Odyssey 2012
The Speaker and Language Recognition Workshop
25-28 June 2012, Singapore

Cisco's Speaker Segmentation and Recognition System

S. Kajarekar, A. Khare, M. Paulik, N. Agrawal, P. Panchapagesan, A. Sankar and S. Gannu

Cisco Systems, Inc, San Jose, CA
{skajarek,apkhare,mapaulik,nehagraw,ppanchap,asankar,sgannu}@cisco.com

Abstract

This paper presents Cisco's speaker segmentation and recognition (SSR) system, which is a part of a commercial product. Cisco SSR uses speaker segmentation and speaker recognition algorithms with a crowd sourcing approach to create speaker metadata. The speaker metadata makes the enterprise videos more accessible and more navigable by itself, and by its combination with other forms of metadata such as keywords. This paper illustrates various functional blocks of SSR and a typical user interface. The paper describes the specific implementations of speaker segmentation and recognition algorithms. The paper also describes the evaluation data and protocols plus results for both speaker segmentation and speaker recognition tasks. Speaker segmentation results show that Cisco SSR performs comparable to the state-of-the-art on RT-03F data. Speaker recognition results show that a small set of user provided labels can be effectively transferred to a continuously expanding set of videos.

1. Introduction

Video is everywhere. People are effortlessly creating, editing, and sharing videos. YouTube® [6] highlights the importance of video in the consumer space. In the enterprise space, solutions like Cisco Show and Share® [5] are providing secure portals for sharing videos between employees.

The success of these portals has created a new problem; making these videos searchable and easier to consume. This is an important problem as the objective of video "sharing" is not really met if a video is not "searchable". Videos are made searchable by indexing them with metadata. This metadata includes title, tags, comments, categories, etc. However, content creaters/uploaders seldom provide exhaustive metadata and the provided metadata tends to be relatively generic [7]. For this reason, a large area of research is focused on automatically generating useful metadata, for example via automatic keyword spotting and automatic topic detection.

Another useful form of metadata is the identity of participants in the video and the locations in the video where the respective participants spoke. We refer to this metadata as a "speaker" metadata. This metadata makes videos searchable by the name of the participants and allows easier navigation within a video, e.g., *find all videos by "John Chambers"*. Note that the speaker metadata can be combined with other metadata such as keywords, e.g., *find all videos where "John Chambers" spoke about "telepresence"*. Speaker metadata is very useful for enterprise videos because they contain structured and loosely structured communications between one or more participants. To the best of our knowledge, no commercial system has automatically generated and used this type of metadata.

This paper describes Cisco's speaker segmentation and recognition (SSR) system. It uses well-known speaker segmentation [2] and recognition [3] algorithms with a novel crowd-sourcing approach for obtaining speaker names. The system automatically creates speaker metadata to improve video indexing and video consumption, and is aimed at the enterprise space. The paper describes the overall architecture of the product, and specific implementation details. It also describes in-house data collection at Cisco. The paper reports the accuracy of speaker segmentation and recognition on this in-house data and also on NIST RT-03F [4] data. It concludes with a short summary section.

2. Cisco SSR

SSR has four functional blocks – speaker segmentation, speaker recognition, model merging and name propagation. Figure 1 shows the connections between these blocks in blue. This section explains SSR by describing the three main workflows.

