AN AUDIO INDEXING SYSTEM FOR ELECTION VIDEO MATERIAL

Christopher Alberti, Michiel Bacchiani, Ari Bezman, Ciprian Chelba, Anastassia Drofa, Hank Liao, Pedro Moreno, Ted Power, Arnaud Sahuguet, Maria Shugrina, Olivier Siohan

Speech Research Group, Google Inc.
79 Ninth Ave, New York, NY

ABSTRACT

In the 2008 presidential election race in the United States, the prospective candidates made extensive use of YouTube to post video material. We developed a scalable system that transcribes this material and makes the content searchable (by indexing the meta-data and transcripts of the videos) and allows the user to navigate through the video material based on content. The system is available as an iGoogle gadget[1] as well as a Labs product (labs.google.com/gaudi). Given the large exposure, special emphasis was put on the scalability and reliability of the system. This paper describes the design and implementation of this system.

***Index Terms*—** Large Vocabulary Automatic Speech Recognition, Information Retrieval, User Interfaces

1. INTRODUCTION

Given the wide audience that is reached by the YouTube (www.youtube.com) video sharing service, the candidates involved in the 2008 United States presidential election race have been making use of this medium, creating channels for election video material they want to disseminate[2]. The popularity of this medium is so large that YouTube has devoted a separate section to the election material named YouChoose (www.youtube.com/youchoose). The corpus of election video material is peculiar in that it is rich in speech, many of it is long form content (videos can be an hour long) and the information retained within is dense. This complicates the task for the end user when attempting to find relevant video material and navigating within a found video. To make this task easier, we developed an audio indexing system allowing both search and navigation of this video material based on content. This is similar to previous audio indexing work [1, 2, 3, 4] however here the focus is on video material, the content of the index is controlled indirectly by the video producers who manage the content on their channels and most importantly, the system is to be designed to scale allowing it to be applied beyond the election video domain. New video material found on the candidate channels is transcribed, videos are then indexed to facilitate search and a user interface allows the end user to navigate through the search results. If the content mangers of the channels take a video down, we remove them from our index. The user interface to interact with this corpus is available for general use on the YouChoose site as well as a Google Labs product (labs.google.com/gaudi). This paper describes the development of this system in more detail. Section 2 describes the system running data aquisition, data processing and the serving infrastructure for the user interface and search and retrieval.

[1]http://www.google.com/ig/adde?moduleurl=www.google.com/ig/modules/elections_video_search.xml or http://tinyurl.com/electiongadget

[2]Democratic candidate Barack Obama and Republican candidate John McCain use channels http://www.youtube.com/user/BarackObamadotcom and http://www.youtube.com/user/JohnMcCaindotcom respectively.

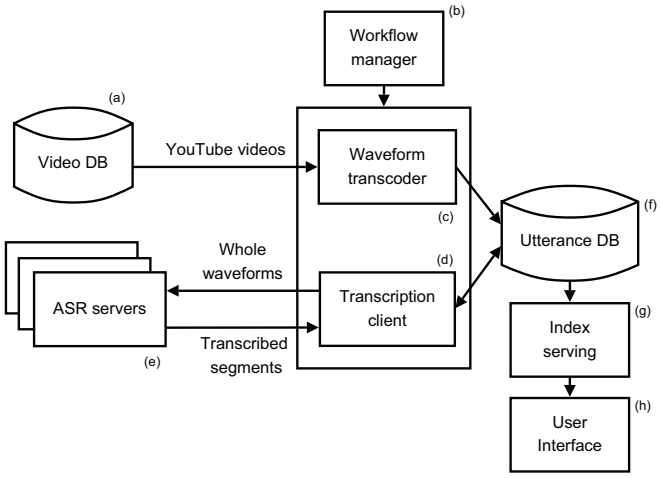

Fig. 1. Block diagram of the complete audio indexing system.

Section 3 describes the transcription system. Section 4 describes the information retrieval system. Finally, section 5 describes the user interface of the system.

