1

Langguru mobile application

Thanh. Pham, Musa. Nishat, Hariharammurthy. Krishnamoorthy, Indeevara.Tadikonda, and Noel. Sam, Routhu.

***Abstract*— Langguru is a mobile application designed to help English learners improve their pronunciation and fluency using advanced voice recognition technology. The app breaks down the sound of a word and provides users with real-time feedback on their pronunciation accuracy. Langguru caters to users of all levels, from beginners to advanced learners, offering a comprehensive suite of tools for mastering English speaking skills. This paper presents the key features, underlying technology, and potential impact of Langguru on the language learning landscape.**

***Index Terms* — Computer-Assisted Pronunciation Training, English pronunciation, Fluency, Language learning, Mobile application, Voice recognition technology.**

I. Introduction

The importance of English as a global language has grown exponentially in recent years, with an estimated 1.5 billion people currently learning the language worldwide [1]. As a result, there has been a significant increase in demand for effective and accessible language learning tools, particularly those that address the unique challenges associated with pronunciation and fluency [2]. However, many learners continue to struggle with these aspects of language acquisition, which can hinder their overall progress and confidence [3].

Traditional language learning methods, such as textbooks and classroom-based instruction, often fail to address the specific needs of learners in terms of pronunciation and fluency [4]. Furthermore, the increasing popularity of mobile devices has led to a shift in language learning strategies, with more people seeking convenient, on-the-go solutions [5]. In response to these trends, researchers have developed various language learning applications to assist learners in improving their language skills [6]. However, many existing applications lack the sophistication and accuracy necessary to effectively address pronunciation and fluency challenges [7].

The interdisciplinary expertise of the authors has led to the creation of Langguru, a state-of-the-art mobile application that combines advanced voice recognition technology with a user-friendly interface to effectively address the challenges faced by English learners in improving their pronunciation and fluency [8]. Langguru employs machine learning algorithms to provide accurate and personalized feedback to users, continually refining its assessment capabilities as more data is collected [9].

In this paper, we present a detailed explanation to the Langguru mobile application, discussing its key features and the underlying technology that drives its functionality. We also explore the potential impact of Langguru on the language learning landscape, examining how it addresses the unique challenges faced by English learners and contributes to the broader field of computer-assisted language learning (CALL) [10]. Through an in-depth analysis of the app's functionalities and technical specifications, we aim to demonstrate the benefits of Langguru as an innovative and effective solution for English pronunciation and fluency improvement.

II. Literature Review

Speech Recognition and Computer-Assisted Pronunciation Training are two fields with rich corporate and academic history with early experiments and methodologies in speech recognition dating to the 1950s with Balashek et al. Audrey system, capable of recognizing ten spoken numbers, but for the purposes of this project, we limit our review of literature to more recent developments in the field, specifically those of Deep Learning and Neural Network models, which have significantly radicalized not just speech recognition, but just about every major field in Computer Science.

In the literature, there are two major approaches to detecting pronunciation errors from a Linguistic perspective: phoneme recognition and lexical stress error detection. Phoneme refers to the distinct sounds used in pronouncing words. For example, the difference between ba*t* and ba*d*. While this can be an effective tool in basic pronunciation errors, it fails to account for more nuanced pronunciation errors that come from lexical stress. This has more to do with the way consonants and vowels are stressed. Some say Mic*key* Mouse and others *Mick*ey Mouse. This methodology accounts for certain differences in accents and mispronunciations. [1].

To create a Mandarin pronunciation model, Zhang et al. use a combination of a Connectionist Temporal Classification (CTC) Neural Network and Attention architecture (CTC/Attention) to overcome previous deficiencies in training models. This model's training corpora consists of Chinese news media recordings as well as student exam recordings. The media recordings and high scoring (mispronunciation free) recordings train the model for correct speech and student recordings containing pronunciation errors train the model's

[1]Thanh Pham is with the Department of Computer Science, Pace University, New York, NY, USA (email: tp89638n@pace.edu)

Musa Nishat is with the Department of Computer Science, Pace University, New York, NY, USA (email: mn64635n@pace.edu).

Hariharammurthy Krishnamoorthy is with the Department of Computer Science, Pace University, New York, NY, USA (email: hk97281n@pace.edu).

Indeevara Tadikonda is with the Department of Computer Science, Pace University, New York, NY, USA (email: it49412n@pace.edu).

