Fast search in large audiovisual archive: The MALACH story

Jan Švec, Pavel Ircing

Department of Cybernetics
Faculty of Applied Sciences
University of West Bohemia

This research was partially supported by the Ministry of Culture Czech Republic, project No.DF12P01OVV022
The story of the archive

- **1993** – Steven Spielberg releases *Schindler’s List*
- **1994** – establishes *Survivor’s of the Shoah Visual History Foundation* (VHF) in order to record the testimonies of survivors and other witnesses of the Holocaust (and also make them accessible to the wide audience)
- **1994 – 1999** – the VHF “field workers” conducted and recorder about 52,000 interviews in 32 languages – 116,000 hours of video in total
- **2006** – the VHF moved (both physically and administratively) to the University of Southern California and became the *USC Shoah Foundation Institute*
## Number of testimonies in individual languages

<table>
<thead>
<tr>
<th></th>
<th>Language</th>
<th>Testimonies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>English</td>
<td>24,872</td>
</tr>
<tr>
<td>2.</td>
<td>Russian</td>
<td>7,052</td>
</tr>
<tr>
<td>3.</td>
<td>Hebrew</td>
<td>6,126</td>
</tr>
<tr>
<td>4.</td>
<td>French</td>
<td>1,875</td>
</tr>
<tr>
<td>5.</td>
<td>Polish</td>
<td>1,549</td>
</tr>
<tr>
<td>6.</td>
<td>Spanish</td>
<td>1,352</td>
</tr>
<tr>
<td>7.</td>
<td>Dutch</td>
<td>1,077</td>
</tr>
<tr>
<td>8.</td>
<td>Hungarian</td>
<td>1,038</td>
</tr>
<tr>
<td>9.</td>
<td>German</td>
<td>686</td>
</tr>
<tr>
<td>10.</td>
<td>Bulgarian</td>
<td>645</td>
</tr>
<tr>
<td>11.</td>
<td>Slovak</td>
<td>583</td>
</tr>
<tr>
<td>12.</td>
<td>Czech</td>
<td>573</td>
</tr>
<tr>
<td>13.</td>
<td>Portuguese</td>
<td>562</td>
</tr>
<tr>
<td>14.</td>
<td>Yiddish</td>
<td>527</td>
</tr>
<tr>
<td>15.</td>
<td>Italian</td>
<td>433</td>
</tr>
<tr>
<td>16.</td>
<td>Serbian</td>
<td>382</td>
</tr>
<tr>
<td>17.</td>
<td>Croatian</td>
<td>353</td>
</tr>
<tr>
<td>18.</td>
<td>Ukrainian</td>
<td>320</td>
</tr>
<tr>
<td>19.</td>
<td>Greek</td>
<td>301</td>
</tr>
<tr>
<td>20.</td>
<td>Swedish</td>
<td>266</td>
</tr>
</tbody>
</table>
Making the archive accessible

• Manual cataloguing of approx. 4,000 English interviews (10,000 hours, about 8% of the archive) by experts:
  1. Division into topically coherent segments
  2. Three-sentence summary
  3. Keyword indexation (elaborate thesaurus - ~ 3,000 core concepts, ~ 30,000 location-time pairs)
<table>
<thead>
<tr>
<th>Location-Time</th>
<th>Subject</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>Berlin-1939</td>
<td>Employment</td>
<td>Josef Stein</td>
</tr>
<tr>
<td>Berlin-1939</td>
<td>Family life</td>
<td>Gretchen Stein</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Anna Stein</td>
</tr>
<tr>
<td>Dresden-1939</td>
<td>Relocation</td>
<td>Gunter Wendt</td>
</tr>
<tr>
<td></td>
<td>Transportation-rail</td>
<td>Maria</td>
</tr>
<tr>
<td>Dresden-1939</td>
<td>Schooling</td>
<td></td>
</tr>
</tbody>
</table>
Making the archive accessible

- It required 150,000 hours (75 person-years) and cost 8 mil. USD to achieve this, proving a manual cataloguing of the entire archive unfeasible (at least at the given level of granularity)
The MALACH project (2001-2007)

- Multilingual Access to Large Spoken Archives
- Large NSF-sponsored project whose aim was to use ASR and IR techniques to improve access to the archive by acceleration and cost-reduction of the cataloguing process
  - or, alternatively, by circumventing the need for cataloguing
- 7 participating institutions
  - VHF (USC Shoah Foundation Institute)
  - IBM Thomas J. Watson Research Center
  - Johns Hopkins University
  - University of Maryland
  - Charles University
  - University of West Bohemia
  - AITIA International, Inc. (joined during the course of the project)
Role of our lab in the project