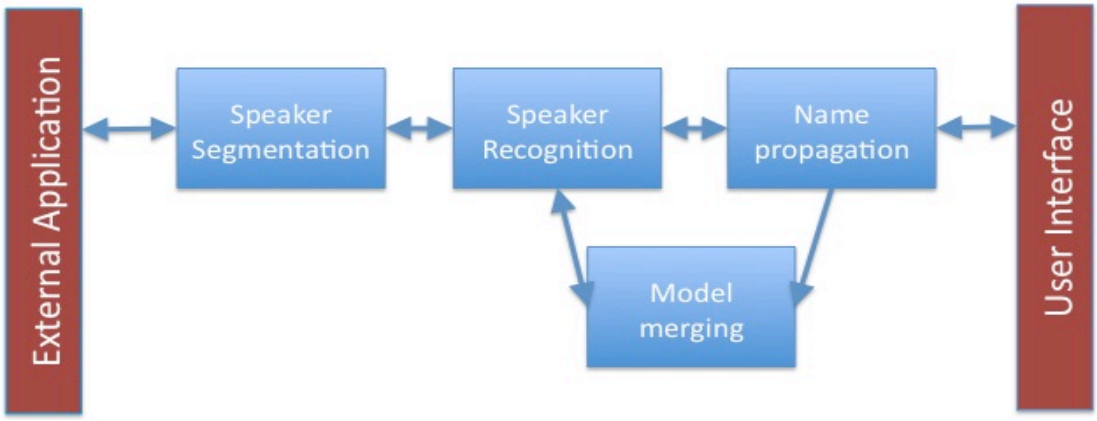

Figure 1 Cisco SSR functional block diagram where SSR blocks are shown in blue.

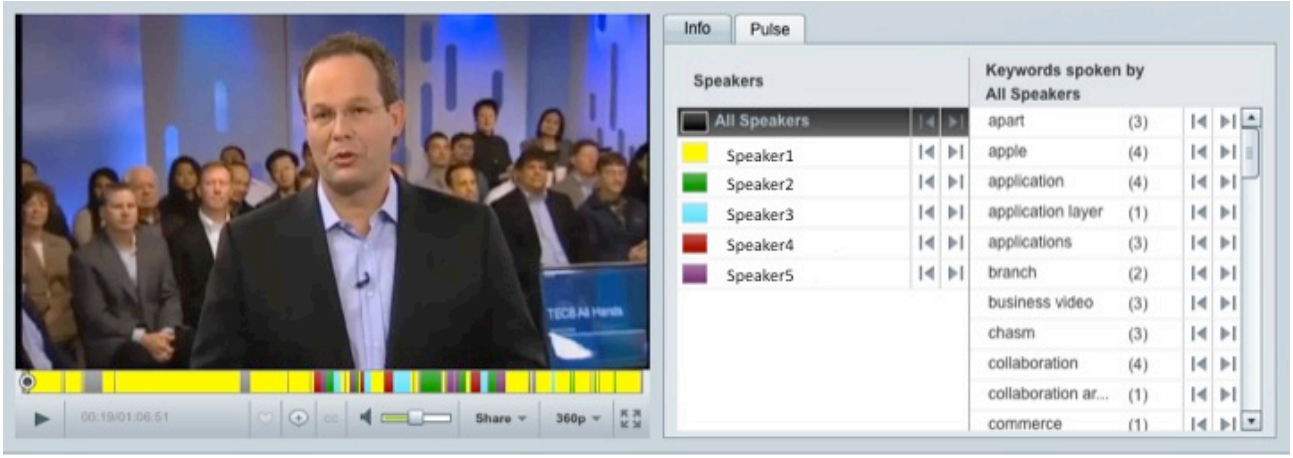

Figure 2 Prototypical UI for displaying SSR information

2.1.1. Speaker segmentation and recognition

External applications send videos to SSR with a unique identifier for that video. An identifier is used to index speaker segmentation information for a video. If the applications re-submit the same video (with the same unique identifier) then SSR simply returns the corresponding information from its database.

Before performing SSR, the video is converted to an audio file. SSR performs feature extraction and speaker segmentation on the audio file. The output consists of speaker homogeneous segments labeled with "SSR speakers". Note that these SSR speakers may or may not refer to unique real-world speakers. The new SSR speakers found in this video are compared to the existing SSR speakers that were created from previously processed videos. A "similar speaker" list is created for each new SSR speaker. Note that the new SSR speakers do not have any name at this stage.

SSR has access to user provided labels for existing SSR speakers (as explained in the following section). It derives a label for each new SSR speaker based on its own list of similar speakers. The updated segmentation, speaker information and label association is stored inside the database. The same information is sent back to the application.

