An Interactive Adaptable Learning Interface for E-Learning Sessions

Elena Odarushchenko[1,] Valentina Butenko[2], Viktor Smolyar[1], Vyacheslav Kharchenko[2]
[1] Poltava State Agrarian Academy, Poltava, UKRAINE
[2] National Aerospace University "KhAI"
Computer Systems and Networks Department
Kharkiv, UKRAINE
elena.odarushchenko@gmail.com; valentinaodarus@gmail.com; v.kharchenko@csn.khai.edu

Received: August 20, 2020. Revised: November 18, 2020. Accepted: November 19, 2020.
Published: November 20, 2020.

Abstract. Introduction of augmented reality into the E-learning systems brings a new era in personal remote education. While E-learning brings a lot of benefits for both instructor and student it also has various difficulties, such as absence of vital personal interaction that can influence on student motivation and inspiration. To decrease this gap we have started the development of an Interactive Adaptable Learning Interface (IALI) that with the help of speech synthesizers, emotions and voice recognition systems tries to adapt the ongoing E-learning session to the student behavior, simulating the way instructor adapt the lesson to the student reactions. As the IALI is initially planned for Ukrainian language students, we have made a deep analysis of nowadays applicable solutions for Ukrainian speech synthesis and recognition. In this paper we present the conducted analysis and application of the most widely applied speech synthesizers, voice and emotions recognition systems that were used to develop first version of IALI.

Keywords: E-learning, speech synthesis, emotion recognition.

I. Introduction

RAPID development of information technologies (IT) have created a great boost for all areas of everyday life. The Internet creation have launched a powerful chain of changes in the way humanity perceive, learn and act. The ease of an access to the tons of information, as well as development of multimedia and IT, have forced the learning process to step out of the classroom and gave birth to the E-learning concept. This concept refers to the use of information technologies to enable an access to the online learning resources [1].

The E-learning have lots of advantages, such as [2]:

1. Flexible. Every student can choose a time and place that perfectly suits him or her.
2. Enhances the knowledge efficacy. The ease of an access to the needed information helps to raise a comprehensively developed specialists.
3. Eliminates discussion barriers. The E-learning motivates student interact with each other with means of discussion forums.
4. Cost effective. Due to E-learning flexibility the additional expenses, such as traveling to the university, renting an apartment etc. are not necessary for students. The instructors can teach as much students as they want without any need to rent a big studying spaces and all supporting expenses.
5. Consider learner differences. Every student of a E-learning course can focus on the part that is most important or interesting for him or her, and work over it with different intensity.

Despite those advantages there are few valuable disadvantages as well. Firstly, the student have to be strongly inspired and self-motivated to completely pass the course as they work remotely with the lack of interaction. Secondly, the control tests are also held remotely, there can be some negative activities as cheating etc. And one of the most noticeable disadvantage is complete absence of vital personal interaction, especially between learner and instructor [2]. One of the possible ways how we can struggle with such disadvantages is to step into new E-learning era – introduce the augmented reality (AR) into the E-learning process. Along with virtual and mixed realities, the AR can increase content understanding, long-term memory retention, learning language associations; improve physical tasks performance, collaboration and motivation [3]. Using the AR in the classrooms and in E-learning can enhance lessons and amplify students' experience through immersive field trips [4].

In this paper we present an Interactive Adaptable Learning Interface (IALI) that by means of speech synthesis and recognition along with user emotions analysis adapts the E-learning session to increase the level of educational material comprehension. As the IALI was originally planned to be applied for learning needs of Ukrainian students we have conducted and additional study on recognition and synthesis of Ukrainian speech. The IALI requires user authorization using face detection and gives tools to manage E-learning session with voice control.

In Section 2 we discuss the basic features of speech synthesis, presents an analysis of the most widely known ones and shows how synthesis was applied in IALI. Section 3 describes how user speech and emotions can be scanned and analyzed for further IALI adaptation. The last section presents conclusions and further research and IALI development stages.

