

Phần mềm đọc báo cho người khiếm thị

SV: Trương Văn Quốc Hoàng

GVHD: TS. Ninh Khánh Duy

06 Tháng 4 , 2018

Lời cảm ơn

Tôi rất cảm ơn giảng viên hướng dẫn TS. Ninh Khánh Duy, đã hướng dẫn tận tâm và nhiệt tình trong suốt quá trình làm đồ án của tôi. Nếu không có sự giúp đỡ của thầy, tôi đã không thể hoàn thành đồ án đúng hạn.

Ngoài ra, tôi cũng rất biết ơn đối với các giảng viên của khoa Công nghệ thông tin, trường Đại Học Bách Khoa Đà Nẵng những người đã truyền cho tôi kiến thức, giúp đỡ trong quá trình học tập và làm việc tại trường.

Tôi cũng muốn gửi lời cảm ơn đến những người bạn đã giúp đỡ tôi trong quá trình học tập, và ba mẹ, gia đình đã động viên, giúp đỡ rất nhiều.

Lời cam đoan liên chính

Tôi hiểu tính chất của đạo, copy và tôi hiểu chính sách của trường với việc đó như thế nào. Tôi xin cam kết rằng:

1. Những nội dung của phần Đồ Án Tốt Nghiệp này được thực hiện bởi chính tôi và được sự hướng dẫn của giảng viên hướng dẫn TS. Ninh Khánh Duy;
2. Mọi tài liệu tham khảo, được dùng trong Đồ Án Tốt Nghiệp này, đều được trích dẫn với đầy đủ tên của tác giả, tên của bào báo/nghiên cứu và được phát hành một cách công khai và trung thực;
3. Mọi sự sao chép không hợp lệ, vi phạm quy chế và gian lận sẽ do chính tôi chịu trách nhiệm hoàn toàn

Sinh viên

Đà Nẵng, 06 tháng 04 2018

Mục lục

1	Giới thiệu	7
1.1	Bối cảnh	7
1.2	Phạm vi của ứng dụng	7
1.3	Mục đích	8
1.4	Mô tả kiến trúc ứng dụng	8
1.4.1	Kiến trúc chung	8
1.4.2	Các phần chính	9
2	Cơ sở lý thuyết	10
2.1	Xử lý ngôn ngữ tự nhiên	10
2.1.1	Định nghĩa	10
2.1.2	Mục đích	10
2.2	Bộ phân tích trang web	11
2.2.1	Vấn đề	11
2.2.2	Giải pháp	11
2.3	Chuẩn hóa tiếng Việt	11
2.3.1	Giới thiệu về chuẩn hóa văn bản	11
2.3.2	Vấn đề	12
2.3.3	Mục đích	12
2.4	Tổng hợp tiếng nói	13
2.4.1	Định nghĩa	13
2.4.2	Mục đích	13
2.5	Nhận dạng tiếng nói	13
2.5.1	Giới thiệu	13
2.5.2	Mục đích	14
2.6	Quản lý trạng thái với máy trạng thái	14

2.6.1	Định nghĩa	14
2.6.2	Mục đích	15
2.7	Chu kỳ hoạt động của ứng dụng	15
2.7.1	Giới thiệu	15
2.8	Xử lý đa luồng	15
2.8.1	Giới thiệu	15
2.8.2	Mục đích	16
3	Phân tích và thiết kế hệ thống	17
3.1	Vấn đề	17
3.1.1	Kết hợp nhiều module thành phần mềm hoàn chỉnh	17
3.1.2	Xây dựng bộ tiếng Việt đặc trưng và có thể thay đổi được	20
3.1.3	Hỗ trợ mở rộng với nhiều trang báo điện tử khác nhau	20
3.2	Thiết kế hệ thống	21
3.2.1	Hệ thống tổng quan	21
3.2.2	Thiết kế máy trạng thái	22
3.2.3	Định nghĩa dữ liệu cho trích xuất trang web	23
3.2.4	Biểu đồ mô tả	29
4	Kết quả chạy phần mềm và Kết luận	31
4.1	Chạy phần mềm	31
4.1.1	Hướng dẫn cài đặt	31
4.1.2	Hình ảnh phần mềm và hướng dẫn sử dụng	32
4.2	Đánh giá và kết luận	35
4.2.1	Đánh giá	35
4.2.2	Kết luận	36
5	Hướng phát triển	37
5.1	Các hướng phát triển tiếp theo	37

Danh sách hình vẽ

1.1	Sơ đồ người sử dụng thao tác với phần mềm	8
1.2	Các thành phần chính của ứng dụng	9
3.1	Các module khác nhau của phần mềm	17
3.2	Thứ tự các module sau khi kết hợp với nhau	19
3.3	Hệ thống tổng quan	21
3.4	Sơ đồ máy trạng thái	22
3.5	Biểu đồ ca sử dụng của phần mềm	29
3.6	Biểu đồ ca sử dụng với các chuyên mục	30
3.7	Biểu đồ ca sử dụng với các bài viết	30
4.1	Hình ảnh phần mềm khi được khởi chạy	32
4.2	Khu vực hiển thị của phần mềm	33
4.3	Các nút chức năng của phần mềm	33
4.4	Màn hình console khi chạy phần mềm	35

Danh sách từ viết tắt

TTS	Text To Speech
GUI	Graphic User Interface
STT	Speech To Text
NLP	Natural Language Processing
FSM	Finite State Machine
API	Application Programming Interface

Chương 1

Giới thiệu

1.1 Bối cảnh

Trong những năm gần đây, công nghệ thay đổi và tiến bộ rất nhanh chóng, những thông tin đa số đều được cập nhật thông qua các trang báo mạng. Hiện tại những người khiếm thị đang rất thiết thòi, họ ít cập nhật được công nghệ và cũng có rất ít các thiết bị để hỗ trợ. Từ đó, tôi nhận thấy một phần mềm đọc báo cho người khiếm thị là rất cần thiết, có thể giúp họ đọc được, cập nhật được thông tin, tin tức hằng ngày thông qua trang báo online.

Ngoài ra, các ứng dụng tiếng Việt và dành cho người Việt vẫn chưa có nhiều. Tôi mong muốn có thể hợp tác với các bạn trong nhóm để tạo ra một ứng dụng dành cho người Việt.

1.2 Phạm vi của ứng dụng

Hiện nay, các công nghệ nhận diện giọng nói, chuyển văn bản từ giọng nói đã phát triển khá nhanh, các công ty lớn đều đã phát triển rất mạnh. Nhưng có một điểm đó là tiếng Việt vẫn chưa thực sự được hỗ trợ, và chưa có các ứng dụng trên đó đưa vào thực tế.

Vậy nên, dự án này sẽ tập trung vào các phần phân tích, nhận diện giọng nói, chuyển đổi chữ thành giọng nói tiếng Việt dựa trên nghiên cứu và áp dụng các thuật toán thay vì sử dụng các API có sẵn của các công ty lớn.

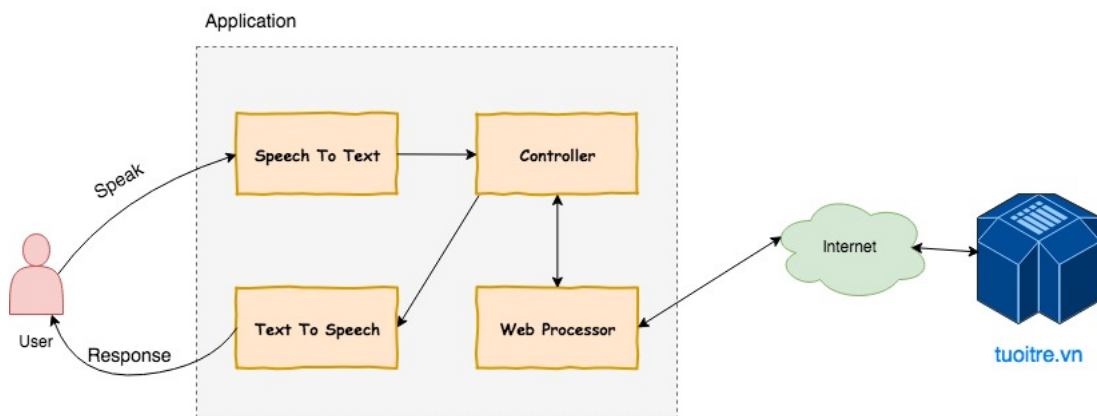
1.3 Mục đích

Ứng dụng đọc báo thành tiếng cho người khiếm thị này sẽ cung cấp được những chức năng sau:

- Tải, bóc tách các bài báo từ các trang báo mạng online
- Đọc lên những bài báo ở các chuyên mục, lựa chọn các bài báo cũng như các chuyên mục khác nhau
- Thông báo ngày, giờ hiện tại
- Nhận diện điều khiển của người khiếm thị thông qua giọng nói

