Text Summarization and Speech Synthesis for the Automated Generation of Personalized Audio Presentations

Séamus Lawless, Peter Lavin, Mostafa Bayomi, João P. Cabral, and M. Rami Ghorab

CNGL Centre for Global Intelligent Content, Knowledge and Data Engineering Group, School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland
{seamus.lawless,peter.lavin,bayomim,cabralj,rami.ghorab}@scss.tcd.ie

Abstract. In today’s fast-paced world, users face the challenge of having to consume a lot of content in a short time. This situation is exacerbated by the fact that content is scattered in a range of different languages and locations. This research addresses these challenges using a number of natural language processing techniques: adapting content using automatic text summarization; enhancing content accessibility through machine translation; and altering the delivery modality through speech synthesis. This paper introduces Lean-back Learning (LbL), an information system that delivers automatically generated audio presentations for consumption in a “lean-back” fashion, i.e. hands-busy, eyes-busy situations. These presentations are personalized and are generated using multilingual multi-document text summarization. The paper discusses the system’s components and algorithms, in addition to initial system evaluations.

Keywords: Lean-back learning · Text summarization · Speech synthesis · Multilingual content adaptation, personalization

1 Introduction

The constantly connected nature of today’s world places increasing demands on people’s time. When coupled with the tsunami of information that is competing for individuals’ attention on a daily basis, users typically face the challenge of having to consume large volumes of content in short periods of time. Moreover, the naturally distributed and inherently multilingual nature of the web creates a further challenge.

The terms “lean forward” and “lean back” can be used to characterize a user’s engagement with a computing device. In lean-forward engagement a user is focused on the device and constantly interacts with the system. An example of this form of engagement is a user reading a report online or using spreadsheet software. In contrast, in lean-back engagement the user’s focus can be elsewhere. Interaction with the device is minimal, yet they are still consuming information. Earlier forms of this type of engagement are radio and television. However, over recent decades, miniaturization coupled with widespread Internet connectivity has meant that media consumption with this low level of engagement can now also take place using mobile technology.
Allowing users to consume traditional web content in a lean-back fashion begins to tackle the challenges outlined above. This paper introduces Lean-back Learning (LbL), an intelligent and responsive information system built upon a set of web services. The system delivers automatically generated audio presentations for consumption in a lean-back fashion, i.e. hands-busy, eyes-busy contexts. Audio presentations are generated using automatic text summarization and speech synthesis. The audio output is provided in multiple languages and is compiled, translated and summarized using multilingual sources. The user can tailor presentations by selecting the preferred level of detail and output language, and specifying their available listening time.

Lean-back Learning allows a user who wishes to learn about a topic, to input one or more search terms. The system then presents the user with a summarized audio presentation, which they can listen to at their leisure. This application can be used in a variety of scenarios, for instance: a tourist who is about to visit a cultural site and who wants some last-minute background information; a student who wishes to get an overview of a new topic that they are about to learn; or a person who wishes to listen to a summary of a piece of news. The users can listen to this information, tailored to their needs, all while being able to carry out other tasks.

This paper discusses the components and algorithms of the proposed LbL system, and presents a usage scenario. It also presents an evaluation of the text summarization component of the framework. This is the first in a series of planned experiments which will evaluate the framework from quantitative and qualitative perspectives. The paper also addresses general questions on the feasibility of dynamically synthesizing summarized text. Areas such as practical and acceptable response times and management of users’ expectations around the amount of content available are discussed.

The remainder of this paper is organized as follows. Section 2 presents state-of-the-art in the areas of Summarization and Speech Synthesis. Section 3 describes the proposed LbL framework (architecture, data flow, and user interface). Sections 4 and 5 discuss the two main underlying services of the framework: SSC (Search, Summarize, and Combine) and SSyn (Speech Synthesis). Following that, as part of the research underpinning this framework, an evaluation of the framework components is presented in Sect. 6. Finally, insights and future work are explored Sect. 7.