II. Speech Synthesizers

The text, video and audio formats are heavily used to present studying information in E-learning systems. The video and audio data are created manually by instructors and to create a high quality course content they need to use an appropriate equipment or even make records in studios. This can increase the cost and time of course development. The media data formats are hosted on some local or remote server which usually brings new expenses. Additionally, the media content cannot be adapted during ongoing E-learning session as only premade data can be played. To bring the interactive adaptation to the E-learning systems sessions we can use text course information. By means of speech synthesizers the appropriate text, generated during lesson, can be voiced without any premade records.

The Text-To-Speech (TTS) systems convert normal language text into speech, while other speech synthesis systems render symbolic linguistic representations into speech. The quality of any speech synthesizer is judged by its similarity to human voice, ability to be clearly understandable. The TTS are widely used as voice assistants that allows people with visual impairment or reading disabilities to listen the written words on a home computer or portable devices. Basically, the speech synthesizers can be split into two types: with limited and unlimited lexicon. The first type implies the premade records of words or sentences. The speech synthesizers with unlimited lexicon use phenoms or syllable at the base. To create a human-like speech the second type of speech synthesizers use phonetic rules. The nowadays four main methods of speech synthesis are presented in Table 1 [5-7].

Table 1. Main speech synthesis methods

Method	About	Advantages	Disadvantages
Parametric synthesis	Performed by constructing the model of an acoustic properties of human speech track and then analyzing speech by determining the values of model parameters	The initial records can be created with help of any speaker on any language	Not appropriate to synthesize previously unrecorded text
Compilation synthesis	Creates text from previously recorded lexicon of initial synthesis elements (not less a word)	Suitable for limited amount of needed responses on an actions, such as giving an information about status of mobile subscriber account, etc.	Synthesized text is bounded by initial lexicon. Needs specific compression methods for wider words amount in the initial lexicon
Rules-based synthesis	Synthes is performed by modeling the speech tract with analog and digital technologies. During the synthesis the modeling parameters values and phoneme connection rules are introduced sequentially at a certain time interval. This method does not directly use the human speech elements.	Suitable to synthesize speech on previously unknown text	Still hard to manage the speech intonation and synthesis result does not match the quality of natural speech due to distortions on boundaries of segments stitching

There are four basic challenges in speech synthesis scope:

1. Text normalization – text are full of heteronyms, numbers and abbreviations that requires expansion into phonetic and semantic representation, that is poorly performed in most nowadays TTS systems.
2. Text-to-phoneme – the TTS systems use combine the dictionary-based and rule-based approaches to determine the pronunciation of word based on spelling that does not properly fit to all languages.
3. Evaluation - the consistent evaluation of speech synthesis systems is difficult because of absence of universally agreed objective evaluation criteria.
4. Prosodics and emotional content – identification of vocal feature that signal emotional content may be used to help synthesized speech sound more natural.

2.1. Google Text-to-Speech and Microsoft Speech *API*

There are two most widely used TTS systems that can synthesize Ukrainian speech –Google Text-to-Speech and Microsoft Speech application programming interface (API).

Google Text-to-Speech is a screen reader app developed by Google for Android operation system (OS) in 2013 and it powers apps to speak the txt on the screen. It can be used by such apps as Google Play Book (to synthesize audiobooks), Google Translate (speak translations, providing useful insight on words pronunciation), Google Talkback (for voice control of the device) and other spoken feedback of accessibility-based and third-party apps. Starting from 2017 the Google TTS system supports the Ukrainian language. Using Google Cloud TTS API under a fixed “pay-as-you-go” payment plan – the third-party developers can integrate the synthesizer interface into their projects. This TTS system requires an internet connection during speech synthesis.

Microsoft Speech API (SAPI) is an interface from Microsoft company for speech recognition and synthesis developed for Windows OS. The SAPI is a freely redistributed component which can be shipped with any Windows app that wish to use this technology. There are two “families” of Microsoft SAPI – versions SAPI 1 to SAPI 4 and SAPI 5. The SAPI 5 was build upon a concept of strict separation of the application and engine. This concept aims to make the API more engine-independent, preventing app from inadvertently depending on features of specific engine.

Unlike Google TTS system the Microsoft Speech API does not require an internet connection for speech synthesis.