2.1.2. User interface

A user interface (UI) shows the SSR output to an end user. An example UI from Cisco Show and Share® is shown in Figure 2. The video is displayed on the left side. SSR information is displayed in two ways. First, a list of speakers is displayed next to the video on the right side. Each speaker is associated with a unique color. Second, a color-coded map is shown below the video, which shows the parts of the video where a particular speaker is speaking. The names for speakers are 1) provided by a previous end user, or 2) derived from a "similar" speaker, or 3) empty. A user can accept the existing names or change them. If user labels two or more SSR speakers with the same name then they are merged into a single speaker and one color represents all of them.

The UI is designed for fast navigation within a video using speaker and keyword information. When a user selects a speaker from the list, the color-coded map shows only the regions where that speaker is speaking. A user then selects these regions either by using the arrows next to the name or by clicking the color-coded map.

The UI also shows the list of keywords that are estimated from a given video. When a user selects a speaker, the keyword list shows only the keywords spoken by the speaker. A user can select keywords to browse within the video. Thus the UI shows an example of how different type of metadata can be combined with speaker metadata to improve navigation within a video.

Any user can label the speakers in a video. User provided labels are sent to SSR and it performs two updates. First, it updates all the SSR speakers with the corresponding user provided labels. Second, it finds all the SSR speakers that are "similar" to the newly labeled SSR speakers and updates their names if necessary.

2.1.3. Merging SSR Speakers

As mentioned before, unique SSR speakers are generated for each video and they are associated with user provided labels. As a result, many SSR speakers have identical labels. An improved statistical model is created for a label by "combining" all, or a sub-set of, the corresponding SSR speakers and by updating this model over time. A model trained from multiple SSR speaker models is referred to as a composite speaker model. Note that there may be multiple composite models for a single real speaker. However, there are many fewer composite models than SSR speaker models, which are generated per video.[1] Following section describes this process in detail.

SSR periodically creates composite speakers models and updates them. It also updates the "similar" speaker associations with respect to the composite models.

2.2. Implementation Details

2.2.1. Speaker Segmentation

Figure 3 shows the block diagram of the segmentation algorithm, which uses standard steps that are well documented in the literature (give citations). Here are some specific details of the system

[1] Composite models can therefore also improve the speed of the search for "similar" speakers.

1. Gender, bandwidth and speech-silence detections are performed as the first step to create gender and bandwidth homogeneous segments.
2. The change detection (CD), linear clustering (LC) and hierarchical clustering (HC) stages use the same features. Linear clustering merges segments that are adjacent in time only, using the Bayesian Information criterion. Hierarchical clustering uses the same clustering criteria, without any restriction on what segments can be clustered together.
3. LC and HC stages do not merge segments with different genders and different bandwidths.
4. Gaussian mixture models (GMM) are used for cross-likelihood ratio (CLR) clustering and Viterbi resegmentation.
5. Post processing removes low-confidence SSR speakers and corresponding speech segments, and improves the segmentation performance on the remaining video. As the results will show, post processing improves SSR accuracy at the cost of increased missed speech detection error.

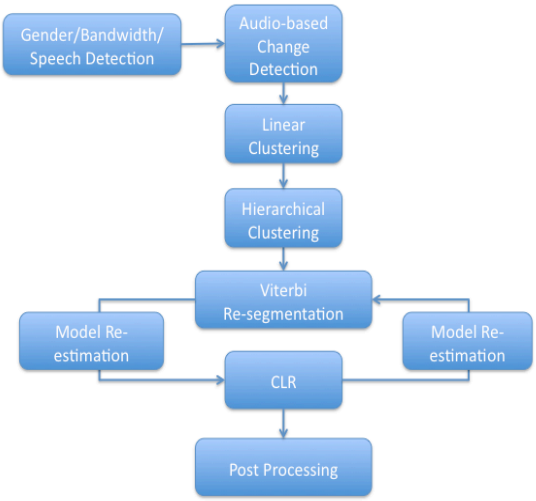

Figure 3 Speaker segmentation algorithm

2.2.2. Speaker Recognition

The first part of speaker recognition is about creating "similar" speakers. Speaker recognition is performed between new and existing SSR speakers. If the distance between two SSR speakers is less than a threshold, then the speakers are marked as 'similar'. Section 4.2 describes the procedure for estimating this threshold. Note that multiple SSR speakers can be "similar" to a new SSR speaker. Also note that SSR speakers are not compared across gender and bandwidth.