2 Background and State-of-the-Art

As Text Summarization and Speech Synthesis are the core services used by the LbL framework, this section provides a state-of-the-art review of research in those areas.

2.1 Text Summarization

Automatic text summarization is a prominent research area in the field of Natural Language Processing (NLP). Research in this area led to the development of various summarization techniques as well as the application of those techniques in diverse domains and on a variety of content bases, such as scientific repositories and meeting recordings [1]. The two main approaches to text summarization are Abstraction [2] and
Extraction [3]. Abstraction is where the summarization system has to “understand” what the text means in order to build an internal semantic representation of the content. Natural language generation techniques are then used to create a summary that is deemed close to what a human being might create. A significant challenge that faces the Abstraction approach is that it requires accurate semantics and training data to be able to automatically interpret the meaning of the content. Because of this challenge, summarization techniques that are based on Abstraction have thus far only exhibited limited success [4]. On the other hand, the Extraction approach, such as LexRank [5] does not require that the summarization system understand the meaning of the text; rather, it attempts to identify the most important sentences in order to extract them into a summary. The system proposed in this paper is based on the Extraction approach as, to-date, it has been shown to be more effective.

Various methods have been discussed in the literature to improve extractive summarization systems. Early systems depended on simple methods, such as: Sentence Location [7], Cue Phrase [8], Most Frequent Words [9], and Sentence Length [8].

The LbL framework introduced in this paper uses TextRank [6] as its foundation. TextRank, one of the most prominent Extraction-based methods, is an unsupervised graph-based ranking model that summarizes text by extracting and ranking the most important sentences according to the number of overlapping words between them.

2.2 Speech Synthesis

The most popular speech synthesis approaches in the literature are Unit-Selection and Hidden Markov Model (HMM) based speech synthesis. A comparison of the two approaches is given in [10]. In summary, the Unit-Selection approach is based on the concatenation of pre-recorded speech units. This approach offers limited flexibility for controlling voice characteristics. Moreover, the type of voice is restricted to that of the recorded speech. On the other hand, the HMM-based approach is fully parametric and the acoustic modelling of speech is learned automatically by training the statistical models from a recorded speech corpus. Its advantage is that it permits the transformation of the synthesizer’s voice type using a small amount of data from the target voice. HMM-based speech synthesis can produce high-quality speech but generally does not sound as natural as that of unit-selection systems [10].

There have been several recent attempts to combine the advantages of statistical approaches with the “naturalness” obtained using unit-selection. Current state-of-the-art systems, such as the system reported in [11], are hybrid systems that are based on the unit-selection approach but use HMMs for selection of the concatenation units. In spite of the relatively high quality voice output produced by these systems, there is a growing demand for increased expressiveness of synthetic speech that is beyond what can be currently produced [12], e.g. audiobooks, spoken dialogue systems, etc. For example, synthesizing voices with different speaking styles using a HMM-based system is one way to improve the speech expressiveness in audiobooks [13]. The prosodic aspects of the synthetic speech, such as pause duration and intonation along the phrase are also important for engaging the listener in the communication process and need to be better modelled by the synthesizers.
The LbL framework uses popular open source speech synthesis systems (HMM-based and unit-selection) for investigating the limitations of this technology in the generation of audio presentations. We aim to identify ways to improve the quality of the standard speech synthesis systems, particularly for this type of application. The HMM-based method is typically preferred in platforms such as LbL, because it is more suitable for mobile devices (faster speech synthesis and lower memory footprint) and it offers higher flexibility for personalizing the voice. For example, using HMM adaptation algorithms it is possible to build a new speaker’s voice using a small amount of speech data from that speaker [14].

3 Lean-Back Learning Portal

LbL aims to provide an end-to-end platform for retrieving, adapting, and narrating concise knowledge. The system therefore orchestrates all the elements of the process: interfacing with the user, retrieving information, translation and summarization, and synthesizing voice narrations. This section discusses how the system controls the underlying components, the workflow and user interface. Details of the underlying algorithms of each component are given in the later sections.

