Research and development

A freely-available authoring system for browser-based CALL with speech recognition

Myles O'Brien
Mie Prefectural College of Nursing, Japan

myles.obrien@mcn.ac.jp

Abstract

A system for authoring browser-based CALL material incorporating Google speech recognition has been developed and made freely available for download. The system provides a teacher with a simple way to set up CALL material, including an optional image, sound or video, which will elicit spoken (and/or typed) answers from the user and check them against a list of specified permitted answers, giving feedback with hints when necessary. The teacher needs no HTML or Javascript expertise, just the facilities and ability to edit text files and upload to the internet. The structure and functioning of the system are explained in detail, and some suggestions are given for practical use. Finally, some of its limitations are described.

Keywords: Automatic speech recognition, CALL authoring tool, computer-assisted language learning, Google speech API.

1. Introduction

The quality of automatic speech recognition (ASR) has been improving steadily with technological advances. The history of ASR-based CALL dates back to the end of the last century (Aist, 1999; Bernstein, Najmi, & Esani, 1999; Strick, 2012, p. 10), when there were experiments with prototype systems, notably FLUENCY (Eskenazi, 1999), a computer-assisted pronunciation training (CAPT) system which used carefully constructed output, free of explicit prompting, to elicit an oral response which was confined to a very narrow range of possibilities. But progress in this area has been rapid, and there are now several commercial systems (Witt, 2012, p. 5) which can evaluate speech with a good correlation to the judgement of human assessors (Bernstein, Van Moere, & Cheng, 2010; Zechner et al., 2014). These, and other systems under development (Penning de Vries, Cucchiarini, Bodnar, Strik, & van Hout, 2015; van Doremalen, Boves, Colpaert, Cucchiarini, & Strik, 2016) attempt to evaluate grammar and content in addition to pronunciation.

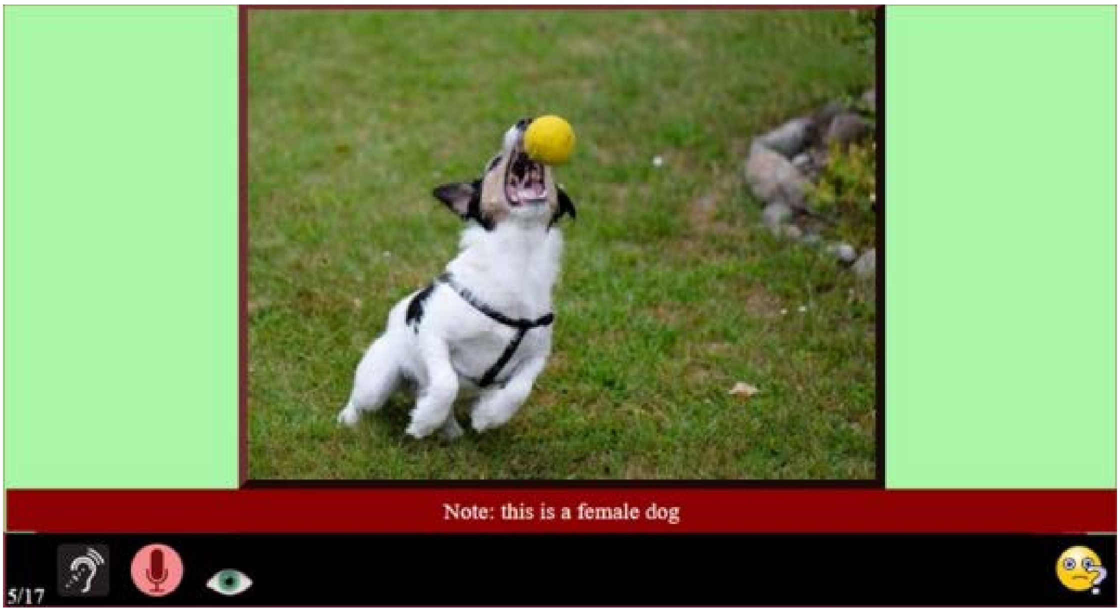
The uses of ASR are not confined to CALL, of course. One company which has been developing ASR for its own various purposes, and is among the leaders in the field, is Google. The ability to perform a Google search by voice alone became available on iPhone in 2008. It has since been greatly extended and improved, now being available in over 80 languages. Google fully opened its Speech API, which gives direct access to its speech recognition capabilities, to developers for use in their applications on a commercial basis in 2016 (https://cloud.google.com/speech). However, a free tier of indirect access to basic ASR functions using JavaScript calls from the Google Chrome browser has been available since the release of version 25 in 2013. This enables speech-to-text conversion within the browser, and so offers the interesting possibility of browser-based CALL incorporating ASR. In order to implement this in as useful a way as

