Lexical speaker identification for Estonian radio talkshows

Kairit Sirts

Macquarie University

LTG seminar 20.06.2016

Outline

1. Speaker identification

2. Lexical speaker identification for Estonian radio talkshows

Outline

Speaker identification

Speaker identification or recognition is the task of automatically annotating speaker turns in the audio with their person names.

Typical process flow

1. Speech recognition - automatic speech transcription
2. Speaker diarization - clustering the turns of the same speaker
3. Speaker recognition - assigning a name to each speaker cluster

Two main types

- Acoustic speaker recognition
- Lexical speaker identification/recognition

Acoustic speaker recognition

- Most common method
- The set of possible speakers is known in advance
- An acoustic model is pre-trained for each speaker
- Standard technique is to use Gaussian mixture models
- Accurate, when the speakers are in the training set
- Impossible to detect speakers not in the training set

Lexical speaker identification (LSI)

- Decide based on transcriptions, which name refers to which speaker
- No pre-trained speaker models
  - Although there have been attempts to use pre-trained speaker-specific topic models
- The set of possible speaker names may or may not be assumed to be known in advance
- Requires more complex annotated training data than acoustic speaker recognition:
  - Need to know for each name in the transcription whom it refers to
  - current speaker, next speaker, previous speaker, some other speaker, no speaker at all
- Can detect speakers not seen in training data

Speaker name detection

- The speakers of the show are known in advance
  - Reduces to the mapping problem only
- The complete set of possible speaker names in all audio documents is assumed to be known in advance
  - Detect which names are mentioned in the current document and map them to the speakers
- Assume a large set of possible speaker names (Wikipedia)
  - Detect the names that occur in the document
  - Decide which of them refer to speakers in this document
  - Map them to the speakers
- Make no assumptions at all
  - Extract all names
  - Decide which of them refer to speakers
  - Map names to speakers

Previous work

- Lots of work on acoustic speaker recognition
  - GMM-UBM (Gaussian mixture model - universal background model)
  - GSV-SVM - (Gaussian supervectors with support vector machines)
- Quite a bit of recent work on television shows:
  - Detect names from transcriptions
  - Overlay texts that typically show the speaker name are used as well
  - Usually combined with acoustic speaker recognition
- Most work on LSI assume the set of speaker names
- Most previous work on LSI has been based on extracting rules
  - From n-grams, e.g THIS IS [NAME] REPORTING FROM
  - Semantic classification trees, where nodes of the tree are regular expressions over words, e.g <+ LIVE + FROM +> matches a sentence that contains words LIVE and FROM in this order

Outline

System for transcribing Estonian radio talkshows

- http://bark.phon.ioc.ee/tsab/
- Created by Tanel Alumäe from Tallinn University of Technology
- Radio talkshow podcasts are automatically downloaded, transcribed and diarized
- A number of speakers are automatically labeled using acoustic speaker recognition

General process for lexical speaker identification

1. Detect all person names from the transcription
   - Do not assume a list of all names that can appear in a show
2. Which of those names correspond to speakers in the show? Binary classification
3. To which speaker each name likely refers to? Multiclass classification (current, previous or next speaker, some other speaker or not a speaker)
4. Combine the scores of two classifiers and map to speakers

Name detection

- Apply a NER system trained to detect person names only
  - Fortunately Estonian NER detection system actually exists
  - Trained on newspaper texts, thus the accuracy on transcriptions is probably considerably lower
  - Automatic transcription leads to even more errors
- Process with morphological analyser to get lemmas
  - Estonian is an inflective language (2 numbers, 14 cases)
- Cluster the mentions of the same name
  - Simplest way is to use some edit distance based approach on lemmas
  - dr Indrek Rätsep, Indrek, Indrekuga, Indrek Rätsepaga
  - Morphological analyser is not always able to give the correct lemma
  - Sometimes a name is never mentioned in the lemma form

Detecting the speakers of the show

- Binary classification task (logistic regression)
- For each name decide whether it refers to a speaker or not
- Features:
  - name lemma identity
  - word and lemma n-grams ($n = 1 \ldots 5$) in the 5 word window centering on the name, the name itself is replaced with generic $<\mathrm{NAME}>$ token
  - Whether the name appears in the first or last dialogue turn in the show
  - Whether the name occurs in the podcast metadata