• originally, our job supposed to be just preparing the ASR training data for a couple of (mostly Slavic) Central and Eastern European languages

• we gradually became involved in essentially all the project research areas and ended up with:
  – ASR systems for Czech, Russian, Slovak, Polish and Hungarian (the last one was actually build by AITIA International under our close supervision)
  – (non-interactive) IR system for Czech
  – interactive search (spoken term detection – STD) system for Czech
<table>
<thead>
<tr>
<th>Language</th>
<th>MALACH Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>24,872</td>
</tr>
<tr>
<td>Russian</td>
<td>7,052</td>
</tr>
<tr>
<td>Hebrew</td>
<td>6,126</td>
</tr>
<tr>
<td>French</td>
<td>1,875</td>
</tr>
<tr>
<td>Polish</td>
<td>1,549</td>
</tr>
<tr>
<td>Spanish</td>
<td>1,352</td>
</tr>
<tr>
<td>Dutch</td>
<td>1,077</td>
</tr>
<tr>
<td>Hungarian</td>
<td>1,038</td>
</tr>
<tr>
<td>German</td>
<td>686</td>
</tr>
<tr>
<td>Bulgarian</td>
<td>645</td>
</tr>
<tr>
<td>Slovak</td>
<td>583</td>
</tr>
<tr>
<td>Czech</td>
<td>573</td>
</tr>
<tr>
<td>Portuguese</td>
<td>562</td>
</tr>
<tr>
<td>Yiddish</td>
<td>527</td>
</tr>
<tr>
<td>Italian</td>
<td>433</td>
</tr>
<tr>
<td>Serbian</td>
<td>382</td>
</tr>
<tr>
<td>Croatian</td>
<td>353</td>
</tr>
<tr>
<td>Ukrainian</td>
<td>320</td>
</tr>
<tr>
<td>Greek</td>
<td>301</td>
</tr>
<tr>
<td>Swedish</td>
<td>266</td>
</tr>
</tbody>
</table>
ASR challenges

*Acoustic related:*
- elderly speakers
- emotional speech with many disfluences and non-speech events (crying, whimpering, etc.)
- accented speech (mostly in English interviews)

*Lexicon (and language model) related:*
- frequent usage of foreign words (especially concentration camp nomenclature in German)
- abundance of colloquial words (mostly in Czech interviews)
- spontaneous nature of the interviews -> lack of appropriate LM training data
Data preparation

- 400 Czech interviews randomly selected
- 15-minute segment starting 30 minutes from the beginning of the interview was transcribed from each of them
- Word by word transcription + non-speech events
- Another 20 testimonies (10 female, 10 male) were transcribed completely for ASR development and test purposes
Acoustic modeling

• in short – state-of-the-art acoustic parameterization and HMM models

• in detail:
  – 3-state left-to-right triphones with Gaussian mixtures
  – 15 PLP cepstral coefficients + $\Delta + \Delta \Delta$ (i.e., 45-dimensional feature vector), computed at a rate of 100 frames/sec.
  – cepstral mean subtraction applied on per-speaker basis
  – speaker-adaptive training and discriminative training
Lexicon creation - handling colloquial words

- during the ASR data preparation, the annotators were instructed to use the orthographic transcription of colloquial words (i.e., not to “standardize” them artificially)
- such approach is beneficial for **acoustic modeling** (transcription is close to the actual phonetic realization) ...
- ... but sometimes leads to unwanted explosion of different “surface” representations of the same standard word form (e.g. *odejet, odject, odjet, vodjet, vodject, ...*) which harms the robustness of the **language model**
- in order to exploit both the advantage of close orthographic transcription and the benefit of standard word forms, we decided to add a “standardized” column to the pronunciation lexicon
### Fragment of lexicon for the word „*odjet“ [Engl. „*to leave“]"

