CORE
Metadata, citation and similar papers at core.ac.uk
Provided by Technical Disclosure Common

Technical Disclosure Commons

Defensive Publications Series

October 04, 2019

INTELLIGENT TELEPROMPTING SYSTEM FOR DELIVERING PRESENTATIONS

Sandro Feuz

Thomas Deselaers

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Feuz, Sandro and Deselaers, Thomas, "INTELLIGENT TELEPROMPTING SYSTEM FOR DELIVERING PRESENTATIONS", Technical Disclosure Commons, (October 04, 2019)
https://www.tdcommons.org/dpubs_series/2548

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

INTELLIGENT TELEPROMPTING SYSTEM FOR DELIVERING PRESENTATIONS

An efficient presentation may depend on concise presentation contents (e.g., visual slides) and an engaging and effective delivery of the contents. Traditionally, presenters often prepare their speech ahead of time and then they may memorize the speech or print it for use during the presentation (e.g., use printed cards during the presentation to refresh their memory). This method may not be the most efficient particularly with long presentations where memorization may not be feasible. Additionally, looking at numerous cards may adversely impact the effective delivery of the presentation as it causes the presenter to look away from the audience to read the prepared cards repeatedly. Another conventional approach may be to use a teleprompter instrument for presenting the speech to presenters while they are delivering the presentation. Teleprompter instruments may depend on a dedicated production assistant to manually advance the displayed text as a presenter is delivering it. Other teleprompter instruments may allow a presenter to estimate their speaking speed and may advance the displayed text at a fixed rate as the presenter is delivering the presentation. These conventional methods do not allow for automatic adaptation of advancing presentation text due to instances where presenters need to deviate from the estimated speed or contents of their speech (for example, to answer questions or to elaborate on a particular topic).

Therefore, a technique is proposed for implementing a teleprompting system that can intelligently and automatically advance displayed text while a presenter is delivering his/her presentation, based on the contents spoken by the presenter. The disclosed technique can be implemented as a software application that may be installed on a computer system (e.g., the computer system where the presentation slides will be displayed). The teleprompter application may receive a text file containing speech of a presentation via a User Interface (UI). The text file

may be in a format that allows for the presence of formatting tokens within the text to highlight certain words or phrases to the presenter. For example, an extended markup language (XML) file with formatting tags may be received. The teleprompter application may then display the contents of the text file on a UI screen for the presenter to see while delivering the presentation. As the presenter speaks, the teleprompter application may record the speech uttered by the presenter according to one or more predetermined criteria. The predetermined criteria may be to capture the presenter speech, as close to real time as possible, and send the captured speech for comparison with the text file contents at an analysis component. In one example, the analysis component may compare each word of the captured speech with matching words in the text file. In this case, the analysis component of the teleprompter application may use other words before and after the word being compared, to build context for comparison with the text file.

The teleprompter application may then run a machine learning speech recognition tool to transcribe the presenter's speech into recognized text. Subsequently, the teleprompter application may compare the recognized text with contents of the text file to determine whether or not to advance the text displayed on the UI screen for the presenter to see the chunk of text following the recognized text.

In order to achieve an efficient comparison that allows for anticipated differences between presenters, the teleprompter application may utilize loose or fuzzy matching algorithms for the comparison process. For example, the teleprompter application may utilize a semantic match algorithm to determine whether the spoken text matches with some content in the text file. Semantic matching is a technique used to identify terms that semantically correspond to one another. For example, the technique can identify that the term "car" is semantically equivalent to the term "automobile" because they are synonyms of each other in the English language. This

semantic matching technique may provide flexibility in recognizing a match between the spoken text and the prepared contents in the provided text file. The flexibility in comparing spoken versus prepared text may be desirable because some presenters may naturally deviate from the prepared speech while delivering the presentation while other presenters may follow their written speech verbatim. If the teleprompter application determines that a matching exist between recognized text and contents of the text file, the teleprompter application may advance the text displayed on the UI Screen, such that the recognized portion is scrolled off screen and the chunk of text that follows the recognized portion within the text file is displayed on the UI Screen for the presenter to see. If, on the other hand, the recognized text does not match content in the text file, the teleprompter application may not advance the displayed text on the UI screen and may continue to record the presenter speech according to the predetermined criteria as explained above.