The second part of speaker recognition is about name propagation, i.e., obtain names for new SSR speakers. The names come from two sources. The first source is a user. A user can label a SSR speaker when viewing the corresponding video. Only a user – who need not be the one who provided the last label - can override this label. In general, SSR allows unlimited name changes using a crowd-sourcing approach. The second source of names is similar speakers, where SSR assigns a name to a new SSR speaker based on names of the corresponding similar speakers.

The details of the name propagation algorithm are given below -

1. A user watches a given video, identifies a SSR speaker as X, and labels the SSR speaker as X.
2. This name is sent to SSR with the corresponding SSR speaker identifier (ID), represented by *id1*. SSR removes the existing name for *id1*, if any, and associates name X with it.
3. SSR finds all other SSR speaker IDs that are similar to this particular ID *id1*.
4. If a given SSR speaker *id2* is similar to *id1* but already has a user given name then *id2* retains its original user given name.
5. If a given SSR speaker *id2* is similar to *id1* and does not have any name associated with it then it gets labeled with the name X.
6. If a given SSR speaker *id2* is similar to *id1* and it has an SSR suggested name that has not been verified by a user, then we find the SSR speaker *id3* from which *id2* derived its name. If the distance between *id2* and *id1* is less than *id2* and *id3*, then *id2* is associated with the name X, otherwise it retains the older name association.

In summary, the algorithm overrides a user given name for an ID with only another user given name. The algorithm derives a label for SSR speaker *id1* based on the closest SSR speaker *id2,* which has a user given name.

3. Evaluation Protocol

The performance of speaker segmentation and name propagation is evaluated separately. The latter includes the evaluation of both speaker recognition and composite model estimation. In the following subsections, we describe the data and the performance metrics used for each task.

3.1. Speaker Segmentation

Evaluation data includes about 75 hrs of video data from Cisco and 3 speech recordings used in the NIST RT03 fall evaluation [1]. We perform gender bandwidth detection based on the true speaker segments for these videos. These labels are assumed to be the true labels and are used for analysis of the results. The data has mixed-gender and mixed-bandwidth conditions. The actual number of speakers varies from 1 to 27.

3.2. Performance Measures

Speaker segmentation error is measured as a sum of three errors: speech false alarm (FA), speech miss detection (MD) and speaker misclassification. The first two are speech silence segmentation errors. In our experiments, these three errors are calculated in two different ways. First, we use the NIST scoring protocol, which assigns one real speaker to only one SSR speaker. If a real speaker is split into two SSR speakers then the dominant SSR speaker is assigned to the real speaker and the time for the other SSR speaker is considered as an error. This is refer to this as a "NIST" error.

In real-life scenarios, the constraint of mapping one real speaker to only one SSR speaker can be relaxed to some extent. If a real speaker is split into multiple SSR speakers then an end-user just needs to provide the same name for multiple SSR speakers. While it can be frustrating to provide the same label for multiple speakers, it is a much more benign

error than the one, where two different real speakers are merged into a single SSR speaker.

Therefore, the NIST scoring script is modified to allow assignment between multiple SSR speakers and a real speaker. This is the second approach. The dominant real speaker is calculated for each SSR speaker, and a mapping is created between the two. The error calculated with the modified NIST protocol is referred to as an "Internal" error. This error measures cluster purity and it indicates potential gains from improved clustering. It also gives the residual error after a user is allowed to give the same name to different SSR speakers. Note that false alarm speech errors and miss detect speech errors are the same with the "NIST" and "Internal" scoring approaches. Also note that a very low internal error at the cost of very high NIST error by producing too many clusters. This is avoided by measuring the overall error as a combination of the two errors.