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ODJET</td>
<td>ODJET</td>
<td>o_d_j_e_t</td>
<td>0.274</td>
</tr>
<tr>
<td>ODJET</td>
<td>ODJET</td>
<td>o_d_j_e_d</td>
<td>0.274</td>
</tr>
<tr>
<td>ODJET</td>
<td>ODEJET</td>
<td>o_d_e_j_e_t</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>ODEJET</td>
<td>o_d_e_j_e_d</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>ODJEC</td>
<td>o_d_j_e_c</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>ODJEC</td>
<td>o_d_j_e_dz</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>ODJECT</td>
<td>o_d_j_e_c_t</td>
<td>0.049</td>
</tr>
<tr>
<td>ODJET</td>
<td>ODJECT</td>
<td>o_d_j_e_dz_d</td>
<td>0.049</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODJET</td>
<td>v_o_d_j_e_t</td>
<td>0.097</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODJET</td>
<td>v_o_d_j_e_d</td>
<td>0.097</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODEJET</td>
<td>v_o_d_e_j_e_t</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODEJET</td>
<td>v_o_d_e_j_e_d</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODJECT</td>
<td>v_o_d_j_e_c_t</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODJECT</td>
<td>v_o_d_j_e_dz_d</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODEJECT</td>
<td>v_o_d_e_j_e_c_t</td>
<td>0.016</td>
</tr>
<tr>
<td>ODJET</td>
<td>VODEJECT</td>
<td>v_o_d_e_j_e_dz_d</td>
<td>0.016</td>
</tr>
</tbody>
</table>
Language modeling

- what training data should be used?
  - large corpora of general texts (newspapers)
  - OR
  - rather small but task-related ("in-domain") data
    (transcripts of the acoustic training data)

- initial experiments showed that LMs trained on the in-domain data
  (although relatively small – approx. 1M tokens) perform much
  better than the ones trained on the large (>30M tokens) newspaper
  corpus

- additional text training data can bring some improvement if they
  are selected carefully
Additional language model data selection

- Czech National Corpus (CNC) – large (over 400M tokens) and very diverse collection of texts
  - it was impractical to use the whole corpus

- we used a simple statistical test to automatically select sentences from the general corpus that were similar to the transcribed testimonies
  - 2 unigram language models were trained – one from the CNC and one from the transcripts (Tr)
  - whenever a sentence $s$ from CNC fulfilled the condition
    \[ P(s/CNC) < t \cdot P(s/Tr) \]
    ($t$ being a variable threshold) we added it to the filtered set (CNC-S)

- a threshold of 0.8 was used to generate a CNC-S containing 16M tokens
Final language model

• interpolation of the trigram LM trained from this CNC-S with the trigram model from the transcripts (Tr:CNS-S interpolation ratio 3:1)

• LM contains:
  – 252k words (unigrams) – 308k pronunciation variants
  – 3.6M bigrams
  – 1.3M trigrams (only trigrams occurring at least twice were included)

• estimated using Kneser-Ney smoothing method

• Word-Error-Rate (WER) - **27.11%**
Non-English ASR Systems

WER [%]

10/01       4/02      10/02      4/03     10/03       4/04     10/04       4/05     10/05       4/06     10/06

Czech     45.91%  84h + LM_{Tr}

Russian   41.15%  + standard

Slovak    38.57%  + adapt

Polish    35.51%  + LM_{Tr+TC}

Hungarian 40.69%  + stand.+LM_{Tr+TC}

45h + LM_{Tr}  57.92%

20h + LM_{Tr}  66.07%

100h + LM_{Tr} 50.82%

100h + LM_{Tr} 45.75%

45h + LM_{Tr} 34.49%
Information retrieval in general

• the most abstract definition of information retrieval system goes along the following lines:
  
  *The goal of the IR system is to satisfy user’s information needs expressed by the query submitted to the system*

• of course the user’s satisfaction is inherently subjective and thus extremely difficult to measure

• we need some means of (preferably automatic) quantitative evaluation of the IR performance to compare different systems

• such evaluation is usually performed on a defined test collection that includes:
  
  – representative set of documents
  – set of pre-defined topics (formalized information needs)
  – judgments of relevance of each document to each topic
Test collection preparation

- relevance assessment is generally extremely labor-intensive
- in our case it’s further complicated by the fact that there are no real document (or topically coherent segments) defined – remember that ASR simply transforms the speech stream into the text stream
- consequently, the traditional task of IR system (“retrieve a relevant document”) was shifted to “find the appropriate start point where the relevant discussion begins”
- the relevance assessors thus did the same thing by hand using a specially designed interactive search system (both the system design and the relevance assessment was done at the Institute of Formal and Applied Linguistics, Charles University)
Example of a defined topic

<top>
<num>1225

title>The liberation of Buchenwald and Dachau

desc>Witness accounts to the liberation of Buchenwald and Dachau concentration camps

narr>The relevant material should include stories by the survivors or liberators that describe the events. Liberation of other camps is not relevant.

