PROJECT WRITE-UP

PROBLEM OVERVIEW

In 2015, there were an estimated 253 million people with visual impairment worldwide. Of these, 36 million were blind and a further 217 million had moderate to severe visual impairment (MSVI). With the world racing towards an era of complete digitalization, a technological advancement that heavily relies on screen based visual outputs, it is time we notice that a lot of people would not be able to reap the complete benefits of this societal change.

Since web experiences are inherently visual, the web is fraught with sites, tools, and apps that are practically unusable for people with visual impairments. It is a known fact that people need to use the web every day to surf, read and write emails, and to do anything else anyone can conceivably do on the internet. Users with visual impairments should not have to adapt their behavior in order to effectively accomplish their goals. Rather, we need to develop tools that should accommodate the needs of all users, including people with visual impairments.

Over the years there has been considerable development to solve these problems. Screen readers are the most popular tool in this domain, and accessibility tools are adapted according to the potential use of a screen reader. Today, developers are advised to provide text hierarchy, pictures now need to have descriptive alternative texts that can be interpreted by the screen reader, websites are structured to ensure they are easily readable and surfable.

However there is still a lot of work that still needs to be done in order to ensure that screen readers can interpret websites effectively. The most common problems with the current accessibility tools arise with inaccurate picture descriptions, interpretation and summarization of further links of these links and decision making while navigating websites.

Wikipedia is one of the most popular online encyclopedias, with a large number of users and visitors every single day. Given its popularity and wide use, our project will be focussing on improving the readability and navigation of Wikipedia from the perspective of a screen reader first. The long term goal is to come up with solutions that could be applied to most websites.

SCOPE OF THE PROJECT

Most of the accessibility tools available currently are not compatible with links in the text, and just read them out as they are. Because a Wikipedia article generally has multiple links, this interferes with the read-out-loud feature and makes the articles difficult to understand.

Currently available tools aren’t compatible with images and other media in the text. We aim to tackle this by either using an image describer and a user-friendly way to read out the alternate text.

While reading out the hyperlinks, we need to present the user with context about the page, so it is easier for them to decide if they want to navigate to that page. Our solution aims to provide a feature for text-summary for a particular link to another Wikipedia page.

Since there are multiple links on a Wikipedia page, we need to present them in a way that ranks them according to their relevance to the parent subject. We need to find an optimal solution to give the user an option to navigate to other pages if needed in a way that doesn't interrupt the text-to-speech often.

SCOPE OF OUR MODULE

To rank the links on their relevance, we must store them in a database, using an appropriate data structure to reduce computational expenses. The links need to be prioritized and presented to the user in descending order of relevance and to do that we need to find how connected they are to the original topic. At the end of each section, these links will be given to the user, and they can decide what link they want to navigate to next. Creating the database for the links and relevance, along with the summary we generate can also be used for training the ML models that other problems require.

GOALS OF THE MODULE

1. For easy access and traversal, consolidate and *create a database of Wikipedia pages*.

2. Screen readers often have problems reading hyperlinks, and Wikipedia has a very high amount of hyperlinks in articles which can disrupt seamless reading, hence creating *a Wikipedia recommendation system from the article*.

3. To create the recommendation system, we would need to see how often do people visit the pages from our article and rank it in order of visitation and relevance. So, we *would include page ranking and links in between pages*.

4. Some pages on Wikipedia don't have pre written summaries, so we would *create a summarisation model for generating summaries.*

5. Finally, this database would need a format in which it can be easily accessed by other modules in the screen reader for traversal and training, hence *creating a requestable API for the database and its contents.*

PROPOSED SOLUTION

To achieve the above mentioned goals, we need to first gather the raw data from wikipedia, create a database, and store the relations between pages in such a way that it facilitates ranking the pages according to relevance.

The data that we have available on wikipedia is the links that exist in a particular wikipedia page and also how many users landed on the page from different wikipedia pages (eg- X people came to the page “Sachin Tendulkar” from the page “Cricket”).

The data about the links present on a page and which section they are present in can be obtained either through web-scraping or certain APIs that wikipedia offers. The number of users is also available as a wikipedia API.

Once we have the raw data, we will be storing this using a graph database with each page as a node containing data about the links on that page and containing a summary of that page. In the edges we will store information about the linkages between the different pages.

Since there is a very large number of pages on wikipedia, we will be using certain techniques to limit the number of pages in our database and will be trying to keep only highly related pages together.

From here we will be using the information about how many times people went from one page to the next and also the properties of the graph formed to determine how relevant two pages really are.

The next step is to train an ML model to generate a summary of the pages that are present that do not have a summary. We can use pages that already have a summary to train a model to generate a summary for pages that don't. These are the various steps we plan to cover in the project to achieve our goals.

SIDs
19103027, Aakriti Aggarwal
19103037, Rishabh Kaushik
19103066, Anmolpreet Singh
19103086, Kavya Malhotra