

STORYTIME - A Story Narrating App For Children

L. Dsa^{1*}, N. Jaiswal², T. Chitodkar³ and Z. Shaikh⁴

^{1*,2,3,4}Department of Computer Engineering, K J Somaiya College of Engineering, Mumbai, India

*Corresponding Author: lynessa.dsa@somaiya.edu

www.jcseonline.org

Received: 27/Mar/2017, Revised: 14/Apr/2017, Accepted: 26/Apr/2017, Published: 30/Apr/2017

Abstract— Audiobooks and Digital eBooks are another way for us to take literature anywhere we want to go, enabling us to tune in to any story anytime we want to. They have become extremely popular recently but they lack emotions in their voice which makes them quite monotonous and tedious. In this paper we present a solution as a mobile application to make reading enjoyable for children. The android application reads the stories aloud with the help of text-to-speech API's along with the insertion of semantic sounds depending on the rules of natural language processing. The stories are read along with appearance of background images to give it a pleasant visual effect. Our application is implemented using Stanford CoreNLP libraries in Java and Text-to-Speech API's, which will strengthen the interest of reading in children and make it more fascinating.

Keywords—Natural Language Processing, Stanford CoreNLP, Java, Text-to-Speech API's, POS Tagger, Basic Dependencies

I. INTRODUCTION

Audiobooks and Digital eBooks in recent years have become extremely popular for reading books and carrying them around. But they pertain mostly to adults. Some of these applications are Scribd, Audible etc. Scribd is a PDF and ePub reader with an integrated virtual library. It is used to read all kinds of written documents and to get huge number of books accessible for direct download through the application itself. The Audible App enables users to download and play the audio books bought via Audible.com. It also allows them to store numerous titles for play on mobile devices. These apps are expensive and they read stories in a dull and monotonous manner. Our application aims to instil a habit of reading in children at a young age. The features available in our application are:

- The playing of missing semantic words/action sounds while the story is read by the application.
- Background images dependent on the context of the story will also appear as the story is being read.
- After the story is read Questions are displayed and further stories are unlocked once the user answers the minimum required questions correctly.
- A different feedback interface appears after each story is read. Based on the feedback the story database is modified.

II. PROPOSED SYSTEM

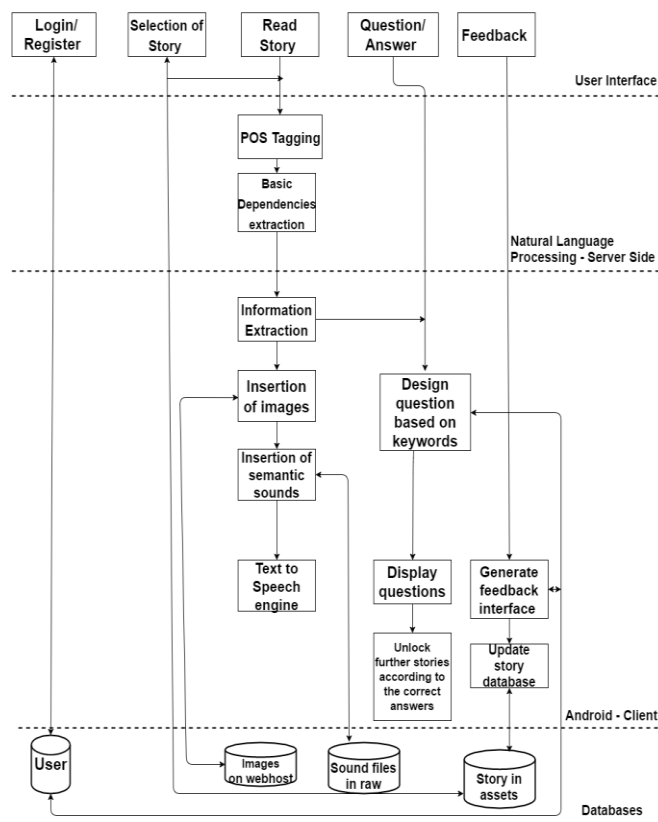


Figure 1

Figure 1 represents the system architecture. The user first needs to register by providing the required details. After the user logs into the application the stories are displayed. The

user then selects any one of the unlocked stories and at the backend; the story is sent to the server database for further processing.

