Spoken Document Retrieval and Browsing

Ciprian Chelba
OpenFst Library

• C++ template library for constructing, combining, optimizing, and searching weighted finite-states transducers (FSTs)
• Goals: Comprehensive, flexible, efficient and scales well to large problems.
• Applications: speech recognition and synthesis, machine translation, optical character recognition, pattern matching, string processing, machine learning, information extraction and retrieval among others.
• Origins: post-AT&T, merged efforts from Google (Riley, Schalkwyk, Skut) and the NYU Courant Institute (Allauzen, Mohri).
• Documentation and Download: http://www.openfst.org
• Open-source project; released under the Apache license.
Organize all the world’s information

and make it universally accessible and useful
Overview

• Why spoken document retrieval and browsing?
• Short overview of text retrieval
• TREC effort on spoken document retrieval
• Indexing ASR lattices for ad-hoc spoken document retrieval
• Summary and conclusions
• Questions + MIT iCampus lecture search demo
Motivation

• In the past decade there has been a dramatic increase in the availability of on-line audio-visual material…
  – More than 50% percent of IP traffic is video
• …and this trend will only continue as cost of producing audio-visual content continues to drop

Broadcast News  Podcasts  Academic Lectures

• Raw audio-visual material is difficult to search and browse
• Keyword driven Spoken Document Retrieval (SDR):
  – User provides a set of relevant query terms
  – Search engine needs to return relevant spoken documents and provide an easy way to navigate them
Spoken Document Processing

• The goal is to enable users to:
  – Search for spoken documents as easily as they search for text
  – Accurately retrieve relevant spoken documents
  – Efficiently browse through returned hits
  – Quickly find segments of spoken documents they would most like to listen to or watch

• Information (or meta-data) to enable search and retrieval:
  – Transcription of speech
  – Text summary of audio-visual material
  – Other relevant information:
    * speakers, time-aligned outline, etc.
    * slides, other relevant text meta-data: title, author, etc.
    * links pointing to spoken document from the www
    * collaborative filtering (who else watched it?)
When Does Automatic Annotation Make Sense?

- **Scale:** Some repositories are too large to manually annotate
  - Collections of lectures collected over many years (Google, Microsoft)
  - WWW video stores (Apple, Google YouTube, MSN, Yahoo)
  - TV: all “new” English language programming is required by the FCC to be closed captioned
  
- **Cost:** A basic text-transcription of a one hour lecture costs ~$100
  - Amateur podcasters
  - Academic or non-profit organizations

- **Privacy:** Some data needs to remain secure
  - corporate customer service telephone conversations
  - business and personal voice-mails, VoIP chats
Text Retrieval

- **Collection of documents:** \( \mathcal{D} = D_1, \ldots, D_N \)
  - “large” N: 10k-1M documents or more (videos, lectures)
  - “small” N: < 1-10k documents (voice-mails, VoIP chats)

- **Query:** \( \mathcal{Q} = q_1 \ldots q_Q \)
  - Ordered set of words in a large vocabulary \( \mathcal{V} \)
  - Restrict ourselves to keyword search; other query types are clearly possible:
    * Speech/audio queries (match waveforms)
    * Collaborative filtering (people who watched X also watched…)
    * Ontology (hierarchical clustering of documents, supervised or unsupervised)
Text Retrieval: Vector Space Model

- Build a term-document co-occurrence (LARGE) matrix (Baeza-Yates, 99)
  - Rows indexed by word
  - Columns indexed by documents

\[
(t_{ij})_{i=1...V, j=1...D}
\]

- TF (term frequency): frequency of word in document
- IDF (inverse document frequency): if a word appears in all documents equally likely, it isn’t very useful for ranking
- For retrieval/ranking one ranks the documents in decreasing order of the relevance score

\[
S(D_j, Q) = \frac{\sum_{i=1}^{Q} t_{ij}}{\text{norm}(t_j)}
\]
Text Retrieval: TF-IDF Shortcomings

• **Hit-or-Miss:**
  – Only documents containing the query words are returned
  – A query for *Coca Cola* will not return a document that reads:
    * “… its Coke brand is the most treasured asset of the *soft drinks* maker …”*

• **Cannot do phrase search: “Coca Cola”**
  – Needs post processing to filter out documents not matching the phrase

• **Ignores word order and proximity**
  – A query for *Object Oriented Programming*:
    * “… the object oriented paradigm makes *programming* a joy …”*
    * “… TV network *programming* transforms the viewer in an *object* and it is *oriented* towards…”*
Probabilistic Models (Robertson, 1976)

- Assume one has a probability model for generating queries and documents.
  
