Diarization-based Speaker Retrieval for Broadcast Television Archives

Marijn Huijbregts and David van Leeuwen

Radboud University Nijmegen, department of Language and Speech Technology
{marijn.huijbregts,d.vanleeuwen}@let.ru.nl

Abstract

In this study we extend a query-by-example diarization-based speaker retrieval system to a full speaker retrieval system for broadcast television. The envisioned system is capable of finding all speakers in an archive using their names instead of example speech fragments. Information extracted from a television guide is used to label speaker clusters that most likely correspond to the found names. As part of the labeling process, all speaker clusters are first classified automatically based on their role in the programs they appear in. The role classification accuracy is 64% on our evaluation set. Speaker names can automatically be attributed to a fraction of the speaker clusters with an accuracy of 70%.

Index Terms: Speaker diarization, large scale speaker diarization, speaker retrieval, data mining

1. Introduction

An important source of information for search in audiovisual material is in what is said, but for the semantic interpretation and acceptance of a verbal statement it is not only useful knowing what was said, but also who said it. Speaker information makes it possible to search for recordings of specific speakers, but it can also aid in various automatic annotation tasks such as topic boundary detection, summarization or automatic speech recognition.

Although automatic recognition of speakers is investigated in a number of research fields, there does not yet exists a system that is able to fully automatically annotate large multimedia archives with the true identity of each speaker. The requirements of such a speaker retrieval system are very demanding. The system should be capable of segmenting all speech in the archive such that each segment contains speech from a single speaker. Then it needs to be able to collect all speech that originates from the same speaker into a cluster, not only within each recording but for the entire archive. Finally the system should be able to label each cluster of speaker segments with the identity of the corresponding speaker. On a smaller scale, when aiming at individual recordings instead of entire archives, the segmenting and clustering steps are addressed in the field of speaker diarization. The goal of speaker diarization is to automatically segment an audio recording into speaker homogeneous regions, and link these together ("Who spoke when?"). Unfortunately, it is not possible to do this collection-wide across recordings with most state-of-the-art speaker diarization systems.

In previous work we have developed a system, based on our diarization component, that is able to perform the first steps of a speaker retrieval system: segmenting and clustering all speech in the collection by speaker [1, 2]. With this system it is not possible to search people using their name, but it is possible to perform query-by-example speaker retrieval. In this kind of retrieval, the user selects a small audio(visual) fragment in which someone is talking and the system then retrieves a list of documents in which the same person is speaking.

In this paper we will report on ongoing work to extend our query-by-example speaker retrieval system to a full speaker retrieval system for broadcast television that is able to find speakers by name instead of by using example speech fragments. We will use a television guide to obtain information on who appear in the various television shows and we will use this information to label speaker clusters that most likely correspond to the names found. Although we concentrate on broadcast television, the methods described in this paper can also be used for other types of archives such as meeting archives with minutes that list participants or movie collections if cast information is available.

The television guide that we use to collect names is not exhaustive, causing inconsistencies in the labeling of speaker clusters. We address this problem by first classifying all speaker clusters in three roles: guest, regular participant and host. In the experiments section we will show that by using this classification, name-labeling accuracy can be improved.

The remainder of this paper is organized as follows. First we will discuss existing speaker retrieval work. In section 3 we will then summarize our query-by-example speaker retrieval method and in section 4 we will describe our speaker classification and name-labeling approach. Section 5 contains experimental results and in section 6 we will discuss remaining challenges and future research directions.

2. Related work

Our goal of retrieving all speech of all speakers in a multimedia collection is closely related to speaker tracking [3, 4]. In speaker tracking, the task is to find spoken segments of a particular speaker for which some training material is given. Most speaker tracking systems solve the task by performing speaker segmentation followed by speaker detection [5, 6]. Because only a selection of a-priori known people are tracked, labeling clusters with corresponding names is straightforward. Note that our approach is different from speaker tracking because we try to retrieve all speakers without the use of labeled training data.