Each sentence of the story is analysed using Parts-Of-Speech tagger[1] and basic dependencies[1]. The keywords are then extracted and stored in the file in the internal storage of the android app. The dependencies between the keywords are found and the appropriate sound file and background image is retrieved using these keywords. The sentence is given to Text-to-Speech API[2] and the sound file is inserted at the appropriate time. After the system has completed reading the story, the questions depending on the context of that story are displayed. Few non-characters are displayed randomly along with one character present in the story and the user has to select the correct option. The number of questions answered correctly by the user is maintained and accordingly further stories will be unlocked. The user is then asked to give feedback to each story.

III. METHODOLOGY

StoryTime is an android application. It is a client- server application. The client consists of the mobile application which is accessed by the users and the server is accessed only by the developers. The server consists of all the java programs and the required jar files required for Natural Language Processing[1].

After the user selects the story from the android[2] app, which they want to read, the Client will send this story file to the server machine.

The Server will then perform the following operations on each line of the story:

1. POS Tagging

By using the Stanford CoreNLP[1] libraries the java program will tag the sentences based on the English Language Parts-of-Speech such as Noun, verb, Adjective etc.

2. Basic dependency

The Stanford CoreNLP[1] libraries are also used to find the basic dependencies in the sentences of the story. The dependencies extracted provide a description of the sentence. For example:

A dobj stands for direct object: The direct object of a verb phrase is the noun phrase which is the object of the verb. So for a sentence like "She gave me an apple" the direct object would be (gave, apple).

Dependencies like nobj, dobj, acomp and xcomp are used to retrieve the keywords and their relation from each sentence.

3. Keyword Extraction

The keywords retrieved are then checked if they are a noun or a verb. The lemma for this keyword is then retrieved from the output of POS Tagger[1].

A string is then created by concatenating the noun and the verb and this string is a filename.

After processing each line of the story the keywords that are extracted are concatenated and sent to the client.

The Operation performed on the Client Side is as follows:

1. Retrieval of Keywords

After receiving the keywords for each sentence. The client will retrieve the appropriate sound file and background image.

2. Insertion of Sound files

Various sound files are stored in the raw folder of Android Studio[2]. These Sound files are then retrieved based on their keywords and played at appropriate location, the text-to-speech engine is paused and the file is played using Android MediaPlayer[2] features.

3. Insertion of Images

The keywords are also used to retrieve the images which are stored on the webhost file manager. These files are then fetched from file manager and displayed on the screen.

Question-Answer module:

After the story is read Questions are generated.

1. Keywords are Extracted by POS[1] tagging .
2. The keywords are mostly nouns.
3. One Keyword images are displayed randomly to the user.
4. An image folder which contains images not related to the story is also stored on the web host file manager of the application.
5. Images from this folder are also displayed to the user.
6. The user has to select the image which was present in the story.
7. The selected image is then checked if it matches the images from the story file.
8. If the Questions are answered correctly then further stories are unlocked.

Feedback Module:

The user is asked to give feedback after every story.

1. A feedback folder is created containing different user interfaces.
2. The interfaces could be stars, chocolates, lollipops etc.
3. The user is asked to select the no of stars (min 1 to max 5).
4. The feedback is then stored in the database for each story.

- If a story has a low rating then that story is either discarded completely or some of the sentences are changed.

IV. PROTOTYPE



Figure 2

Figure 2 represents the registration page of a new user. They have to enter their details such as username, nickname and age. The user also has to enter a CAPTCHA. The username has to be unique.