2. WORKFLOW SYSTEM

Figure 1 gives an overview of our workflow system. The inventory of YouTube videos resides in a database (block a in figure 1), internally accessible at Google. This database contains for every YouTube video, among other information, the title, the description entered by the video owner, and a number of formats and locations where the video is available. The system periodically scans this database for changes (additions or deletions) of entries among the 79 politician channels of interest[3]. The workflow manager (block b in figure 1) coordinates the actions of a set of workers (blocks c and d in figure 1). It runs a periodic scan of the video database where the "waveform transcoder" workers (block c) extract the audio signal from new videos and downsamples them to a 16kHz sampled, 16 bit linear signal. Special attention is paid to select the least compressed source format since experimental results showed as much as a 10% WER degradation for video transcoded from compressed formats like the commonly used FLV format. In addition to periodic scanning, the workflow keeps a queue of transcoded videos which are processed by "transcription client" workers (block d in figure 1). These workers perform transcription by submitting the audio to the Automatic Speech Recognition (ASR) service (block e).

The ASR service segments the audio, discards music and noise

[3]http://www.youtube.com/members?s=mv&t=a&g=8 or http://tinyurl.com/politician-channels

and then performs the transcription using a multi-pass decoding strategy. The ASR service is described in more detail in section 3. The service uses Google infrastructure to allow scaling to large scale processing by simply expanding the number of machines that run the transcription service. The result of the transcription process consists of a time aligned transcript and confidence scores for each word. This information is stored in a system-local utterance database (block f) and serves as the basis for the information retrieval index (block g). The index allows search as well navigation within videos in the user interface (block h). These systems are described in more detail in sections 4 and 5 respectively.

There were two important requirements when designing the workflow system: reliability of the data storage and availability. We want to minimize the risk of losing data and ensure we are robust against machine outages. To do so, we replicated the workflow in two geographically distant locations meaning there are two identical workflow systems and two utterance databases that stay in sync through a replication mechanism. More precisely, every mutation made to one copy of the database is propagated to the other and vice versa. The replication provides for redundancy in storage providing some safeguard against data loss and robustness against machine outages. In addition, the replication provides for a load balancing mechanism between the two workflow systems when both workflows are "healthy". All individual system components are built upon scalable Google infrastructure and as a result the system capacity to handle queries or process videos scales by increasing the number of machines.

3. AUTOMATIC SPEECH RECOGNITION

The ASR service implements an audio segmentation and multi-pass ASR decoding strategy transcription engine. A video is first segmented into speech utterances and the utterances are subsequently transcribed.

The audio segmentation is based on 64-component Text Independent Gaussian Mixture Models (TIGMM). An ergodic HMM with state models representing the TIGMMs segments the audio stream into regions of speech, music, noise or silence by computing the Viterbi alignment. Utterances are then defined as the found speech regions. The speech utterances are clustered based on full covariance Gaussians on the Perceptual Linear Prediction (PLP) feature stream. Each cluster is forced to contain at least 10 seconds of speech and utterance in these clusters share adaptation parameters in the subsequent transcription process.

The baseline transcription system was a Broadcast News (BN) system trained on the 96 and 97 DARPA Hub4 acoustic model training sets (about 150 hours of data) and the 1996 Hub4 CSR language model training set (128M words)[4]. This system uses a Good-Turing smoothed 4-gram language model, pruned using the Seymore-Rosenfeld algorithm [5] to about 8M n-grams for a vocabulary of about 71k words. The baseline acoustic model is trained on PLP cepstra, uses a linear discriminative analysis transform to project from 9 consecutive 13-dimensional frames to a 39-dimensional feature space and uses Semi-tied Covariances [6]. The acoustic model uses triphone state tying with about 8k distinct distributions. Distributions are modeling emissions using 16-component Gaussian mixture densities. In addition to the baseline acoustic model, a feature space speaker adaptive model is used [6]. The decoding strategy obtains a first transcript using the baseline model running with a narrow beam (about 0.3 times realtime on a Pentium IV), then computes a feature space transform and maximum likelihood linear regression transform [6] and then decodes the data with a larger beam (about realtime on a Pentium IV). This final decoding stage produces word-lattices and word confidence scores are computed based on the lattice arc posterior probabilities. On the 1997 Broadcast news evaluation set[5], this system obtains a 17.7% WER.