The framework currently uses Wikipedia as its content source. The following reasons were behind the choice of Wikipedia: (1) it is an open and multi-domain content source; (2) it features content in multiple languages; and (3) Wikipedia articles are well and consistently structured, which facilitated producing tailored summaries that vary in the amount of content and level of detail (depending on the user’s needs). The framework is, however, not functionally tied to Wikipedia as a content source. LbL could function just as effectively on any body of structured data.

3.1 Framework Architecture

The LbL framework is made up of three components, each of which can stand alone as an independent web service: Search, Summarize & Combine (SSC); Speech Synthesis (SSyn); and the Sequence Controller (SC), which also incorporates the user interface (UI) component. The overview is shown in Fig. 1.

The framework’s ‘modular’ design is intentional and uses standard Internet protocols (RESTful and HTTP posts) for communication between components. This allows components to be altered, added or replaced without impacting on other parts of the framework. The SC service also uses a database to store details of each presentation prepared. For design and performance monitoring purposes, it also records times taken for the SSyn and SSC components to complete their roles.

3.2 Workflow and User Interface

The LbL UI is developed using responsive design to ensure a consistent experience across all devices. The user has the ability to sign in using their Google + account. Once logged in, s/he can specify the topic using free-text keyword search, then select
the level of detail required: High-level Overview, Detailed Introduction, or All Details. The ‘level of detail’ selected governs the number of sections and the depth of sub-sections from the Wikipedia article to be included in the automatically generated summary. Next, the user selects their required output language (three languages are currently supported by the system: English, French, and German). The user’s input is submitted to the SC component which passes it on to the SSC service. SSC carries out an initial evaluation for this topic and estimates the amount of content available. Three word-count values are returned, each one corresponding to one of the three levels of detail available in the UI.

Back at the UI, estimated minimum, intermediate and maximum durations for the audio presentation are calculated and presented to the user for selection; these are based on the aforementioned word-counts and use an empirically obtained ‘word-count to audio seconds (duration)’ ratio. Figure 2 shows the UI and a typical range of durations. The user can then select their desired duration. The presentation length selected is used within the SSC service to determine what level of summarization is applied to the sections used in the relevant article. For this work, summarization varies between 5 % and 80 % of the source text.

This sequence is interactive. If the user’s initial expectations on presentation duration are not met or are exceeded, the user has two options as follows. They can select another level of detail as appropriate and thereby obtain a greater or lesser number of sections and sub-sections from the source article. Alternatively, they can select a different level of summarization by choosing different time options. The initial feedback provided, along with the user interaction around the amount of available content serves to ameliorate the unpredictability caused by variation in articles, both in terms of their length and depth of detail. When the user is satisfied and specifies their final desired presentation length, the details are submitted for processing. From this point onwards, the sequence of events is controlled by the SC. The topic strings, level of detail and selected presentation duration are sent to the SSC service as a tuple. The SSC then returns the summarized content in XML format.
3.3 User Response Times

For usability, it is desirable that the user has the shortest possible wait-time before audio playback begins. This time must include the time taken for content retrieval, summarization, and speech synthesis. A long wait-time here would have a significant impact on the usability of the system. To address this, rather than synthesizing the entire presentation at once, the XML output of the SSC component is parsed by the SC and divided into a number of small chunks of text. As soon as the first segment is created, it is immediately sent to the SSyn service to allow the user to start listening.

In LbL, Speech Synthesis takes place faster than audio playback. The first audio files are created relatively shorter than subsequent ones. This ensures that, once listening has commenced, there is always more synthesized audio available than the user has consumed. After the first segment is synthesized, listening and synthesis take place simultaneously and items on the playlist become available.