https://doi.org/10.4995/eurocall.2017.6830

possible, it was decided by the author to attempt the development of a very flexible system which could be used with ease by anybody, without JavaScript knowledge, to make their own ASR-based CALL material for internet deployment. The system has been successfully implemented and made freely available, and is described in detail in this paper.

2. Outline of the system

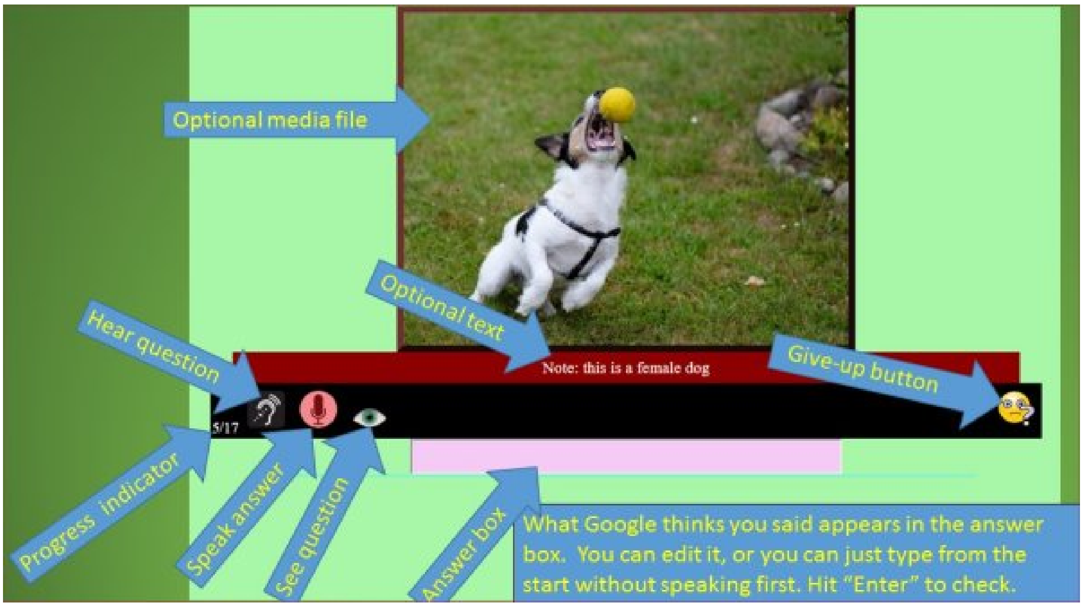
The system is tentatively named *QAspeak*. At its most basic, it consists of one html file (and a folder containing the image files it uses) plus a plain text file containing a list of questions with the acceptable answer(s) for each. The person making the CALL material (the teacher) needs to edit only the text file. The person using the material for study (the user) can elect to listen to and/or read the question, and answer by speaking or typing. The teacher has the option of adding a media file (image, sound, or video) and/or text of any length to each question. Another option is to add a sound file of the question. If this is not included, the question text will be read by the device's text-to-speech function. Figure 1 shows an example of the interface for a question which includes both image and text options.



Note: this is a female dog

Figure 1. An example of the interface for a question.

The user hears "What's the dog doing?" and is expected to provide an appropriate response, like "It's catching the ball." The user interacts through the icons, which are in the black strip, and the answer box, which is just below it. Figure 2 shows an annotated version of the same interface.



Note: this is a female dog

What Google thinks you said appears in the answer box. You can edit it, or you can just type from the start without speaking first. Hit "Enter" to check.

Figure 2. An annotated version of the interface in Figure 1.

