Fennec: Enhancing English Listening Skills with Video-Watching Experiences

Lam Pham, Chen-Hsiang Yu, and Mira Yun

Department of Computer Science and Networking
Wentworth Institute of Technology
Boston, MA 02115, USA
{phaml1, yuj6, yunm}@wit.edu

Abstract

In traditional English education, instructors mainly focus on teaching reading and writing rather than listening and speaking skills. Specifically, there is no standard way to improve listening skills and learners can easily lose their passion in learning. In this research, we try to answer two fundamental questions: (1) how to enhance English listening skill by using available online videos, and (2) how to provide an engaging learning environment for English learners. *Fennec* is a web application that utilizes automatic transcription technology to help non-native English speaker actively engage with various online videos to improve their listening skills. The application is a user-centric design and has gone through traditional Human-Computer Interaction (HCI) practices to improve its usability. The current result shows that *Fennec* not only provides lecture on demand like services with existing online videos, but it also offers automatically generated assessment questions for the users.

Keywords—YouTube videos, transcription, English, listening skills.

I. INTRODUCTION

Listening is a natural way to learn a new language. Previous study has shown that when a person is engaged in communication, approximately 9% is devoted to writing, 16% to reading, 30% to speaking, and 45% to listening [1]. However, traditional English teaching for foreigners normally starts from reading and writing rather than listening and speaking. Specifically, listening is one of the most difficult skills for English learners to improve. In order to improve listening skills, different methodologies have been proposed, such as class activities, online group studies and exercises, etc. Unfortunately, not all people in different countries or environments can freely enjoy such educational resources.

Based on the learners' various demands, many different types of learning tools have been introduced, such as YouTube videos, podcasts, books and mobile applications [1-3]. By using appropriate learning tools, English learners can improve their language skills effectively. Different from speaking and writing skills, however, improving listening skills can be very frustrating and learners can easily lose their passion and engagement in the lesson. Our research aims at answering two specific questions: (1) how to utilize learning tools, such as YouTube videos, as a useful resource for English learner to improve their listening skills, and (2) how to provide learners a different environment where they can actively engage into the lesson rather than just listening.

In this paper, we propose the *Fennec,* a web application that utilizes a transcription technology, to help non-native English speakers actively engage with various online YouTube videos to improve their listening skills. The *Fennec* is capable of automatically generating different assessment questions for the selected YouTube video. The rest of this paper is organized as the following. Section II provides related work that shows various learning tools for learners. Section III presents the details about our web application design and implementation. Finally, we conclude our work in Section IV.

II. Related Work

Different approaches have been used to improve English language. For example, Artyushina et.al [1] introduced podcast as a leaning tool at the undergraduate level. The survey of their study has shown that podcasting is a creative and entertaining technology for individuals in e-Education. Podcasting has also demonstrated its great educational purpose as it widens the students' lexical and grammatical context.

Reading comics books for language learning is another approach. Kocacs and Miller [2] developed the Foreign Manga Reader - a system that helps readers comprehend foreign-language written materials and learn the grammar and vocabularies. By providing the sentence-structure visualization for a dialog, learners can understand the grammar, pronunciations, phrases and individual words. (Figure 1)

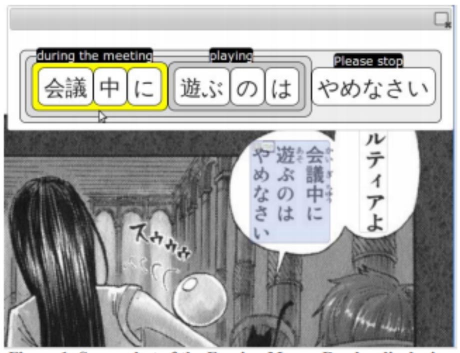


Figure 1. Foreign Manga Reader: Grammar visualization for a dialog [2]

Not only for language learners, how-to videos on the web are one of the most popular and powerful tools. Since millions of learners today use how-to videos to learn new skills in a variety of domains, Kim et. al. [3] created ToolScape application to provide how-to videos with step-by-step annotations. The application offers an interactive video player to display step descriptions and intermediate result thumbnails in the video timeline. It enhances learning experience via existing how-to videos.