Instead of concentrating on an entire collection, it is also possible to use information in individual recordings to label speakers. In [7] it was shown that, by making use of the structure of broadcast news speech and automatic speech recognition, it is possible to automatically obtain the names of speakers in individual news shows. For example, the fact that a reporter is often introduced by the anchor-person can be used to recover the name of the reporter. In another recent study, instead of using speech recognition results, name captions were used to determine the names of speakers. Diarization and face recognition techniques were combined to obtain relevant shots of the target
speaker [8]. Note that these techniques rely on the structure of particular shows. Our approach is intended to work for all types of audiovisual archives where some metadata with global speaker information is available.

3. Query-by-example

In previous work we have developed a query-by-example speaker retrieval system for broadcast television. In this section we will summarize our system. An in-depth description is provided in [1, 2].

![Diagram of the four steps of the data preparation procedure.](image)

Before retrieval is possible, data preparation is done in the four steps depicted in figure 1. Firstly, in the Speech Activity Detection (SAD) step, all non-speech regions are excluded. Secondly, the speech in the recordings is cut-up in evenly sized chunks of 20 minutes. In the third step, speaker diarization is performed on each chunk. The reason for chunking audio is to make it possible to perform diarization on audio streams of indefinite length. Fourth, for each speaker cluster that is found by the diarization system, a speaker recognition model is trained (called speaker enrollment). After enrollment, the data of all speakers is scored against each speaker model, resulting in a list of scores for each speaker model.

Query-by-example retrieval is then carried out as follows. Firstly, it is determined to which speaker cluster (created in step 4) a name belongs, and note that this step is found by the diarization system. The query result is cut-off at 90 or if the score of the cluster is beneath 2.5. In the third step, for each cluster in the query result its corresponding name-vector is retrieved. In the final step, all name-vectors are summed together. The name that was linked to the most clusters is selected. If multiple names score the highest, the cluster is labeled with all alternatives.

4. Cluster labeling approach

If we are able to label the speaker cluster in the above system with corresponding names, it becomes possible to perform speaker retrieval with names as queries instead of example speech fragments. The first step in the labeling process is to determine which names occur in the archive. For our broadcast television retrieval system, we obtain this information using the teletext television guide information that provides a short description of most shows, sometimes including information on who participated in the show.

4.1. Guide information

The television guide provides begin-time and end-time for each show and it gives the title of the show. In the remainder of this paper we will refer to an occurrence of such a show as an episode. We will refer to the series of shows that have the same name in the guide, such as news or the various episodes of a sitcom, as the series.

From Dutch Wikipedia sites, we have collected a list of 2257 Dutch celebrities. We have used this list to search for names in the descriptions of all the episodes in the guide. In this manner we have extracted a list of 320 names of Dutch celebrities. For each episode we have stored the channel, the start- and end-times and the names that occurred in the episode description. The names occurring in the episode are encoded as binary valued elements in a name-vector associated to this episode.

4.2. Name labeling

On average 10–20 speaker clusters occur in the same episode. If a name is found in the description of an episode, it is not very effective to guess which of the speaker clusters should be labeled with that name. Instead, we use the procedure depicted in figure 2.

In the first two steps a list of the most similar speaker clusters, the query result, is generated in the same manner as we did in the query-by-example system. The query result is cut-off at 90 or if the score of the cluster is beneath 2.5. In the third step, for each cluster in the query result its corresponding name-vector is retrieved. In the final step, all name-vectors are summed together. The name that was linked to the most clusters is selected. If multiple names score the highest, the cluster is labeled with all alternatives.

![Diagram of the procedure for name labeling.](image)

Names that occur often in the television guide, e.g., talk-show hosts, have a higher prior of occurring as false positive in the query result lists of other speaker queries. In order to reduce errors caused by this, we require that the most occurring name for a query covers more than 50% of all occurrences of that name in the guide.

In our development experiments we noticed that, even after filtering out these low coverage names, the name labeling system tended to make errors where show hosts are not mentioned in the television guide. If the speaker cluster of a host can not be labeled with the correct name because the name is simply not in the guide, it will be labeled with the name of the guest in the show that was mentioned most often in the guide. Because the
host will also have participated in all shows in which the guest was present, the percentage of name occurrences covered by its query result can be reasonably high.