Leftmost in the black strip is the progress indicator, which shows how far the user has advanced through the set of questions. Next is the ear icon, which allows the user to hear the question through text-to-speech, or a sound file, if available. Then the microphone icon activates Google speech recognition for the user to speak the answer. The ASR system's interpretation of the speech appears in the pink answer box, and the answer is checked against the list of acceptable alternatives the teacher has set up. If the answer is correct, a congratulatory image and a green arrow icon, to move on to the next question, appear. Also, the other remaining permitted answers are displayed. If the answer is incorrect, corrective feedback (described in detail later) is displayed, and the user can try speaking the full answer again, or edit the current text in the answer box, hitting the "Enter" key on the device to have the answer checked. This process can be repeated until a correct answer is obtained, or the user resorts to the "Give Up" button, which shows the full list of permitted answers and allows progress to the next question. One more source of help is available along the way: hitting the eye icon will display the text of the question at any stage, which may assist a user who is having trouble understanding the audio of the question. The user is also permitted to type their answer directly at any stage, even from the beginning without speaking at all.

3. Anatomy of the controlling text file

To set up the questions, the teacher needs to make a plain text file, with a very simple format, specifying the questions, permitted answers, and additional media files. Each question-answer set extends over 3 or more lines. The first line must begin with a question mark. This specifies the beginning of a new question. If a media file is to be included, its name is entered after the question mark, and optional text may be typed on the same line. The next line should contain the text of the question, and the sound file name, if one is supplied, to be used instead of text-to-speech. The third and subsequent lines contain the permitted answers. So, in general, the format of a question is like this (where square brackets signify an optional item):

? [media file name] [text]
Question text [mp3 file name]
Answer 1
Answer 2
Answer 3
etc.

The example shown in Figure 1 corresponds to the following lines:

? dog.jpg Note: this is a female dog
What's the dog doing?
She's catching a ball.
It's catching a ball.

Lines where the first character is not "?" or alphabetic (including blank lines) are ignored, so that dividers or comment lines can be added anywhere. Any number of questions may be included in one text file.

These are the first 3 questions from the text file for the online example (http://www.mcn-moodle.org/asr), which shows that the format is very simple, yet flexible:

?
What browser does this have to be?
Chrome
It has to be Chrome.

? Jack can't stand carrots.
Does Jack like carrots?
No, he doesn't.
------XXXXXXXXXX-------

? This is my sister's cat. His name is Nando. nando6.mp4
What is he swinging?
His tail.
He's swinging his tail.
He is swinging his tail.

The first is a minimal example with no media files at all and just 2 permitted answers. The second adds a little optional explanation. It accepts only one answer. "------XXXXXXXXXX-------" will be ignored. The third has optional text and a video file.

While direct editing of the text file affords the advantages of maximum speed, simplicity, and flexibility, the procedure may not appear very user-friendly to many less computer-oriented teachers, and could well discourage some from using the system at all. Therefore, if the initial version of the system attracts attention and proves successful, a high-priority addition to the next version should be a form-like application to enable structured, guided input of the text items and file names, and automatic generation of the corresponding controlling text file.

4. Uploading to the web

The html file and "images" folder which are supplied as the core of the system, the controlling text file, and all specified media files should be uploaded to the same directory. The html file may be freely renamed, but the controlling text file must have the same name, with a ".txt" extension in place of ".html". Figure 3 shows a schematic example:

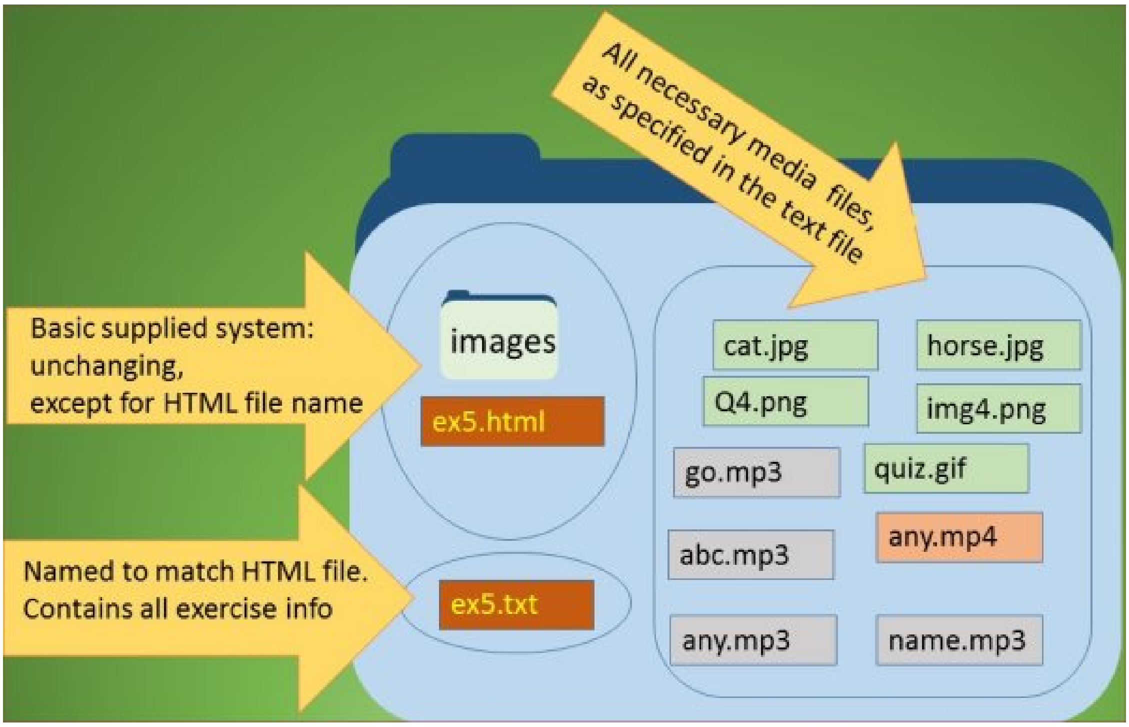

All necessary media files, as specified in the text file
Basic supplied system: unchanging, except for HTML file name
Named to match HTML file. Contains all exercise info
images
ex5.html
ex5.txt
cat.jpg
horse.jpg
Q4.png
img4.png
go.mp3
quiz.gif
abc.mp3
any.mp4
any.mp3
name.mp3

Figure 3. An exercise on the web.

Using this system, deploying ASR-based CALL material to the web requires minimal technical expertise from the teacher. The required files just need to be dropped into a folder which is already online, or which will then be uploaded. How well the material which can be made with the system meets the requirements of each teacher is, of course, a separate question. There is sufficient flexibility in the system to allow it to be used in quite a variety of different ways, some of which will be outlined in a later section. Before that, the description of the functioning of the system will be completed, with an account of the feedback it produces.

5. Feedback

When the user has finished speaking or hits the “enter” key to have an answer checked, the system first checks if the user’s answer matches any of the teacher’s listed permissible answers exactly. If it does, a message appears, stating how many attempts were needed, and whether the user looked at the question text or did some typing. Also, the full list of permitted answers for that question is displayed, and the arrow icon to allow progression to the next question appears. Finally, one of four images appears, depending on how smoothly the user has arrived at the answer. The user incurs a penalty for: (1) looking at the question text, (2) typing an answer completely or in part, or (3) requiring more than one attempt to answer correctly. The 4 images supplied with the system are shown in Figure 4. These are kept in the “images” folder, and the teacher may replace any or all of them with different .jpg images of the same filename. The image *score3.jpg* appears when the user incurs no penalties, i.e., gives correct answer first time by listening and speaking only. The other images, *score2.jpg, score1.jpg* and *score0.jpg* appear when 1, 2, or 3 penalties, respectively, are incurred. Note that extra penalties are not incurred for repeated “offences” of the same type, so the maximum is 3.