  - We would like to rank documents according to the point-wise mutual information.

  \[
  MI(D_j, Q) = \log \frac{P(D_j, Q)}{P(D_j) \cdot P(Q)}
  = \log P(Q|D_j) - \log P(Q)_{\text{const}}
  \]

- One can model \(P(Q|D_j)\) using a language model built from each document (Ponte, 1998).

- Takes word order into account:
  - models query N-grams but not more general proximity features.
  - expensive to store.
Ad-Hoc (Early Google) Model (Brin, 1998)

- HIT = an occurrence of a query word in a document
- Store context in which a certain HIT happens (including integer position in document)
  - Title hits are probably more relevant than content hits
  - Hits in the text-metadata accompanying a video may be more relevant than those occurring in the speech reco transcription
- Relevance score for every document uses proximity info
  - weighted linear combination of counts binned by type
    * proximity based types (binned by distance between hits) for multiple word queries
    * context based types (title, anchor text, font)
- Drawbacks:
  - ad-hoc, no principled way of tuning the weights for each type of hit
Text Retrieval: Scaling Up

- Linear scan of document collection is not an option for compiling the ranked list of relevant documents
  - Compiling a short list of relevant documents \textit{may} allow for relevance score calculation on the document side
- Inverted index is critical for scaling up to large collections of documents
  - think index at end of a book as opposed to leafing through it!

All methods are amenable to some form of indexing:
- **TF-IDF/SVD**: compact index, drawbacks mentioned
- **LM-IR**: storing all N-grams in each document is very expensive
  - significantly more storage than the original document collection
- **Early Google**: compact index that maintains word order information and hit context
  - relevance calculation, phrase based matching using only the index
Text Retrieval: Evaluation

- \texttt{trec\_eval} (NIST) package requires reference annotations for documents with \textit{binary relevance judgments} for each query
  - Standard Precision/Recall and Precision@N documents
  - Mean Average Precision (MAP)
  - R-precision (R=number of relevant documents for the query)

\begin{itemize}
  \item \texttt{d1} \quad \texttt{r1} \quad P_1; R_1
  \item \texttt{d1} \quad \texttt{r2} \quad P_2; R_3
  \item \texttt{d1} \quad \texttt{rM} \quad P_n; R_n
  \item \texttt{dN}
\end{itemize}

\begin{itemize}
  \item Ranking on reference side is flat (ignored)
\end{itemize}
Evaluation for Search in Spoken Documents

• In addition to the standard IR evaluation setup one could also use the output on transcription

• Reference list of relevant documents to be the one obtained by running a state-of-the-art text IR system

• How close are we matching the text-side search experience?
  – Assuming that we have transcriptions available

• Drawbacks of using trec_eval in this setup:
  – Precision/Recall, Precision@N, Mean Average Precision (MAP) and R-precision: they all assume binary relevance ranking on the reference side
  – Inadequate for large collections of spoken documents where ranking is very important

• (Fagin et al., 2003) suggest metrics that take ranking into account using Kendall’s tau or Spearman’s footrule
TREC SDR: “A Success Story”

• The Text Retrieval Conference (TREC)
  – Pioneering work in spoken document retrieval (SDR)
  – SDR evaluations from 1997-2000 (TREC-6 to TREC-9)

• TREC-8 evaluation:
  – Focused on broadcast news data
  – 22,000 stories from 500 hours of audio
  – Even fairly high ASR error rates produced document retrieval performance close to human generated transcripts
  – Key contributions:
    * Recognizer expansion using N-best lists
    * Query expansion, and document expansion
  – Conclusion: SDR is “A success story” (Garofolo et al, 2000)

• Why don’t ASR errors hurt performance?
  – Content words are often repeated providing redundancy
  – Semantically related words can offer support (Allan, 2003)
Broadcast News: SDR Best-case Scenario

- Broadcast news SDR is a best-case scenario for ASR:
  - Primarily prepared speech read by professional speakers
  - Spontaneous speech artifacts are largely absent
  - Language usage is similar to written materials
  - New vocabulary can be learned from daily text news articles

State-of-the-art recognizers have word error rates ~10%
* comparable to the closed captioning WER (used as reference)

- TREC queries were fairly long (10 words) and have low out-of-vocabulary (OOV) rate
  - Impact of query OOV rate on retrieval performance is high (Woodland et al., 2000)

- Vast amount of content is closed captioned
Search in Spoken Documents

• TREC-SDR approach:
  – treat both ASR and IR as black-boxes
  – run ASR and then index 1-best output for retrieval
  – evaluate MAP/R-precision against human relevance judgments for a given query set