This type of error is unfortunate, because in a lot of use cases we are especially interested in finding occurrences of people that do not occur on television at regular slots. Confusing these people with show hosts will significantly clutter up the query results. We therefore add another post-processing step to the algorithm: classifying the role of each speaker cluster.

4.3. Role classification

We have developed a method to classify each speaker cluster in the archive as either a host, a regular participant or a guest. The host of a television program is the person that presents the program. In broadcast news this is the anchor person, in a documentary this is typically the voice-over. Note that it is possible that a program has multiple hosts, either presenting together or each presenting their own episodes. The regular participant is someone that is clearly linked to a program, but not as prominently as the host. Examples of regular participants are journalists in broadcast news and sidekicks in talkshows. Finally, guests are people that do not occur often in a particular show. For most retrieval applications, the guest class, particularly celebrities such as politicians or experts on some relevant daily affair, is the most interesting group.

4.3.1. Baseline

Before we discuss our role classification approach that makes use of the query-by-example system, we will introduce two straightforward baseline systems that only use the speaker disarization information and episode timing information. The first baseline approach simply classifies all speaker clusters as guests. This classification will give us the obtainable classification precision when no knowledge is put into the system except that the guest class has the highest prior.

The second baseline system classifies each episode separately. For each episode, all speaker clusters are collected. Using K-means with $K = 3$ on the total speaking time of each cluster, the clusters are divided into three groups. The clusters that have most speaking time are classified as hosts, the clusters with the least speaking time are classified as guests and the remaining clusters are classified as regular participants.

4.3.2. Approach

The role classification approach extends the second baseline method in two ways. Firstly, instead of looking at the clusters in each episode separately, for each cluster we will use the query result of the query-by-example system: the list of the most similar speaker clusters. Secondly, we will perform K-means clustering twice: from the query result lists of each speaker cluster in an episode we will compute the average speaking time in the series that belongs to that particular episode and we will also compute the number of times that the speaker cluster was linked to an episode from the series.

Performing K-means clustering, with $K = 3$, on the average speaking time of a speaker cluster presumably will result in better classification precision than classifying on speaking time of one single episode as we do in the baseline system. Also, information on how often a speaker participated in a series should be valuable for determining the role of a speaker cluster.

As shown in figure 3, taking the intersection of each class, nine classes can be made from the two K-means clustering results. The final task of the role classification system is to assign a role to each of these nine classes. We investigated all 3⁹ possible combinations using the development set. The optimal result was obtained with the configuration shown in figure 3, which has an intuitive interpretation.

4.4. role-based name labeling

We use the role classification results to improve the accuracy of the name labeling as follows. We split the names that we found in the guide in two: names that are found more or less than five times in one series of television episodes. We assume that if someone appears five times or more in a series within the three months, this person is either a regular or a host. Using this assumption we restrict the name labeling.

At first, speaker clusters that are classified as host or regular of a certain series, are only allowed to be labeled with a name that is also considered a regular or host of that particular series on basis of the guide information.

Next, the guests of series are considered. It is possible that clusters classified as regular, also occur as guest in another series. Therefore after the first labeling step, now all speaker clusters that are classified as guest or regular are allowed to be labeled with a name that is considered to be guest on basis of the guide information. For this labeling step we only consider names that occur more than three times in the guide (in any series). On our development set we noticed that it is not accurately possible to label names that occur three times or less in the three months of recordings.

5. Experiments

All experiments described in this section are performed on our three months television broadcast archive, excluding commercials. For the role classification task we have annotated a small development set of 268 speech fragments, each originating from a unique speaker cluster of all three channels on one of the recorded days. For the evaluation set, we annotated another 1000 fragments, randomly selected from the archive.

The development set for the name labeling task consist of the same data as for the role classification (the three channels on a single day). The evaluation set consists of a query for each speaker cluster in all recordings of 42 days (all three channels). We did not manually label each speaker cluster with its name. Instead, we inspected the system results and annotated each result as correct or incorrect. This method is considerably quicker than annotating all clusters by hand, although it needs to be done for each new experiment. Note that it is possible that

\footnote{This is also possible for hosts, but we exclude this group because of the problems noted in section 4.2.}
a speaker cluster is labeled with multiple names (see section 4). In this case, the labeling is considered correct if no more than two names are assigned to the cluster and one of these names is correct.