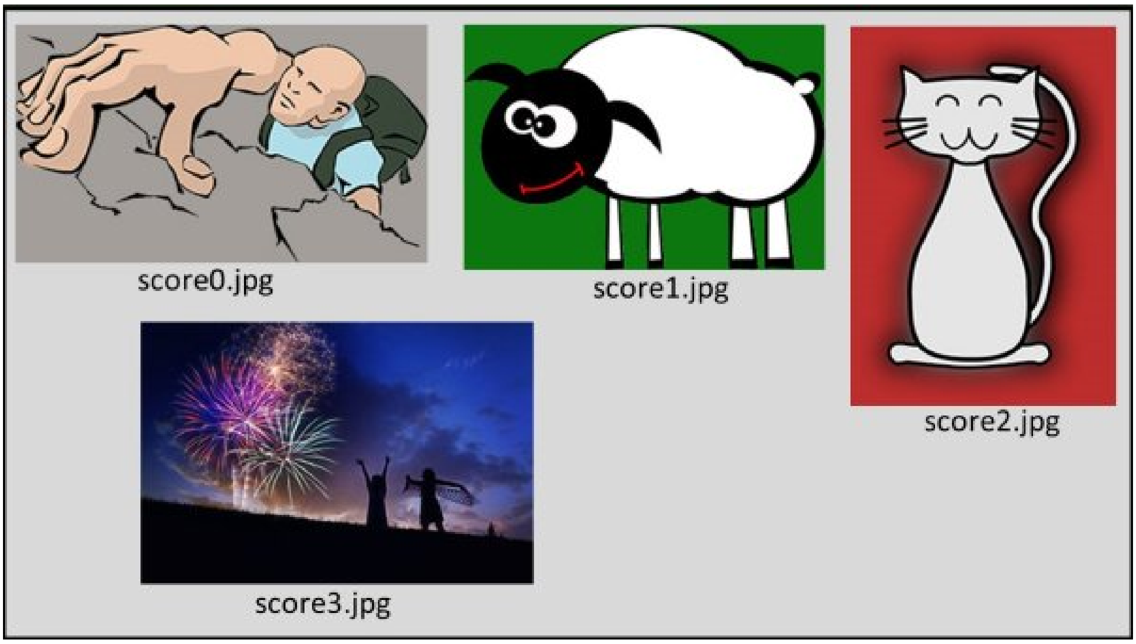
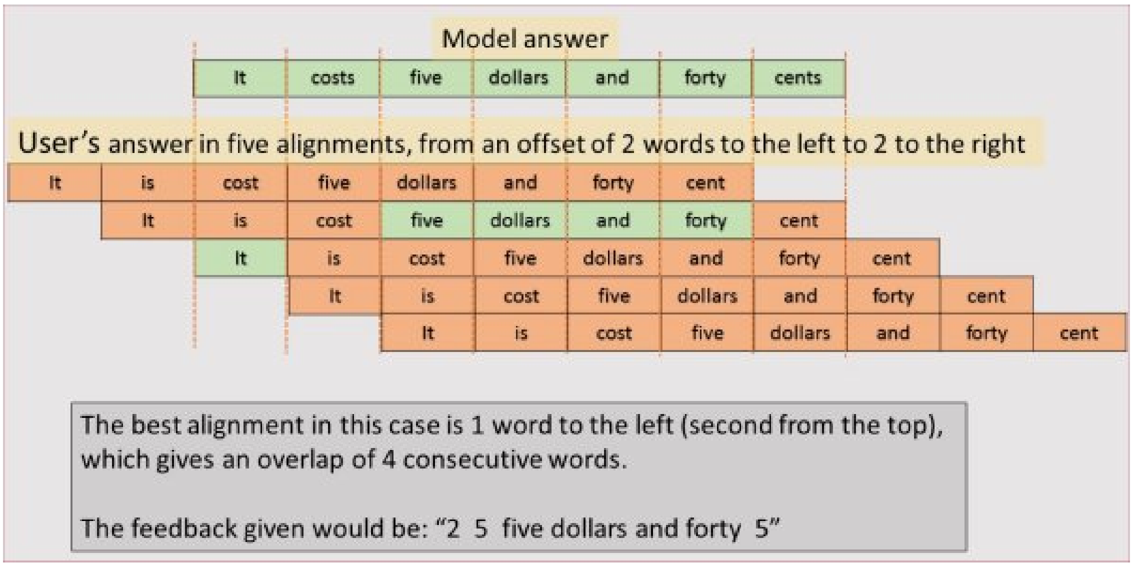


Figure 4. The supplied feedback images.

When the user's answer does not match exactly any of the designated correct answers, corrective feedback appears. If the length of the user's answer is too far from the length of any of the correct answers (word count less than 0.6 times or greater than 1.5 times the word count of the correct answer), a message appears saying this, and the user must try again. Otherwise, feedback is based on the correct answer which most closely matches the user's attempt, and shows the words which correspond exactly in both, in their correct positions. Incorrect words are substituted by a number indicating the number of letters in the correct word. To allow for a sequence of correct words in the user's answer which is slightly offset as regards position in the sentence, it is compared to the model answers in five alignments, with offsets ranging from 2 words to the left to 2 to the right, to find the longest matching sequence of words (best match). Figure 5 illustrates the process.