Various commercial and free voices were developed for this TTS system. All available Ukrainian voices were created by third-party companies, thus they are not preinstalled in basic system and have to be additionally added to it. The most commonly used Ukrainian voices for Microsoft Speech API are as follows:

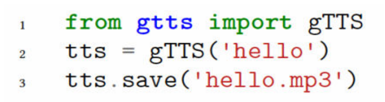
1. UkrVox Igor Murashko – software that convert written Ukrainian text into spoken works. As the basis was taken a voice of famous Ukrainian radio announcer Igor Murashko. The UkrVox has a strong linguistic base, wide lexicon of pre-basic words, supports a word formation and morphological text analysis. It supports .txt and .xml formats and provides an ability to control speech speed and volume along with selection of generated sound quality. The last version was published in 2011 and it is distributed under a free license [8].
2. RHVoice Anatol’ – free male voice Anatol’ for Microsoft Speech API 5 on Android. Uses the voice of professional announcer Anatoliy Podorozhko. This synthesizer is available for OS Windows, Linux and Android. Does not support the manual adding accents to the text and English transcription. The last version was published in 2016 and it goes under a free license [9].
3. CyberMova Natalka – the female voice Natalka for OS Window. With CyberMova Natalka user can add accents to the text and control the speed of reading. The synthesizer can read dates, email addresses and English words. The last version was published in 2015 and it is provided under four payment plans – standard, family, class and site [10]. For visually impaired users the CyberMova Natalke is distributed under a free license.

2.2. Synthesizing Speech in an Interactive Adaptable Learning Interface

The Interactive Adaptable Learning Interface (IALI) is built with Python. The user interface was created on QML with the help of PyQt5 framework. The IALI works on MS Windows 7, 8, 8.1 and 10 for both x86 and x64 platforms, but can be built also for Linux and MacOS with minor changes in core code.

To synthesize speech during E-learning session the Interactive Adaptable Learning Interface (IALI) uses the online gTTS library, that connects to Text-to-Speech API from Google Translate. This library sends a request with the text file to the server and waits for the respond in a form of mp3 file. This approach heavily relies on strong internet connection, which is commonly required during E-learning sessions, and also takes a free memory space to store the temporary audio file. On one hand this approach can pose a problem in case of bad Internet connection, as generation of the text file and loading the created audio file can take too much time. On the other hand - gTTS library is a cross-platform solution that requires no third-party components.

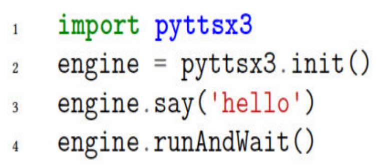
The code presented on Fig.1 shows how easily string line ‘hello’ can be converted by means of gTTS library into “hello.mp3” file.



```
from gtts import gTTS
tts = gTTS('hello')
tts.save('hello.mp3')
```

Fig. 1. Example of gTTS application

We can also synthesized the speech autonomously using pyttsx3 library [11] with pypiwin32 library [12] to grant direct access to SAPI 5. Due to cross-platform base the pyttsx3 can use a eSpeak and nsss instead of SAPI 5. The Fig.2 shows code that voices a string ‘hello’ with pyttsx3.



```
import pyttsx3
engine = pyttsx3.init()
engine.say('hello')
engine.runAndWait()
```

Fig. 2. Example of pyttsx3 library application

The offline solution helps to eliminate the main gTTS disadvantages but created a need to set up additional components, which is not always possible. Thus, we have decided to give a IALI user an ability to choose between two possible speech synthesis options – Google Text-to-Speech or SAPI5. In case user have selected a SAPI5 for offline speech synthesis there further can be selected a voice for a specific language, as default English voice is not appropriate for voicing Ukrainian language.

III. Speech and Emotions Recognition

The speech recognition has made an outstanding progress in the past years with such widely used services as Google Voice Search, that support about 120 languages. Various methods, methods and algorithms were introduce to improve accuracy and speed of speech recognition process [13, 14].

An adaptability of IALI is based on various user activities and reactions on presented material during E-learning session. Emotions are extremely important during the real communication between teacher and student, for example the calm or happiness can show that student undestand the studying material or high concentration can express that studying material is rather complex. Based on those expressions teacher can change the class pace and spend more time on some complicated information. The knowledge-based techniques, statistical methods and different hybrid approaches are striving to reach the high accuracy in recognizing emotions based on photo, video, voice and text [15, 16].