III. *FENNEC*

Fennec is a web application that utilizes a transcription technology to improve English listening skills through watching online videos. We choose YouTube as an example to demonstrate the idea, but in fact the idea of the system can be generalized to all kinds of online videos. *Fennec* automatically generates different assessment questions for the selected YouTube video.

3.1 Design

To generate assessment questions automatically for the selected video, the system needs to apply a speech recognition technology to the selected video. There are different available services for speech recognition, such as Google Speech API [4] and Web Speech API [5], and *Fennec* uses Web Speech API to simplify the process. (Figure 2)

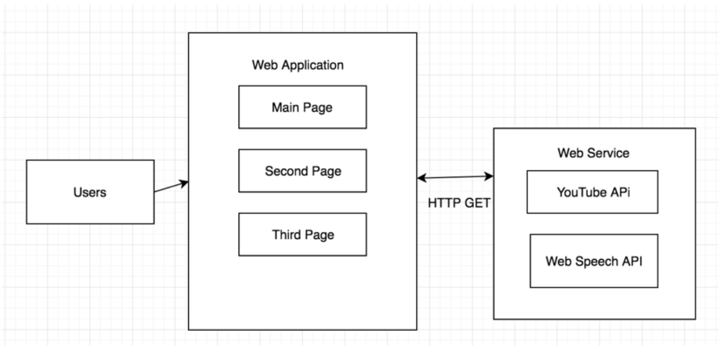


Figure 2. System Architecture of *Fennec*

In terms of the UI design, the *Fennec* cares about the user experience and follows traditional Human-Computer Interaction (HCI) practices to improve its usability. The system is user-centric, and we start from creating a low-fidelity paper prototype to get inputs from the users. The main system is simplified to have only three main pages. (Figure 3)

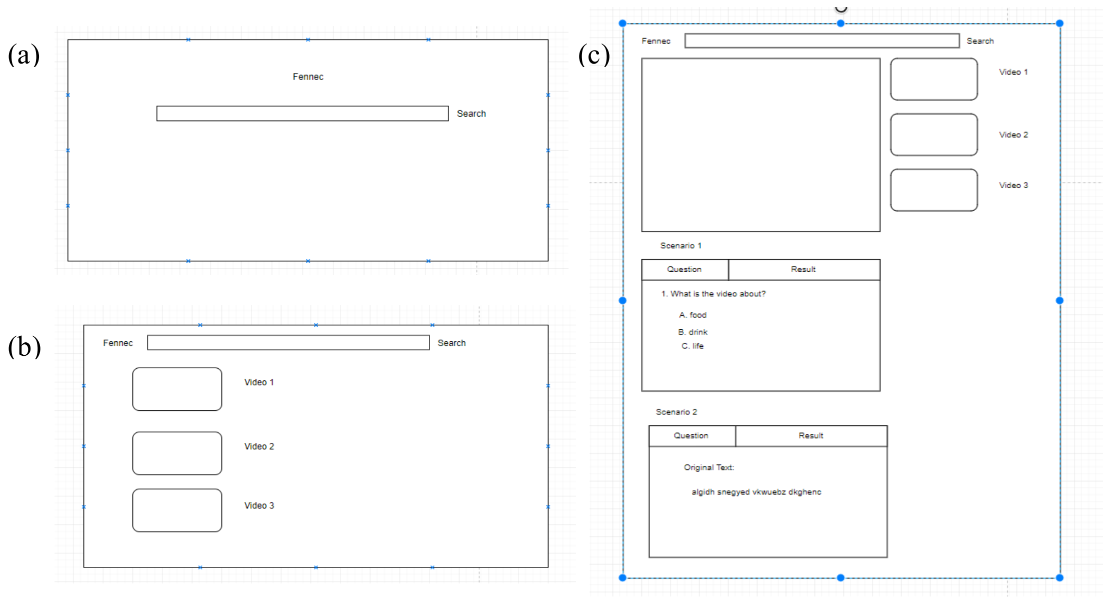
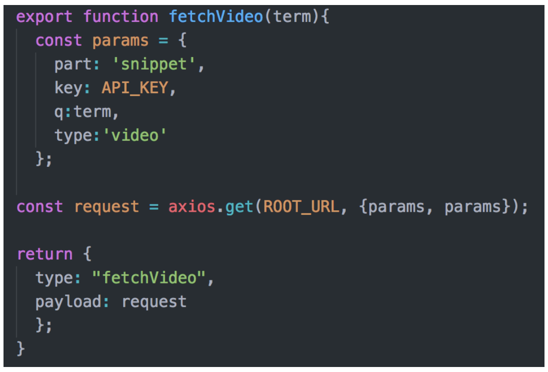