• Issues with this approach:
  – 1-best WER is usually high when ASR system is not tuned to a given domain
    * 0-15% WER is unrealistic
    * iCampus experiments (lecture material) using a general purpose dictation ASR system show 50% WER!
  – OOV query words at a rate of 5-15% (frequent words are not good search words)
    * average query length is 2 words
    * 1 in 5 queries contains an OOV word
Domain Mismatch Hurts Retrieval Performance

### SI BN system on BN data

<table>
<thead>
<tr>
<th>Percent Total Error</th>
<th>22.3% (7319)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Substitution</td>
<td>15.2% (5005)</td>
</tr>
<tr>
<td>Percent Deletions</td>
<td>5.1% (1675)</td>
</tr>
<tr>
<td>Percent Insertions</td>
<td>1.9% (639)</td>
</tr>
</tbody>
</table>

1:  61 -> a ==> the (1.2%)
2:  61 -> and ==> in
3:  35 -> (hesitation) ==> of
4:  35 -> in ==> and
5:  34 -> (hesitation) ==> that
6:  32 -> the ==> a
7:  24 -> (hesitation) ==> the
8:  21 -> (hesitation) ==> a
9:  17 -> as ==> is
10: 16 -> that ==> the
11: 16 -> the ==> that
12: 14 -> (hesitation) ==> and
13: 12 -> a ==> of
14: 12 -> two ==> to
15: 10 -> it ==> that
16:  9 -> (hesitation) ==> on
17:  9 -> an ==> and
18:  9 -> and ==> the
19:  9 -> that ==> it
20:  9 -> the ==> and

### SI BN system on MIT lecture

**Introduction to Computer Science**

<table>
<thead>
<tr>
<th>Percent Total Error</th>
<th>45.6% (4633)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percent Substitution</td>
<td>27.8% (2823)</td>
</tr>
<tr>
<td>Percent Deletions</td>
<td>13.4% (1364)</td>
</tr>
<tr>
<td>Percent Insertions</td>
<td>4.4% (446)</td>
</tr>
</tbody>
</table>

1:  19 -> lisp ==> list (0.6%)
2:  16 -> square ==> where
3:  14 -> the ==> a
4:  13 -> the ==> to
5:  12 -> ok ==> okay
6:  10 -> a ==> the
7:  10 -> root ==> spirit
8:  10 -> two ==> to
9:   9 -> square ==> this
10:  9 -> x ==> tax
11:  8 -> and ==> in
12:  8 -> guess ==> guest
13:  8 -> to ==> a
14:  7 -> about ==> that
15:  7 -> define ==> find
16:  7 -> is ==> to
17:  7 -> of ==> it
18:  7 -> root ==> is
19:  7 -> root ==> worried
20:  7 -> sum ==> some
Trip to Mars: what clothes should you bring?


“The average recorded temperature on Mars is -63 °C (-81 °F) with a maximum temperature of 20 °C (68 °F) and a minimum of -140 °C (-220 °F).”

A measurement is meaningless without knowledge of the uncertainty

Best case scenario: good estimate for probability distribution $P(T|Mars)$
ASR as Black-Box Technology

A. 1-best word sequence $W$
   - every word is wrong with probability $P=0.4$
   - need to guess it out of $V$ (100k) candidates

B. 1-best word sequence with probability of correct/incorrect attached to each word (confidence)
   - need to guess for only 4/10 words

C. N-best/lattices containing alternate word sequences with probability
   - reduces guess to much less than 100k, and only for the uncertain words

A WSpeech recognizer operating at 40% WER

How much information do we get (in sense)?

a measurement is meaningless without knowledge of the uncertainty
ASR Lattices for Search in Spoken Documents

Lattices contain paths with much lower WER than ASR 1-best:
- dictation ASR engine on iCampus (lecture material) 55% lattice vs. 30% 1-best
- sequence of words is uncertain but may contain more information than the 1-best

Cannot easily evaluate:
- counts of query terms or Ngrams
- proximity of hits
Vector Space Models Using ASR Lattices

• Straightforward extension once we can calculate the sufficient statistics “expected count in document” and “does word happen in document?”
  – Dynamic programming algorithms exist for both

\[
\text{count}(w_i|D_j) = E_{P(W|A)}[1(w_i)]
\]
\[
P(w_i \in D_j) = E_{P(W|A)}[1(w_i \in D_j)]
\]