3.1. Recognizing Ukrainian Speech in an Interactive Adaptable Learning Interface

As the most newly developed models are mainly trained on the English vocabulary, so the most significant results can be found in recognition of English language. Meanwhile, only a few nowadays popular speech recognition systems supports the Ukrainian language.

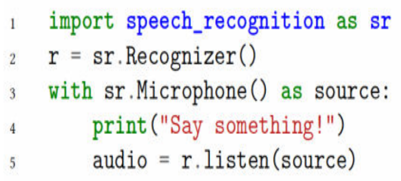
The Cloud Speech-to-Text API support over 120 languages, including Ukrainian, under special payment plan for every 15 seconds over one free hour of speech recognition from audio and video content.

The voice virtual assistant Alice created by Yandex are expected to support Ukrainian language in late 2019. The Yandex have already provide the SpeechKit Cloud API, that successfully supports English, Russian, Ukrainian and Turkish languages. This interface provides a free payment plan under a 10100 daily requests. One of the main difficulties of applying the SpeechKit Cloud API on Ukraine territory is a block of all Yandex services since 2017.

The next service that supports recognition of Ukrainian language is VoiceTypist from CyberMova [10]. The application is built on vocabulary that contains over 200000 words and is expected to be enlarged up to 1000000. Using VoiceTypist user can add punctuation marks to the text during it dictation. The VoiceTypist cannot be added to the third party programs and can only be used as a stand-alone product.

CMUSphinx [17] is an open source speech recognition toolkit applicable for mobile and server applications that supports C, C++, C#, Python, Ruby, Java and JavaScript language. It provides a Pocketshinx library for Python projects. Comparing to the other systems the CMUSphinx library does not require an Internet connection, but still needs a lot of memory space to keep the voice files. The official site provides developers with all needed information on acoustic and voice models, supported by CMUSphinx, for 14 languages. Initially this toolkit does not support detection the Ukrainian language, but in IALI we have created a limited vocabulary to add a Pocketsphinx speech recognition with Python.

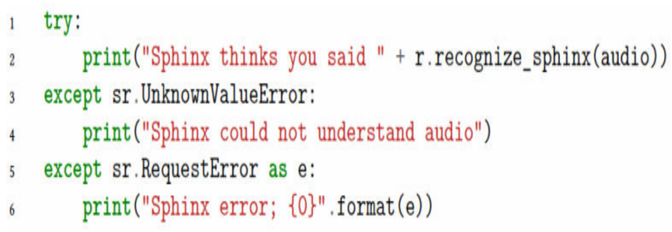
The IALI uses Speech Recognition library that gives and API to most popular speech recognition systems, such as CMUSphinx, Google Speech Recognition, Google Cloud Speech API, Wit.ai, Microsoft Bing Voice Recognition, Houndify API, IBM Speech to Text and Snowboy Hotword Detection. As it was previously mentioned, only few among those supports Ukrainian language and just CMUSphinx can perform without Internet connection. By default the IALI uses a microphone as an audio input source. The Fig. 4 shows how recorded with microphone speech fragment is stored in audio variable for further recognition.



```python
import speech_recognition as sr
r = sr.Recognizer()
with sr.Microphone() as source:
    print("Say something!")
    audio = r.listen(source)
```

Fig. 3. Listening the user

With IALI user can select the system for recorded files recognition – Google Speech Recognition of CMUSphinx. Example of how CMUSphinx is applied to recognition of recorder from microphone audio file (Fig.5).