Figure 3. The proposed idea of the system

The main page only has a search bar for retrieving videos with a specific search term. (Figure 3 - (a)) After taking a search term input, the system will be redirected to a second page that has a list of recommended videos. This design references YouTube layout to lower the learnability to the learners. (Figure 3 - (b)) When the user clicks on a specific video, the selected video will be shown on the third page, along with a list of videos on the right-hand side. As Figure 3 – (c) indicates, a multi-tab design is included next to the main video. While the Question tab is used to show assessment questions, which is automatically generated by the *Fennec*, the Result tab can provide the transcribed content.

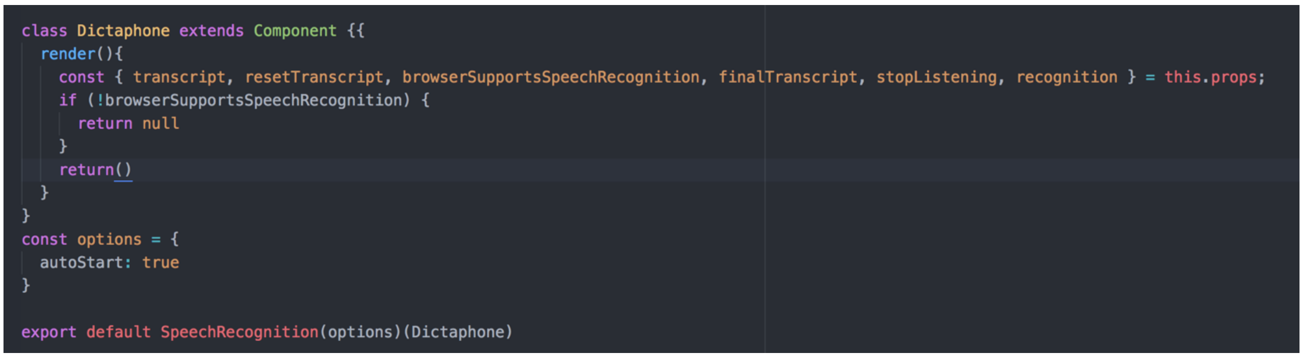
3.2 Implementation

The *Fennec* uses existing web services, including YouTube APIs and Web Speech APIs, to demonstrate the idea. To make the system comply with the state-of-the-art Web technologies, we use React, a JavaScript library for building user interfaces [6], to design the system. Basically, the system uses `fetchVideo()` function with predefined parameters to send a request to YouTube server for retrieving designed information. (Figure 4) In addition, to make an asynchronous HTTP request from the browser, we use `axios`, a promise-based HTTP client for the browser and Node.js. [7]



```
export function fetchVideo(term){
  const params = {
    part: 'snippet',
    key: API_KEY,
    q:term,
    type:'video'
  };

const request = axios.get(ROOT_URL, {params, params});

return {
  type: "fetchVideo",
  payload: request
  };
}
```

Figure 4. Handle HTTP GET request to YouTube

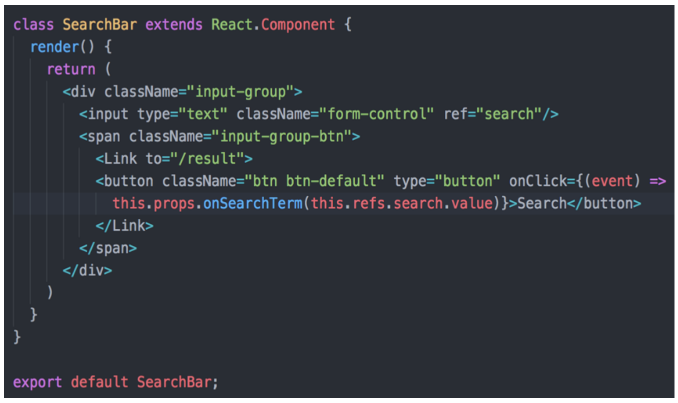


```
class Dictaphone extends Component {{
  render(){
    const { transcript, resetTranscript, browserSupportsSpeechRecognition, finalTranscript, stopListening, recognition } = this.props;
    if (!browserSupportsSpeechRecognition) {
      return null
    }
    return()
  }
}
const options = {
  autoStart: true
}

export default SpeechRecognition(options)(Dictaphone)
```