Classifying names to speakers

- For each name predict whether it refers to:
  - current speaker
  - next speaker
  - previous speaker
  - some other speaker in the show
  - not a speaker in the show
- Multiclass logistic regression
- Same features as in speaker detection model

Fusing scores together

- Both models make predictions for each name occurrence independently
- Predictions related to the same name must be combined
- Let n_{ij} be the jth occurrence of the name N_i
- $\{N_i\} = \{n_{i1}, \ldots, n_{iJ_i}\}$ - set of all occurrences related to name N_i
- $P(n_{ij})$ - probability that name occurrence n_{ij} refers to a speaker
- $P(N_i)$ - probability that name cluster N_i refers to a speaker

$$P(N_i) = 1 - \prod_{n_{ij} \in \{N_i\}} (1 - P(n_{ij}))$$

Fusing scores together

- Fusing the speaker classification model scores is a bit more complex
- Transform the probabilities over (CURRENT, PREVIOUS, NEXT, OTHER, NO SPEAKER) to probabilities over diarized speaker clusters
  - $P(S_k|n_{ij}) = P(\text{CURRENT}|n_{ij})$ - current speaker is S_k
  - $P(S_k|n_{ij}) = P(\text{PREVIOUS}|n_{ij})$ - previous speaker is S_k
  - $P(S_k|n_{ij}) = P(\text{NEXT}|n_{ij})$ - next speaker is k
  - $P(S_k|n_{ij}) = P(\text{PREVIOUS}|n_{ij}) + P(\text{NEXT}|n_{ij})$ both previous and next speaker are S_k
  - $P(S_k|n_{ij}) = \frac{P(\text{OTHER}|n_{ij})}{|\text{OTHER}|}$ - divide the probability equally among possible "other" speakers
  - Other speakers are those that are neither current, next, nor previous.

Fusing the scores

- Construct $\hat{P}(S_k|N_i)$ via its complement

$$\hat{P}(S_k|N_i) \propto 1 - \prod_{n_{ij} \in \{N_i\}} (1 - P(S_k|n_{ij}))$$

- Finally normalize

$$P(S_k|N_i) = \frac{\hat{P}(S_k|N_i)}{\sum_{k'=1\ldots K} \hat{P}(S_{k'}|N_i)}$$

Combine the scores of both models

- We can view the models as components of a generative model (?)
- Name detection model - prior distribution $P(N)$ over names
- Speaker classification model - likelihood $P(S|N)$ of speakers given a name
- Thus the posterior is $P(N|S) \propto P(S|N)P(N)$ - distribution over names for a speaker

Map names to speakers

- Simplest approach: greedily
- Another possibility: find global optimum using Munkres algorithm
- Introduce a parameter to exclude name-speaker pairs where prior is below a specified threshold
- Introduce a parameter to exclude name-speaker pairs where posterior is below a specified threshold

Training and parameter tuning

- Two logistic regression models - trained independently
- Parameters to tune:
  - Regularization norm (l1/l2)
  - Regularization parameters for both models
  - Prior threshold
  - Posterior threshold

Data

- 50 manually transcribed and diarized talkshows
- I labelled the names semi-automatically with the labels: CURRENT, PREVIOUS, NEXT, OTHER, NO SPEAKER
- Trained and tested using 5-fold cross-validation
- There is a separate test set that hasn't been used so far

Preliminary results

Model	**Precision**	**Recall**
Name detection	86.3 (1.6)	53.5 (1.4)
Full system	ca 80	ca 25

- Need to tune parameters more carefully
- Perform error analysis on false positives
- Goal is to get very high precision by trading off recall

What else could be done?

- Use a list of full names extracted from Wikipedia
- Use sex information to rule out inconsistent speaker-name pairs
- Feature engineering (although morphological features such as suffixes, POS tags or morphological tags have not seemed to be helpful so far)

What else could be done?

- Construct context features using dependency parses (although the accuracy of the Estonian dependency parser is only ca 75%)
- Use word embeddings somehow - construct context representation using an RNN etc
- Reformulate the problem in some other way?
- ???