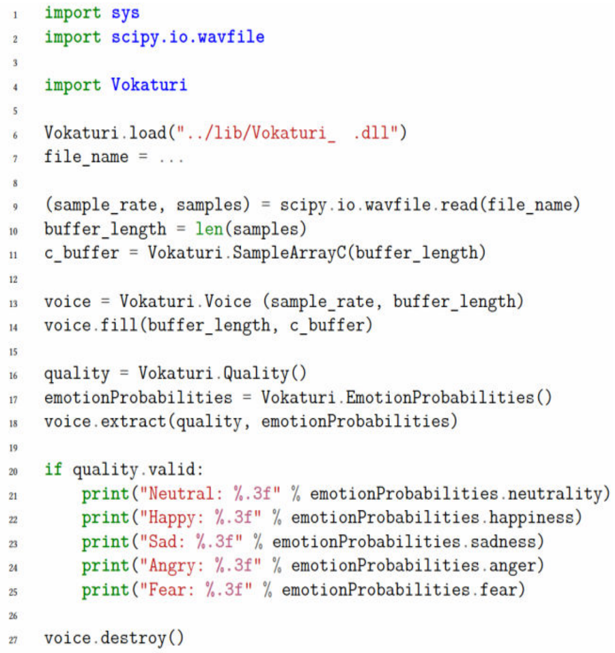
```python
try:
    print("Sphinx thinks you said " + r.recognize_sphinx(audio))
except sr.UnknownValueError:
    print("Sphinx could not understand audio")
except sr.RequestError as e:
    print("Sphinx error; {0}".format(e))
```

Fig. 4. Recorded speech recognition with CMUSphinx

3.2. User Emotions Recognition in an Interactive Adaptable Learning Interface

The IALI performs basic user emotions detection using voice records and live camera records. We use Vokaturi emotion recognition software [18] to make analysis of user emotions based on voice records. Vokaturi libraries can be added to the third party software created with C, C++ and Python for iOS, MacOS, Android, Windows and Linux operation systems. There are three Vokaturi licenses – OpenVokaturi, VokaturiPlus and VokaturiPro. The first one is distributed with general public license (GPL) and uses Internet connection to detect basic user emotions – neutral, happy, sad, angry and frightened. VokaturiPlus and VokaturiPro use deep-learning algorithms to detect more subtle emotions.

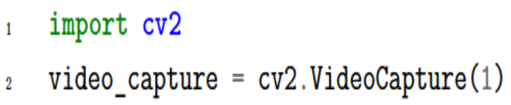
The Fig.6 presents how WAV audio file can be read (with scipy.io.wavfile module) and analyzed using OpenVokaturi library. The Vokatury.py and OpenVokaturi.dll files have to be stored in the same directory as the basic application module.



```
import sys
import scipy.io.wavfile

import Vokaturi

Vokaturi.load("../lib/Vokaturi_  .dll")
file_name = ...

(sample_rate, samples) = scipy.io.wavfile.read(file_name)
buffer_length = len(samples)
c_buffer = Vokaturi.SampleArrayC(buffer_length)

voice = Vokaturi.Voice (sample_rate, buffer_length)
voice.fill(buffer_length, c_buffer)

quality = Vokaturi.Quality()
emotionProbabilities = Vokaturi.EmotionProbabilities()
voice.extract(quality, emotionProbabilities)

if quality.valid:
    print("Neutral: %.3f" % emotionProbabilities.neutrality)
    print("Happy: %.3f" % emotionProbabilities.happiness)
    print("Sad: %.3f" % emotionProbabilities.sadness)
    print("Angry: %.3f" % emotionProbabilities.anger)
    print("Fear: %.3f" % emotionProbabilities.fear)

voice.destroy()
```

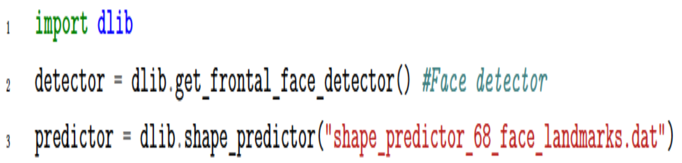
Fig. 5. Analysis of audio file using OpenVokaturi

Additionally, IALI uses OpenCV and dlib libraries to read user emotions based on live camera records. Starting with capturing video from camera with OpenCV (Fig. 7). then we use dlib methods for detection of 68 anchor points on human face (Fig. 8) and again OpenCV to read video from camera (Fig. 9). Based on anchor points location the applied dlib deep-learning algorithms will make an emotions recognition and classification.