Speaker recognition performance is also measured with FA and MD errors. The errors are referred to as speaker FA and speaker MD respectively. The errors are measured at different operating points using different thresholds. Typically a threshold is chosen to minimize a certain cost function, which is a combination of speaker FA and MD with other parameters. In this paper, the results are reported using equal error rate (EER), where the cost function is an average of speaker FA and speaker MD. In other words, the operating point assumes the same cost and priors for the two types of errors.

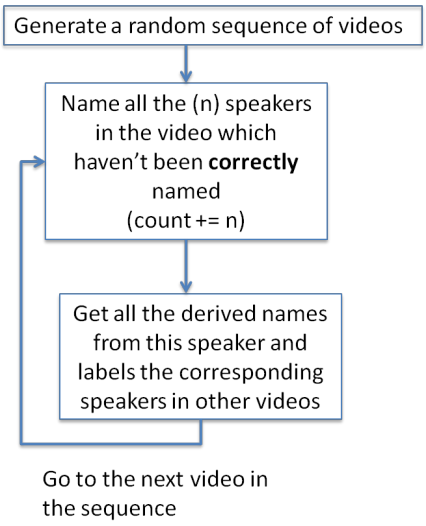

Figure 4 Name propagation evaluation protocol

3.3. Name propagation and model merging

The goal of this evaluation is to measure the accuracy of speaker recognition. The speaker recognition algorithm inside SSR compares SSR speakers from different videos to derive names for new SSR speakers. Thus speaker recognition accuracy directly translates to the efficacy of name propagation. Higher speaker recognition accuracy is important for minimizing user input and improving the usability of speaker metadata.

The evaluation data consists of 78 videos from Cisco. The speakers are mostly executives, many of whom appear in numerous videos. The videos are either corporate announcements or interviews. There are 68 unique speakers across 78 videos. There are 8 speakers (executives) that speak across multiple videos, and 60 unique speakers (interviewers and others) that speak in only one video.

The evaluation protocol assumes that users will upload videos in batches; they will watch videos in a random sequence; and label SSR speakers as needed. Figure 4 shows the evaluation protocol, which is described in detail below.

1. Process all the videos to get the segmentation and similar speaker information. Note that none of the SSR speakers have any names associated with them.
2. Generate a random sequence of the 78 videos from the dataset; this sequence determines the order in which a user might watch and label the speakers in the videos.
3. Take the next video in the list, name all the SSR speakers that haven't been named or have been labeled wrong. Keep a count of how many labels were empty or wrong. Confirm all the correct names.
4. Propagate the speaker names to all the similar SSR speakers in other videos. Note that all the previous videos in this sequence have correct labels for all the SSR speakers. So name propagation does not affect those labels. It only affects SSR speakers in the videos that have not been processed yet.
5. Repeat steps 3, 4 and 5 for all the videos in the random sequence.

For each random sequence of videos, three types of errors are measured.

1. Number of user inputs - The number of user inputs required to label all the videos in a sequence. The goal is to label all of the speakers in the videos with minimal user input.
2. False acceptance error - If the speaker is labeled incorrectly due to incorrect similar speaker association then it is counted as a false acceptance error.
3. False rejection error - If a speaker has been labeled in another video but the name has not propagated to the current video, then it is counted as a false rejection error.

An experiment is performed over multiple random sequences of videos, and mean and the standard deviation of each type of the error is calculated.

The benefit of the composite models is measured with and without name propagation. The results from our "name propagation" evaluation, where we did not merge SSR speakers, are used as a baseline. The setup is modified with an additional step after step 3. In this step, all the speaker models with the same names are used to get the corresponding composite models (if possible), and the speaker similarity is recomputed for all speaker models.

Second, speaker recognition performance is measured without name propagation (and the random sequence). This is similar to a more traditional speaker recognition evaluation setup. The videos are divided into two sets – train and test. True names are obtained for all the SSR speakers in the train set. Speaker recognition is performed with SSR speakers in the test set. The speaker recognition experiment is repeated by switching the train and test sets. The final performance measure is the average performance of these two speaker

recognition experiments. This performance measure is referred to as a baseline.

