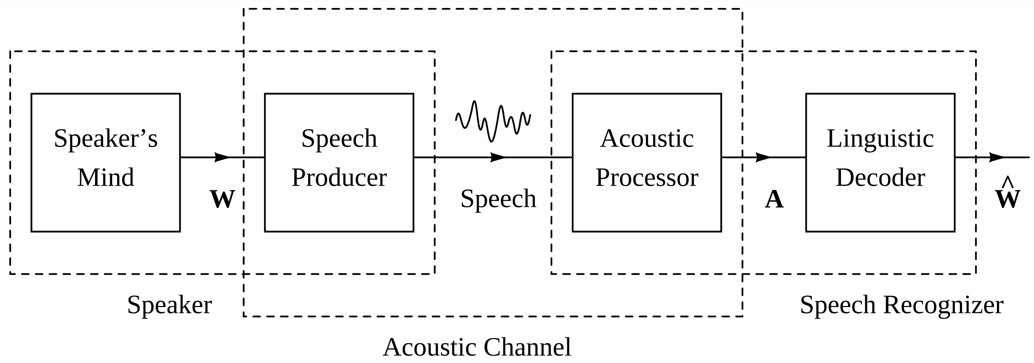
Large Scale Distributed Acoustic Modeling With Back-off N-grams

Google Search by Voice

Ciprian Chelba, Peng Xu, Fernando Pereira, Thomas Richardson

Statistical Modeling in Automatic Speech Recognition



$$\hat{W} = \mathrm{argmax}_{\mathrm{W}} P(W|A) = \mathrm{argmax}_{\mathrm{W}} P(A|W) \cdot P(W)$$

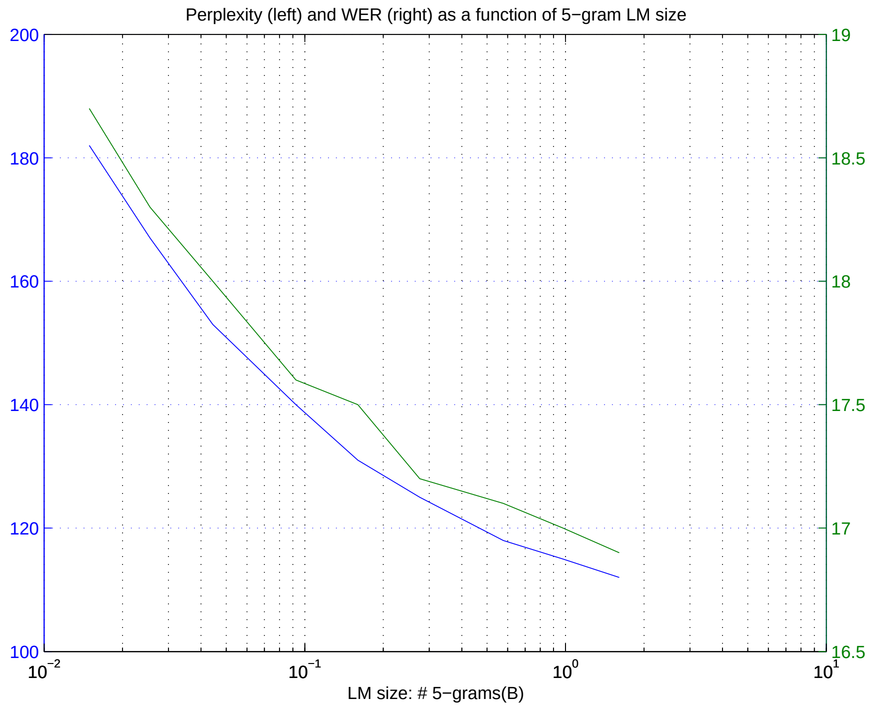
- $P(A|W)$ *acoustic model* (Hidden Markov Model)
- $P(W)$ *language model* (Markov chain)
- *search* for the most likely word string $\hat{W}$
  - due to the large vocabulary size—1M words—an exhaustive search is intractable

