;
  bool _isAvailable = false;
  bool _isListening = false;

  String resultText = "";

  @override
  void initState() {
    super.initState();
    initSpeechRecognizer();
  }

  void initSpeechRecognizer() {
    _speechRecognition = SpeechRecognition();

    _speechRecognition.setAvailabilityHandler(
      (bool result) {
        print("setAvailabilityHandler: " + result.toString());
//        setState(() => _isAvailable = result);
      },
    );

    _speechRecognition.setRecognitionStartedHandler(
      () => setState(() => _isListening = true),
    );

    _speechRecognition.setRecognitionResultHandler(
      (String speech) {
        print("setResultText");
        setState(() => resultText = speech);
      },
```



Is a commercial heater being performed in a non-sacred place in a purpose built place of the features built to be theaters as company played it was called the theater on and it was built in 1576 and it's Kimber is were torn down at the end of the 16th century to build a ship is associated with Shakespeare was the name of that hair glue gloves here and it is billed out of

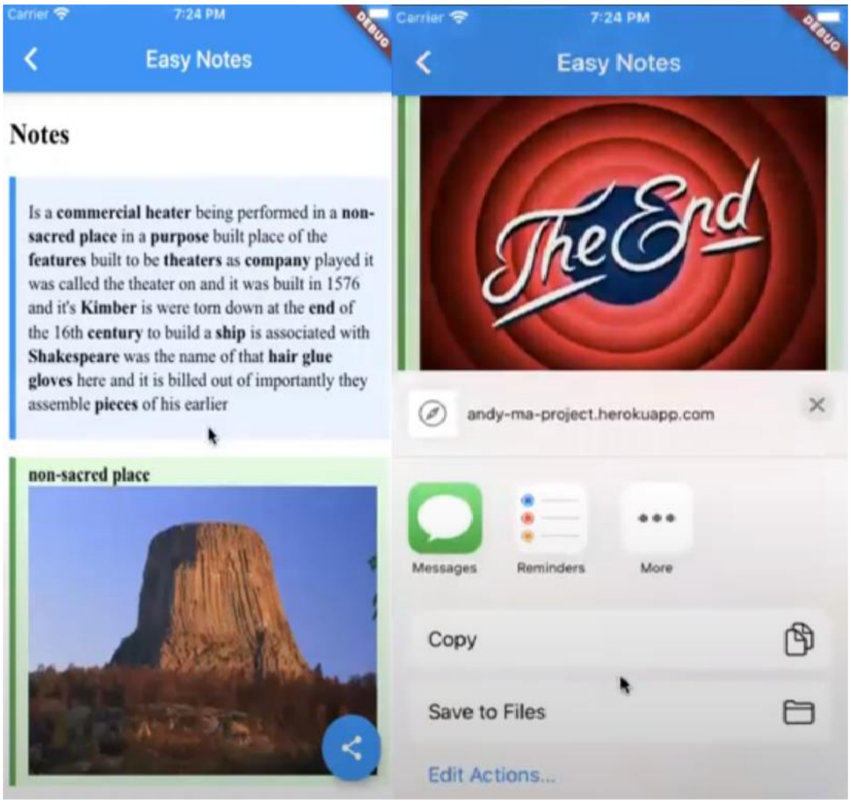
Figures 1-2. Flutter speech recognition code (left); Notespeak App UI transcription (right)

Notespeak image display is used to depict visuals for keywords. These become available once the user selects the “convert to notes” blue button, shown in the lower right of figure 2. The transcribed text is tokenized and filtered to only keep the keywords. Keywords are then sent to an image search function, where an image is rendered using Bing. A portion of code showing this utilization is shown in Figure 3. An example of how the UI displays to the user is shown in Figure 4.



```
img = search_image(name)
if img != None:
    innerHtml += '<div class ="success"><center><p class = "title">' + (name.upper()) + '</p></center><br><img src = "' + img + '" /></div>'
    #innerHtml += '<div class ="success"><strong>' + name + '</strong><br><img src = "' + img + '" /></div>'
```

Figure 3. Flutter image display / selection code



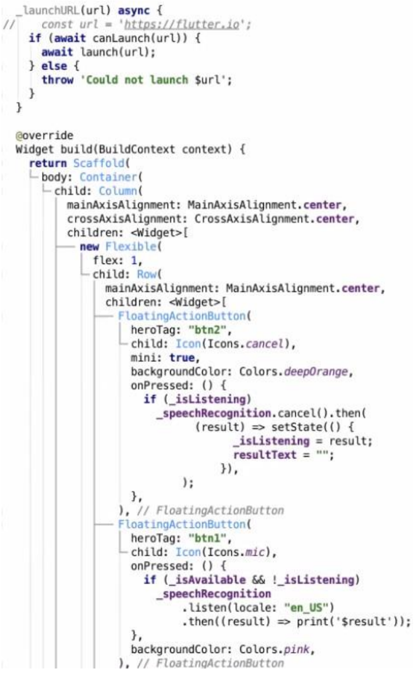
Notes

Is a **commercial heater** being performed in a **non-sacred place** in a **purpose** built place of the **features** built to be **theaters** as **company** played it was called the theater on and it was built in 1576 and it's **Kimber** is were torn down at the **end** of the 16th **century** to build a **ship** is associated with **Shakespeare** was the name of that **hair glue gloves** here and it is billed out of importantly they assemble **pieces** of his earlier

non-sacred place

Figures 4-5. Notespeak App UI images of keywords (left); Notespeak App share button options (right)

A sharing function is available by clicking the share button displayed on the note transcription screen, as shown in figure 4. When the button is pressed, a sub-screen will be displayed at the bottom that provides multiple options. The copy option saves the entire text and image data to the device's clipboard. The save to files option stores the data on the phone's local storage, if available. The send-to-application option directly transfers data from Notespeak to a selected secondary application. The airdrop feature can send the data to devices nearby if the phone allows this. Figure 6 shows the code used to make the data display and share button features.