The benefit of composite models is tested as follows. In the train set, speaker models are merged based on their true names to obtain a smaller set of composite models. Speaker recognition is performed on the speakers in the test set with only the composite models. This experiment is repeated by switching the train and test sets. The results are averaged across the two experiments and are then compared to the baseline.

3.3.1. Algorithm for Model Merging

The algorithm to perform model merging on speakers, which have the same user given name, is described below –

1. Store the sufficient statistics of the GMM representing each SSR speaker in each video.
2. Get all models that have the same *user given name*.
3. Perform a bottom up clustering on these speaker models. At each clustering step, find the two closest models, combine these models by adding the sufficient statistics of the GMM representing each model, and estimate the mean of a single new model. Continue clustering until the closest distance between the speaker models is greater than a predetermined threshold. This results in one or more composite speaker models for a given speaker.
4. Repeat steps 1 through 3 periodically in the system.

Note that "user given name" refers to not only names that a user manually provides for a speaker, but also to names that SSR has suggested for speakers in a given video and that were confirmed by a user.

4. Results

4.1. Speaker segmentation

Table 1 shows results for all the videos (RT03+Cisco) before and after post-processing. Table 2 shows results on RT03 before and after post-processing (as explained in Section 2.2.1). In both cases, we compare performance with the NIST and internal scoring scripts. Note that all these results were produced with the same SSR parameter settings.

The results show the effect of post-processing as described in section 2.2.1. For both RT03 and Cisco videos, post-processing reduces the speaker error at the cost of false rejection. The confidence threshold was selected to ensure that FR error does not exceed 10%.

The results also show the difference between the NIST and internal scoring. These two scorings strategies yield the same FA and FR for speech-silence segmentation. The only difference is in the speaker error. Internal error is always lower than NIST error by design. As mentioned before, the internal error can be interpreted as the error after improved clustering or residual error after the user labels multiple speakers with the same name.

4.2. Name propagation and model merging

Table 3 shows the results obtained for speaker name propagation using the evaluation protocol described in Section 3.3. The results were obtained using 50 different permutations of video sequences. Note that the theoretical minimum number of names required to label all speakers in the videos is equal to the unique number of speakers in the data set, which in this case is 68. The maximum number of names required to label all speakers, assuming perfect speaker segmentation and no name propagation across videos, is equal to 201. This is equal to all of the SSR speakers found in all of the videos. The operating point is chosen to be where the false acceptance error rate is equal to the false rejection error rate. Table shows that the corresponding threshold is between 0.6 and 0.65. The experiments show that it is consistent across two-fold cross validation. Table 3 also shows that name propagation using speaker recognition has brought the average number of user provided labels to 77.4 to 74.94. This is significantly better than the number without speaker recognition. This number is also very close to the minimum number of speaker labels.

Table 1 Segmentation error for all videos (RT03+Cisco)

Stage	Speech FA error (%)	Speech FR error (%)	Speaker error(%)	
			NIST	Internal
Before post processing	3.26	1.44	11.53	3.61
After post processing	1.38	10.92	4.09	2.82

Table 2 Segmentation error for RT03

Stage	Speech FA error (%)	Speech FR error (%)	Speaker error (%)	
			NIST	Internal
Before post processing	2.66	0.19	5.97	4.72
After post processing	1.84	9.79	3.41	3.41

Table 3 Name propagation errors without model merging

Threshold	Average number of user labels required	FA speaker error %	FR speaker error %
0.5	106.26	0.59	18.08
0.55	94.06	1.84	12.35
0.6	83.4	4.63	7.23
0.65	77.94	7.30	4.61
0.7	74.94	13.23	3.09
0.75	72.9	20.42	1.82

The system is evaluated with the same set of video sequences that were used in the model merging evaluation. Table 4 shows the results after merging the models with the threshold set to 0.65.

These results suggest that the merging of models improves the false rejection rate and reduces the number of user provided labels. However the difference is not significant because there is not enough evaluation data. This experiment will be performed with more data in the future.