To evaluate the transcription accuracy on the election video material, we sampled about 10 hours of material from 6 candidate channels on YouTube. In terms of duration, the videos in the sample ranged from 14 seconds to 3095 seconds with a mean duration of 167 seconds. Duration and count statistics are shown table 1. We had the sample manually transcribed resulting in 91138 word tokens and a vocabulary of 6978.

Channel	Number of videos	Sample duration (s)
JohnMcCaindotcom	85	10790
BarackObamadotcom	46	10712
explorehuckabee	41	3455
hillaryclintondotcom	32	5328
RonPaul2008dotcom	8	3583
gravel2008	3	642
Total	214	34510

Table 1. Duration and count of the videos sampled to evaluate transcription accuracy.

There were several unexpected results when applying the BN baseline system on the election data set and in the attempts to tune the system. The error rate of the baseline system is 40.1%, much higher than the BN test set. However, the retrieved videos did not appear very poorly transcribed. An error analysis using an histogram of error rates comparing BN utterances with election videos is shown in figure 2. It shows a much larger variance in error rates of the video material compared to the BN utterances. Another striking difference is the deletion rate, 3.0% for the BN system vs. 18.6% for the election set. Further inspection showed that some of the videos have a large deletion rate because the segmentation process removes noisy parts of the video (eg. a candidate in conversation with people on the street). Other high error rate videos show very mismatched recording conditions (eg. a microphone in the middle of a meeting room rather than a close talking microphone used in training and most test videos). This clearly shows that a corpus like this, be it constrained in terms of domain, is still much less controlled than a typical DARPA corpus.

Another surprising result is the out-of-vocabulary (OOV) measurement. The baseline system has an OOV rate of 1.42% (1284 OOV tokens out of 90109), 239 unique OOV words. Although this number is low, the impact of the OOVs is significant on the user perception. For example, lexical items like "Obama", "superdelegate" and "Putin" are important even if their token count will not dramatically swing the error rate.

To adapt our transcription engine to better suit the election video data, we obtained data from Google News[6]. We retrieved a sampling of the articles classified as election news (designated for the election part of the News site) for a 3 month period ending May 18, 2008. This text was normalized, most prominently mapping quantitative measures (eg. "$1.6 billion" to "one point six billion dollars") and common acronyms like state abbreviations (eg. "NJ" to "new jersey"). This resulted in a sample of 7.7M words.

[4]LDC corpora: LDC97S44 + LDC97T22 (AM96), LDC98S71 + LDC98T28 (AM97) and LDC98T31 (LM96).

[5]LDC corpus LDC2002S11

[6]http://news.google.com

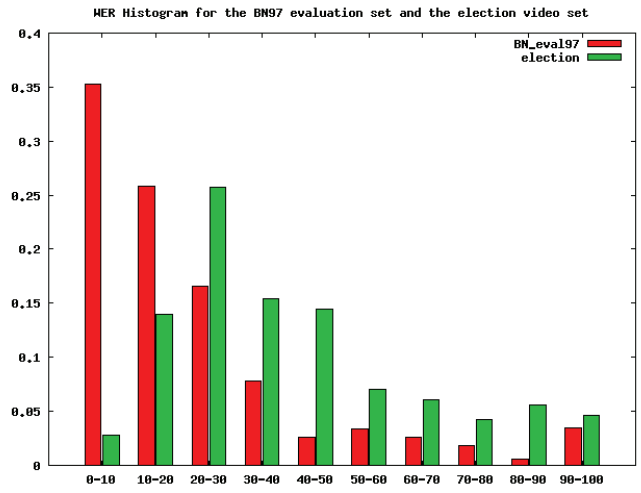

Fig. 2. Word error rate histogram of BN utterances and videos.

We built a Good-Turing smoothed 4-gram on the News data and measured a 509 perplexity on the election test set as opposed to the 174 we obtained with the baseline language model. This clearly shows that the transcripts of the speeches differs quite significantly in style from the political news content. Computing a perplexity minimizing mixing weight, we interpolated the News and baseline LM with a 0.11 weight, marginally improving the perplexity.