1.4 Mô tả kiến trúc ứng dụng

1.4.1 Kiến trúc chung



Hình 1.1: Sơ đồ người sử dụng thao tác với phần mềm

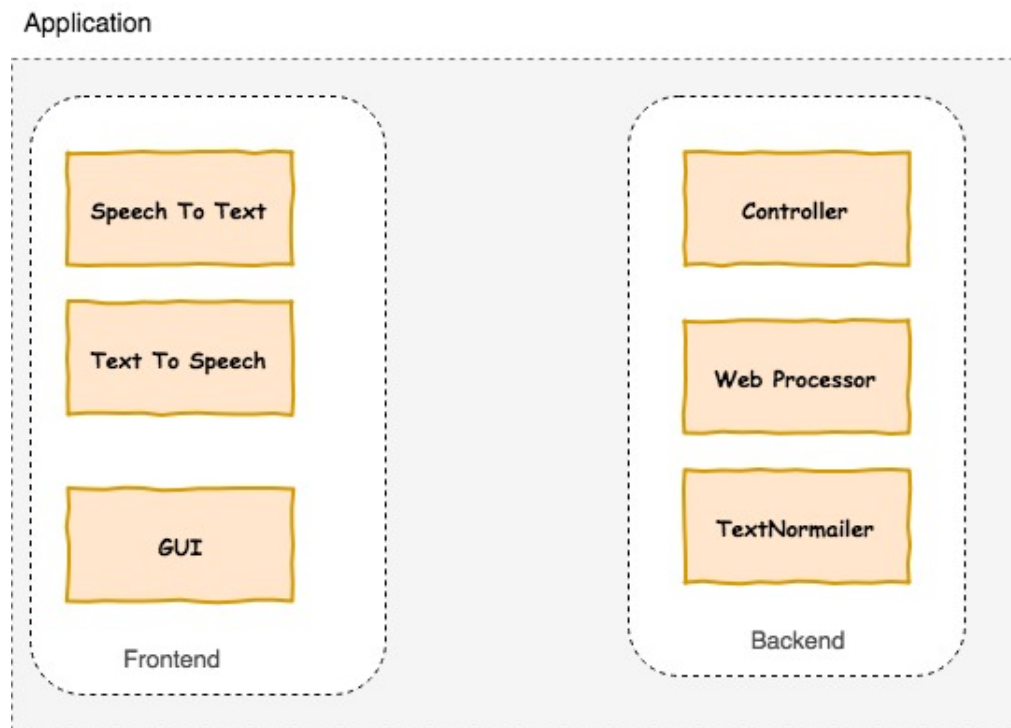
Hình trên mô tả hoạt động thao tác của người khiếm thị với hệ thống. Hệ thống ứng dụng sẽ đảm nhận các công việc sau:

- Nhận thông tin yêu cầu từ người dùng
- Phân biệt các lệnh và thực thi
- Tải trang báo online và bóc tách nội dung từ đó
- Chuẩn hóa nội dung văn bản

- Chuyển văn bản đã chuẩn hóa thành giọng nói

Ứng dụng sẽ gồm các phần chính đó là nhận dạng tiếng nói, bộ phân tích trang web, bộ chuẩn hóa tiếng Việt, và bộ chuyển đổi văn bản thành giọng nói tiếng Việt

1.4.2 Các phần chính



Hình 1.2: Các thành phần chính của ứng dụng

Dựa vào hình vẽ ở trên ta thấy kiến trúc hệ thống bao gồm 2 phần

- Phần Front-end

Phần này bao gồm giao diện được dùng có thay thế cho module nhận diện giọng nói, Module nhận diện giọng nói, và module tổng hợp giọng nói để chuyển từ văn bản sang giọng nói.

Để thuận lợi cho việc thực thi trên các nền tảng khác nhau, tôi đang sử dụng ngôn ngữ lập trình Python

- Phần Back-end

Phần này sẽ xử lý các công việc bao gồm chuẩn hóa tiếng Việt, lấy thông tin bài báo, trang web, và module quản lý điều hướng chính

Phần Back-end cũng được sử dụng ngôn ngữ lập trình Python

Chương 2

Cơ sở lý thuyết

2.1 Xử lý ngôn ngữ tự nhiên

2.1.1 Định nghĩa

Xử lý ngôn ngữ tự nhiên (natural language processing - NLP) là một nhánh của trí tuệ nhân tạo tập trung vào các ứng dụng trên ngôn ngữ của con người. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ - công cụ hoàn hảo nhất của tư duy và giao tiếp.

Xử lý ngôn ngữ chính là xử lý thông tin khi đầu vào là “dữ liệu ngôn ngữ” (dữ liệu cần biến đổi), tức dữ liệu “văn bản” hay “tiếng nói”. Các dữ liệu liên quan đến ngôn ngữ viết (văn bản) và nói (tiếng nói) đang dần trở nên kiểu dữ liệu chính con người có và lưu trữ dưới dạng điện tử. Đặc điểm chính của các kiểu dữ liệu này là không có cấu trúc hoặc nửa cấu trúc và chúng không thể lưu trữ trong các khuôn dạng cố định như các bảng biểu.

Xử lý ngôn ngữ là một kỹ thuật quan trọng nhằm giúp máy tính hiểu được ngôn ngữ của con người, qua đó hướng dẫn máy tính thực hiện và giúp đỡ con người trong những công việc có liên quan đến ngôn ngữ như: dịch thuật, phân tích dữ liệu văn bản, nhận dạng tiếng nói, tìm kiếm thông tin,...

2.1.2 Mục đích

Trong phạm vi ứng dụng này, xử lý ngôn ngữ tự nhiên [5] giúp cho phần mềm hiểu được các lệnh cơ bản từ người khiếm thị, từ đó hoạt động trên những yêu cầu mà người khiếm thị đưa ra.

Đây là cách duy nhất ứng dụng có thể tương tác với người khiếm thị bởi vì người khiếm thị không thể nhìn thấy và thao tác trên phần mềm như người bình thường được.

2.2 Bộ phân tích trang web

2.2.1 Vấn đề

Các trang báo online hiện nay bao gồm rất nhiều chuyên mục khác nhau như thời sự, kinh tế, văn hóa, v.v.... Trong mỗi chuyên mục lại có nhiều bài viết và nhiều thông tin, chúng luôn được cập nhật hằng ngày.

Khi tải một trang web ta sẽ nhận được các thông tin hữu ích nói trên và ngoài ra còn có những thông tin khác như quảng cáo,... Chúng ta có thể nhìn thấy được và lọc, đọc được thông tin cần thiết. Nhưng người khiếm thị thì không. Vậy nên ta cần một bộ phân tích trang web để có thể trích xuất những thông tin cần thiết, có ích và loại bỏ những thông tin không cần thiết để phục vụ cho người khiếm thị.

2.2.2 Giải pháp

Xây dựng một bộ phân tích trang web có thể tải trang và phân tích dữ liệu nhận được (các thẻ HTML, CSS, Javascript) và từ đó trích xuất vừa đủ thông tin mà ứng dụng cần. Chi tiết sẽ được mô tả cụ thể trong 3.2.3 phần "Định nghĩa dữ liệu cho trích xuất trang web"

2.3 Chuẩn hóa tiếng Việt

2.3.1 Giới thiệu về chuẩn hóa văn bản

Trong tất cả các lĩnh vực công nghệ ngôn ngữ và tiếng nói đều phải đối mặt với các văn bản thực tế, ví dụ trong dịch máy, phát hiện chủ đề hoặc hệ thống Tổng hợp tiếng nói (Text-to-Speech) thì văn bản chính là đầu vào của chúng. Trong trường hợp khác, nhận diện tiếng nói tự động thường phụ thuộc vào mô hình ngôn ngữ được huấn luyện trên văn bản. Trong thế giới lý tưởng, văn bản “sạch” nghĩa là nó sẽ gồm toàn các từ viết đầy đủ, các cách viết rõ ràng. Nhưng thật không may, ngôn ngữ viết lệch ra khỏi ý nghĩa của nó theo hai hướng quan trọng. Thứ nhất, trong hầu hết các ngôn ngữ đều xuất hiện sự nhập nhằng trong các từ bình thường ví dụ như trong tiếng Anh: từ bass trong văn bản, người đọc có thể hiểu là một loại cá, hoặc bass là một dụng cụ chơi nhạc. Thứ hai, trong hầu

hết các loại văn bản, chúng ta thấy xuất hiện các từ không chuẩn (Non Standard Word), ví dụ như : chữ số, ký tự la mã, viết tắt của chuỗi ký tự, địa chỉ email,... v.v. Chuẩn hóa văn bản [2][3] là một quá trình quyết định xem làm thế nào có thể nhận ra và phân loại những từ không chuẩn này, những từ này vốn là những từ mà không thể xử lý bằng việc áp dụng các quy tắc “từ thành tiếng” chẳng hạn như “Nato” (Na tô), “WTO” (vê kếp-tê-ô).