```
  _launchURL(url) async {
//    const url = 'https://flutter.io';
    if (await canLaunch(url)) {
      await launch(url);
    } else {
      throw 'Could not launch $url';
    }
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
    └body: Container(
      └child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
        ── new Flexible(
          flex: 1,
          └child: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: <Widget>[
            ── FloatingActionButton(
              heroTag: "btn2",
              └child: Icon(Icons.cancel),
              mini: true,
              backgroundColor: Colors.deepOrange,
              onPressed: () {
                if (_isListening)
                  _speechRecognition.cancel().then(
                        (result) => setState(() {
                              _isListening = result;
                              resultText = "";
                            }),
                      );
              },
            ), // FloatingActionButton
            ── FloatingActionButton(
              heroTag: "btn1",
              └child: Icon(Icons.mic),
              onPressed: () {
                if (_isAvailable && !_isListening)
                  _speechRecognition
                      .listen(locale: "en_US")
                      .then((result) => print('$result'));
              },
              backgroundColor: Colors.pink,
            ), // FloatingActionButton
```

Figure 6. Code to create the data display and share button features

4. EXPERIMENT

Notespeak should be able to aid people in accomplishing all their note-taking goals. However, we need the features to provide efficient results, including transcription word accuracy. To evaluate transcription accuracy, we tested the detection of technical terms spoken in real time to assure that they were being transcribed appropriately within their given context. The experiment was applied to a series of files that contained audio from different settings involving individuals giving speeches in English. We transcribed three speeches by ear using a team of two people. Each excerpt had an exact duration of five minutes. The first of these was a business meeting with a single person taking the lead describing a product. The second consisted of a mathematics class lecture. The last audio was a debate between two opposing presidential candidates. The transcription done by humans was then compared to the one made by Notespeak. All missing, additional, and incorrect words were subtracted from the grand total identified by humans. The results are shown in Table 1.

Table 1. Word correctness for the three audio files

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

Table 1 shows a high success rate for all three videos, averaging 87.81% overall. This indicates that Google speech recognition services are very accurate [13, 14]. When compared to studies conducted by Emil Protalinski, there was some discrepancy [2]. He indicated a 4.9% error rate as of 2017 versus our 12.19%. This may suggest that our experiment was not executed under the best conditions, but it did approximate the expected result.

Images displayed for the notes sometimes differed from the given context. For example, the word "bat" could refer to either a flying mammal or a piece of baseball equipment. Weidentified the quantity of correct images given the context of the notes taken. To accomplish this, we conducted a new test utilizing the same three audio files from our first experiment. The images displayed were carefully analysed by two testers who labelled them as correct orincorrect within their given contexts. The results are shown in Table 2.

Table 2. Image correctness for the same audio files

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

The rate of overall correctness averages 85.72%. This suggests that Notespeak is reliable in producing images that match the context of the notes, which are helpful in understanding and visualizing the notes [12]. The images were selected by a search algorithm that identifies select keywords in the notes. This algorithm could possibly be further improved by utilizing some of the search strategies employed by Bing.

The experiment results demonstrate that Notespeak's main features, transcription and image creation, are reliable but can be improved. Speech recognition services provided improved results in environments with low background/white noise. This was especially observed with our target audiences within classroom and business settings. Other scenarios also produced good results, however (see Tables 1 and 2). This correlates with the outcomes achieved by Bokhove, Christian, and Christopher, who deem a good digital transcription as achieving within the 90-percentile range in terms of accuracy [6]. This is approximately 2.2% off from our average.

Correct image selection based on keywords averaged 85.72%, when we expected it to be closer to 70% [7]. This is a good rate, although image filtering would likely further improve the selection and relation of the images to the context of the notes. We also have access topseudocode that could improve our current algorithm in future updates.

5. Related Work

