SAIVT WebDemo

Project Proposal

Francesco Ferraioli n8323143

TABLE OF CONTENTS

PROJECT OUTLINE

Speech Audio Image Video Techologies (SAIVT) has been researching and implementing numerous techniques and algorithms for entity extraction from video's. Entities can be faces shown on the video, voices that are heard, objects and many more. Recently SAIVT has decided to showcase all of these research findings and algorithm through a web site. The project that I will undertake involves various improvements and feature implementation and enhancements to this web site.

The first task I will undertake is implementing enhancements to the website in terms of the User Interface (UI). I have been informed that the front end of the application is not written following best practices. First and foremost, my task will be to change this and ensure that the code for the front end is written well, easy to understand and renders as intended. Furthermore, once that is complete and the front end of the application is written well, I will begin to make the UI more user friendly and nice to look at. This will involve changing the style and the look and feel of the pages.

Moreover, there is currently some functionality in place for editing tags for a video. However, the functionality in place is not very user friendly and requires the user to follow specific rules. Part of this task will be changing that to a more user friendly implementation, allowing the user to simply remove them and add them individually and very easily.

The second and probably largest, most difficult and time demanding task will be implementing the functionality to add video's through the website itself. In this task a user will be able to navigate to a page to upload a video of their own to the website.

Once the user has filled in the various information related to the video and then selected the video to upload and submitted the form, the server must then be able to receive this data. Once the data is received, the server must then store the metadata temporarily and send the video off to be processed by a High Performance Computer (HPC) and extract the various entities. Once the processing is done, the entities extracted must be clustered with the current entities in the database from the other videos. What that means is checking if an entity that has been extracted from the video is the same as an entity extracted in another.

The third and final task of the project is to add enhancements to the displaying of the entity extraction results on the video play page. Once a face has been detected by the system during the processing, information is stored about the whereabouts on the video the face is shown. The system then creates another video from that video but adds on top a hollow square around the face, a different color for each face entity it sees on that video. This new video is the video that is displayed on the video play page. Part of this task would be to make the face on the video clickable and that once the user click it, it will link the user to a page with all the videos where that entity has been found.

Another enhancement to that page is to show a label on the bottom left hand of the video with the current speaker and a small image of them. This label will also be clickable and clicking on it, will link the user to a page with all the videos where the entity that was at that moment detected and stated as speaker has been found.

A possible fourth task would be to add users and add user permission. This would allow for actions to be limited only to users who are logged in and who have special permission rather than any user of the site.

PROJECT PLAN

As discussed in the Project Outline there are three main tasks and one possible other. I will now discuss my plan on how I will achieve all the tasks explained.

The first main task is that of implementing improvements to the UI of the website. This is a good initial task because it will also gives me time to get to understand firstly how Django works, but also get a feel of how the website functions and renders the views. Initially I would start by encapsulating all the inline styling to classes, making the html much neater and more professional. Furthermore, there is a lot of inline javascript which I will take out to one main global javascript file which gets loaded on page load. Once all of that is complete I will start integrating the Twitter Bootstrap Framework into the system. This would involve possibly taking out some of the current classes from the elements and adding others. Primarily, integrate the grid layout of the Bootstrap framework in place which is very specific and thus needs some time to integrate into the current system. Once the grid layout is in place I can begin to make the UI more user friendly. Bootstrap has many classes which can be added to buttons and other elements to make them look more appealing to look at.

The last improvement I will make as part of this task is enhancing the way a user can edit the tags of a video. Currently the user is provided with the list of tags as links and underneath a text area with all the tags listed out and a save button under that. The user can then edit the tags in the text area but they need to make sure they put each tag on a different line. However, I plan to implement this feature with an alternate method which is much simpler for the user. The plan is to add a remove link next to each tag link and to add a button which once clicked will show a form to allow the user to add another tag to that video. All this, both the adding and the removing of tags will be done through ajax, this means that there will be no need of page reload for these changes to take effect and the page to display these changes. Implementing this with ajax gives the user a much a better experience when using the website and wanting to add or remove a tag of a video.

As for the second task, the implementation of the feature which allows the user to add their own videos is much more specific than the last but much more time consuming. The first step would be to add a page for the user to submit the video along with the various data that would need to be stored with it (i.e. title, summary, etc.). Along side that is adding a link that the user can access anywhere that will link them to that page.

The processing of the video could take a long time and thus, as soon as the form has been submitted, as part of the form, the user will be asked if they wish to receive an email once the processing of the video is complete and if so, will need to enter the email. This will increase the user experience as they do not need to wait for the whole process to be done for them to continue browsing.