Có rất nhiều nghiên cứu về chuẩn hóa văn bản ở nhiều ngôn ngữ như tiếng Anh, tiếng Hindi, tiếng Nhật, tiếng Trung... Ở Việt Nam, đã có nhiều hệ thống tổng hợp tiếng nói đã đạt được nhiều kết quả như SAOMAI, HOASUNG, VOICE OF SOUTHERN, VieTalk, ..., nhưng hầu hết các hệ thống này vẫn chưa dành đủ sự quan tâm cho chuẩn hóa văn bản.

2.3.2 Vấn đề

Trong các bài viết thu thập từ trang báo thường có những cách viết khác nhau, và bao gồm nhiều dạng dữ liệu mà ta chưa thể đọc lên được. Dẫn đến chúng ta cần phải chuẩn hóa [3] và xử lý các chữ viết tắt trước khi đọc lên để đảm bảo người nghe có thể hiểu được.

Ví dụ:

- Xử lý chữ viết tắt [3] trước khi đọc lên, ví dụ: TW sẽ đọc lên là "trung ương", UBND sẽ đọc là "ủy ban nhân dân",...
- Đọc số khác nhau tùy vào ngữ cảnh, ví dụ 1 sẽ đọc là "một", 10 sẽ đọc là "mười" thay vì "một không", $\frac{1}{3}$ sẽ đọc là "một phần ba" thay vì đọc "một xuyệt chéo ba",...
- Xử lý dấu câu trước khi đọc
- Có từ điển phiên âm từ phù hợp cho các từ tiếng nước ngoài [4].
- Xử lý đọc các đơn vị, ví dụ: kg sẽ đọc là "ki lô gam", USB sẽ đọc lên là "đô la Mỹ"

2.3.3 Mục đích

Với việc áp dụng chuẩn hóa tiếng Việt, sẽ giúp cho việc đọc thành tiếng hiệu quả hơn. Cũng như giúp cho người nghe sẽ hiểu được bài báo. Ứng dụng sẽ trở nên tự nhiên và đọc nhịp nhàng, thoải mái cho người nghe.

2.4 Tổng hợp tiếng nói

2.4.1 Định nghĩa

Tổng hợp tiếng nói [1] là việc tạo ra tiếng nói của con người một cách nhân tạo. Một hệ thống máy tính thực hiện mục đích này được gọi là một hệ thống tổng hợp tiếng nói. Tổng hợp tiếng nói có thể được thực hiện bằng phần mềm hay nhúng vào phần cứng của máy tính.

Việc tổng hợp tiếng nói có thể được thực hiện bằng nhiều phương pháp. Phương pháp phổ biến nhất hiện nay là phương pháp tổng hợp bằng cách ghép nối các đoạn tiếng nói nhỏ hơn được lưu trữ trong cơ sở dữ liệu. Việc lưu trữ cơ sở dữ liệu nhiều hay ít làm ảnh hưởng rất lớn đến kết quả thu được tốt hay không. Đôi khi, vì mục đích cần phải giảm độ lớn của cơ sở dữ liệu, người ta chấp nhận làm giảm chất lượng của tiếng nói thu được trong một mức cho phép.

Chất lượng của một hệ thống tổng hợp tiếng nói được đánh giá dựa trên độ “giống” đối với tiếng nói của người thật và khả năng để người nghe có thể hiểu được hết ý nghĩa của văn bản.

Một hệ thống chuyển văn bản thành tiếng nói (tiếng Anh là Text To Speech, trong đó sẽ được viết tắt là TTS) là một hệ thống có đầu vào là một văn bản và đầu ra là một sóng âm thanh.

2.4.2 Mục đích

Bộ tổng hợp tiếng nói được dùng để đọc thông tin lên cho người khiếm thị, giúp ứng dụng giao tiếp với người khiếm thị.

2.5 Nhận dạng tiếng nói

2.5.1 Giới thiệu

Tiếng nói là phương tiện giao tiếp cơ bản và rộng rãi nhất của loài người, nó hình thành và phát triển song song với quá trình tiến hóa của loài người. Đối với con người, sử dụng lời nói là một cách diễn đạt đơn giản và hiệu quả nhất. Ưu điểm của việc giao tiếp bằng tiếng nói trước tiên là ở tốc độ giao tiếp, tiếng nói từ người nói được người nghe hiểu ngay lập tức sau khi được phát ra. Từ khi ngành công nghiệp máy tính phát triển, nhiều công trình nghiên cứu trên tiếng nói nhằm khai thác các thông tin từ tiếng nói để ứng

dụng trong nhiều lĩnh vực như hệ thống trả lời điện thoại tự động, dịch vụ tra cứu thông tin du lịch bằng tiếng nói, và ứng dụng nhận dạng tiếng nói trong các hệ thống bảo mật đã đem lại nhiều lợi ích và cách thức giao tiếp thuận tiện hơn cho con người.

Nhận dạng tiếng nói [5] ra đời đã góp phần thay đổi cách người dùng điều khiển máy tính cũng như các thiết bị điện tử khác. Không cần phải thao tác trên màn hình hay bàn phím như thông thường, hệ thống nhận dạng tiếng nói giúp chuyển đổi tín hiệu tiếng nói từ người dùng thành câu lệnh tương ứng. Dựa vào khả năng này, việc áp dụng nhận dạng tiếng nói cho người khiếm thị điều khiển máy tính là hoàn toàn phù hợp.

Hiện nay, khi thế giới đang ngày càng phẳng dần, mọi người ai cũng có nhu cầu tiếp cận nguồn thông tin vô tận trên Internet, kể cả người khiếm thị. Ý tưởng tạo ra ứng dụng đọc báo điện tử cũng được hình thành từ đó. .

2.5.2 Mục đích

Nhận dạng của loài người là một quá trình hoàn hảo, đó là sự quan sát đối tượng cần nhận dạng, ghi nhận lại những đặc trưng của đối tượng, phân lớp đối tượng và có sử dụng khả năng phán đoán suy luận để phân biệt đối tượng đó với đối tượng khác.

Nhưng đối với người khiếm thị, việc thao tác với phần mềm điện tử là một việc cực kỳ khó khăn. Việc tương tác với ứng dụng bằng giọng nói là cần thiết vì người khiếm thị không có khả năng dùng màn hình.

Do đó, cần tạo ra một hệ thống điều khiển bằng giọng nói [5] mà có thể thay thế các thao tác trên giao diện.

2.6 Quản lý trạng thái với máy trạng thái

2.6.1 Định nghĩa

Máy trạng thái hữu hạn (finite-state machine FSM) [6][7] hoặc Máy tự động trạng thái hữu hạn (finite-state automaton FSA), hoặc là máy tự động hữu hạn, hoặc gọi đơn giản là máy trạng thái, là một mô hình tính toán toán học. Nó là một máy trừu tượng luôn có trạng thái nằm trong tổng hữu hạn các trạng thái tại bất kỳ thời điểm nào. Máy trạng thái hữu hạn có thể chuyển từ trạng thái này sang trạng thái khác để phù hợp với đầu vào; sự thay đổi này được gọi là quá trình chuyển đổi. Máy trạng thái hữu hạn được xác định bởi danh sách các trạng thái của nó, trạng thái khởi đầu, và các điều kiện cho từng sự chuyển đổi trạng thái.

Hành vi của máy trạng thái có thể được quan sát qua nhiều thiết bị hiện đại, đó là việc thực hiện một chuỗi các hành động định trước tùy vào chuỗi sự kiện mà chúng được lập trình.

2.6.2 Mục đích

Với ứng dụng đọc báo hiện tại, chúng ta có thao tác giữa các phần khác nhau (bộ nhận dạng giọng nói, bộ phân tích trang web, bộ tổng hợp tiếng nói); các phần này là những module độc lập và cần phải hoạt động trơn tru với nhau, phối hợp để giúp phần mềm hoạt động hoàn chỉnh.

Phần mềm có rất nhiều trạng thái chuyển đổi của các bài viết, các chuyên mục, và các thao tác chức năng, nên việc quản lý trạng thái là một việc rất quan trọng. Áp dụng máy trạng thái sẽ giúp phần mềm hoạt động hiệu quả và chính xác.

Để tìm hiểu thêm về cách cài đặt máy trạng thái, xem tiếp cụ thể ở 3.2.2.

2.7 Chu kỳ hoạt động của ứng dụng

2.7.1 Giới thiệu

Ứng dụng sẽ được bật lên và tải các trang web, các bài báo theo lượt dùng hiện tại và không lưu lại bất kỳ thông tin gì.

Ứng dụng luôn tải và phân tích trang web để lấy các bài viết mới nhất hiện tại trên đó.