Voice Search LM Training Setup

- correct google.com queries, normalized for ASR, e.g. `5th -> fifth`
- vocabulary size: 1M words, OoV rate 0.57% (!), excellent n-gram hit ratios
- training data: 230B words

Order	no. n-grams	pruning	PPL	n-gram hit-ratios
3	15M	entropy	190	47/93/100
3	7.7B	none	132	97/99/100
5	12.7B	1-1-2-2-2	108	77/88/97/99/100

Is a *Bigger* *LM Better? YES!*



- PPL is really well correlated with WER.
- It is critical to let model capacity (number of parameters) grow with the data.

Back to Acoustic Modeling: How Much Model Can We Afford?

- typical amounts of training data for AM in ASR vary from 100 to 1000 hours
- frame rate in most systems is 100 Hz (every 10ms)
- assuming 1000 frames are sufficient for robustly estimating a single Gaussian
- 1000 hours of speech would allow for training about 0.36 million Gaussians (quite close to actual systems!)
- We have 100,000 hours of speech! Where is the 40 million Gaussians AM?

Google

Previous Work

- GMM sizing: [a]
  $\log(\text{num. components}) = \log(\beta) + \alpha \cdot \log(n)$
  typical values: $\alpha = 0.3$, $\beta = 2.2$ or $\alpha = 0.7$, $\beta = 0.1$
- same approach to getting training data as CU-HTK [b]
- they report diminishing returns past 1350 hours, 9k states/300k Gaussians
- we use 87,000 hours and build models up to 1.1M states/40M Gaussians.

---

[a] Kim et al., "Recent advances in broadcast news transcription," in *IEEE Workshop on Automatic Speech Recognition and Understanding, 2003.*

[b] Gales at al., "Progress in the CU-HTK broadcast news transcription system," *IEEE Transactions on Audio, Speech, and Language Processing, 2006.*

Back-off N-gram Acoustic Model (BAM)

$W =$ <S> action </S>, sil ae k sh ih n sil

BAM with $M = 3$ extracts :

```
ih_1 / ae k sh ___ n sil     frames
ih_1 /    k sh ___ n sil     frames
ih_1 /      sh ___ n         frames
```

Back-off strategy:

- back-off at both ends if the M-phone is symmetric
- if not, back-off from the longer end until the M-phone becomes symmetric