Figure 5. Speech Recognition feature

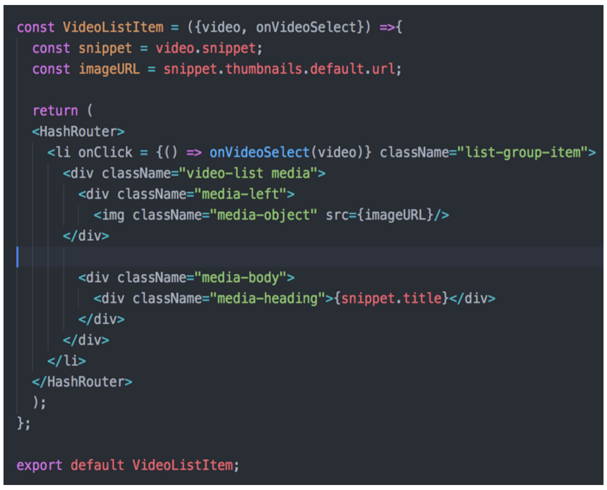
Some main functionalities of Web Speech API are injected to `this.props` for passing down to some child components. `Props` is a single object in a component that is passed as a JSX attribute. Unfortunately, Speech Recognition does not work well as a component, so it is set as a container. Hence, other components can be rendered within the Third Page Component.

(a)



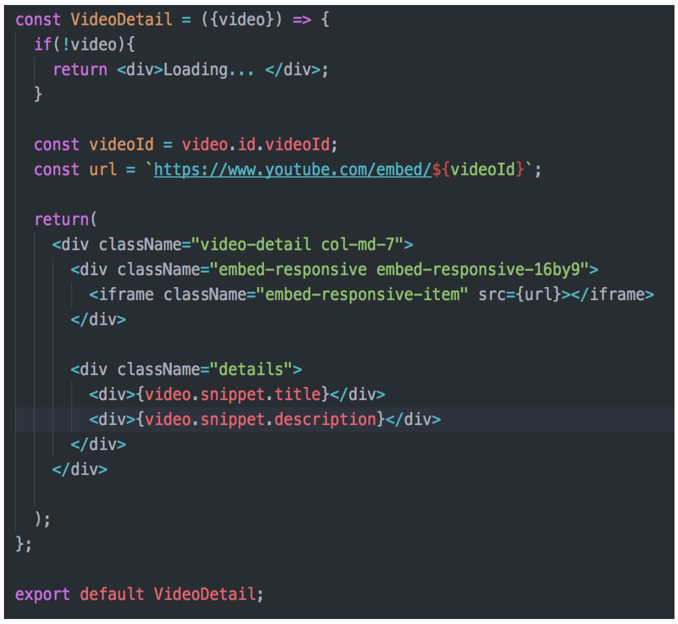
```
class SearchBar extends React.Component {
  render() {
    return (
      <div className="input-group">
        <input type="text" className="form-control" ref="search"/>
        <span className="input-group-btn">
          <Link to="/result">
          <button className="btn btn-default" type="button" onClick={(event) =>
            this.props.onSearchTerm(this.refs.search.value)}>Search</button>
          </Link>
        </span>
      </div>
    )
  }
}

export default SearchBar;
```

(b)