Table 4 Name propagation performance after model merging

Threshold	Average number of user labels required	FA Speaker error %	FR Speaker error %
0.65	75.88	7.82	3.55

Table 5 shows the speaker recognition results with and without model merging and Figure 5 shows the speaker FA and speaker FR curves as a function of the threshold.

These results show that the merging operation does not improve the recognition accuracy, but it does not degrade it either. The thresholds are very similar across the two setups. Note that a very simple approach for creating composite models and to compute the distance between the models. Improved model merging and distance computation approaches should yield improvements to performance. These results show a potential for significant reduction in speaker recognition time.

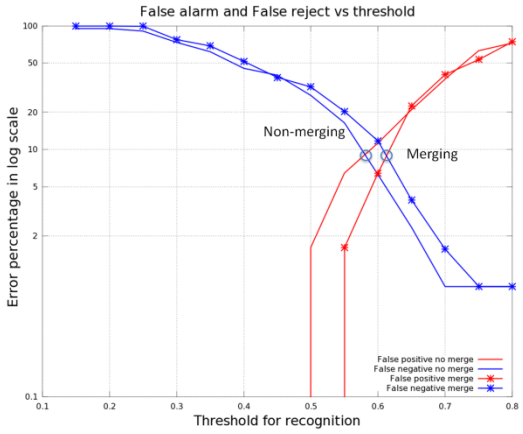

Figure 5 False acceptance and false rejection errors as a function of threshold

5. Summary and Future Work

Usability studies at Cisco have shown that speaker metadata is useful for video search and navigation. This paper described Cisco's speaker segmentation and recognition (SSR) system that generates this metadata. To the best of author's knowledge, it is the first commercial system that 1) automatically discovers speakers from videos and 2) uses crowd sourcing to label these speakers, so that a growing set of videos become searchable using speaker metadata.

The paper described the functional blocks of SSR using different use cases – segmentation and recognition; UI; and model merging. Segmentation evaluation was performed using 75 hours of internal Cisco data and 3 shows from RT03. This data is a mix of different genders and bandwidths. The numbers of speakers vary from 1 to 27. In addition, Cisco internal data was used to measure accuracy of speaker recognition and name propagation. Novel error measures and evaluation protocols were described to measure the performance.

Our segmentation errors on RT03 were comparable to the state-of-the-art [1]. SSR has shown acceptable performance on Cisco internal data and deployments inside Cisco. Name propagation results show that SSR effectively minimizes the labeling effort required from end users.

Table 5 Speaker recognition performance

Experiment	False positive %	False negative %	Total error %
w/o model merging	11.29	6.25	17.54
w/ model merging	6.45	11.71	18.16

6. References

[1] Meignier, S., and Merlin, T., "LIUM SPKDIARIZATION: An Open Source Toolkit For Diarization", *Proceedings of CMU SPUD Workshop, 2010,* http://www.cs.cmu.edu/~sphinx/Sphinx2010/papers/104.unblinded.pdf.

[2] Tritschler, A., Gopinath, R. A., "Improved speaker segmentation and segments clustering using the bayesian information criterion", In proceedings of *EUROSPEECH*, 679-682, 1999.

[3] Bimbot, F., Bonastre, J.-F., Fredouille, C., et al., "A Tutorial on Text-Independent Speaker Verification," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 4, pp. 430-451, 2004

[4] RT-03 fall evaluation plan, http://www.itl.nist.gov/iad/mig/tests/rt/2003-fall/docs/rt03-fall-eval-plan-v9.pdf

[5] Cisco Show and Share, http://www.cisco.com/en/US/prod/video/ps9339/ps6681/show_and_share.html

[6] YouTube website, www.youtube.com

[7] Aradhye, H.; Toderici, G.; Yagnik, J., "Video2Text: Learning to Annotate Video Content," *Data Mining Workshops, 2009. ICDMW '09,* pp.144-151, 6-6 Dec. 2009, http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/35638.pdf