March 2020

SYSTEM AND METHOD FOR GLOBAL TRAINING

Thomas Stern

Evan Jones

Tyler Saunders

Supriya Iyer

Sherri Schachter

See next page for additional authors

Inventor(s)
Thomas Stern, Evan Jones, Tyler Saunders, Supriya Iyer, Sherri Schachter, and Kathryn Kearney

System and Method for Global Training

Creating and maintaining translated documents targeted to global markets can be a daunting task. Many businesses strive to disseminate information to their international offices simultaneously, for example to train all employees on new matter using slide-based presentations. It can be expensive, time-consuming, and cumbersome to provide translated versions of these presentations to offices around the world. Oftentimes, as technology changes rapidly, the content of the presentations can be outdated by the time the presentation is translated. Maintenance of the presentation is also difficult because it is time-consuming and uses up resourced needed to develop the next training document, for example. Current translation services fail to address the underlying business and production issues companies face when translating content for their employees. The current solutions result in delay to market of up-to-date training, significant cost, slow speed of translation, and quality control issues.

The system and method described herein disclose a cloud-based service that receives slide content as input and produces a translated video as output. The proposed technique uses artificial intelligence products as well as a phrase-based translation application programming interface (API) and text-to-speech voice synthesis to create intermediate artifacts that can be used "as is," or that can be preserved, modified, and reused for production efficiency. The intermediate artifacts may include structured original text (for example, in JavaScipt Object Notation (JSON) format); image files (for example, in Portable Network Graphics (PNG) format); translated text script in either Unicode text or JSON format (or both); MP3 audio of narration produced by speech synthesis in all target languages; MP4 video artifacts, one file per sentence, per slide, or per video/slide deck; MP4 video artifacts with or without captions and/or subtitles; SRT files for subtitles or captions, etc.

The technique presented herein solves translation production issues by turning the production process into a build process. That is, a user can change text for one slide, for example, regenerate only a file for that one file and then reassemble the final video. It should be noted that while the following description uses a slide presentation with individual slides as an example, the proposed technique can be applied to other types of documents and media files (e.g., documents with individual pages or the like).

Figure 1 illustrates a flow diagram of a method for creating a translated version of a slideshow-based document. A user may upload source files to cloud storage from either a web-based front-end interface, or from an add-on or feature in an application of the content platform. The method described below may then be performed on the source files by a collection of software services hosted on a cloud computing platform.

The method may be performed by the following components: a presentation intake component, a translation component, a synthesizing component, and a video creator component. The components may be part of a cloud storage application or any other application. Alternatively, each component or any subset of the components can be a separate application hosted on a cloud computing platform.

At block 105, the presentation intake component can receive source files from a user device. In the illustrated example, the source file may be a slide presentation document including one or more slides. The user device can be any computing device that can access the cloud computing platform. The slide presentation document may be generated using, for example, a presentation application of the cloud computing platform.

At block 110, the presentation intake component creates an image corresponding to each slide. Each image file, which may be in PNG or other format, is saved as an intermediate artifact which may be downloaded, altered, and/or uploaded by the user during the production process. The presentation intake component may drop the background of the slide and make it transparent.

At block 115, the presentation intake component extracts the script, such as the speaker notes, for each slide. The presentation intake component saves the scripts as plain text, and processes the plaint text files into structured files such as JSON files. Alternatively, the user may upload a plain text file for each slide to be processed into structured JSON files. The structured original text in JSON format is saved as another intermediate artifact which may be downloaded, altered, and/or uploaded by the user during the production process. According to one example, the presentation intake component may detect that a slide does not contain speaker notes, in which case the presentation intake component may create a blank text file.

At block 120, the translation component translates the JSON file into one or more target languages, creating a text file for each language. The user may indicated one or more target languages in which to translate the content. The translated text script may be saved in Unicode text or JSON format, for example. The translated text is saved as another intermediate artifact, which may be downloaded, altered and/or uploaded during the production process. In some implementations, a machine learning model may be trained to fix common mistakes in the translated text, whether based on jargon, commonly used expressions or idioms, for example.