5.1. Role classification

The results of the role classification experiments are listed in Table 1. For the development set, we have tested the two baselines, the two individual K-mean results (“average speaking time” and “appearance occurrences”) and the proposed system. As can be seen from the baseline experiments, classifying the speaker clusters based on speaking time in single episodes is not effective. The improvement of the diarization baseline is only 1% over the all-guest baseline. The individual K-means results are also not helping much, but combined the total precision is 10% better than the baseline. Looking at the precision of the individual classes, classifying guests is easier than classifying hosts and regulars. The reason for this might be that the distinction between regulars and hosts is subjective in some cases (for example in movies or soaps). The evaluation set results show that we did not over-tune the algorithm on our small development set. The precisions are comparable to the development set results.

<table>
<thead>
<tr>
<th>Experiment</th>
<th>%G</th>
<th>%R</th>
<th>%H</th>
<th>%Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-guest baseline</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>48</td>
</tr>
<tr>
<td>Diarization baseline</td>
<td>60</td>
<td>41</td>
<td>34</td>
<td>49</td>
</tr>
<tr>
<td>Average speaking time</td>
<td>55</td>
<td>20</td>
<td>52</td>
<td>48</td>
</tr>
<tr>
<td>Appearance occurrences</td>
<td>61</td>
<td>32</td>
<td>63</td>
<td>50</td>
</tr>
<tr>
<td>Proposed system (devset)</td>
<td>80</td>
<td>34</td>
<td>48</td>
<td>59</td>
</tr>
<tr>
<td>Proposed system (eval)</td>
<td>75</td>
<td>57</td>
<td>53</td>
<td>64</td>
</tr>
</tbody>
</table>

Table 1: The results of the role classification task.

5.2. Name labeling

First, we applied the proposed algorithm without restricting the labeling to the role classes. In total, 20 clusters were labeled with a name of which 11 were correct. After inspecting the errors, we decided to implement the role-based restricted method. With this method, only 11 clusters were labeled, but now 10 of these labels were correct.

We also applied the role-based restricted method on the evaluation set of 42 days. In total, the system labeled 668 speaker clusters with a regular/host name and 123 speaker clusters with a guest name. The labeling accuracy was 70% for both classes. Note that we have only labeled the query speaker clusters themselves. If we would also label the query result list with the found names, the number of labeled clusters would be much higher.

Inspection of the labeling errors taught us that most labeling errors were due to names missing from either the list of names used to search the guide or from the guide itself. Also role classification errors caused errors in labeling. Typically, errors were made for series where the guests talk significantly more than the host. These guests are sometimes misclassified as regular and labeled with the name of the host.

6. Discussion

In this paper we have reported on our ongoing work on diarization-based speaker retrieval. We have shown that it is possible to use television guide information to label speaker clusters created with a diarization component, with the actual names of the speakers. Because the television guide did not contain exhaustive lists of speakers, we first classified each speaker cluster on its role (guest, regularly occurring or host) in order to improve name-labeling accuracy.

On the evaluation set of 42 days, 70% of the name labels were correct. Unfortunately, because of the thresholds involved, in these 42 days only 791 speaker clusters were labeled with in total 70 unique names. The number of labeled clusters can be improved if not only the queries themselves are labeled, but also the list of speaker clusters that make up the query result.

There was only little speaker information in the guide available. In total, 320 unique names were found in the guide, but only 96 names occurred more than three times which we used as a selection criterion. The algorithm was capable of attributing 70 names out of these 96.

In future research we will apply our name labeling method using more exhaustive textual information sources such as subtitles, automatic speech recognition results, caption information or background episode information from websites. We will also investigate if we can use the obtained speaker information to train better fitting speaker and speech recognition models.

It would also be interesting to investigate if our labeling method could be applied for different types of classification. For example, to find out if we are able to classify speakers on basis of their occupation, using the conversation topics in episodes they attend.

7. Acknowledgments

This paper is based on research carried out for the BATS project within the research program IM-Pact funded by IBBT and ICTRegie.

8. References