• One can then easily calculate term-frequencies (TF) and inverse document frequencies (IDF)
• Easily extended to the latent semantic indexing family of algorithms
• (Saraclar, 2004) show improvements using ASR lattices instead of 1-best
## SOFT-HITS for Ad-Hoc SDR

### Graphical Representation

![Graphical Diagram](image)

### Table of Time (

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>0.00</th>
<th>0.50</th>
<th>1.00</th>
<th>1.50</th>
<th>2.00</th>
<th>2.25</th>
<th>2.85</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table Data

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIL</td>
<td>TO 0.5</td>
<td>IT 0.2</td>
<td>Didn’t 0.5</td>
<td>Elaborate 0.7</td>
</tr>
<tr>
<td>IN</td>
<td>0.3</td>
<td>IN 0.2</td>
<td>IT 0.4</td>
<td>Didn’t 0.2</td>
</tr>
<tr>
<td>AN</td>
<td>0.1</td>
<td>A 0.1</td>
<td>Elaborate 0.1</td>
<td>Sil 0.1</td>
</tr>
<tr>
<td>BUT</td>
<td>0.1</td>
<td>BUT 0.1</td>
<td>Didn’t 0.1</td>
<td>Sil 0.1</td>
</tr>
</tbody>
</table>
Soft-Indexing of ASR Lattices

- Lossy encoding of ASR recognition lattices (Chelba, 2005)
- Preserve word order information without indexing N-grams
- **SOFT-HIT:** posterior probability that a word $w$ happens at a position $n$ in the spoken document $A$

$$P(w, n|\text{LAT}(A))$$

- Minor change to text inverted index: store probability along with regular hits
- Can easily evaluate proximity features ("is query word i within three words of query word j?") and phrase hits
- Drawbacks:
  - approximate representation of posterior probability $P(W|A)$
  - unclear how to integrate phone- and word-level hits
Position-Specific Word Posteriors

- Split forward probability based on path length
- Link scores are flattened

\[
\alpha_n[l] = \sum_{\pi: end(\pi) = n, length(\pi) = l} P(\pi)
\]

\[
\alpha_e[l + 1] = \sum_{i=1}^{q} \alpha_{s_i}[l + \delta(l_i, \epsilon)] \cdot P(l_i)
\]

\[
P(n, l|LAT) = \frac{\alpha_n[l] \cdot \beta_n}{norm(LAT)}
\]
Experiments on iCampus Data

• Our own work (Chelba 2005) (Silva et al., 2006)
  – Carried out while at Microsoft Research

• Indexed 170 hrs of iCampus data
  – lapel mic
  – transcriptions available

• dictation AM (wideband), LM (110Kwds vocabulary, newswire text)

• dvd1/L01 - L20 lectures (Intro CS)
  – 1-best WER ~ 55%, Lattice WER ~ 30%, 2.4% OOV rate
  – *.wav files (uncompressed) 2,500MB
  – 3-gram word lattices 322MB
  – soft-hit index (unpruned) 60MB
  – transcription index 2MB
  (20% lat, 3% *wav)
Document Relevance using Soft Hits (Chelba, 2005)

- Query
- N-gram hits, \( N = 1 \ldots \ Q \)
- full document score is a weighted linear combination of N-gram scores
- Weights increase linearly with order \( N \) but other values are likely to be optimal
- Allows use of context (title, abstract, speech) specific weights

\[
S(D, q_i \ldots q_{i+N-1}) = \log \left[ 1 + \sum_{\text{segment}} \sum_{\text{position}} \prod_{k=0}^{N-1} P(w_{k+l}(s) = q_{i+l}|D) \right]
\]

\[
S_{N-gram}(D, Q) = \sum_{i=1}^{Q-N+1} S(D, q_i \ldots q_{i+N-1})
\]

\[
S(D, Q) = \sum_{N} w_N \cdot S_{N-gram}(D, Q)
\]
How well do we bridge the gap between speech and text IR?

**Mean Average Precision**

- **REFERENCE**: Ranking output on transcript using TF-IDF IR engine
- 116 queries: 5.2% OOV word rate, 1.97 words/query
- Removed queries w/ OOV words for now (10/116)

<table>
<thead>
<tr>
<th>Our ranker</th>
<th>transcript</th>
<th>1-best</th>
<th>lattices</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.99</td>
<td>0.53</td>
<td>0.62</td>
</tr>
</tbody>
</table>

(17% over 1-best)
Retrieval Results: Phrase Search

How well do we bridge the gap between speech and text IR?