Model answer

		It	costs	five	dollars	and	forty	cents			

User's answer in five alignments, from an offset of 2 words to the left to 2 to the right

It	is	cost	five	dollars	and	forty	cent				
	It	is	cost	five	dollars	and	forty	cent			
		It	is	cost	five	dollars	and	forty	cent		
			It	is	cost	five	dollars	and	forty	cent	
				It	is	cost	five	dollars	and	forty	cent

The best alignment in this case is 1 word to the left (second from the top), which gives an overlap of 4 consecutive words.

The feedback given would be: "2 5 five dollars and forty 5"

Figure 5. An example of checking the user's answer in the 5 alignments.

A couple of examples are now given, for further explanation:

(1) Correct answer: *They've gone to the movies.*

User's answer: *They have gone to the movies.*

Feedback: *7 gone to the movies.*

Note that the user's answer is correct as regards grammar and meaning, but if it is not listed among the teacher's permitted answers, it is regarded as wrong. In this case, the teacher has decided to enforce the use of a contracted form. "gone to the movies" is correct, so it is displayed in full, and "7" signifies "They've", which has seven characters including the apostrophe.

(2) Correct answer: *He went there yesterday.*

User's answer: *He went to there yesterday.*

Feedback: *He went 5 9.*

Note that there are 2 correct word sequences of 2 words each in the user's answer, but only one is shown. Showing both would give away the complete correct answer in this case. Of course, if the student's answer had been "She went to there yesterday", the feedback would have been "2 4 there yesterday".

This feedback is intended to provide the user with a solid basis to improve the next attempt, without making the correct answer too obvious. In fact, disclosure of the number of letters in each missing word will narrow down the possibilities considerably and, in many cases, allow the user to answer next time with a high degree of confidence. It might be argued that this makes the challenge too easy, and displaying something like "BLANK" for each word instead of the number of letters would be preferable. This would correspond to the "indirect CF" of Ellis (2009), specifically his type 2a, "indicating + locating the error" (p. 98), whereas the method used could be said to fall between this and his "direct CF", where the correct answer is provided. However, if the user does find that the feedback makes the correct answer very easy to determine, this probably means that they already have quite a good idea what it should be, from a narrow range of possibilities. Also, an additional factor which should be considered with this system is that, where the user is trying to answer by speaking, even if the correct answer is known, an additional challenge is to pronounce it aloud so that Google ASR will recognize it as intended. This factor swung the decision in favor of using what we might call "semi-direct CF" instead of indirect CF in this original version this system. A variant version would, of course, be very simple to produce. As the system allows "hybrid" input by speaking and/or writing, and also can be used in a variety of different ways, as will be discussed in the next section, it is difficult to specify an optimum one-size-fits-all type of feedback. It is all the more difficult since, even in a given situation, such as pronunciation (Golonka, Bowles, Frank, Richardson, & Freynik, 2014, p.81-82; Levy & Stockwell, 2006, p189-190) or composition (Ellis, 2009; Guenette, 2007) training, the question of what kind of CF (including none) is most effective remains contentious.

6. Usage possibilities

Basically, the system provides the teacher with a simple way to set up browser-based CALL material which will elicit spoken or written answers from the user and check them against a list of specified permitted answers, giving feedback with hints when necessary.

Typically, the response will be elicited through a direct question, which the user can listen to and/or read, and the user may need to refer to the image/sound/video and/or extra text, if provided. In the simplest case (e.g., "What's the day after Tuesday?" to elicit "It's Wednesday"), no extra information is required. In the example shown in Figure 1, the user will need to refer to the image to answer the question "What's the dog doing?" Straightforward reading or listening comprehension questions may be based on text or media, or the questions may be designed to give practice in particular sentence patterns, rather than testing understanding.

A possibility that doesn't use direct questions to elicit the response is "listen and repeat", which combines listening and pronunciation practice. The text could also be shown, to remove the listening comprehension element and concentrate on pronunciation, though the sound file would still serve as a pronunciation model. Conversely, the user could be required to type the answer to put the focus on listening, making a classic dictation exercise. Of course this system could not provide very rigorous CAPT, since the feedback is entirely based on the results of Google's ASR, which uses AI techniques to make sense of the input by combining results from its Acoustic, Pronunciation, and Language models (https://www.google.com/about/careers/stories/how-one-team-turned-the-dream-of-speech-recognition-into-a-reality), so that intelligible, rather than native-like, pronunciation (Munro, 2011; Witt, 2012) is sufficient to produce a correct result.