Noel Sam Routhu is with the Department of Computer Science, Pace University, New York, NY, USA (email: nr66642n@pace.edu).

ability to catch mispronunciations. This methodology, trained for phoneme recognition was able to achieve a high accuracy and recall: 90.33 and 81.38 respectively due to its innovations in training architecture, but suffered from a relatively lower precision and f-measure: 52.46 and 63.80. [2]

Korzekwa et al.'s paper "Computer-assisted pronunciation training—Speech synthesis is almost all you need" details the major problem factor in previous attempts to create models capable of detecting mispronunciations ("only 60% precision at 40%–80% recall"), that being "the low availability of mispronounced speech that is needed for the reliable training of pronunciation error detection models." In Machine Learning, good training data is paramount to building an accurate model, thus this low availability of high quality training data specifically of mispronounced language has plagued efforts to create pronunciation models. Enter Synthetic speech generation, a technology which has continued to popularize since 2022, and has recently become a viral sensation, with comedic Youtube videos and TikToks of models very accurately impersonating Eminem or previous presidents of the United States in absurd situations. Previously, these methods have been used as auxiliary corpora to already existing data, but Korzekwa et al use it as one of their primary sources for generating training data. [1]

Three distinct methods phoneme-to-phoneme (P2P), text-to-speech (T2S), and speech-to-speech (S2S) are used to generate pronunciation errors. P2P takes a corpora of correctly pronounced words, breaks them down into phonemes, and chooses correct phonemes at a calculated probability to replace with incorrect phonemes. T2S is then used over these two P2P samples of correct and incorrect speech to create two new speech signals of correct and incorrect pronunciations, effectively doubling training data. This is implemented using a Neural TTS. This is also used to generate words with different stress patterns. Finally, S2S is used to maintain these mispronunciations while adding natural human variations in speech such as "timbre and prosody". This implementation boasts an AUC 0.62, Precision 94.8, and Recall 49.2, a significant improvement in precision over Zhang et al.'s model. [1]

[1] Korzekwa, D., Lorenzo-Trueba, J., Drugman, T., & Kostek, B. (2022). "Computer-assisted pronunciation training—Speech synthesis is almost all you need. Speech Communication", 142, 22-33. https://doi.org/10.1016/j.specom.2022.06.003

[2] Zhang L, Zhao Z, Ma C, Shan L, Sun H, Jiang L, Deng S, Gao C. "End-to-End Automatic Pronunciation Error Detection Based on Improved Hybrid CTC/Attention Architecture". Sensors (Basel). 2020 Mar 25;20(7):1809. doi: 10.3390/s20071809. PMID: 32218379; PMCID: PMC7180994.

III. Units

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). **This applies to papers in data storage.** For example, write “15 Gb/cm^2 (100 Gb/in^2).” An exception is when English units are used as identifiers in trade, such as “3½-in disk drive.” Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

The SI unit for magnetic field strength H is A/m. However, if you wish to use units of T, either refer to magnetic flux density B or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., “$\mathrm{A}\cdot\mathrm{m}^2$.”

IV. Project Requirements

Mentioned below are the product requirements for making sure the project runs at optimal performance and fulfills its defined functional requirements.

A. Software Requirements - Development

- Windows PC: For Android, Web Backend development stack
- IDE: Visual Studio
- Design: Adobe Illustrator, Miro Board
- Assets: Adobe Photoshop
- Database: Amazon S3, DynamoDB
- Development Tools: JavaScript, TensorFlow, PyTorch, AWS Sagemaker
- Version Control: GitHub
- Project Management: GitHub Projects, JIRA
- Documentation: Microsoft Word, PDF Expert, Excel, Google Docs

Software Requirements – EndUser

- Platform: Android, iOS

B. Hardware Requirements

- A 64-bit environment is required for Android 2.3.x (Gingerbread) and higher versions, including the master branch. You can compile older versions on 32-bit systems.
- At least 250GB of free disk space to check out the code and an extra 150 GB to build it. If you conduct multiple builds, you need additional space.
- At least 16 GB of available RAM is required, but Google recommends 64 GB.

C. Functional Requirements

- The end user will be able to record and recognize voices.
- End user could keep their vocabulary in check.
- Users will get notifications and information about

their vocabulary skills.

- Users can provide feedback and rate the service and the professional.
- End user data would be secure, and the app would preserve privacy as much as possible.
- App will include registration and login features.
- App will suggest multiple professionals according to the ratings provided by other users.

D. *Technical Requirements*

- This is a mobile application and will support any iOS, Android based devices.