4 Search, Summarize, and Combine

This section describes the underlying process for retrieving, summarizing, translating, and combining content. The section also describes how the content that is output by the summarization process is prepared for speech synthesis.

The SSC service has a number of modules which interact in order to deliver the required summarized output. The service pipeline is shown in Fig. 3. An Information Retrieval (IR) module is used to search across Wikipedia and retrieve links to pages in multiple languages (English, French and German) which are relevant to the topic in question. Wikipedia APIs are used to retrieve the text of these pages. The information in Wikipedia articles of different languages is not necessarily the same (i.e. the articles are typically natively authored in each language and are not translations of each other). So the advantages of initially sourcing content in different languages are: (1) to extend the search to a wider content-base; and (2) obtain multiple perspectives on the same topics (e.g. political events). If this approach is extended to minority languages, detailed presentations could be generated where very limited source content exists.

A Slicer module is used to slice each page into sections using the embedded Wikipedia structure. The Multilingual Summarizer component automatically generates summaries...
for each content slice in the source language of the text. The Machine Translation (MT) component\(^1\) then translates the summaries from their source language into the output language that was specified by the user. The Merger combines the summaries from all the documents and merges those which contain information about the same aspect of the topic. Finally, the Multi-document Summarizer produces a final summary by eliminating redundant information that may exist after merging.

![Diagram of the SSC service pipeline](image)

**Fig. 3.** The SSC service pipeline

The summarizer module uses a number of processing steps in order to identify and extract the most important sentences from an article. This extractive task is unsupervised and uses the TextRank algorithm. TextRank measures the similarity between sentences by calculating the number of overlapping words between them. Before the comparison of sentences takes place, stopwords are removed and then the remaining words are stemmed using the Lancaster stemmer (following on the successful approach presented in [23]). The sentence location approach [7] is applied to extract the first sentence from the original article to be used as the first sentence in the summary. The SSC service works as follows:

1. The SC sends four parameters to the SSC service: (a) the user’s query; (b) desired level of detail; (c) desired presentation duration; and (d) output language.
2. The service starts by calling language detector API to identify the query language.
3. The IR module is then called to search across Wikipedia. The result is a Wikipedia link to a page that contains the article that the user searched for.
4. This link is used to retrieve links to the same article in the remaining two languages. The articles (one for each language) are retrieved as HTML.
5. The slicer segments each article into slices according to the page structure in Wikipedia (i.e. main sections, sub-sections and sub-sub sections). This is used to personalize the final presentation. For example, if a user requests a “High-level

---

\(^1\) The MT service used in the experiments is Bing Translation API. The following are the reasons for choosing Bing: (a) it is generally known to perform relatively well in terms of translation quality and speed; (b) it supports a range of languages; and (c) it provides a well-defined RESTful Web service to communicate with it.
Overview”, only the top-level hierarchical sections are used. The final output from the slicer is three groups of slices: all the slices of the article for each of the languages.

6. Each slice is summarized using the TextRank algorithm. The length of the summary generated is dictated by the desired duration that the user has specified.

7. After the slices are summarized, a machine translation module starts to translate slices from their source language to the output (user-specified) language.

8. After translation, the SSC has three groups of slices, all in the output language. The merger module combines slices from these groups by comparing the titles of slice pairs. Slices that have no match are included in the final output without merging. The merging process can produce slices which have duplicated or redundant information. The multi-document summarizer extracts the most important sentences from the merged content after ranking them. It calculates the score of each sentence by comparing it with all the other sentences in the slice and calculates the word overlap between sentences. If the overlap score between sentences exceeds a pre-defined threshold (0.5), the two sentences are considered to be near duplicates, and the translated sentence is removed.

After the final summarization, slices are combined in one single document and then converted into an XML file for delivery to the SC component.

5 The Speech Synthesis Process

The LbL framework is designed so that users can consume information in a lean-back fashion, i.e. hands-busy, eyes-busy contexts. Therefore, one of the main features of the implemented system is the generation of audio presentations based on the summarized content received from the previous step.