2.8 Xử lý đa luồng

2.8.1 Giới thiệu

Một ứng dụng có thể bao hàm nhiều luồng. Mỗi luồng được đăng ký một công việc riêng biệt, mà chúng được thực thi đồng thời với các luồng khác [8].

Đa luồng giữ thời gian nhàn rỗi của hệ thống thành nhỏ nhất. Điều này cho phép bạn viết các chương trình có hiệu quả cao với sự tận dụng CPU là tối đa. Mỗi phần của chương trình được gọi một luồng, mỗi luồng định nghĩa một đường dẫn khác nhau của sự thực hiện. Đây là một thiết kế chuyên dùng của sự đa nhiệm.

Trong sự đa nhiệm, nhiều chương trình chạy đồng thời, mỗi chương trình có ít nhất một luồng trong nó. Một vi xử lý thực thi tất cả các chương trình. Cho dù nó có

thể xuất hiện mà các chương trình đã được thực thi đồng thời, trên thực tế bộ vi xử lý nhảy qua lại giữa các tiến trình.

2.8.2 Mục đích

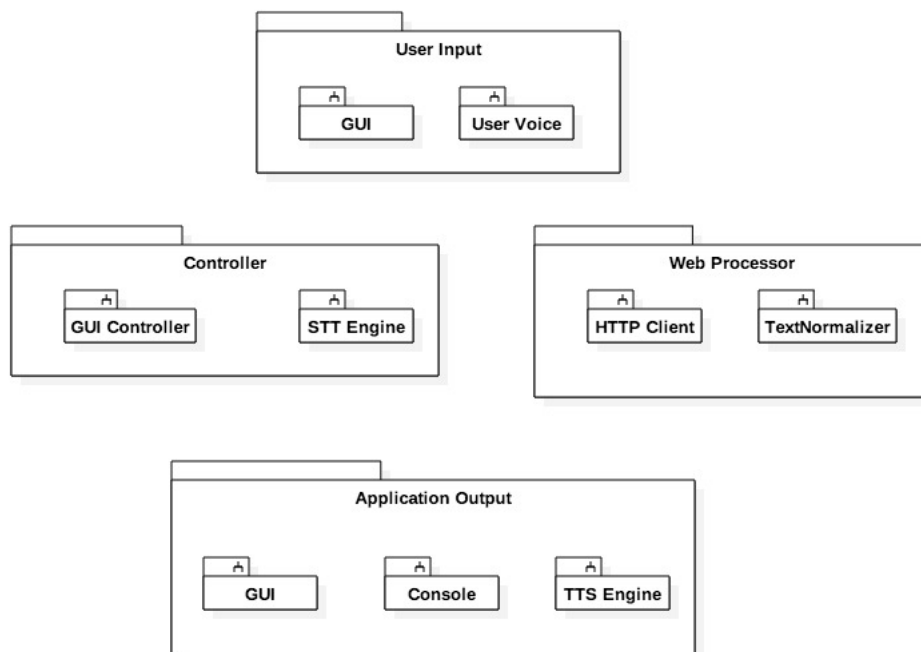
Như đã mô tả ở phần trên, phần mềm này có rất nhiều module khác nhau, mỗi module đều xử lý những tác vụ phức tạp và cần phải phối hợp với nhau. Đầu ra của module này là đầu vào của module kia. Vậy nên việc vận dụng tốt thời gian xử lý của CPU để cho phép chạy đa luồng ứng dụng rất cần thiết cho phần mềm này.

Chương 3

Phân tích và thiết kế hệ thống

3.1 Vấn đề

3.1.1 Kết hợp nhiều module thành phần mềm hoàn chỉnh



Hình 3.1: Các module khác nhau của phần mềm

Hình trên cho thấy rất nhiều module khác nhau được kết hợp để xây dựng nên phần mềm này, được chia theo từng chức năng cụ thể:

User Input (Dữ liệu người dùng nhập vào)

- **GUI (giao diện)**

Module giao diện giúp hiển thị các thông tin, có các nút bấm để theo tác với phần mềm. Giao diện được thêm vào để giúp người khác có thể thao tác giúp cho người khiếm thị, cũng như giúp ích cho quá trình phát triển phần mềm.

- **User Voice (tiếng nói của người dùng)**

Người dùng dùng giọng nói ra lệnh trực tiếp cho người dùng

Controller (Khối xử lý)

- **GUI controller (bộ xử lý giao diện)**

Module GUI controller xử lý các nút bấm trên giao diện, xử lý việc nhận dữ liệu từ các module khác và phản hồi hiển thị lên lại trên màn hình.

- **STT Engine (Speech To Text xử lý ngôn ngữ)**

Giọng nói do người dùng nói ra sẽ do module này giúp nhận diện và xử lý, sau đó trích xuất ra câu lệnh để đưa vào phần mềm xử lý

Web processor (Bộ phân tích trang web)

- **HTTP Client (thư viện thao tác với tài nguyên internet)**

Thư viện giúp thao tác với các trang báo điện tử trên Internet, giúp ta lấy các dữ liệu từ internet và bóc tách các dữ liệu đó ra những thông tin cần thiết

- **TextNormalizer (bộ chuẩn hóa văn bản)**

Khi trích xuất được nội dung văn bản từ trang báo, các văn bản sẽ được module này xử lý, chuẩn hóa thành văn bản hoàn chỉnh với các từ viết tắt, số, đơn vị,... đã được chuẩn hóa hoàn toàn

Application Output (Đầu ra của chương trình)

- **GUI (giao diện)**

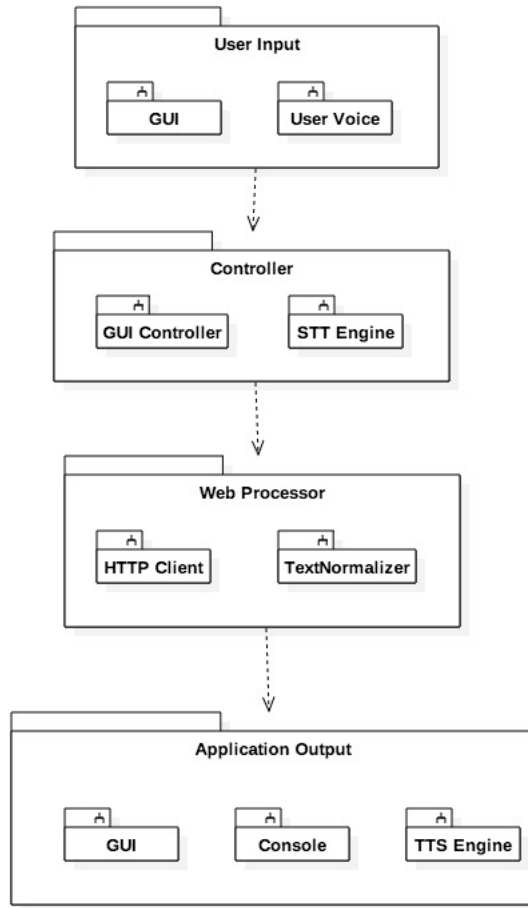
Đầu ra sau khi xử lý sẽ hiển thị trên giao diện

- **Console (Cửa sổ console)**

Hiển thị các thông tin đầu ra cho quá trình phát triển phần mềm

- TTS Engine (Text To Speech chuyển văn bản thành giọng nói)

Những thông tin đầu ra sau khi được chuẩn hóa thì sẽ được module này đọc lên, thông tin đến người dùng, giúp người khiếm thị có thể nghe được và hiểu được.



Hình 3.2: Thứ tự các module sau khi kết hợp với nhau

Hình 3.1 cho ta thấy tất cả các module đang sử dụng và Hình 3.2 cho ta thấy được cách các module này hoạt động và thao tác với các module khác.

Dấu mũi tên biểu diễn đầu ra của phần này sẽ là đầu vào của phần kia, và tất cả các module trong hình đều hoạt động đồng thời.

Để giải quyết được vấn đề các module hoạt động đồng thời và đảm bảo các dữ liệu di chuyển một cách khép kín, đúng với các module tương ứng, giải pháp để giải quyết ở đây là áp dụng mô hình máy trạng thái, và xử lý đa luồng. Chi tiết sẽ được mô tả rõ hơn ở phần thiết kế hệ thống mục sau

3.1.2 Xây dựng bộ tiếng Việt đặc trưng và có thể thay đổi được

Hiện nay với sự tiến bộ của khoa học công nghệ và sự phát triển của các công ty lớn như Microsoft, Google, có rất nhiều phần mềm và module tổng hợp tiếng nói cho tiếng Việt, và các module đó đều được cung cấp API miễn phí cho người dùng.

Tuy nhiên các module đó giọng đọc vẫn chưa được chuẩn, dễ nghe, hơn nữa bị phụ thuộc vào các nhà cung cấp. Ta không thể tùy chọn thay đổi được.