In one embodiment, the technique can translate the content of the image, in addition to translating the script. That is, the presentation intake component may create another plain text intermediate artifact containing the text from the content of the slide itself. The presentation intake component may use image identification or optical character recognition (OCR) to extract the text from the slide image and create a separate structured JSON file. The translation component may then translate the separate structured JSON file into one or more target languages, creating a text file for each language. The translation component may then replace the language on the slide with the translated text in the image file.

At block 125, the synthesizing component synthesizes a voice from each JSON file, creating an audio file corresponding to each slide using text-to-speech conversion on the translated structured files. The synthesizing component uses the JSON file as input, and creates an audio file, such as an MP3 file, of the narration. The user may choose a female voice, a male voice, or a diversity option which alternates between female and male voices with each slide. The user may also choose an accent. For example, English may be American English, British English, or Australian English. Each audio file is saved as an intermediate artifact, which can be downloaded, altered and/or uploaded by a user during the production process.

At block 130, the synthesizing component may determine the duration of the audio files corresponding to each slide. The synthesizing component may determine that a slide does not include a script, in which case it may set duration of silent audio, for example three seconds of white noise. Other durations may be set by the user.

At block 135, the video creator component generates a silent video for each slide by including the still image of each slide for the determined duration of the audio file corresponding to the slide.

At block 140, the video creator component merges the silent video file with the audio files of the slide to create a single voice-over video file for each slide.

At block 145, the video creator component merges the voice-over slide video files corresponding to each slide to create a final voice-over slide video containing all the slides in the presentation.

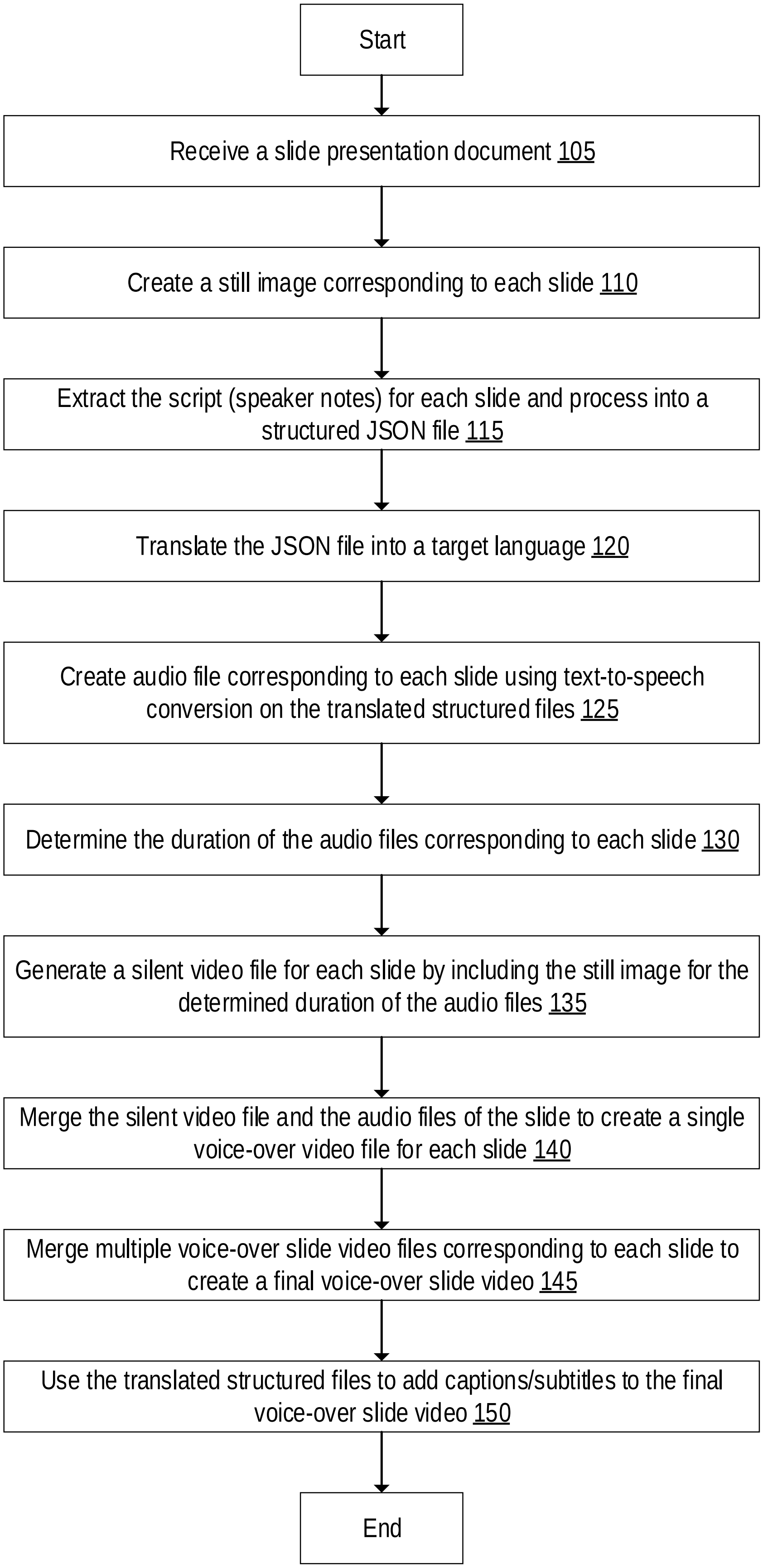
At block 150, the video creator component uses the translated structured files to add captions or subtitles to the final voice-over slide video. Captions are in the same language as the audio, whereas subtitles are in a different language from the audio. Captions can be used to help when the viewer has difficulty hearing or understanding the audio, whereas subtitles can be used to help when the viewer doesn’t understand the language of the audio. The user may choose to add either subtitles or captions.