The next step would be the the implementation of the back end, which will accept the information submitted along with the video and store it. The next step would be to send the video to a HPC to do the processing of the entity extraction. A possible task here would be to increase the efficiency of the clustering of the entities with the current entities found in the other videos. Once all the extraction and clustering is complete the next step would be to send all the data that was collected back to the server and store it there. Once that happens, the user will now be notified via email if they had previously selected it. At this stage the processed video will be part of the collection of video's displayed on the website.

The last major task for this project is enhancing the display of the entity extraction results on the video play page. I plan to start with adding a label somewhere to display the current speaker. After that I can begin to work on the enhancement of the face tracking, that is, having the box that follows the faces be a HTML element instead of a box simply part of the video. Both these tasks would involve delving into exactly what data we store about the entities in the videos and firstly determining if we need more data and then, once we have enough, use it to implement these enhancements.

The possible task is that of adding users and user permissions features to the website. This would mainly involve research on the Admin plugin that comes with Django and configure it and implement it on the website.

TIME PLANNING

In this section I will describe, through the use of a gantt chart, how I plan to manage my time and the time allocation I have place for each task.

The following gantt chart is separated weekly (i.e. each box is a week). This project is a year long project and thus I have 28 weeks as outlined in the gantt chart.

I plan to meet with Dr David Dean - who is a senior researcher at SAIVT and who is currently working on the webdemo – at least once a week. The day to meet has been decided to be Monday and thus the dates shown on the gantt chart is the monday of each week. On top of that I plan to complete work at home in my own time in order to complete the tasks.

I have developed this gantt chart from only the 3 major tasks. The fourth task will only be considered if I complete the tasks before the project ends.

SEMESTER 1 - BEB801

		Project Week 1	Project Week 2	Project Week 3	Project Week 4	Project Week 5	Project Week 6	Project Week 7	Project Week 8	Project Week 9	Project Week 10	Project Week 11	Project Week 12	Project Week 13	Project Week 14
		24/02/14	03/03/14	10/03/14	17/03/14	24/03/14	31/03/14	07/04/14	14/04/14	21/04/14	28/04/14	05/05/14	12/05/14	19/05/14	26/05/14
Task	**Sub-Task**	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Mid-Sem Break	Week 9	Week 10	Week 11	Week 12	Week 13
UI Enhancements	Clean up														
	Bootstrap Integration														
	Tag CRUD														
Add Video	Page setup														
	Back-end setup														
	File transfer to Lyra														
	Processing														
	File transfer to server														
	Back-end setup														
Entity display enhancements	Current speaker label														
	Face tracking html element														

SEMESTER 2 - BEB802

		Project Week 15	Project Week 16	Project Week 17	Project Week 18	Project Week 19	Project Week 20	Project Week 21	Project Week 22	Project Week 23	Project Week 24	Project Week 25	Project Week 26	Project Week 27	Project Week 28
		21/07/14	28/07/14	04/08/14	11/08/14	18/08/14	25/08/14	01/09/14	08/09/14	15/09/14	22/09/14	29/09/14	06/10/14	13/10/14	20/10/14
Task	**Sub-Task**	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Mid-Sem Break	Week 11	Week 12	Week 13
UI Enhancements	Clean up														
	Bootstrap Integration														
	Tag CRUD														
Add Video	Page setup														
	Back-end setup														
	File transfer to Lyra														
	Processing														
	File transfer to server														
	Back-end setup														
Entity display enhancements	Current speaker label														
	Face tracking html element														

LITERATURE REVIEW

As discussed, the system is a website which showcases a lot of SAIVT's work. SAIVT has taken numerous news related video's and has run them through many different script that they have engineered to extract entities from videos. These script include a combination of various speaker extraction, face extraction and optical character recognition scripts as well as speaker and face clustering to link the video's together as best as possible. I will now discuss some of my findings on both speaker and face recognition.

SPEAKER RECOGNITION:

Speaker recognition is defined as not only detecting when an individual is speaking but moreover the identification of the person, the individual, who is speaking. Over the years there has been much research in the area of the use of machines to automate speaker recognition (Campbell, J., 1997). There have been a vast quantity of algorithms and techniques implemented for automated speaker recognition, even over coming major difficulties like background noise (i.e. recognizing the speaker in a noisy environment), some more robust than others (Campbell, J., 1997).

A method used for speaker recognition, and one that is highly researched and used at SAIVT is that of Joint Factor Analysis (JFA) Modeling. This technique involves combining the appropriate estimation of speaker variability and session variability subspaces. (Vogt, R., et al, 2008). Each subspace are then “trained on a database containing a large number of speakers each with several independently recorded sessions.” (Vogt, R., et al, 2008).