The speech synthesis process comprises a text analysis component which extracts the linguistic features that are used for building the synthetic voice and for processing input text at the speech synthesis stage. This module includes tools for text normalization, and Natural Language Processing (e.g. part-of-speech tagging, grapheme-to-phoneme conversion and intonation prediction). Typically, this part of the synthesizer is strongly language-dependent and it is necessary for the system to function effectively, regardless of the approach used (e.g. unit-selection or HMM-based).

The English HMM-based synthetic voice used in this framework was built using the HTS-2.2 toolkit [15]. The speech data used to build the English voice is the female US SLT subset of the CMU ARCTIC speech database [16]. The text analysis part of voice building was performed using Festival Multisyn [17]. The system uses the STRAIGHT vocoder [18] to extract spectrum and aperiodicity parameters from the signal during analysis. F0 is the other speech parameter which is estimated using the Entropic Speech Tools implementation of the RAPT algorithm [19].

For acoustic modelling, the system uses a five-state Hidden Semi-Markov Model (HSMM) structure. The F0 parameter vector (including its delta and delta-delta features) is modelled by multi-space probability distribution HMM (MSD-HSMM), whereas the spectrum and aperiodicity streams (including dynamic features) are
modelled by HSMM using continuous distributions respectively. The F0, spectrum and aperiodicity parameters are clustered using different decision trees, because these parameters have their own contextual factors.

During synthesis, the speech parameters are generated from the input text and trained HSMMs, using a parameter generation algorithm based on the maximum likelihood criterion. Finally, the speech waveform is produced from the speech parameters using the STRAIGHT vocoder. For the provision of French and German languages in the service, the LbL framework currently uses the publically available Unit-Selection MARY TTS system [20].

6 Evaluation

Due to the multi-layered nature of the LbL system, a single method of evaluation would be insufficient to effectively evaluate the performance of each component. Therefore, the plan for evaluation consists of several phases that involve both quantitative and qualitative evaluation. The quantitative evaluation is concerned with the performance and effectiveness of the various components and algorithms that make up the framework. The qualitative evaluation is concerned with the user’s perception of the service in a lean-back situation, the informational value of the generated summaries, and preference for synthetic audio presentations as compared to text form. The aim is to evaluate both the combination and the compartmentalization of all the elements that make up the LbL system. In this paper, we focus on the evaluation of the SSC component, which is a core component to the framework. We also provide an outline of the planned evaluation for the SSyn component.

6.1 SSC Component Evaluation

Throughout the evolution of approaches to automatic text summarization, many evaluation studies [3, 21] have been based on automated evaluation systems such as ROUGE [22]. ROUGE is a tool which includes several automatic evaluation methods that measure the similarity between summaries. However, automatic evaluation lacks the fine-grained, nuanced judgments of quality which can only be achieved through human evaluation. This includes quantified assessments of qualities such as readability/understandability, informativeness, conciseness, and the overall quality of the summary. To this effect, in this paper a human evaluation is carried out to assess the quality of the summaries produced by the SSC service. This is achieved via human judgments of the overall quality of the summaries produced as well as the quality of the summaries in specific subject domains. As different subject domains (e.g. Politics, Sports) use different terminology, language-styles, and structure, these factors would affect the performance of automatic summarization algorithms; hence, we evaluate the knock-on impact of this on users.

While one of the elements in the evaluation is the quality of the generated summary with respect to the source text, the quality of the source text itself is out of scope of this
study. Furthermore, LbL functions independent of the structure of the underlying content source and is designed to allow different content sources to be utilised.

**Experimental Setup.** The aim of the experiment is to evaluate the quality of summaries in general and for specific subject domains. A document-set of 25 abstracts of Wikipedia articles was selected from six subject domains. Abstracts have different lengths (ranging from approx. 180 words to more than 560 words.) The subject domains were selected at random and the articles were randomly chosen from these domains. The selected domains were: Accidents, Natural Disasters, Politics, Famous People, Sports, and Animals. After selecting the articles from Wikipedia, a summary was generated for each abstract using the summarization module discussed earlier.