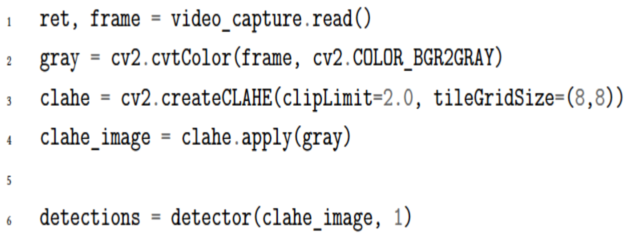
```
import cv2
video_capture = cv2.VideoCapture(1)
```

Fig. 6. Video capture with OpenCV



```
import dlib
detector = dlib.get_frontal_face_detector() #Face detector
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")
```

Fig. 7. Detecting anchor points



```
ret, frame = video_capture.read()
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
clahe_image = clahe.apply(gray)

detections = detector(clahe_image, 1)
```

Fig. 8. Reading video from camera

IV. Conclusions

As a result of conducted studies we have developed first version of IALI that can be connected to the ongoing E-learning session. Application of this interface aims to improve level of material understanding due to constant analysis of student reactions. Due to the voice control feature users can focus on studying material and manage basic settings of the session.

We have analyzed the available speech synthesis, voice and emotions recognition systems that support Ukrainian language, as IALI will be introduced into the studying process of Poltava State Agrarian Academy and National Aerospace University "KhAI".

In our future work we intend to improve accuracy of the user emotions analysis, expand the voice control over E-learning sessions settings and test various sceneries of IALI reactions to different user behaviors.

References

[1] Arkorful, V., Abaidoo, N.: The role of e-learning, the advantages and disadvantages of its adoption in Higher Education. International Journal of Education and Research 2(12), 397-410 (2014).

[2] Holmes, B., Gardner, J.: E-Learning: concepts and practice. 1st edn. SAGE Publications Ltd., London (2006).

[3] Radu, I.: Augmented reality in education: a meta-review and cross-media analysis. Personal and Ubiquitous Computing 18(6), 1533-1543 (2014).

[4] Donally, J.: Learning transported: augmented, virtual and mixed reality for all classrooms. 1st edn. ISTE, London (2018)

[5] EECS20N: Signals and systems, parametric speech synthesis, https://ptolemy.berkeley.edu/eecs20/speech/voder.html, last accessed 2019/10/12.

[6] Taylor, P.: Text to speech synthesis. 1st edn. Cambridge University Press, Cambridge (2009).

[7] Dutoit, T.: An introduction to text-to-speech synthesis. 1st edn. Springer, Dordrecht (1997).

[8] BIBLRPROG Windows, UkrVox page, https://biblprog.org.ua/ua/ukrvox, last accessed 2019/03/20.

[9] Trosti, Anatol' homepage, http://www.trosti.com.ua/ua/anatol.html, last accessed 2019/03/20.

[10] CyberMova homepage, http://cybermova.com, last accessed 2019/03/20.

[11] pyttsx3 2.7 project, https://pypi.org/project/pyttsx3/2.7, last accessed 2019/04/10.

[12] pypiwin32 219 project, https://pypi.org/project/pypiwin32/219, last accessed 2019/04/10.

[13] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y.: End-to-end attention-based large vocabulary speech recognition. In: IEEE International Confer-

ence on Acoustics, Speech and Signal Processing (ICASSP) Proceeding, pp. 1-8, IEEE, Shanghai, China (2016).

[14] Toshniwal, S., Weiss, R., Sainath, T., et al.: Multilingual speech recognition with a single end-to-end model. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) Proceeding, pp. 1-8, IEEE, Shanghai, China (2016).

[15] Katsigiannis, S., Ramzan, N.: DREAMER: a database for emotion recognition through EEG and ECG signals from wireless low-cost off-the-shelf devices. IEEE Journal of Biomedical and Health Informatics 22(1), 98-107 (2017).

[16] Cambria, E., Hussain, A.: Sentic computing: a common-sense-based framework for concept-level sentiment analysis. 1st edn. Springer, Heidelberg (2015).

[17] CMUSphinx Homepage, https://cmusphinx.github.io, last accessed 2019/10/12.

[18] Vokaturi Homepage, https://vokaturi.com, last accessed 2019/10/12