The system allows a teacher to harness the power of Google's ASR in a very flexible and simple way, though it has several limitations.

7. Limitations

The system works well on Windows, Macintosh, and Linux desktop or laptop computers with a fast internet connection, but the Google Chrome browser must be used. ASR does not work with other browsers, though all the other features do. Although it is designed to adapt to all screen sizes, the system's ASR will not work at all on iOS devices, even if Chrome is used. It does work on Android, but the ASR results can be much inferior to those obtained on desktop or laptop machines. Even under the best conditions, Google ASR is, of course, not perfect. It may be very difficult for a speaker to get it to successfully interpret a short, single word like "two", as there are so many variations in how individual speakers may pronounce it, and there is no context supplied to aid interpretation. However, a phrase like "one, two, three" will be recognized far more easily. Similarly, unless it is pronounced in a very specific way, the single word "lung" will be taken as the much more common word "long", but "heart, lung, kidney" causes no such problem. Of course, much the same would apply to a human listener. The teacher should bear this in mind when designing material.

For maximum compatibility across platforms, image files are limited to the three types, jpg, png, and gif; sound files must be mp3, and video files mp4. When suitable media files are available, they are reasonably easy to deploy by just including their filenames at the appropriate points in the setup file and uploading the files themselves with the other system files. However, the teacher must be careful to avoid copyright breaches, and making customized media files is time-consuming and requires some technical expertise. Each media file must be separately included. There is no way of, for instance, using different sections of a large sound or video file.

The teacher needs to have access to a website to which they can freely upload files, and the minimal expertise to do so. If HTTPS protocol is not used, each time the user hits the microphone icon to input speech, they may get an additional screen message requiring confirmation of permission to use the microphone, and it will not be activated until they hit the *Allow* button. Recent versions of the Chrome browser seem to have become much less intrusive in this regard, requiring confirmation only once at the start of a session. Also, HTTPS support is becoming more common as a free option on hosting services. If it is not available by default, an SSL certificate to enable it can be obtained from a Certificate Authority such as *Let's Encrypt*, https://letsencrypt.org, which is a non-commercial organization offering a free, automated service.

The teacher's model answers may contain punctuation and capitalization, but because of the difficulty of including these through ASR, they will be ignored in checking the user's answer.

8. Pilot study

A small pilot study to gauge user reaction was carried out individually with six Japanese nursing students, of mixed English ability, whose course includes several compulsory English subjects. The functioning of the system was explained with a few examples, and then they were asked to try it out, using a set of 25 questions, a few very simple (*Are you a student?* → "Yes, I am") and others a little more challenging (being shown a picture of a nurse with a stethoscope around her neck, and asked *What does she have around her neck?* → "She has a stethoscope"), but none were very difficult, in order to avoid the difficulty of the material itself distracting from the central consideration of the principles of the system's functioning and its interface. When they had finished all the questions, they were asked to complete a short web questionnaire in Japanese consisting of three 5-point scale items, (in English translation) "Is likely to help your study of English?", "Is easy to use?", "Is enjoyable?", and three free input items, "Good points", "Bad points", "Any other comments". The researcher observed their use of the system, providing assistance if needed, but the questionnaires were completed in private, to avoid pressure on the students.

The responses for the 5-point scale items were very positive, an average of 4.5 for "Is easy to use" and 4.8 for the other two items. In the free input section, positive points mentioned were, "helpful for pronunciation training" (most frequent), "enjoyable to use", "alternative answers and hints shown", and "can enjoy studying even if weak at English". Negative points or suggestions for improvement included desires for an enhanced hints option showing what words should be used (not just how many letters in each) and allowing their pronunciation to be heard, better microphone sensitivity (low voices were not picked up well), and a greater range of acceptable answers (though this last point is not inherent to the system, but at the teacher's discretion).