A similar method is called “i-vector speaker modeling” (Ghaemmaghami, H., et al, 2013) and also this is heavily researched and used at SAIVT. I-vector speaker modeling differs from JFA modeling in that it “represent the GMM super-vector by a single total-variability space” (Kanagasundaram, A., et al, 2011). This total variability space contains both the speaker and the channel variability but requires “additional intersession compensation approaches” (Kanagasundaram, A., et al, 2011). There have been some existing approaches that have shown effective for this task including WCCN, LDA and NAP (Kanagasundaram, A., et al, 2011).

FACE RECOGNITION

Face recognition has been a hot topic, especially in the field of Human Computer interactions (Castrillon-Santana, M., et al, 2008). Face recognition was defined as “visual pattern recognition problem where the face, represented as a three-dimensional object that is subject to varying illumination, pose, expression and other factors, needs to be identified based on acquired images” (Jain, A., 2005).

The Viola-Jones object detection framework has drastically increased the number of facial processing applications (Castrillon-Santana, M., et al, 2008). All these applications have provided very reliable approaches to the problem of face recognition.

The basic idea behind the Viola-Jones framework, more specifically the Haar-Cascade technique, is based on the idea of “boosted cascade of weak classifiers” (Castrillon-Santana, M., et al, 2008). It is the process of sequentially testing the image against numerous classifiers and rejecting those that fail. These classifiers include multiple “Haar-like features” which is an encapsulation of rectangular patterns (Kasinski, A, 2010). There are three types of patterns, edge features, line features and centre-surround features (Kasinski, A, 2010).

SPEECH CLUSTERING

Speech Clustering is the process of linking the entities found in an audio file based on identity (Ghaemmaghami, H., et al, 2013). This is achieved "using speaker diarization and speaker linking" (Ghaemmaghami, H., et al, 2013).

The most extensively utilized method for speaker clustering in speaker diarization is agglomerative clustering with retraining (ACR) (Ghaemmaghami, H., 2013). Even though ACR is the most widely used technique, it was found to be very inefficient. For this reason, when conducting speaker linking in large dataset, another method is preferably used (Ghaemmaghami, H., 2013).

This more efficient method is called complete-linkage clustering (CLC). This is achieved through merging based on a highest similarity, or lowest distance, score. (Ghaemmaghami, H., et al, 2013). Moreover, another technique for speech clustering is named cross likelihood ratio (CLR) (Ghaemmaghami, H., 2013). This method has shown to be robust and effective especially when "incorporated into the JFA modeling framework" (Ghaemmaghami, H., 2013).

OPTICAL CHARACTER RECOGNITION (OCR)

Optical Character Recognition (OCR) is the process of converting typewritten or printed images to computer readable text (Mori S., 1997). This technique is mostly used in data entry, where it is necessary to import vast information stored in scanned images that needs to be computer readable. However, this process can be also very useful in entity extraction. (Mori S., 1997).

The most widely used engine for OCR is Tesseract, not only because it is open source, but also because it is very efficient (Smith R., 2007). Tesseract performs OCR by running various processes including: Line and Word Finding and Word Recognition (Smith R., 2007). Line and Word Finding are algorithms which essentially tell the computer where the lines of words are in the image. The Word Recognition process ultimately tries to understand exactly what computer readable word is a particular imaged word. (Smith R., 2007)

REFERENCES:

Campbell, J., (1997) Speaker Recognition: A Tutorial

Castrillon-Santana M., Deniz-Suarez O., Anton-Canalıs L., Lorenzo-Navarro J. (2008). Face and Facial Feature Detection Evalucation

Ghaemmaghami, H. (2013). Robust automatic speaker linking and attribution.

Ghaemmaghami H., Dean D., Sridharan S. (2013). Speaker Attribution of Australian Broadcast News Data

Jain A., Li, S. (2005). Handbook of Face Recognition.

Kanagasundaram A., Vogt R., Dean D., Sridharan S., Mason M. (2011). i-vector Based Speaker Recognition on Short Utterances

Kasinski, A., Schmidt, A. (2010) The architecture and performance of the face and eyes detection system based on the Haar cascade classifiers

Mori S., Nishida H., Yamada H. (1999) Optical Character Recognition

Smith R. (2007) An Overview of the Tesseract OCR Engine

Vogt R., Baker B., Sridharan S.. (2008). Factor Analysis Subspace Estimation for Speaker Verification with Short Utterances

Waheed K., Weaver K., Salam F. (2002). A robust algorithm for detecting speech segments using an entropic contrast