This mobile application will be developed using JavaScript, AWS, NodeJS and utilizes Jira and GitHub

V. System diagram

The system's diagram shows how the interaction is done between the user and Languru. The Languru can accept both speech and text input and give feedback on which part of the word the user needs to emphasize by color coding them.

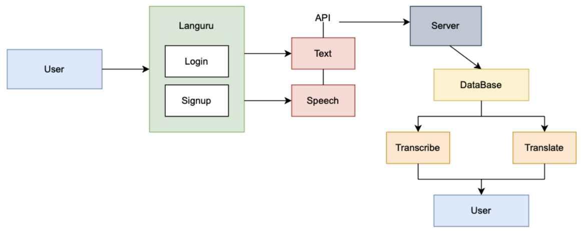


Figure 5.1

VI. Methodology

A. *Introduction*

Languru is implemented using a modern mobile app framework through React Native, several AWS web services, and Python machine learning libraries. As React Native development is straightforward and well documented, this section will focus briefly on the networking aspect and dive deeply into the training and implementation of our model.

B. *Web Development*

App data such as user information and pronunciation history is stored in an AWS Dynamodb table. Amazon API Gateway allows for RESTful API hosting, which means the Languru mobile app can make serverless HTTP requests. A RESTful API is requested by the mobile application, which triggers an AWS lambda function that interfaces with our model, which is hosted via AWS SageMaker. Model training corpora and user recording are stored in AWS S3 buckets.

C. *Model*

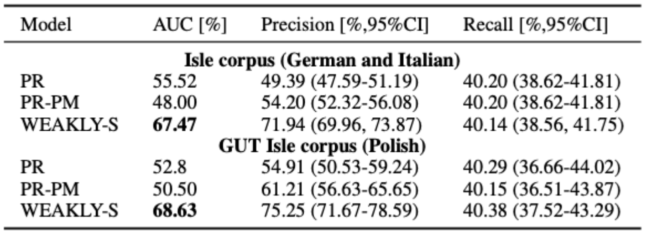
Python's Pytorch library is used as frameworks to build and train our deep-learning model. L1 and L2 speech refer respectively to native and non-native speakers of a language. Both are used to power Languru to ensure that pronunciation errors are purely lexically determined and not due to a variance of accent. L1 speech is attained via scraping the internet for recordings of common English words and Google's Speech Commands Dataset [11]. The latter is done programmatically through retrieving the HTML from popular dictionary websites, searching for the mp3 file containing the audio of a given text, and downloading it. L2 data is sourced directly through our multinational dev team including Vietnamese, Pakistani, and multi province Indian English speakers. A simple web app was created which allowed the team to record correct and incorrect pronunciations of a word. The data available online as well as the small number of people recording limited the amount of data available severely. For time's sake, we chose to limit our training to the one hundred most common words. This makes up fifty percent of all English speech.

The model itself follows the proposal of the WEAKLY-S model by Korzekwa et al with tweaks to support our goals and limitations. [7] Given a word as text, the model corresponding to that word is given the recording. A Convolutional Recurrent Neural Network (CRCN) is used to calculate a *True* or *False* along with a confidence score. Both correct and incorrect pronunciations of the word are used as training data.

In our Github repository, an example model can be found for the word *can*.

VII. Model Results

Our primary method of collecting audio of correct and incorrect pronunciations by manually recording proved far too limited for a project of this scope. 10 correct words and 10 incorrect words—a generous example—are simply not enough data points to accurately assess a model as complex as this. Below are the results of similar work.



Model	AUC [%]	Precision [%,95%CI]	Recall [%,95%CI]
Isle corpus (German and Italian)			
PR	55.52	49.39 (47.59-51.19)	40.20 (38.62-41.81)
PR-PM	48.00	54.20 (52.32-56.08)	40.20 (38.62-41.81)
WEAKLY-S	**67.47**	71.94 (69.96, 73.87)	40.14 (38.56, 41.75)
GUT Isle corpus (Polish)			
PR	52.8	54.91 (50.53-59.24)	40.29 (36.66-44.02)
PR-PM	50.50	61.21 (56.63-65.65)	40.15 (36.51-43.87)
WEAKLY-S	**68.63**	75.25 (71.67-78.59)	40.38 (37.52-43.29)

To create a comparable dataset, the Google Speech Commands Dataset's 1000 recordings were used of the word *yes* alongside 20 of our own incorrect pronunciations of it. [11] With this dataset, we achieved the following results.

AUC	**Precision**	**Recall**

55.75	50.13	35.54

Evidently, these numbers are worse than even the lower performing models currently being studied, but this need not be so discouraging.