In order to add captions to the final video product, the synthesizing component may create one MP3 audio file for each sentence in the target language. The synthesizing component may then determine the length of that audio file, which will determine how long the caption will remain on the screen. That is, the caption will remain on the screen for the amount of time necessary for the voice to speak the words. The synthesizing component may then use the JSON artifact, plus the time determination, to create an SRT file for each sentence. An SRT file is a plain-text file that contains the information needed to execute the captions or subtitles, including the start and end timecodes of the text to ensure the captions or subtitles match the audio as well as the sequential number of the captions or subtitles. The video creator component may then create silent MP4 videos for the duration of each spoken sentence. Multiple video files may share the same still image file as an image may have more than one sentence of script. Each individual silent video file is generated with the SRT file. The video creator component then merges the audio and silent video files including the captions to create a final video. The process can be similar to add subtitles to the final video product, however the technique can use translated text as input when creating the SRT file.

ABSTRACT

A technique is proposed to translate slide-based documents, for example, into videos with or without captions or subtitles. The presentation intake component extracts a script or speaker notes from the content inputted by the user, and creates a text file corresponding to each slide or image. The translation component then translates the script into multiple target languages. The synthesizing component uses a text-to-speech conversion to create an audio file for each slide or image. The video creator component creates a silent video for each image for the duration of the audio file corresponding to each still image, with or without captions or subtitles created using the translated text. The video creator component merges the silent video files with the audio files in order to produce a final video translation. Along each step of the way, the technique saves each file as intermediate artifacts, which may be downloaded, altered and/or uploaded by the user during the production process.

Keywords: training documents, slide translation, voice-over-slide video, intermediate artifacts,


Start
Receive a slide presentation document 105
Create a still image corresponding to each slide 110
Extract the script (speaker notes) for each slide and process into a structured JSON file 115
Translate the JSON file into a target language 120
Create audio file corresponding to each slide using text-to-speech conversion on the translated structured files 125
Determine the duration of the audio files corresponding to each slide 130
Generate a silent video file for each slide by including the still image for the determined duration of the audio files 135
Merge the silent video file and the audio files of the slide to create a single voice-over video file for each slide 140
Merge multiple voice-over slide video files corresponding to each slide to create a final voice-over slide video 145
Use the translated structured files to add captions/subtitles to the final voice-over slide video 150
End