To conduct the evaluation of the summaries, a web application was developed. The articles were divided into groups; each group had five articles, each article from a different domain. When the first user (experiment participant) logged in, s/he was randomly assigned a group. The next user was then randomly assigned a group from the remaining unassigned groups. This continued until all the groups were assigned to users. The process was then repeated for the next set of users who logged in to the system. This ensured an even spread of assessment. Each article in the group that is presented to the user was followed by the generated summary. The users were asked to evaluate each summary according to the following characteristics:

1. **Readability & Understand-ability:** the user was asked to assess whether the grammar and the spelling of the summary is convenient or not.
2. **Informativeness:** assess how much information from the source text is preserved in the summary.
3. **Conciseness:** assess if this summary does not contain any unnecessary or redundant information.
4. **Overall Quality:** evaluate the overall quality of the summary.

The users were asked to evaluate each characteristic on a mean Likert scale (ranging from 1 to 6, where 1 is the lowest quality and 6 is the highest). An open call for participation in the experiment was made through mailing lists and social media. The participants came from different countries and were from academia and industry. They had different educational backgrounds and disciplines. Their ages ranged from 27 to 40.

**Results and Discussion.** Thirty eight users ultimately participated in the experiment. Each user evaluated at least four articles in different domains. The final result was analyzed regarding: (a) general summary quality and (b) domain-specific summary quality. Table 1 reports the mean and the standard deviation scores of the user evaluations. The scores show that, in general (“All Domains Combined”), users were satisfied with the summaries produced by our summarization system, as the summaries received an average mean score of over four in all criteria measured. The results also show the mean and the standard deviation scores of the user evaluations for each domain separately. Some domains exhibited higher mean scores than others (e.g. Accidents and Natural Disasters vs. the other domains).
6.2 SSyn Component Evaluation

The HTS system used in this work for synthesizing the English voice is a very popular HMM-based speech synthesizer, which performed very well against other speech synthesizers in the Blizzard Challenge 2005 [24].

The Blizzard Challenge is an annual event in which participants are provided with a speech corpus and have to synthesize a set of test utterances. Then, an overall evaluation of the synthesizers is conducted and the results can be examined in the Blizzard Challenge Workshop. The HTS system has been used as the benchmark HMM-based speech synthesizer in the Blizzard Challenge since 2006. For example, in the recent Blizzard Challenge 2013 [25], only two systems out of ten obtained better results in terms of naturalness and similarity to the target speaker than HTS and this benchmark system was in the group of the four equally-intelligible systems which obtained the best intelligibility results (among a total of 11 systems).

The MARY TTS system [27] which was also used in this work for synthesis of German and French has also taken part in several editions of the Blizzard Challenge evaluation, e.g. [26]. However, while the system itself has been evaluated, these languages have never been evaluated in the Blizzard Challenge. Therefore the system’s speech quality for these languages will need to be evaluated against the benchmark systems in order to gain a complete picture of the performance of the SSyn service in LbL.

The current speech synthesizers used in the LbL framework acts as a baseline synthesizers in ongoing experiments which are being conducted to evaluate our own improved versions of the speech synthesis service in multiple languages. There are two hypotheses that we plan to test through the ongoing experimentation. One is to test the hypothesis that an audio presentation is better than the written one. This is being evaluated both in terms of user preference and listener comprehension, especially in the dual tasking situation where the user is listening and trying to understand a spoken