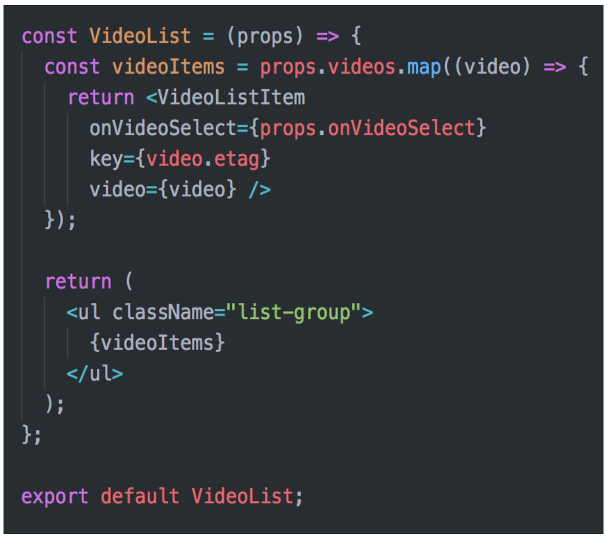
```
const VideoListItem = ({video, onVideoSelect}) =>{
  const snippet = video.snippet;
  const imageURL = snippet.thumbnails.default.url;

  return (
  <HashRouter>
    <li onClick = {() => onVideoSelect(video)} className="list-group-item">
      <div className="video-list media">
        <div className="media-left">
          <img className="media-object" src={imageURL}/>
      </div>

        <div className="media-body">
          <div className="media-heading">{snippet.title}</div>
        </div>
      </div>
    </li>
  </HashRouter>
  );
};

export default VideoListItem;
```

(c)



```
const VideoDetail = ({video}) => {
  if(!video){
    return <div>Loading... </div>;
  }

  const videoId = video.id.videoId;
  const url = `https://www.youtube.com/embed/${videoId}`;

  return(
    <div className="video-detail col-md-7">
      <div className="embed-responsive embed-responsive-16by9">
        <iframe className="embed-responsive-item" src={url}></iframe>
      </div>

      <div className="details">
        <div>{video.snippet.title}</div>
        <div>{video.snippet.description}</div>
      </div>
    </div>

  );
};

export default VideoDetail;
```

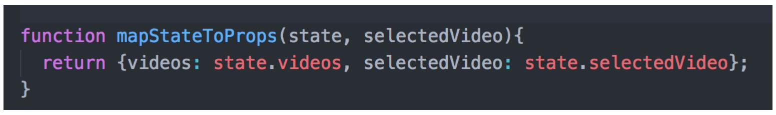
(d)



```
const VideoList = (props) => {
  const videoItems = props.videos.map((video) => {
    return <VideoListItem
      onVideoSelect={props.onVideoSelect}
      key={video.etag}
      video={video} />
  });

  return (
    <ul className="list-group">
      {videoItems}
    </ul>
  );
};

export default VideoList;
```

Figure 6. Code snippets of some main components in the *Fennec*. (a) search bar: get the user's input query for retrieving videos, (b) list video items: load information for each video in the list, (c) video detail: load a specific videoID with its title and description, (d) video list: return a list of 5 videos.

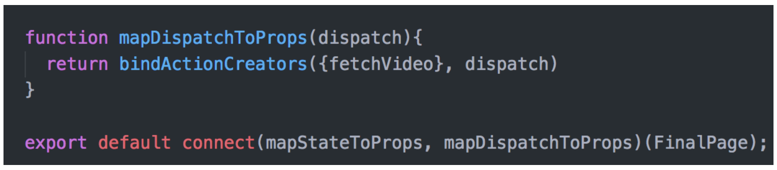
These main components will be rendered into three containers: Main Page, Second Page, and Third Page respectively. `mapStateToProps()` function is the most important function defined for containers. It stores data into components. As Figure 7 indicates, it returns two application states, i.e. listed videos and the selected video.



```
function mapStateToProps(state, selectedVideo){
  return {videos: state.videos, selectedVideo: state.selectedVideo};
}
```

Figure 7. mapStateToProps() function

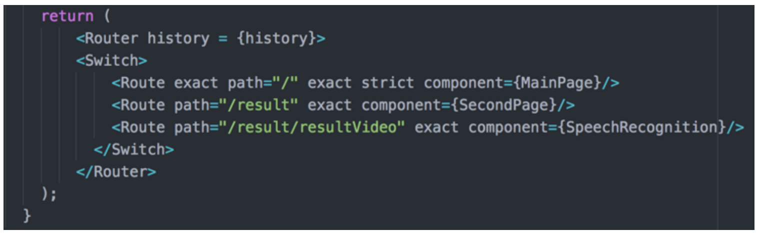
Another essential function is `mapDispatchToProps()`, which provides action creators as a props to the components. In this case. `fetchVideo()` function is called as an action creator that is required to be bind with a dispatch call to merge into components.



```
function mapDispatchToProps(dispatch){
  return bindActionCreators({fetchVideo}, dispatch)
}

export default connect(mapStateToProps, mapDispatchToProps)(FinalPage);
```