In addition, we added all lexical items seen in the News sample but not present in our BN baseline vocabulary. This expanded our vocabulary from 71k to 88k. Pronunciations for the new lexical items were generated by Pronunciation By Analogy [7] which was trained on the base Pronlex derived vocabulary[7]. Although this performed well on important novel lexical items like "superdelegate", it did poorly on some of the names. For example "Barack" was initially "/b/ /ae/ /r/ /ae/ /k/" as opposed to "/b/ /aa/ /r/ /aa/ /k/" and "Putin" was "/p/ /ah/ /t/ /ih/ /n/" as opposed to "/p/ /uw/ /t/ /ih/ /n/". We manually checked and corrected the most frequent items from the test set. The resulting adapted system obtained a 36.4% WER and the OOV rate of 0.5%.

4. INDEXING AND RETRIEVAL

The ASR 1-best output from the transcription system described in 3 contains richer information than regular text alone. For each video, besides having text meta-data such as title and a short description, we have a "spoken document" consisting of the utterances produced by the ASR system. Each utterance has a start and end time as determined by the segmentation and a time-aligned transcript where each word has an associated confidence score. For indexing and retrieval of "spoken documents" we extended the standard Google infrastructure such that it stores the extra information above.

One possible route would be "HTML-ize" the "spoken document" by filling in an HTML template with the title, description, and ASR 1-best output, and then use the Google indexing and retrieval modules as black boxes. The problem with this approach is that we lose control over the text tokenization, which is fixed by the vocabulary of the speech recognizer.

Changing the ASR output to match the text tokenization output by the Google indexing pipeline—also the one expected by the Google ranking algorithms—would mean altering the start/end times and confidence values to match those of the new text tokenization.

Instead, we preferred to build the inverted index using the ASR engine tokenization, and check that our inverted index was not too far from the tokenization expected by the Google retrieval algorithm. To do so we ran a set of queries against a "witness" index built from the "HTML-ized" version of the documents, and compared the two retrieval outputs using a simple distance metric between ranked lists of documents that weighs each element by its reciprocal rank. The average distance over a large set (10k) of queries randomly sampled from our query logs was computed to be less that 0.05, a satisfyingly low value given the distance metric is properly normalized to take values in the interval $[0.0, 1.0]$.

More importantly, this approach allows us to synchronize at the word position level the text-only part of our index with the extra information output by the ASR system: word start/end times and confidences.

The context of each word in the document is also saved (whether it is part of the title, description or ASR output), along with the position information in the document, both used for ranking the relevance of a document to a given query, as described in a precursor of the Google ranking algorithm [8]. In parallel with the index for the text component of the documents, we build a position-based index that stores all the additional information for each word, if available: start/end time and confidence. That way, once we identify query keyword hits in the document we can access the extra-information made available by the ASR system in constant time and make use of it in ranking and/or presenting snippets.

5. USER INTERFACE

The web accessible user interface (UI) initially shows an automatically generated HTML page for a default or URL-configured query. Subsequent queries the user enter into the page are sent asynchronously by the javascript and rendered on the client-side without reloading the page. We have made two variations of our web UI for audio indexing available. One is an iGoogle gadget, where space limitations are of particular importance. The other is our Labs Page, where more space can be devoted for exploring the videos' spoken content.

There is a wide range of use cases for the transcript in the UI. Our application is designed to let the user find relevant videos and to quickly navigate to the relevant sections of a particular video. The significant design challenges here are to

- provide an overview of each search result sufficient to make a choice
- smoothly integrate time-aligned textual information with video playback for more targeted in-video navigation.

Our UI provides a joint solution to these problems. Each of the results lists the key information about the video - thumbnail, title, duration, the date the video was uploaded, and the approximate number of spoken mentions. The Ajax client-side processing nature of our UI allows the user to click on each result in order to see details and start playing a video without navigating away from the results list or reloading the web page.

Displaying all the details provided by the speech recognizer in a useable manner is a UI challenge integrating ASR output with video playback. The transcripts lack punctuation and contain errors that can make them hard to read. However they can dramatically improve the video browsing experience. As a first step, we have borrowed the common search concept of a snippet and applied it to spoken content, making snippets with bolded hit terms available. As with text snippets, they are intended to be skimmed and not carefully read

[7]LDC catalog: LDC97L20

Fig. 3. Google Audio Indexing page on labs.google.com/gaudi. Snippet appears on mouseover of a marker in the playbar.

and a rough transcription suffices. For a more detailed overview, we return the ten best snippets, selected primarily on the number of token hits in the snippet and the confidence scores associated with those hits.