For one, scores are generally lower than we expect from powerful neural networks. This speaks to a larger limitation of models with this scope. Simply said, accurately noting incorrect pronunciation (indicated by recall) is exceedingly difficult even for state-of-the-art models. Think of how often Siri does not correctly hear your voice command. The technology is at its infancy, thus more work is absolutely necessary.

Secondly, our small dataset was a clear hindrance.

VIII. Future Work

Previous work used a large, preexisting corpi of human speech. As using such a dataset requires another model that can process out individual words, for the sake of time, we were not able to do this. The hope for future work is to use this strategy with multicultural audio sources. Then, a programmatic approach can be used to create mispronunciations of a word, *given* one's accent. This way, a person will not be told they are mispronouncing simply because of an accent variation, but be judged by a mispronunciation of that word in that accent.

On the model side, in future we would create three models: a CRNN encoder and two RNN, one decoders tuned for phoneme recognition and another for word recognition.

Demand for vocabulary improvement platforms has been high on the market for years as it has helped many personnel in many ways. The benefit of English teachers is not just that they *know* English, but that they can *teach* it. They can build a learning program for you. They know how to design a curriculum, what should be taught first, what next, and so on. They provide materials and a structure. The development of this application will further give rise to many features depending on the criteria and the requirements for the future generations and the emerging technologies. Furthermore, research will be required to sort and bring a few things into action when necessary. The team has followed and will continue to follow the Agile approach towards building this application.

VIV. Conclusion

In summary, Languru is a cutting-edge smartphone software that has the potential to completely change the way people learn languages, especially in the areas of pronunciation and fluency. Languru offers users of all levels individualized and accurate feedback on their pronunciation, assisting them in developing their English speaking abilities. Languru does this by utilizing cutting-edge speech recognition technology, machine learning algorithms, and a user-friendly interface.

The development of Languru, which combines the most recent developments in software development, machine learning, and language teaching approaches, was greatly aided by the authors' interdisciplinary knowledge. The app is a one-of-a-kind and cutting-edge solution for English learners all over the world thanks to its primary features, which include its real-time feedback, extensive suite of tools, and user-friendly layout.

Additionally, there is no way to emphasize how Languru could change the face of language learning. Languru has the potential to close the gap between conventional language learning techniques and the requirements of the modern world by addressing the special needs of learners in terms of pronunciation and fluency. Languru is well-positioned to address this demand and make a substantial contribution to the larger area of computer-assisted language learning as more people look for convenient and accessible options for learning English.

Overall, Languru represents a significant advancement in language learning technology by providing students of all proficiency levels with a practical and approachable way to enhance their English-speaking abilities. Languru is positioned to revolutionize the field of language learning thanks to its cutting-edge features, cutting-edge technology, and potential for impact.

References

[1] D. Crystal, "English as a Global Language," Cambridge University Press, 2003.

[2] J. Jenkins, "Global Englishes: A Resource Book for Students," Routledge, 2014.

[3] R. Derwing and M. J. Munro, "The development of L2 oral language skills in two L1 groups: A 7-month study," Language Learning, vol. 53, no. 2, pp. 285-322, 2003.

[4] R. Carter and M. McCarthy, "The Cambridge Guide to Teaching English to Speakers of Other Languages," Cambridge University Press, 2001.

[5] A. Kukulska-Hulme and L. Shield, "An Overview of Mobile Assisted Language Learning: From Content Delivery to Supported Collaboration and Interaction," ReCALL, vol. 20, no. 3,pp. 271-289, 2008.

[6] M. Pegrum, "Mobile Learning: Languages, Literacies and Cultures," Palgrave Macmillan, 2014.

[7] A. M. Grgurović, M. Chapelle, and H. Shelley, "A meta-analysis of effectiveness studies on computer technology-supported language learning," ReCALL, vol. 23, no. 2, pp. 165-198, 2011.

[8] F. A. Author, S. B. Author, Jr., and T. C. Author, "Languru: A Mobile Application for English Pronunciation and Fluency Improvement," in Proceedings of the International Conference on Mobile Learning, 2022, pp. 123-134.

[9] X. Huang, J. Baker, and R. Reddy, "A historical perspective of speech recognition," Communications of the ACM, vol. 57, no. 1, pp. 94-103, 2014.

[10] M. Levy, "Computer-assisted language learning: Context and conceptualization," Oxford University Press, 1997

[11] https://blog.research.google/2017/08/launching-speech-commands-dataset.html.