Figure 1 illustrates a flow diagram of a method for intelligent teleprompting system for delivering presentations. First, at block 100, the teleprompting system may receive, via a user interface (UI) hosted on a computing device, a text file containing speech of a presenter. The speech may be prepared by a presenter of the presentation to serve as notes for aiding the presenter in delivering the presentation, for example. The text file may be received by the teleprompting system before the presentation starts. In one example, the text file may be uploaded to the teleprompting system from a memory device (e.g. a memory stick). In another example, the teleprompting system may receive a hyper link to a remote location where the text file is hosted. In this case, the teleprompting system may download a copy of the text file into local memory so that the text file may be locally accessible during the presentation.

In one example, the contents of the text file may consist of pure text with no formatting tags or special tokens. In another example, the contents of the text file may consist of text representing the actual speech, as well as formatting tags for stressing some words or phrases within the speech. For example, the text file may contain an opening tag and a closing tag . The opening and closing tags may be interpreted by the teleprompting system to mean that the text between the opening tag and the closing tag may be displayed to the presenter in a bold font, in a bigger size font, in underlined format, etc. The presenter may then remember to stress the importance of the highlighted text when delivering the presentation. In another example, the text file may further contain special tokens within the text to mark significant locations within the text including the beginning of a slide, the end of a slide, a slide number, etc. In this case, when a presenter say "next slide" for example, the teleprompting system may recognize that the presenter is making a request to advance the displayed text such that at least a beginning portion of the slide following the currently displayed slide may be displayed on the teleprompter UI screen for the presenter to see.

In order to support the special tokens and the formatting tags, the text file may be in a known format that may allow for markup tags within the text (e.g. extended markup language (XML) formatted file, hypertext markup language (HTML) file, etc.).

Subsequently, at block 110, the teleprompting system may display contents of the text file on the UI screen of the teleprompting system. The teleprompting system may initially display a first chunk of text that can fit within the UI screen. The first chunk of text may be at least a portion of the first slide within the text file contents. As will be discussed in more details below, the teleprompting system may then advance or scroll the displayed text such that the following chunk of text is displayed, replacing the first chunk of text, as the presenter utters the

contents of each chunk. As another example, the first chunk of text may be replaced with another chunk of text in an out of order fashion if the presenter requests the teleprompter system to jump to a particular slide that may be a few slides before or after the currently displayed slide.

At block 120, as the presenter speaks while delivering the presentation, the teleprompting system may record the presenter speech according to a predetermined criteria, such that the spoken content may be compared with contents within the text file. The predetermined criteria may dictate how to record the presenter's speech before sending the recorded speech for comparison with the text file contents. In one example, the predetermined criteria may be to capture the presenter speech, as close to real time as possible, and send the captured speech for comparison with the text file contents at an analysis component. In one example, the analysis component may compare each word of the captured speech with matching words in the text file. In this case, the analysis component of the teleprompter application may use other words before and after the word being compared, to build context for comparison with the text file.

When there is an audio segment representing a portion of the presenter speech ready for processing, the teleprompting system at block 125 may run a machine learning speech recognition tool to transcribe the audio segment into recognized text. In one example, the machine learning speech recognition tool may be a known speech recognition system that may accept an audio input and may generate a chunk of text as an output. In another example, the speech recognition tool may accept another form of media content as an input (e.g. video stream), and may extract the audio component of the media content for transcribing into a text format. In another example, the speech recognition tool may also generate a presenter identifier based on the voice of the presenter in the audio contents, In this example, the teleprompting

system may be configured to identify the main presenter as the presenter that speaks first, for example, and may accept only the audio segments uttered by the main presenter for comparison and further processing.