</top>
Retrieval experiments

• retrieval from a stream with unknown topic boundaries is extremely challenging
• the task was simplified by dividing the interviews into passages by sliding a window of fixed length across the text stream

resulting segments were treated as “documents”, making the usage of the traditional (document-oriented) IR systems possible
• this approach was used in CLEF 2006 and 2007 CL-SR tracks – albeit succesfully (in terms of winning the competition), it has many drawbacks
Scheme of the MALACH IR system

- Speech signal
- Transcribed speech
- Search system
- Relevant records
... tak sem tady maminkou hned tom čtyřicátém druhém jsme sem přišli tak jsme tady byli ubytovaný v pokoj ...
</ASRTEXT>
</DOC>
Scheme of the system

Speech signal → transcribed speech → Automatic Speech Recognition → Transcript repository → Search system

VHF04106-0047.18
VHF04167-0146.32
VHF05103-0192.98

<top>
<num>1225
<title>The liberation of Buchenwald and Dachau
<desc>Witness accounts to the liberation of Buchenwald and Dachau concentration camps ...

relevant records
Drawbacks of the “document IR” approach

- the “documents” created by sliding a fixed window are not meaningful (topically coherent)
- we have complex structured queries, yet we are not able to exploit the structure properly
  - our team, the Charles university team and I believe everyone else in the CLEF used some type of “bag-of-words” approach that disregards the query structure altogether
- it exploits only the plain text output from the ASR (i.e., the most probable transcription), not the alternative hypotheses
- words missing from the lexicon cannot be found as they do not have a chance to be recognized
- the system is not interactive – it just returns the list of (encoded) segment starting points but does not allow to play it directly
Creating “Google-like” search system

• people usually do not put such complicated queries when they are searching the Internet – they usually just type a few words

• we would like to have a list of retrieved results (starting points in the recordings) that could be replayed right away
Search engine

- Supports fast searching for words and phrases in the archive
- Uses SQL database – optimized for large data and complex queries
- **Indexing task** – ASR output is processed
  - Recognition lattices used instead of 1-best hypothesis
- **Searching task** – words and phrases are searched in the index
- Current index – 553 speakers, 937 hours of recordings
- Both the word-based and phoneme-based search is used
  - word-based search – for in-vocabulary words, supports search of all morphological forms of a word, 252k words
  - phoneme-based search – arbitrary out-of-vocabulary words, ex. geographic locations, names, slang
Word lattice example
Indexing task

- **Word lattices**
  - 3k arcs per minute
  - 252k different units (vocabulary size)
  - In-vocabulary words

- **Phoneme lattices**
  - 21k arcs per minute
  - 44 different units (phoneme alphabet)
  - Out-of-vocabulary words
Word index

- Simpler task
  - 2-stage pruning of word lattice
  - Absolute threshold $\theta_w$ ($\theta_w = 0.05$)
    - arcs with lower posterior prob. are not indexed
  - Discarding the overlapping arcs labeled with the same word

- Result
  - 5-tuples: ($start_t$, $end_t$, $word$, $posterior$, $item_id$)
  - Lattice structure is ignored
  - 1k hours – 12M db records
Phoneme index (1)

- Indexing single phonemes – infeasible
- Conversion of lattice into set of adjacent arcs – set of **triplets**
  - Triplet – trigram of phonemes
- Pruning of the set of triplets – discard triplet if:
  - One or more phonemes in triplet is silence
  - Two adjacent phonemes are identical
  - Posterior prob. of phoneme is lower than $\theta_P$ ($\theta_P = 0.05$)
  - Combined score of triplet is lower than $\theta_C$ ($\theta_C = 0.1$)
  - Within the time window ($\Delta P = 0.03s$) is another triplet with higher combined score
Phoneme index (2)

- Combined score
  - Geometric mean of scores of phonemes in triplet
  - Eliminates least promising paths through the lattice
  - Phoneme with very low score – whole triplet gains a low score

- Result
  - 5-tuple: (start_t, end_t, triplet, score, item_id)
  - Lattice structure is partially ignored
  - 44 phonemes – 63k different triplets
  - 1k hours – 88M db records
Phoneme index
Example

adl/.73     edl/.23     atl/.03     etl/.03
dle/.54     dl#/.23    dle/.64     ler/.52
l#e/.21     lem/.35    #em/.18     ler/.61
Discard triplets containing silence
Phoneme index
Example

Discard triplets with low score of phoneme
or low combined score
Discard low probability variants of triplets inside a time window
Searching task

- Searching of isolated words
  - In-vocabulary – word index
  - Out-of-vocabulary – phoneme index

- Searching of phrases
  - Operators – mandatory words (+), exact form („“)
  - Split phrase into isolated words
  - Analyze hits of isolated words
Searching the word index