Từ vấn đề đó, phần mềm này được sử dụng một thư viện mã nguồn mở là Pocket-Sphinx và được phát triển bởi một nhóm SV trường Đại học Bách Khoa Đà Nẵng. Với các dữ liệu được thu thập và phát triển từ đầu. Với mong muốn có thể có một bộ xử lý tiếng Việt đặc trưng và có thể làm chủ, tùy biến và thay đổi được. Một điểm lợi của module này nữa đó là khả năng hoạt động offline, độ phản hồi nhanh.

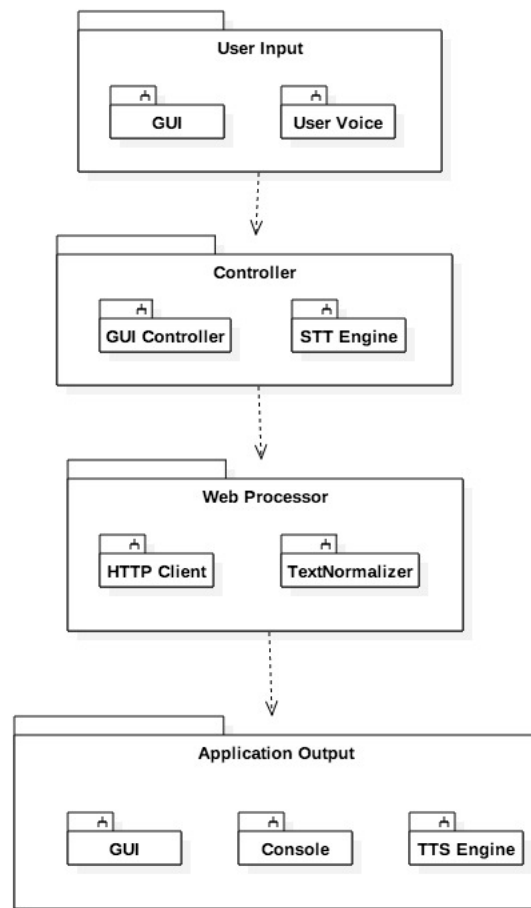
3.1.3 Hỗ trợ mở rộng với nhiều trang báo điện tử khác nhau

Hiện nay có rất nhiều trang báo điện tử nổi tiếng và cung cấp nhiều thông tin như **tuoitre.vn**, **vnexpress.net**,.... Các trang web đó đều có cấu trúc gần như tương tự nhau. Hiện tại phần mềm chỉ đọc và thao tác trên trang báo **tuoitre.vn**

3.2 Thiết kế hệ thống

Thiết kế cụ thể của hệ thống phần mềm

3.2.1 Hệ thống tổng quan



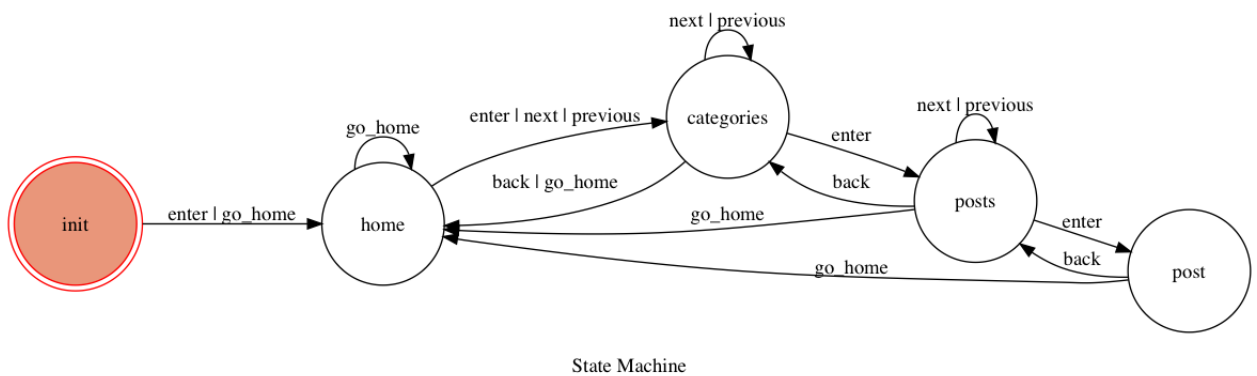
Hình 3.3: Hệ thống tổng quan

- **User Input:** Phần này sẽ nhận dữ liệu từ người dùng, bao gồm từ giao diện và từ giọng nói của người dùng cùng một lúc, sau đó sẽ đưa dữ liệu đến tiếp cho phần điều khiển
- **Controller:** Phần này sẽ nhận dữ liệu từ phần *User Input* và xử lý điều hướng cho chương trình. Đây là phần chính, quan trọng quyết định và gọi và tắt mở các module của phần mềm
- **Web Processor:** Phần này sẽ nhận lệnh từ *Controller* và thực hiện gửi các yêu cầu

lên trang web, sau đó sẽ bóc ra phần nội dung cần thiết. Các nội dung sau đó sẽ được module *TextNormalizer* chuẩn hóa trước khi đưa ra Output

- Application Output: Đây là phần đầu ra của chương trình. Phần này sẽ nhận dữ liệu từ *Controller* và *Web Processor* để hiển thị và đọc ra cho người dùng.

3.2.2 Thiết kế máy trạng thái



Hình 3.4: Sơ đồ máy trạng thái

Các trạng thái hiện có Chúng ta có 5 trạng thái được mô tả ở hình trên.

- Init: Trạng thái này là trạng thái đầu tiên, chưa có gì. Mục đích để cho dễ lập trình
- Home: Đây là trạng thái lúc người dùng mới mở ứng dụng lên, họ vào trang chủ của trang web cũng như trang chủ của ứng dụng. Ở trạng thái này, phần mềm sẽ liệt kê tất cả các chuyên mục hiện có của trang web. Ví dụ các chuyên mục của báo *Tuổi Trẻ* như: Thời sự, thể thao, giáo dục,... Người dùng nói *Tải* hoặc nhấn nút tải, sẽ chuyển sang trạng thái *Categories* tiếp theo
- Categories: Đây là trạng thái duyệt các chuyên mục, giúp người dùng chọn các chuyên mục mà họ thích để vào đọc các bài báo cụ thể trong đó. Khi người dùng chọn chuyên mục, phần mềm sẽ chuyển sang trạng thái *Posts* tiếp theo
- Posts: Trạng thái các bài viết trong cùng một chuyên mục, ở đây người dùng sẽ có thể duyệt giữa các bài viết để chọn tải và đọc bài viết. Khi chọn một bài viết, phần mềm sẽ chuyển đến trạng thái *Post* tiếp theo
- Post: Đây là trạng thái tại một bài báo cụ thể, bao gồm tiêu đề, nội dung trang báo, đây là trạng thái sâu nhất, từ đây chỉ có thể trở về các trạng thái trước.

Dựa vào *Hình 3.4* về sơ đồ máy trạng thái ở trên, dấu mũi tên thể hiện sự chuyển đổi 1 chiều từ trạng thái này sang trạng thái kia. Chữ mô tả ngay trên mũi tên cho thấy hành động sẽ dẫn tới sự thay đổi trạng thái tương ứng. Ta có thể mô tả máy trạng thái với chi tiết các hành động như sau:

- Khi phần mềm khởi động, sẽ từ trạng thái **Init** gọi *enter* để đến trạng thái **Home**, lúc này phần mềm sẽ tải các thông tin ở trang chủ và phần mềm sẽ có dữ liệu của các chuyên mục.
- Tại trạng thái **Home** ta có thể chuyển sang trạng thái **Categories** bằng cách gọi một trong các lệnh *enter*, *next*, *previous*. Lúc này, phần mềm sẽ chọn chuyên mục đầu tiên của bài báo làm chuyên mục hiện tại, và có các chuyên mục khác được lưu trữ để chờ thực hiện các câu lệnh tiếp theo
- Tại trạng thái **Categories** sẽ chứa dữ liệu của 1 chuyên mục hiện tại, tại đó ta có thể gọi *next/previous* để có thể chọn đến chuyên mục kế tiếp hoặc trước đó, khi đã chọn được chuyên mục ưng ý, người dùng gọi *enter* để vào trạng thái **Posts** tiếp theo
- Tại trạng thái **Posts** sẽ chứa dữ liệu của 1 bài viết hiện tại, tại đó ta có thể gọi *next/previous* để có thể chọn đến bài viết kế tiếp hoặc trước đó, gọi *enter* để vào trạng thái **Post**
- Tại trạng thái **Post**, phần mềm sẽ tiến hành tải về nội dung bài viết đã chọn trước đó ở trạng thái **Posts**
- Ở các trạng thái (trừ **Home** và **Init**) ta đều có thể gọi *back* để chuyển về trạng thái ngay trước đó
- Mọi trạng thái đều có thể gọi *gohome* để trở về trạng thái **Home**.