Figure 8. mapDispatchToProps() functions

For the Main Page, Router is initialized for the other two containers. Basically, Router initializes a routing path that associates to specific containers. This part is only required to define once in the first container, then it can be used throughout the whole application.



```
  return (
      <Router history = {history}>
      <Switch>
          <Route exact path="/" exact strict component={MainPage}/>
          <Route path="/result" exact component={SecondPage}/>
          <Route path="/result/resultVideo" exact component={SpeechRecognition}/>
        </Switch>
      </Router>
  );
}
```

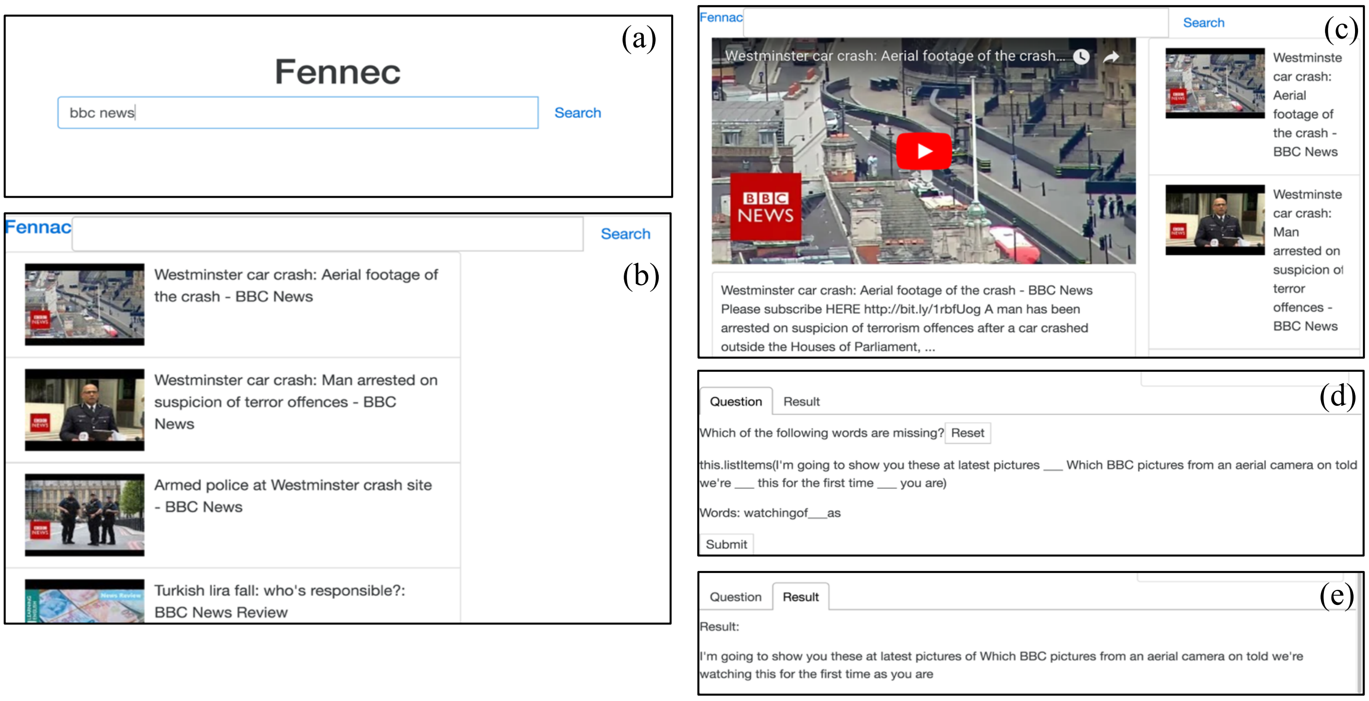
Figure 9. Routing between containers

3.3 Results

The *Fennec*, a Web application for enhancing English listening skills with video-watching experiences, is illustrated as Figure 10. The application is user-centric and easy-to-use. It simply contains three main pages, as explain in section 3.1. The key features of the system include:

1. Query a key word and retrieve a list of related videos
2. Provide a learning mode to watch the selected video
3. Transcribe the watching video with Speech Recognition technique
4. Generate assessment questions automatically

The *Fennec* project has an impact for the users whose first language is not English. From technical perspective, handling data flow and managing states of the whole application are trivial. Compared with traditional HTML and JavaScript approaches, it took several steps to bind transitions between pages using React [6]. There are some improvements that can be made for the system. For example, when clicking on a certain video on page two (Figure 10 - (b)), the transitioning does not function smoothly because it keeps loading the layout before getting back data. Moreover, although generating assessment questions from a transcribed text is a success, the ideal operation of the drag and drop is not finished at this moment. Due to the inconsistent states of React state, the search results often return back unrelated videos.



Westminster car crash: Aerial footage of the crash - BBC News

Westminster car crash: Man arrested on suspicion of terror offences - BBC News

Armed police at Westminster crash site - BBC News

Turkish lira fall: who's responsible?: BBC News Review

Westminster car crash: Aerial footage of the crash - BBC News
Please subscribe HERE http://bit.ly/1rbfUog A man has been arrested on suspicion of terrorism offences after a car crashed outside the Houses of Parliament, ...

Westminster car crash: Aerial footage of the crash - BBC News

Westminster car crash: Man arrested on suspicion of terror offences - BBC News

Question Result

Which of the following words are missing? Reset

this.listItems(I'm going to show you these at latest pictures ___ Which BBC pictures from an aerial camera on told we're ___ this for the first time ___ you are)

Words: watchingof___as

Submit

Result:

I'm going to show you these at latest pictures of Which BBC pictures from an aerial camera on told we're watching this for the first time as you are

Figure 10. (a) Fennec main page with a search query, (b) a list of returned videos, (c) a learning mode of video watching, (d) assessment questions, and (e) a transcript generated by speech recognition technique

IV. Conclusion and Future Work

Learning English should be an interesting experience, but traditional English education mainly focuses on teaching reading and writing rather than listening and speaking skills. As indicated in [1], when a person is engaged in communication, 45% are devoted to listening. There is a gap between time used in training listening skills and the importance of this skill. Unfortunately, there are not available tools designed to improve English listening comprehension. In this research, we try to answer two fundamental questions: (1) how to enhance English listening skills by using available online videos, and (2) how to provide an engaging learning environment for English learners.

We propose *Fennec*, a new web application designed for enhancing listening skills for English learners. The application is simple and user-centric. It not only went through traditional HCI practices for usability enhancement, but it also investigated possible techniques to automatically generate assessment questions for listening comprehension. The current result indicates that *Fennec* can provide lecture on demand like services with online videos. In addition, the assessment questions can be automatically created for any selected video. The future work of this research includes improving UI design for interactions, extending the system to support speaking enhancement and adding support for mobile devices.

References

[1] Artyushina, Galina G, et al. "Podcasting as a Good Way to Learn Second Language in e-Learning." Communications of the ACM, ACM.

[2] Kovacs Geze and Robert C. Miller. "Foreign Manga Reader: Learn Grammar and Pronunciation while Reading Comics." Proceedings of the adjunct publication of the 26th annual ACM symposium on User interface software and technology (UIST), 2013.

[3] Juho Kim, Phu Tran Nguyen, Sarah Weir, Philip J. Guo, Robert C. Miller and Krzysztof Z. Gajos. "Crowdsourcing step-by-step information extraction to enhance existing how-to videos," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, April 26-May 01, 2014, Toronto, Ontario, Canada.

[4] “Cloud Speech-to-Text Documentation | Cloud Speech-to-Text API | Google Cloud.” Cloud Speech-to-Text Documentation, Google, cloud.google.com/speech-to-text/docs/

[5] len, Shires, and Hans Wennborg. “Web Speech API Specification.” Web Speech API Specification, 19 Oct. 2012

[6] React - A JavaScript library for building user interfaces: https://reactjs.org/

[7] axios - Promise based HTTP client for the browser and node.js: https://github.com/axios/axios