Mean Average Precision

- **REFERENCE** = Ranking output on transcript using our own engine (to allow phrase search)
- Preserved only 41 quoted queries:
  - "OBJECT ORIENTED" PROGRAMMING
  - "SPEECH RECOGNITION TECHNOLOGY"

<table>
<thead>
<tr>
<th>Our ranker</th>
<th>1-best</th>
<th>lattices</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>0.58</td>
<td>0.73</td>
</tr>
</tbody>
</table>

(26% over 1-best)
Why Would This Work?

<table>
<thead>
<tr>
<th>[30]:</th>
<th>[31]:</th>
<th>[32]:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BALLISTIC = -8.2e-006</td>
<td>MISSILE = -8.2e-006</td>
<td>TREATY = -8.2e-006</td>
</tr>
<tr>
<td>MISSILE = -11.7412</td>
<td>TREATY = -11.7412</td>
<td>AND = -11.7645</td>
</tr>
<tr>
<td>TREATY = -53.1494</td>
<td>AND = -53.1726</td>
<td>COUNCIL = -15.5136</td>
</tr>
<tr>
<td>ANTIBALLISTIC = -64.189</td>
<td>COUNCIL = -56.9218</td>
<td>ON = -48.5217</td>
</tr>
<tr>
<td>AND = -64.9143</td>
<td>SELL = -64.9143</td>
<td>SELL = -53.1726</td>
</tr>
<tr>
<td>COUNCIL = -68.6634</td>
<td>FOR = -68.6634</td>
<td>HIMSELF = -54.1291</td>
</tr>
<tr>
<td>ON = -101.671</td>
<td>FOUR = -78.2904</td>
<td>UNTIL = -55.0891</td>
</tr>
<tr>
<td>HIMSELF = -107.279</td>
<td>SOFT = -84.1746</td>
<td>FOR = -56.9218</td>
</tr>
<tr>
<td>UNTIL = -108.239</td>
<td>FELL = -87.2558</td>
<td>HAS = -58.7475</td>
</tr>
<tr>
<td>HAS = -111.897</td>
<td>SELF = -88.9871</td>
<td>FOUR = -64.7539</td>
</tr>
<tr>
<td>SELL = -129.48</td>
<td>ON = -89.9298</td>
<td>&lt;/s&gt; = -68.6634</td>
</tr>
<tr>
<td>FOR = -133.229</td>
<td>SAW = -91.7152</td>
<td>SOFT = -72.433</td>
</tr>
<tr>
<td>FOUR = -142.856</td>
<td>[...]</td>
<td>FELL = -75.5142</td>
</tr>
</tbody>
</table>

Search for “ANTIBALLISTIC MISSILE TREATY” fails on 1-best but succeeds on PSPL.
Precision/Recall Tuning (runtime)

(Joint Work with Jorge Silva Sanchez, UCLA)

- User can choose Precision vs. Recall trade-off at query run-time
Speech Content or just Text-Meta Data?

(Joint Work with Jorge Silva Sanchez, UCLA)

• **Corpus:**
  – MIT iCampus: 79 Assorted MIT World seminars (89.9 hours)
  – Metadata: title, abstract, speaker bibliography (less than 1% of the transcription)

• Multiple data streams
  – similar to (Oard et al., 2004):
    – **speech**: PSPL word lattices from ASR
    – **metadata**: title, abstract, speaker bibliography (text data)
  – linear interpolation of relevance scores

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Metadata</td>
<td>1</td>
<td>0.056</td>
</tr>
<tr>
<td>Speech</td>
<td>0.319</td>
<td>0.815</td>
</tr>
<tr>
<td>Meta - Speech.</td>
<td>0.323</td>
<td>0.826</td>
</tr>
</tbody>
</table>
Enriching Meta-data

(Joint Work with Jorge Silva Sanchez, UCLA)

- Artificially add text meta-data to each spoken document by sampling from the document manual transcription.
Spoken Document Retrieval: Conclusion

- Tight Integration between ASR and TF-IDF technology holds great promise for general SDR technology
  - Error tolerant approach with respect to ASR output
  - ASR Lattices
  - Better solution to OOV problem is needed
- Better evaluation metrics for the SDR scenario:
  - Take into account the ranking of documents on the reference side
  - Use state of the art retrieval technology to obtain reference ranking
- Integrate other streams of information
  - Links pointing to documents (www)
  - Slides, abstract and other text meta-data relevant to spoken document
  - Collaborative filtering
MIT Lecture Browser [www.galaxy.csail.mit.edu/lectures](http://www.galaxy.csail.mit.edu/lectures)

(Thanks to TJ Hazen, MIT, Spoken Lecture Processing Project)