3.2.3 Định nghĩa dữ liệu cho trích xuất trang web

Hiện tại phần mềm chỉ hoạt động với trang web **tuoitre.vn** nhưng với mục tiêu hướng tới hỗ trợ nhiều trang web hơn nữa, ta cần có một định nghĩa dữ liệu chuẩn để có thể mở rộng sau này.

3.2.3.1 Dữ liệu chuyên mục

a. Định nghĩa dữ liệu (schema)


```
{
  "$schema": "http://json-schema.org/",
  "description": "Definition for category",
  "title": "Categories",
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "name": {
        "type": "string",
        "description": "name of category"
      },
      "url": {
        "type": "string",
        "description": "url to category"
      }
    },
    "required": ["name", "url"]
  }
}
```

b. Dữ liệu mẫu ví dụ

```
[
  {
    "name": "Home",
    "url": "/"
  },
  {
    "name": "Media",
    "url": "/media.htm"
  },
  {
    "name": "Thời sự",
    "url": "/thoi-su.htm"
  },
]
```

```
{
  "name": "Thế giới",
  "url": "/the-gioi.htm"
},
{
  "name": "Pháp luật",
  "url": "/phap-luat.htm"
},
{
  "name": "Kinh doanh",
  "url": "/kinh-doanh.htm"
},
{
  "name": "Xe",
  "url": "/xe.htm"
},
{
  "name": "Nhịp sống trẻ",
  "url": "/nhip-song-tre.htm"
},
{
  "name": "Văn hóa",
  "url": "/van-hoa.htm"
},
{
  "name": "Giải trí",
  "url": "/giai-tri.htm"
},
{
  "name": "Khoa học",
  "url": "/khoa-hoc.htm"
},
{
  "name": "Sức khỏe",
  "url": "/suc-khoe.htm"
}
```

```

    },
    {
      "name": "Giả-thật",
      "url": "/gia-that.htm"
    },
    {
      "name": "Thư giãn",
      "url": "/thu-gian.htm"
    }
  ]

```

3.2.3.2 Dữ liệu danh sách bài viết

a. Định nghĩa dữ liệu (schema)

```

{
  "title": "Posts",
  "description": "Describe all posts in a category",
  "type": "object",
  "properties": {
    "category": {
      "type": "string",
      "description": "name of current category"
    },
    "url": {
      "type": "string",
      "description": "url to this category"
    },
    "posts": {
      "type": "array",
      "description": "list of post inside",
      "items": {
        "type": "object",
        "description": "representation of a post",
        "properties": {
          "name": {

```

```

        "type": "string",
        "description": "Title of the post",
    },
    "url": {
        "type": "string",
        "description": "url to this category",
    }
},
"required": ["name", "url"]
}
},
"required": ["category", "url", "posts"]
}
}

```

b. Dữ liệu mẫu ví dụ Chú ý dữ liệu dưới đây đã bị cắt bớt và thay bằng dấu ... để phù hợp trang

```

{
  "category": "Pháp Luật",
  "url": "/phap-luat.htm",
  "post": [
    {
      "name": "Dừng chờ đèn đỏ,...",
      "url": "/dung-cho-den-do-m..."
    },
    {
      "name": "Trả hồ sơ, điều tra bổ....",
      "url": "/tra-ho-so-dieu-tra-bo-....."
    },
    {
      "name": "Người cha bắt kẻ đột nhập.....",
      "url": "/nguoi-cha-bat-ke-dot-nhap-vao-...."
    },
    {
      "name": "Thất thoát hàng trăm tỉ tại Agribank...",

```

```
"url": "/that-thoat-hang-tram-ti-tai-agribank-..."
},
{
  "name": "Truy bắt nhóm cướp tài sản....",
  "url": "/truy-bat-nhom-cuop-tai-san-bat-giu-...htm"
},
{
  "name": "Bắt quả tang đại úy 'dỏm'...",
  "url": "/bat-qua-tang-dai-uy-dom-lua-b...."
},
{
  "name": "Xét xử cựu phó giám đốc Sở....",
  "url": "/xet-xu-cuu-pho-giam-doc-so-nong..."
},
{
  "name": "Công an Long An không khởi tố....",
  "url": "/cong-an-long-an-khong-khoi-to-vu-phong-vien-..."
},
{
  "name": "Khởi tố bà nội bé ...",
  "url": "/khoi-to-ba-noi-be-23-ngay-tuoi...htm"
},
{
  "name": "Chủ xe khách đòi công an...",
  "url": "/chu-xe-khach-doi-cong-an-boi....htm"
},
{
  "name": "Làm rõ vụ 2 phóng viên...",
  "url": "/lam-ro-vu-2-phong-vien-nghi-nhan-tien-....htm"
},
{
  "name": "Trục xuất 52 người Trung Quốc...",
  "url": "/truc-xuat-52-nguoi-trung-quoc-nhap-....htm"
},
{
```

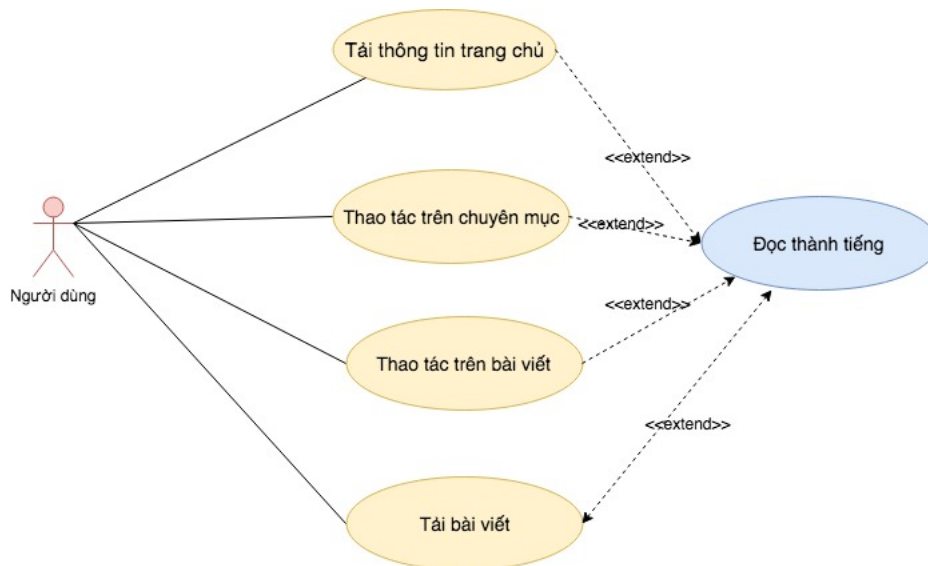
```

    "name": "Truy tổ nguyên lãnh đạo TP....",
    "url": "/truy-to-nguyen-lanh-dao-tp-vung-tau....htm"
  },
  {
    "name": "Tiền Giang triệt phá...",
    "url": "/tien-giang-triet-pha-nhom-chuyen....htm"
  }
]
}

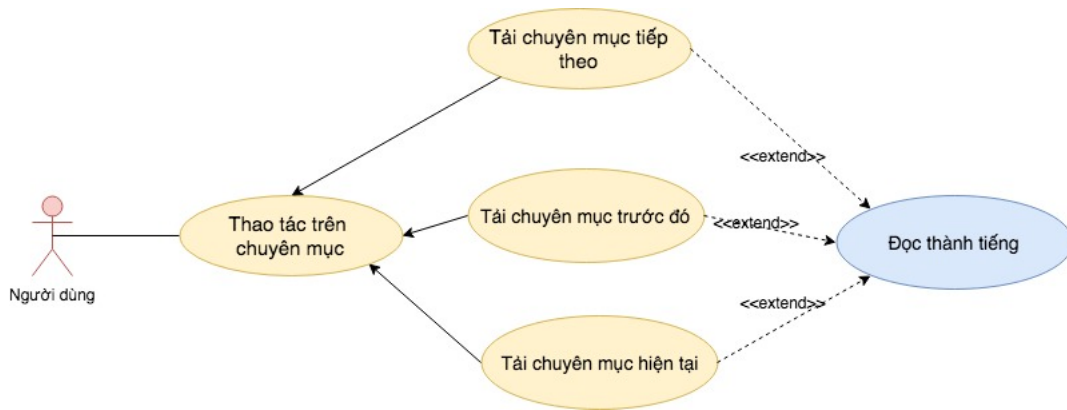
```