At block 130, the teleprompting system may compare the recognized text with contents of the text file to determine whether or not to advance the displayed text on the UI Screen. Because presenters differ in how closely they follow their notes while delivering the presentation, an efficient comparison process may need to utilize similarity metrics that may allow for acceptable deviation from the prepared text. For example, the teleprompting system may utilize a semantic match algorithm to determine whether the spoken text matches with some content in the text file. Semantic matching is a technique used to identify terms that semantically correspond to one another. For example, the technique can identify that the term "car" is semantically equivalent to the term "automobile" because they are synonyms in the English language. This semantic matching technique is desired because it provides flexibility in recognizing a match between the recognized text versus the contents in the provided text file. This flexibility may accommodate a variety of presentation styles from strictly following the prepared text to loosely following the notes while adding contents and answering questions when delivering the presentations, as explained in more details herein.

At block 140, if the semantic matching tool determined a match between the recognized text and contents of the text file, the teleprompting system may advance the displayed text on the UI screen at block 150 to show new chunk of text to the presenter. For example, the teleprompting system may scroll at least a portion of the recognized text off the UI screen and replace it with the chunk of text that follows the recognized text within the text file, such that the presenter may see the following chunk of text on the UI Screen. In another example, if the

presenter requested the teleprompter to advance to a particular material within the text file contents (e.g. by saying "next slide"), the teleprompting system may scroll a portion of the recognized text off the UI screen and replace it with a chunk of text from the requested slide for the presenter to see. If, on the other hand, the recognized text does not match content in the text file, the teleprompting system at block 160 may stop advancing the displayed text on the UI screen. The teleprompting application may then continue to record presenter speech, transcribe recorded audio into recognized text, and compare recognized text with contents of the text file until a new match is found or the presentation concludes. If a new match is found, the teleprompting application may then resume advancing the text displayed on the UI screen based on the new match.

Further to the description above, a presenter may be provided with controls allowing the presenter to make an election as to both if and when systems, programs or features described herein may enable collection of presenter information (e.g., information about presenter activities on the cloud-based content management platform, information about content of documents stored in the cloud storage, information about presenter schedule, a presenter preferences, or a presenter current location), and if the presenter has sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a presenter's identity may be treated so that no personally identifiable information can be determined for the presenter. Thus, the presenter may have control over what information is collected about the presenter, how that information is used, and what information is provided to the presenter.

ABSTRACT

A system is proposed for intelligently and automatically advancing presentation text on a teleprompter User Interface (UI) as the presenter speaks. The system includes receiving, by a teleprompter system, a text file comprising speech of a presenter via the UI. The system further includes displaying contents of the text file on a UI screen. As the presenter speaks, the system further includes recording presenter speech according to predetermined criteria. The system also include running machine learning speech recognition tool to transcribe the recorded speech into recognized text. The system also includes semantically comparing text from the text file with the recognized text. If the recognized text matches content in the text file, the system includes advancing displayed text on the UI screen to show the following chunk of text to the presenter. On the other hand, if the recognized text does not match content in the text file, the system includes stopping the advancement of the text on the UI screen.

Keywords: Teleprompter, delivering presentation, machine learning, speech recognition, semantic matching.