To facilitate rapid video navigation, the snippets should index into the video and provide points to start playback at their occurrences. Our UI implements this through custom javascript controls of the YouTube player letting any element on the page seek to arbitrary points in the video. As shown in figure 3, due to space limitations in the gadget we have integrated the snippets with the player controls, placing markers on the playbar at the start time of each snippet. Clicking on the marker initiates playback two seconds before the start time of the first snippet word. The added offset prevents missing the snippet audio due to timing imprecision and/or seeking irregularities. The yellow markers on the playbar provide a good one-glance overview of the frequency of the query terms in the video. Because the markers correspond to relevant video sections, they provide a high level overview token hits.

The playbar is the logical point of integration of text with video timing, but not ideal due to the combination of a small playbar size and a large density of transcript information. Especially for longer videos, a pixel in the playbar width may correspond to 10 or more seconds, causing some markers to overlap which detracts from the mouseover and browsing experience. We have experimented with merging the overlapping markers, but the solution was clumsy since the underlying video segments are actually temporally distinct. Therefore, we show only one of the overlapping snippets, and recalculate overlaps whenever the player is resized. On our labs page, we have utilized the extra space to give immediate access to all the snippets as shown at the bottom of figure 3.

The second feature that audio indexing provides is search within the selected video, especially useful for longer videos, where localization of information is time-consuming. This is useful both to people interested in a particular topic covered in the video, or for bloggers and reporters, searching for a particular quote.

Finally, we provide a feature to link to a video with a particular search query filled in. We do not provide links to the search page as a sharing options, because the underlying corpus is changing and the link may not return what the person sharing it expected. We are planning to expand the linking feature to link to particular video segments as well.

Our UI lets the user easily navigate among the search results and uses transcripts with time alignments to allow for easy and fast in-video navigation. However, the best way to integrate transcripts with the video browsing experience is still an open question. In particular, it is important to assess the impact ASR can make on video accessibility, to provide more readable transcripts and greater precision in navigating to a particular video segment.

6. CONCLUSIONS

This paper describes the design and development of a scalable system underlying the election video search application that allows end users to search and navigate through the YouTube video material surrounding the 2008 US presidential election. It makes an inventory of videos, maintained by the election candidates, more accessible by use of automatic speech recognition, information retrieval and user interface technologies. Although these technologies are imperfect, the system provides a utility to the end user in that it allows the user to do content-based searches and navigation which, particularly for the dense long-form content, makes the information more readily available. The choice of consistently building the system on Google infrastructure allows us to scale this application to much larger corpora in the future.

7. REFERENCES

[1] J.V. Thong, P.J. Moreno, B. Logan, B. Fidler, K. Maffeu and M. Moores, "SPEECHBOT: An Experimental Speech-Based Search Engine for Multimedia Content in the Web," *Compaq Cambridge Research Lab, Tech. Rep. CRL 2001/06*, 2001.

[2] M. Bacchiani, J. Hirschberg, A. Rosenberg, S. Whittaker, D. Hindle, P. Isenhour, M. Jones, L. Stark and G. Zamchick, "SCANMail: Audio navigation in the voicemail domain," *HLT2001*, 2001.

[3] S. Renals, D. Abberley, D. Kirby and T. Robinson, "Indexing and retrival of broadcast news," *Speech Communication*, 2000.

[4] J. Makhoul, F. Kubala, T. Leek, D. Liu, L. Nguyen, R. Schwartz and A. Srivastava, "Speech and Language technologies for audio indexing and retrieval," *Proc. IEEE*, vol. 88, no. 8, pp. 1338–1353, 2000.

[5] K. Seymore and R. Rosenfeld, "Scalable Backoff Language Models," in *Proc. Int. Conf. on Spoken Language Processing*, Philadelphia, USA, 1996.

[6] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[7] Y. Marchand and R.I. Damper, "A multistrategy approach to improving pronunciation by analogy," *Comput. Linguist.*, vol. 26, no. 2, pp. 195–219, 2000.

[8] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.