<table>
<thead>
<tr>
<th>Domain</th>
<th>Mean &amp; Std. Dev</th>
<th>Readability</th>
<th>Informativeness</th>
<th>Conciseness</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accidents</td>
<td>Mean</td>
<td>5.21</td>
<td>4.50</td>
<td>4.71</td>
<td>4.50</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.8926</td>
<td>0.7596</td>
<td>0.9945</td>
<td>0.6504</td>
</tr>
<tr>
<td>Natural disasters</td>
<td>Mean</td>
<td>5.00</td>
<td>4.60</td>
<td>4.33</td>
<td>4.56</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.8528</td>
<td>0.9145</td>
<td>1.044</td>
<td>0.8675</td>
</tr>
<tr>
<td>Politics</td>
<td>Mean</td>
<td>4.61</td>
<td>4.33</td>
<td>3.56</td>
<td>4.11</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>1.243</td>
<td>1.138</td>
<td>1.042</td>
<td>1.023</td>
</tr>
<tr>
<td>Famous people</td>
<td>Mean</td>
<td>4.79</td>
<td>4.21</td>
<td>4.34</td>
<td>4.31</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>1.048</td>
<td>0.9403</td>
<td>1.010</td>
<td>0.8906</td>
</tr>
<tr>
<td>Sports</td>
<td>Mean</td>
<td>4.63</td>
<td>4.43</td>
<td>4.27</td>
<td>4.30</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.9994</td>
<td>0.8172</td>
<td>1.172</td>
<td>0.9523</td>
</tr>
<tr>
<td>Animals</td>
<td>Mean</td>
<td>4.96</td>
<td>4.13</td>
<td>4.13</td>
<td>4.21</td>
</tr>
<tr>
<td></td>
<td>SD</td>
<td>0.9991</td>
<td>0.7974</td>
<td>0.9470</td>
<td>0.8330</td>
</tr>
<tr>
<td>All domains</td>
<td>Mean</td>
<td>4.86</td>
<td>4.39</td>
<td>4.24</td>
<td>4.36</td>
</tr>
<tr>
<td>combined</td>
<td>SD</td>
<td>0.9936</td>
<td>0.9041</td>
<td>1.067</td>
<td>0.8856</td>
</tr>
</tbody>
</table>
summary while doing something else. Another evaluation is to test if the synthetic voice does not significantly affect the user satisfaction with the LbL service compared with a recorded voice from a person reading the summary. We also plan to conduct experiments to evaluate the quality of synthetic voice in the context of spoken summaries. This will be used to identify important factors which affect this type of audio presentation, such as pause position and length which are predicted from the text, speech naturalness and variation of intonation along and between sentences.

6.3 System-Level Evaluation

While this paper focuses on the evaluation of the two main components of the LbL system, this is not intended to downplay the importance of whole-system user-focused evaluation. A qualitative user study is planned which will evaluate the user perception of LbL with respect to the function of the user interface and the interaction mechanisms, the informational value of the summaries which are generated, and the quality of the synthesized voice and audio presentations that are delivered by the service.

7 Conclusion and Future Work

This paper introduced Lean-back Learning: an information system that delivers automatically generated audio presentations for consumption in hands-busy, eyes-busy contexts. The system is based on two core NLP components: an automatic text summarization service and a speech synthesis service. The paper presented the LbL architecture, its component services and the algorithms which are used by those services. In addition, a series of initial system evaluations were discussed.

There are a number of future areas of research which are planned, with the aim of improving the LbL framework. Features extracted from sentence parsing will be used to extend the feature set currently used in tree-based clustering for speech synthesis. A syntax parser can generate a large number of features from which the most appropriate can be selected to model prosody. For example, features could be derived from the syntax trees which are correlated with pause position and duration.

Further work is planned in the area of personalization. In the current version of the system, the audio presentation is adapted based upon explicit user input. This can be extended to utilize user history and inferred preferences to personalize both the content and synthesized voice used in generating the audio presentations. Lean-back Learning may also be used as a tool for evaluating the differences between the effects of learning from audio and written media.

Acknowledgements. This research is supported by the Science Foundation Ireland (grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College, Dublin.
References