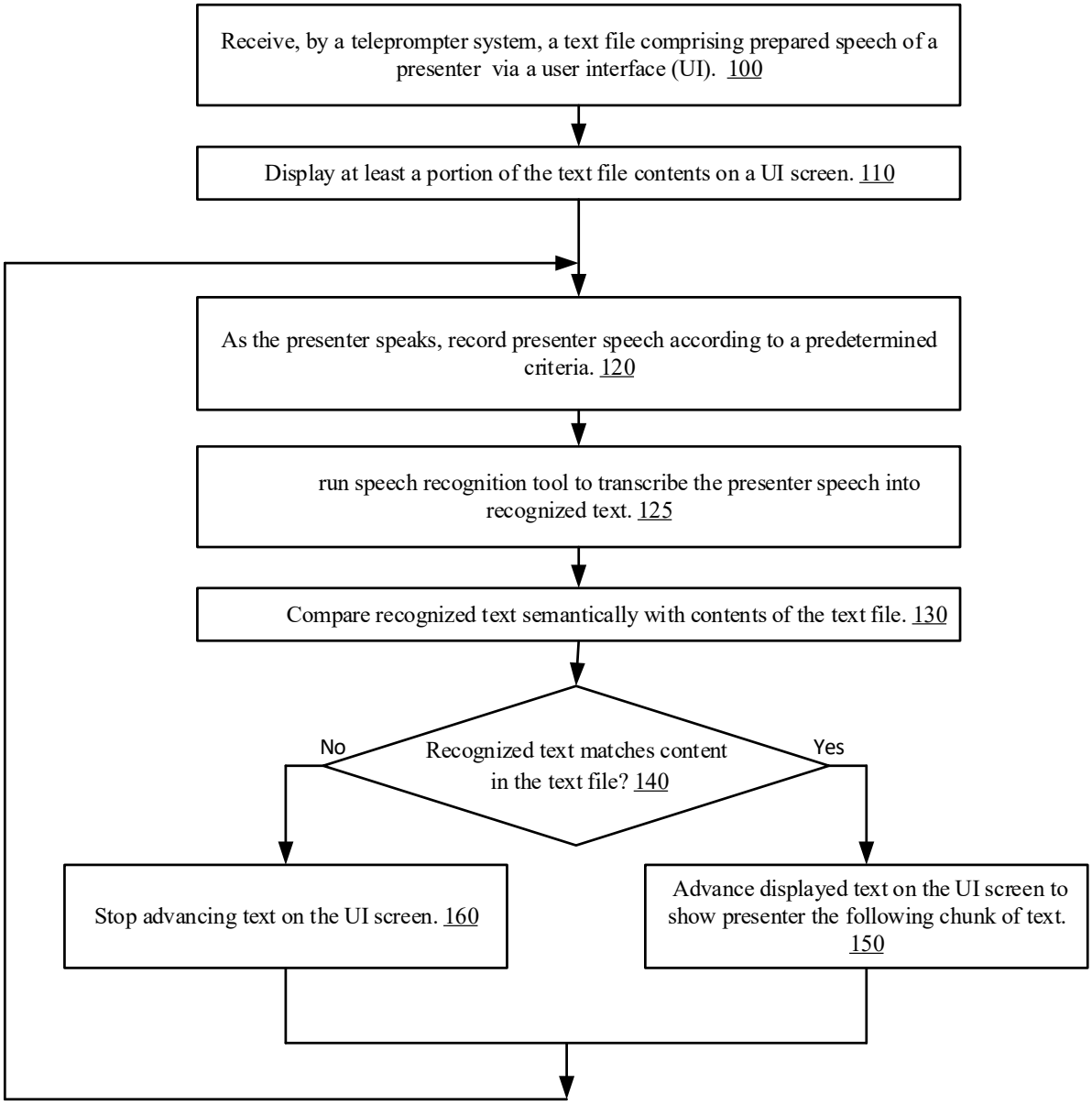
Receive, by a teleprompter system, a text file comprising prepared speech of a presenter via a user interface (UI). 100
Display at least a portion of the text file contents on a UI screen. 110
As the presenter speaks, record presenter speech according to a predetermined criteria. 120
run speech recognition tool to transcribe the presenter speech into recognized text. 125
Compare recognized text semantically with contents of the text file. 130
Recognized text matches content in the text file? 140
No
Yes
Stop advancing text on the UI screen. 160
Advance displayed text on the UI screen to show presenter the following chunk of text. 150

FIG. 1