Observation of the students' trial suggested that the system has the potential to be a useful tool for language study. Some common shortcomings of Japanese speakers' pronunciation of English were apparent in particular items, for instance words with the "l" sound, like "yellow" and "cold", or the word "would", which the weaker students tend to pronounce as "ud". The Google ASR interpretation was often absurdly far from the intended utterance (for example, an intended "It's yellow" was interpreted as "8 year old" for several of the students), but they were very pleased if they could eventually get their intended words across after several attempts. In the worst cases they resorted to typing, but nobody in the trial went as far as using the "Give up" button.

The overall reaction was quite positive, indicating that the system may have a lot of potential as a tool for the language teacher. It is hoped that it will be applied and prove beneficial in many different teaching environments, and that improved or variant versions will make it all the more useful and adaptable.

9. Download

The system can be downloaded from http://www.mcn-moodle.org/asr. The zip file includes the HTML file and images folder, which contains all the image files used in the user interface. These may left unchanged, though teachers are free to make their own customizations by editing the HTML file or replacing any image files with their own. The controlling text file used in the online example is also included. The teacher will need to edit this, or make a new one, to set up new material. The author also gives permission for teachers to make modified versions of the system for educational use by editing the HTML or JavaScript, provided they do not claim the original or modified system as their own work.

References

Aist, G., (1999). Speech recognition in Computer-Assisted Language Learning. In Cameron, K. (Ed.), *CALL: Media, design & applications* (pp. 165-181). Lisse: Swets & Zeitlinger.

Bernstein, J., Najmi, A., Ehsani, F. (1999). Subarashii: Encounters in Japanese Spoken Language Education. *CALICO Journal*, *16(3)*, 361-384. Retrieved from https://calico.org/html/article_619.pdf.

Bernstein, J., Van Moere, A., & Cheng, J. (2010). Validating automated speaking tests. *Language Testing*, *27*(3), 355-377.

Ellis, R. (2009). A typology of written corrective feedback types. *ELT Journal*, *63(2)*, 97-107.

Eskenazi, M. (1999). Using automatic speech processing for foreign language pronunciation tutoring: Some issues and a prototype. *Language Learning & Technology*, *2*(2), 62-76.

Golonka, E. M., Bowles, A. R., Frank, V. M., Richardson, D. L., & Freynik, S. (2014). Technologies for foreign language learning: a review of technology types and their effectiveness. *Computer Assisted Language Learning, 27*(1), 70-105.

Guenette, D. (2007). Is feedback pedagogically correct?: Research design issues in studies of feedback on writing. *Journal of Second Language Writing, (16)*, 40-53.

Levy, M. & Stockwell, G. (2006). *CALL dimensions: Options and issues in computer-assisted language learning*. Mahwah, NJ: Lawrence Erlbaum Associates.

Munro, M. J. (2011). Intelligibility: Buzzword or buzzworthy? In. J. Levis & K. LeVelle (Eds.). *Proceedings of the 2nd Pronunciation in Second Language Learning and Teaching Conference*, Sept. 2010. (pp.7-16),Ames,IA: Iowa State University. Retrieved from http://jlevis.public.iastate.edu/2010%20Proceedings%2010-25-11%20-%20B.pdf.

Penning de Vries, B., Cucchiarini, C., Bodnar, S., Strik, H., & van Hout, R. (2015). Spoken grammar practice and feedback in an ASR-based CALL system. *Computer Assisted Language Learning, 28*(6), 550-576.

Strick, H. (2012). ASR-based systems for language learning and therapy. In O. Engwall (Ed.), *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training* (pp. 9-20). Retrieved from http://www.speech.kth.se/isadept/ISADEPT-proceedings.pdf.

van Doremalen, J., Boves, L., Colpaert, J., Cucchiarini, C, & Strik, H. (2016). Evaluating automatic speech recognition-based language learning systems: A case study. *Computer Assisted Language Learning*, 29(4), 833-851.

Witt, S.M. (2012). Automatic Error Detection in Pronunciation Training: Where we are and where we need to go. In O. Engwall (Ed.), *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training* (pp. 1-8). Retrieved from http://www.speech.kth.se/isadept/ISADEPT-proceedings.pdf.

Zechner, K., Evanini, K., Yoon, S., Davis, L., Wang, X., Chen, L., Leong, C. W. (2014). Automated Scoring of Speaking Items in an Assessment for Teachers of English as a Foreign Language. *Proceedings of the Ninth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 134–142). Retrieved from http://www.aclweb.org/anthology/W14-1816.