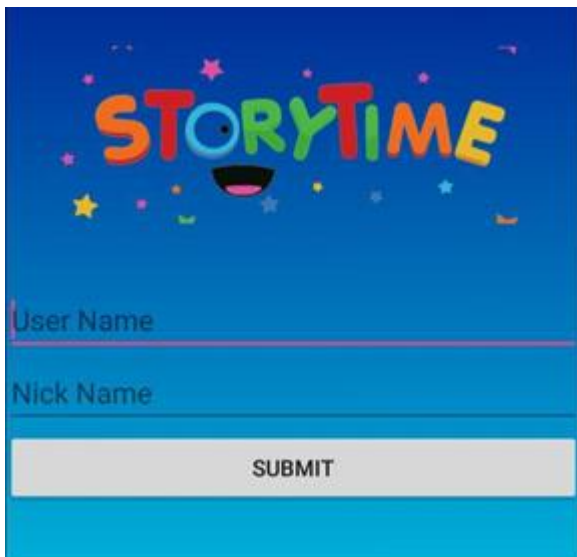


Figure 3

Figure 3 represents the login page. The user provides the username and their corresponding nickname. The system validates and matches the input permitting access only if result is positive. Otherwise the error message will get displayed.



Figure 4

Figure 4 represents the list of stories which will be displayed to the user. The user will then select any one of the unlocked story

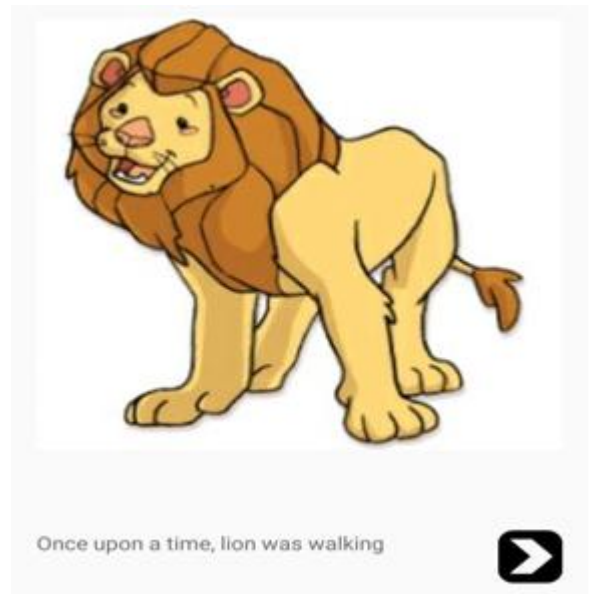


Figure 5

Figure 5 shows a sentence of the story which is selected and it is read along with insertion of sound at the appropriate place along with background images based on the context of the story.



Figure 6

Figure 6 shows the Question and answer Module. Five questions per story are generated after the story has been read. On successful submission of minimum number of correct answers, the next list of stories would be unlocked for the user.



Figure 7

Figure 7 represents the Feedback Module. Feedback Interfaces are randomly generated in different formats, for example, the user can be asked to give chocolates or stars so as to rate the story that was read to them. According to the feedback received, modifications are done in the story format if required.

V. CONCLUSION AND FUTURE SCOPE

Thus, our application can be used by children to read and listen to stories. We began our implementation by using Parts-Of-Speech Tagger and basic dependencies on the story text files to extract keywords with the help of StanfordCoreNLP libraries. The dependencies manuals then helped us pick out the appropriate sound file. They are then inserted with synchronisation, while the API's convert the story from text to speech. Based on these keywords the background images and questions are also generated.

This project can be further expanded to take stories as user input and insert appropriate sound and image files. The Question- Answer interface could also be expanded, so that the user can ask the application Questions.

REFERENCES

- [1] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit", In the Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, USA, pp.55-60, 2014.
- [2] W. Hu, T. Chen, Q. Shi, X. Lou, "Smartphone Software Development Course Design Based on Android", In the Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT 2010), China, pp.2180-2184,2010
- [3] F. Hill, A. Bordes, S. Chopra, J. Weston, "The Goldilocks Principle: Reading Children's Books With Explicit Memory Representations", Published as a conference paper at ICLR 2016, NY, pp.1-13,2016.
- [4] J. Dodge, A. Gane, X. Zhang, A. Bordes, S. Chopra, A. Miller, A. Szlam, J. Weston, "Evaluating Prerequisite Qualities For Learning End-To-End Dialog Systems", Published as a conference paper at ICLR 2016, NY, pp.1-15,2016.
- [5] A. Budholia, R. Mehta, M. Godhrawala, Z. Shaikh, "Mobile Field Service Engineer Application", International Journal of Computer Sciences and Engineering, Vol.4, Issue.2, pp. 100-104, 2016