Yu Fu, et al. demonstrated how Mobile Application UI is perceived by the public and designers alike, with a focused analysis on users' preferences: "Selective user involvement which treats users mainly as information sources is adopted to efficiently incorporate users' insights in practical UI designs" [3]. They found that UI is more impactful as it relates to certain functions, such as Multi-icon or activity, and that color patterns also have impact. This article explores multiple apps in controlled environments and compares them to one another with detailed results. Since Notespeak focuses on providing a good user interface, UI research is important. Although Notespeak is different in comparison to the apps studied by Yu Fu, et al., it does contain some of the same features [3]. Notespeak also caters to the note-taking needs of students and businesspeople.

Jolanda-Pieta van Arnhem demonstrated how Evernote, a note-taking app, offers various useful services, which include sketching, multiplatform access, and text/image/audio integration [4]. Evernote offers more services than Notespeak, such as sketching restaurant information, Notebook services, etc. [11]. Notespeak, on the other hand, focuses on the simplicity and notational aspects of quick and easy note-taking. Overall, both applications have their strengths and weaknesses. Whereas Notespeak is better suited for people who only need accessible notes, Evernote offers extra features that may be needed by others willing to pay more for them [15].

R. Ranchal, et al. studied the benefits and constraints of speech recognition between real-time captioning (RTC) and post lecture transcription (PLT) for classroom settings [5]. Note taking in PLT was executed using a video recorder, while RTC employed a note taking application similar to Notespeak. Their investigation found that PLT is better than RTC by a considerable margin for various functions, including word error rate and recognition accuracy, 22% and 78% respectively. Notespeak provided even better results, but we used updated speech recognition services available seven years after this paper was published. Notespeak's average audio recognition accuracy was found to be 87.81%. The PLT gave 85% accuracy, which is lower than Notespeak's. According to Ranchal, et al., "students felt that RTC improved teaching and learning in class as long as word recognition was greater than 85 percent and the transcription and display lag was negligible' [5].

6. CONCLUSION AND FUTURE WORK

Notespeak is a phone application used to take down live notes and save them for users to use. We experimented with users' input, errors, and feedback based on usage of the app. The results indicate that Notespeak can collect audio data and transcribe it efficiently with an average success rate of 87.81%, provided the user's phone has the minimal requirements to run the program.

The application is currently limited by the fact that its services can be impacted by outside sources. The audio recognition, for example, is heavily dependent on each phone's microphone quality, so the transcription accuracy may be impacted by this. The app is practical, but can also be affected by background noise; further testing of and search for optimal audio service continues. Optimization of data organization and image filters continues as well, since some images may still not be appropriate given the audio context.

To solve these issues, we plan to implement an error report feature to collect phone data. This data will be used to identify which microphones don't work properly and allow the app to send messages to users alerting them that the minimal technical requirements are not being met. Using an improved image search and filtering algorithm will also allow the app to select images that better fit the context of the audio transcription.

REFERENCES

[1] Klappenbach, Anna. “Most Spoken Languages in the World 2020.” Busuu Blog, 20 Dec. 2019, blog.busuu.com/most-spoken-languages-in-the-world/.

[2] Protalinski, Emil. “ProBeat: Has Google's Word Error Rate Progress Stalled?” VentureBeat, VentureBeat, 10 May 2019, venturebeat.com/2019/05/10/probeat-has-googles-word-error-rate-progress-stalled/.

[3] 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.

[4] Van Arnhem, Jolanda-Pieta. "Unpacking Evernote: Apps for note-taking and a repository for note-keeping." The Charleston Advisor 15.1 (2013): 55-57.

[5] R. Ranchal et al., "Using speech recognition for real-time captioning and lecture transcription in the classroom," in IEEE Transactions on Learning Technologies, vol. 6, no. 4, pp. 299-311, Oct.- Dec. 2013, doi: 10.1109/TLT.2013.21.

[6] 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.

[7] K. Wnuk and S. Soatto, "Filtering Internet image search results towards keyword based category recognition," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, 2008, pp. 1-8, doi: 10.1109/CVPR.2008.4587621.

[8] Saindon, Richard, and Stephen Brand. "Systems and methods for automated audio transcription, translation, and transfer." U.S. Patent Application No. 11/410,380.

[9] Cloran, Michael Eric, et al. "Real-time transcription system utilizing divided audio chunks." U.S. Patent No. 9,710,819. 18 Jul. 2017.

[10] Viitaniemi, Ville, and Jorma Laaksonen. "Keyword-detection approach to automatic image annotation." (2005): 15-22.

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

[12] Keegan, Shobana Nair. "Importance of visual images in lectures: case study on tourism management students." *Journal of hospitality, leisure, sport and tourism education* 6.1 (2007): 58-65.

[13] 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.

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

[15] Van Arnhem, Jolanda-Pieta. "Unpacking Evernote: Apps for note-taking and a repository for note-keeping." *The Charleston Advisor* 15.1 (2013): 55-57.