3.2.4 Biểu đồ mô tả

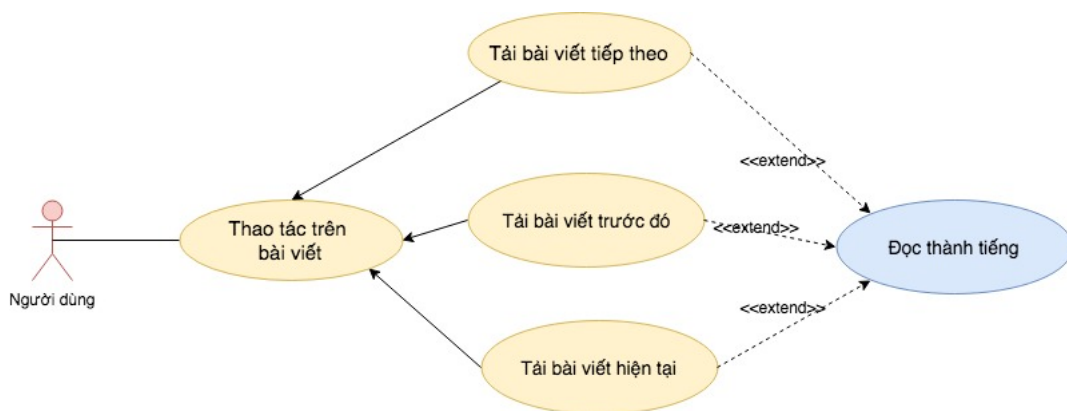
3.2.4.1 Biểu đồ ca sử dụng (Usecase)



Hình 3.5: Biểu đồ ca sử dụng của phần mềm



Hình 3.6: Biểu đồ ca sử dụng với các chuyên mục



Hình 3.7: Biểu đồ ca sử dụng với các bài viết

Chương 4

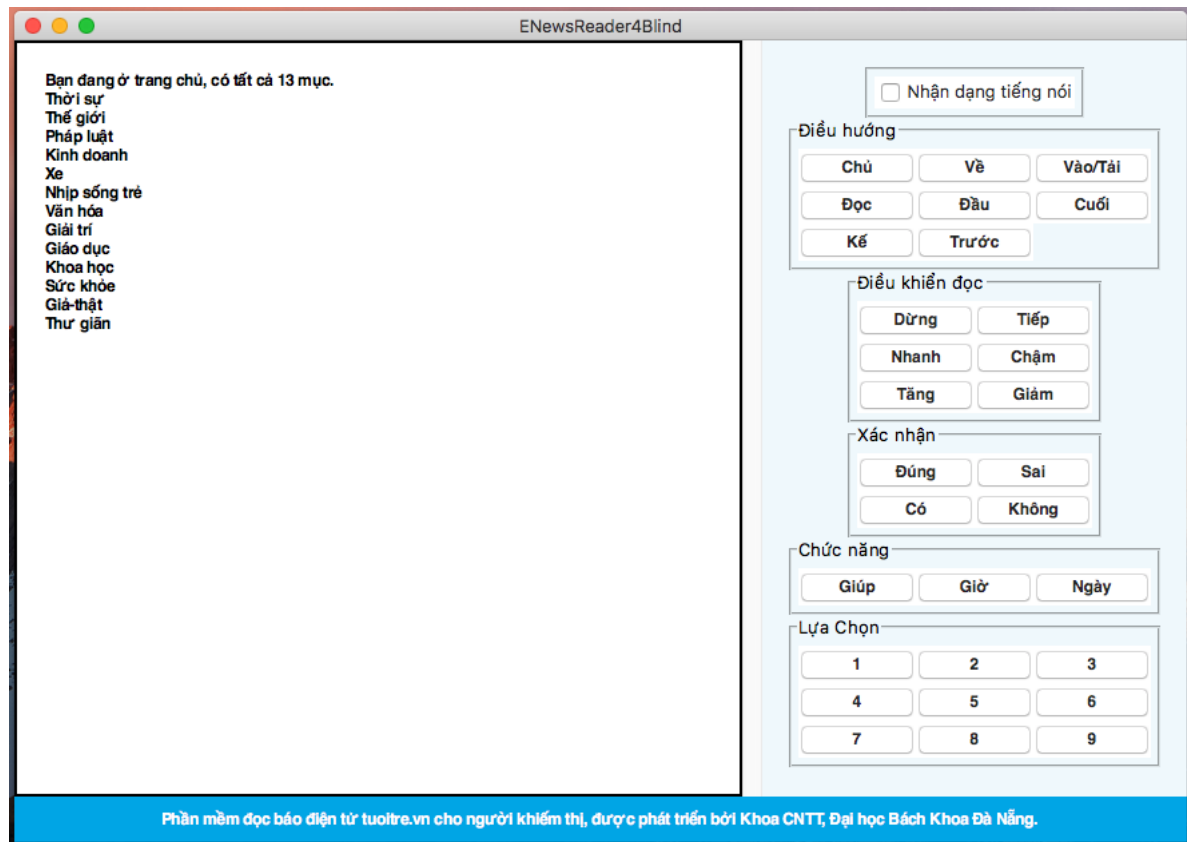
Kết quả chạy phần mềm và Kết luận

4.1 Chạy phần mềm

4.1.1 Hướng dẫn cài đặt

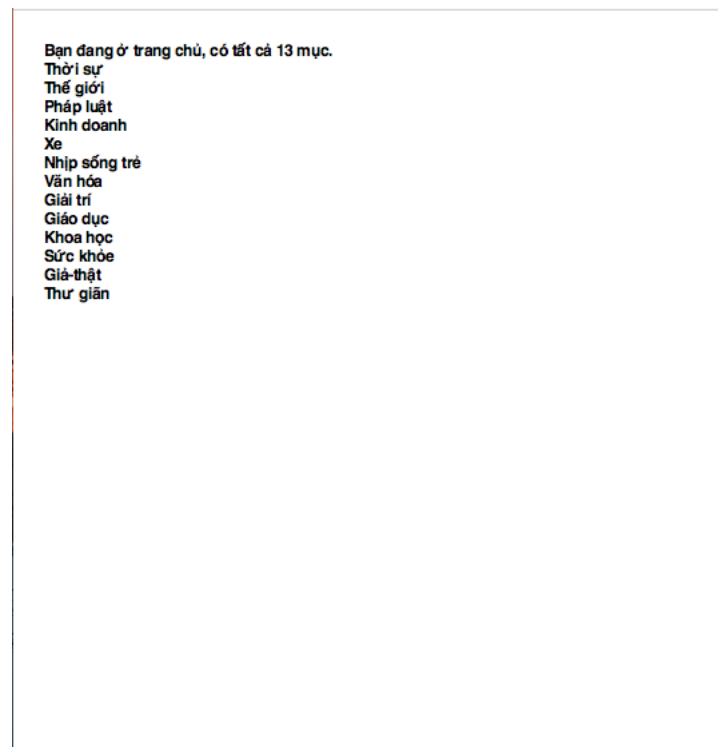
1. Copy bộ mã nguồn vào máy tính
2. Đảm bảo máy của bạn đã cài python3
3. Chạy lệnh: `pip install -r requirements.txt`
4. Chạy ứng dụng bằng cách đứng ở thư mục gốc và chạy: **make run-app**

4.1.2 Hình ảnh phần mềm và hướng dẫn sử dụng

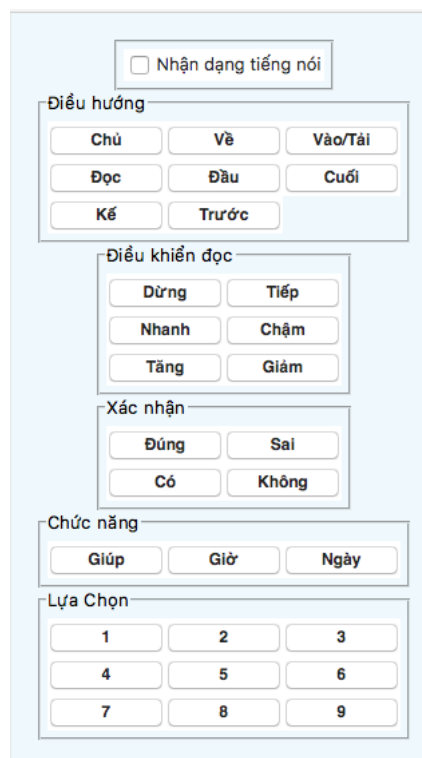


Hình 4.1: Hình ảnh phần mềm khi được khởi chạy

Hình 4.1 là giao diện của phần mềm, bao gồm phần hiển thị ở bên trái, các nút chức năng ở phía bên phải, và thông tin phần mềm ở dưới cùng