- Straightforward
  - Query the SQL database for occurrence of a given word

- Morphology and pronunciation
  - vocabulary – \((word, pronunciation, lemma)\)
  - perform automatic phonetic transcription of searched word
  - look up words with the same pronunciations (Šindler, Schindler)
  - generate set of lemmas for these words
  - map set of lemmas back to a set of word forms (ie. all morphological variants of word in the vocabulary)
Searching the phoneme index (1)

- Automatic phonetic transcription of word
- Decomposition of pronunciation into a sequence of phoneme triplets
- Query the phoneme index for these triplets
- Sort and cluster the occurrences given the time and item_id
- Generate combined score for the cluster (word result)
Searching the phoneme index (2)

- Combined score of a cluster
  \[ score_{\text{comb}} = (1-\lambda) \, score_{\text{ACM}} + \lambda \, score_{\text{hit}} \quad (\lambda = 0.6) \]
  - \( score_{\text{ACM}} \) – arithmetic mean of scores in a cluster
  - \( score_{\text{hit}} \) – ratio between the number of triplets in a cluster and the number of triplets representing the searched word

- Algorithm does not require the presence of all triplets in the phoneme index
Searching the phoneme index
Example

- Searched word: ádler
- Phonetic transcription: a d l e r
- Triplets: adl, dle, ler
- Combined score: \((1-.6) \times .66 + .6 \times 1 = 0.86\)
Sorting and clustering

Sort by item_id

12008

22801

35612

Sort by time offset

Cluster 1

Cluster 2

Cluster 3

\[ \Delta t \]
Searching for multiword phrases

- Split phrase into isolated words
- Perform word- or phoneme- based search
- Sort and cluster the isolated words given *time* and *item_id*
- Discard clusters not containing all mandatory words
- Score the clusters using the arithmetic mean of cluster items
- Show the textual representation of cluster in GUI
## Search engine evaluation

<table>
<thead>
<tr>
<th>Lattice type</th>
<th>In-vocab. words</th>
<th>OOV words</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FOM [%]</td>
<td>EER [%]</td>
</tr>
<tr>
<td>1-best ASR</td>
<td>79.60</td>
<td>20.86</td>
</tr>
<tr>
<td>Word lattice</td>
<td>93.73</td>
<td>19.79</td>
</tr>
<tr>
<td>Phoneme lattice</td>
<td>73.69</td>
<td>45.99</td>
</tr>
</tbody>
</table>

![Graph showing detection rate vs. false alarms]

- Word search
- Phonetic search
- 1best search
System architecture

- Speed, portability, extensibility

- Effective algorithms, rapid prototyping and development

- Voiar library (Voice archive) – Python
  - Python Algorithms + SQL data
  - SQL database – structured data storage
    - MySQL, SQLite
  - Filesystem – unstructured data storage
Work in progress

• Focus on weighted finite state transducers (WFSTs)

• Why?
  • ASR lattices are acyclic WFSTs

• Optimization algorithm already developed & tested
  • Incl. determinization and minimization

• Effective representation of ASR uncertainty
Factor automaton

- Automaton created from the input lattice
- Contains all subsequences (factors) of the lattice
- Special care of:
  - How to encode the lattice id
  - How to encode the factor timing
- Encoding posterior probability is straightforward
Optimum index

• Create a factor automaton from each lattice

• Encode lattice ID and timing

• Make a union of all factor automata
  • Over the set of all lattices

• Optimize the resulting index
  • Make it deterministic and minimal
Search process

• Optimal
  • Complexity depends only on the input length and the number of results

• Word lattices
  • Create query (represented as WFST)
  • Compose with index

• Phoneme lattices
  • Not every phoneme sequence occur in the lattice!
  • Approximate search
Open questions ...

- Phoneme lattice index optimization (size!)
- Index updates (adding/removing of lattices)
  - Keep the index optimal
- Cloud-based implementation
- Other uses of factor automaton
  - Unsupervised lattice clustering
  - Lattice classification (uses SVMs)
  - Search of named entities
    - dates, times, addresses, company names
  - Voice search (search by an example)
Current project - AMALACH

• MALACH follow-up, supported by the Ministry of Culture of the Czech Republic (program NAKI, grant no. DF12P01OVV022)

• joint project of UFAL, Charles University and Department of Cybernetics, UWB

• Goals:
  – improve the Czech search system
  – build an analogous system for English
  – employ machine translation to allow cross-language searching
Applications of the search technology

• Malach Center for Visual History, Prague
  – Czech, pilot application

• Prospective applications in English:
  – USC Shoah Foundation, Los Angeles
  – Jewish Holocaust Centre, Melbourne