Hình 4.2: Khu vực hiển thị của phần mềm



Hình 4.3: Các nút chức năng của phần mềm

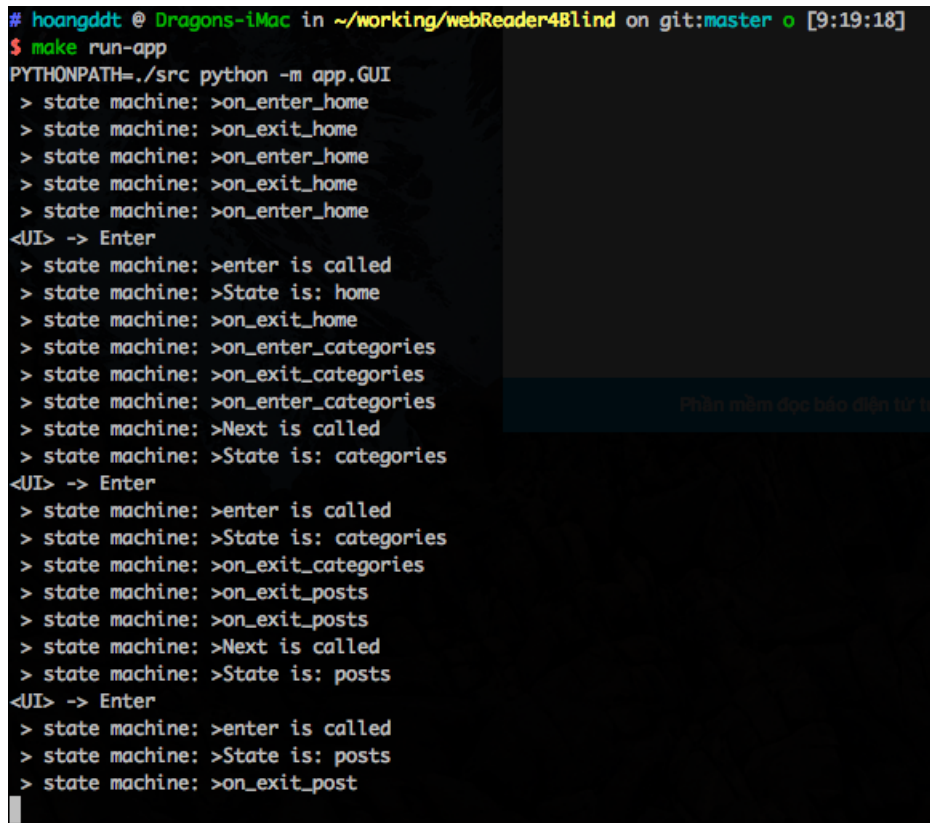
Mô tả chức năng của các nút ở hình 4.3:

- Nhận dạng tiếng nói: Đây là 1 nút giúp cho bật/tắt module nhận dạng tiếng nói.

Khi bật phần mềm sẽ hỗ trợ nhận dạng tiếng nói và thực thi bằng lệnh từ tiếng nói (phần mềm vẫn hoạt động với thao tác trên giao diện). Các tiếng nói để ra lệnh tương tự như nhấn của các nút. VD: ra lệnh bằng việc đọc: *Chủ, Về*,...

Khi tắt thì phần mềm sẽ chỉ hoạt động bằng thao tác trên giao diện.

- Điều hướng: ở đây bao gồm các nút điều hướng trên việc thao tác với trang web, thực hiện các chức năng như tải chuyên mục, bài viết, chọn các chuyên mục bài viết tương ứng.
- Điều khiển đọc: Bao gồm lệnh đọc cho phép đọc lên trạng thái hiện tại, các lệnh khác giúp dừng việc đọc, tăng tốc độ đọc nhanh, chậm và tăng giảm âm lượng của phần mềm.
- Xác nhận: Bao gồm các nút giúp cho việc xác nhận lại các lệnh
- Chức năng: *Giúp* sẽ đọc lên thông tin phần mềm, *Giờ* và *Ngày* giúp thông tin về ngày và giờ hiện tại cho người dùng
- Lựa chọn: Đây là các nút lựa chọn để thao tác với các chuyên mục



```
# hoangddt @ Dragons-iMac in ~/working/webReader48lind on git:master o [9:19:18]
$ make run-app
PYTHONPATH=./src python -m app.GUI
> state machine: >on_enter_home
> state machine: >on_exit_home
> state machine: >on_enter_home
> state machine: >on_exit_home
> state machine: >on_enter_home
<UI> -> Enter
> state machine: >enter is called
> state machine: >State is: home
> state machine: >on_exit_home
> state machine: >on_enter_categories
> state machine: >on_exit_categories
> state machine: >on_enter_categories
> state machine: >Next is called
> state machine: >State is: categories
<UI> -> Enter
> state machine: >enter is called
> state machine: >State is: categories
> state machine: >on_exit_categories
> state machine: >on_exit_posts
> state machine: >on_exit_posts
> state machine: >Next is called
> state machine: >State is: posts
<UI> -> Enter
> state machine: >enter is called
> state machine: >State is: posts
> state machine: >on_exit_post
```

Hình 4.4: Màn hình console khi chạy phần mềm

Màn hình console chủ yếu dùng để in thông tin trạng thái và để debug khi phát triển chương trình, phần này sẽ không cung cấp cho người dùng.

4.2 Đánh giá và kết luận

4.2.1 Đánh giá

Phần mềm đã chạy và đáp ứng được yêu cầu sử dụng của người khiếm thị.

Ưu điểm

- Đã chạy và tải được các chuyên mục, bài viết của báo tuổi trẻ
- Duyệt được giữa các chuyên mục, bài viết và giữ được trạng thái ứng dụng
- Nhận diện giọng nói điều khiển của người dùng khá chính xác trong điều kiện ít nhiễu
- Hỗ trợ cả điều khiển bằng giọng nói và điều khiển bằng nút bấm trên giao diện

- Đọc các trạng thái hiện tại và các bài viết hiệu quả
- Toàn bộ các module phối hợp nhịp nhàng, chạy trơn tru ít khi xảy ra lỗi

Nhược điểm

- Giọng đọc vẫn chưa được hay
- Trong điều kiện ngoài trời, vẫn chưa xử lý được các nhiễu
- Phần giao diện có khá nhiều nút gây khó khăn cho người sử dụng
- Có nhiều module nên khá tốn tài nguyên của máy

4.2.2 Kết luận

Với sự kết hợp giữa các công nghệ hiện đại, phần mềm đã đáp ứng được nhu cầu sử dụng của người dùng khiếm thị. Góp phần giúp những người khiếm thị có thể tự mình thao tác bằng giọng nói, điều khiển để có thể đọc báo online một cách dễ dàng. Giúp họ có thể cập nhật các tin tức hằng ngày một cách nhanh nhất.

Ngoài ra, việc nghiên cứu và áp dụng các module nhận dạng, tổng hợp tiếng nói và phối hợp một cách hoàn chỉnh là một tiền đề để có thể phát triển thêm nhiều phần mềm khác cho người khiếm thị cũng như cho các ứng dụng thông minh hằng ngày.

Chương 5

Hướng phát triển

5.1 Các hướng phát triển tiếp theo

Những mặt cần cải thiện

- Cải thiện module Text To Speech để có giọng đọc ra tốt hơn
- Cải thiện module Speech To Text để có thể xử lý nhiều tốt hơn, xử lý ngôn ngữ tự nhiên tốt hơn
- Phát triển tối ưu hơn để có thể chạy trên các môi trường, hệ thống nhúng khác giúp cho thiết bị có thể đóng gói nhỏ gọn như 1 thiết bị cầm tay hỗ trợ cho người dùng
- Mở rộng phần mềm hỗ trợ có thể hoạt động trên nhiều trang web online khác nữa
- Bổ sung thêm từ điển cho bộ chuẩn hóa Tiếng Việt

Tài liệu tham khảo

- [1] Text-to-speech Synthesis 1st edition.
Paul Taylor
Cambridge University Press, 2009
- [2] Normalization of non-standard words.
Richard Sproat, Alan Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, Christopher Richards
Computer Speech and Language, Volume 15, Issue 3, pp. 287-333, 2001.
- [3] Biểu diễn ngữ cảnh trong khai triển chữ viết tắt dùng tiếp cận học máy.
Ninh Khánh Duy, Nguyễn Văn Quý
Tạp chí Khoa học và công nghệ Đà Nẵng, số 05(114), 2017
- [4] Một giải pháp Việt hóa cách phát âm các từ vựng tiếng Anh trong văn bản tiếng Việt.
Triệu Thị Ly Ly, Bùi Thanh Sơn, Lê Thị Hà Bình, Ninh Khánh Duy
Kỷ yếu Hội nghị quốc gia FAIR, pp. 502-510, Đà Nẵng, 2017.
- [5] Xây dựng module điều khiển bằng giọng nói trong ứng dụng đọc báo điện tử cho người khiếm thị.
Lê Vũ Công Hòa, Hoàng Thị Minh Khanh, Lê Quang Tam, Ninh Khánh Duy
- [6] State Machine Design.
<https://github.com/pytransitions/transitions>
- [7] Finite-State Machines,
<http://www.cse.chalmers.se/~coquand/AUTOMATA/book.pdf>,
, November 20th 2017
- [8] Python Thread-based parallelism,
<https://docs.python.org/3/library/threading.html>