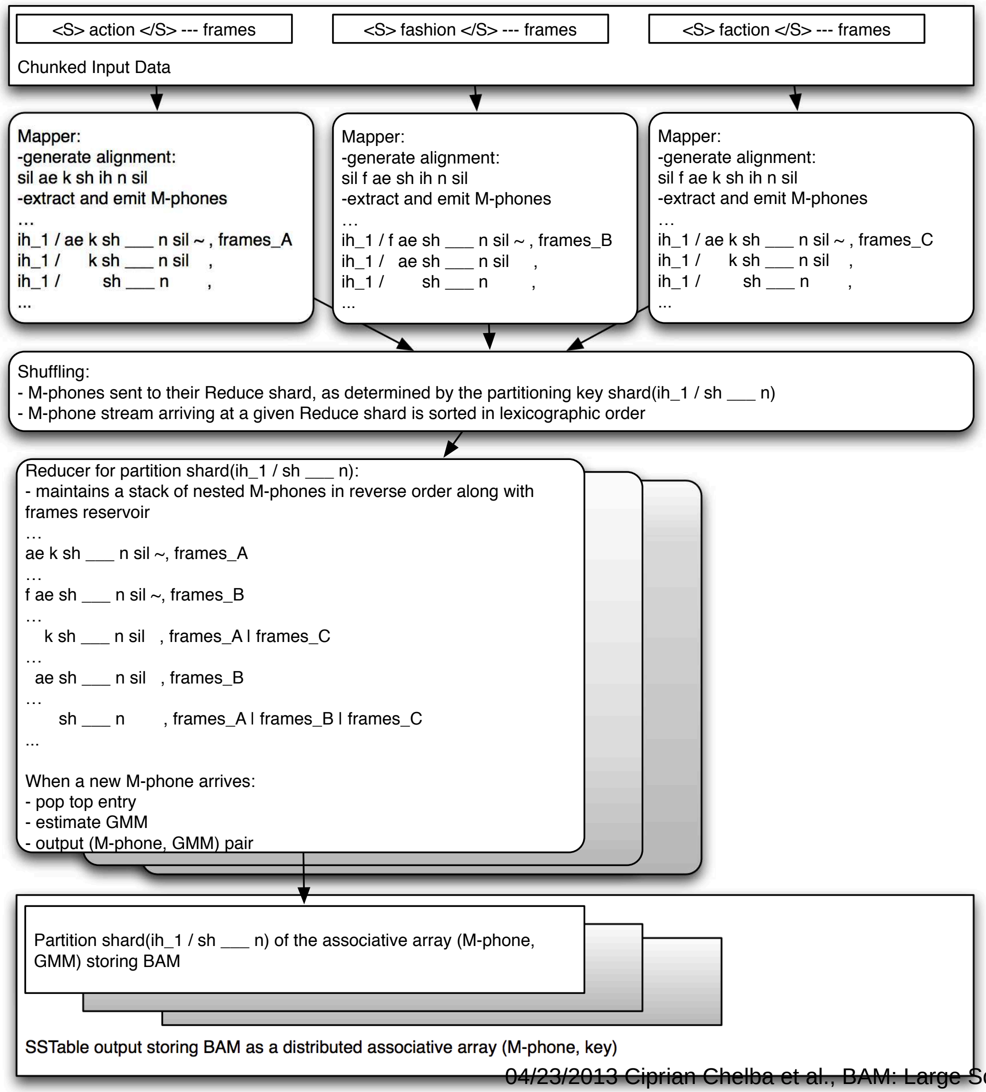
Rich Schwartz et al., Improved Hidden Markov modeling of phonemes for continuous speech recognition, in Proceedings of ICASSP, 1984.

Back-off Acoustic Model Training

- generate context-dependent state-level Viterbi alignment using: $H \circ C \circ L \circ W$ and the first-pass AM
- extract maximal order M-phones along with speech frames, and output (M-phone key, frames) pairs
- compute back-off M-phones and output (M-phone key, empty) pairs
- to avoid sending the frame data M times, we sort the stream of M-phones arriving at Reducer in nesting order
- cashe frames arriving on maximal order M-phones for use with lower order M-phones when they arrive.

Google

MapReduce for BAM Training



N-best Rescoring

- load model into an in-memory key-value serving system (SSTable service) with S servers each holding $1/S$-th of the data
- query SSTable service with batch requests for all M-phones (including back-off) in an N-best list

$$\begin{aligned}
\log P_{AM}(A|W) &= \lambda \cdot \log P_{first\ pass}(A|W) + \\
&\quad (1.0 - \lambda) \cdot \log P_{second\ pass}(A|W) \\
\log P(W, A) &= 1/lmw \cdot \log P_{AM}(A|W) + \\
&\quad \log P_{LM}(W)
\end{aligned}$$

Google

Experimental Setup

- training data
  - baseline ML AM : 1 million manually transcribed Voice Search spoken queries—approx. 1,000 hours of speech
  - filtered logs: 110 million Voice Search spoken queries + 1-best ASR transcript, filtered at 0.8 confidence (approx. 87,000 hours)
- dev/test data: manually transcribed data, each about 27,000 spoken queries (87,000 words)
- $N = 10$-best rescoring:
  - 7% oracle WER on dev set, on 15% WER baseline
  - 80% of the test set has 0%-WER at 10-best

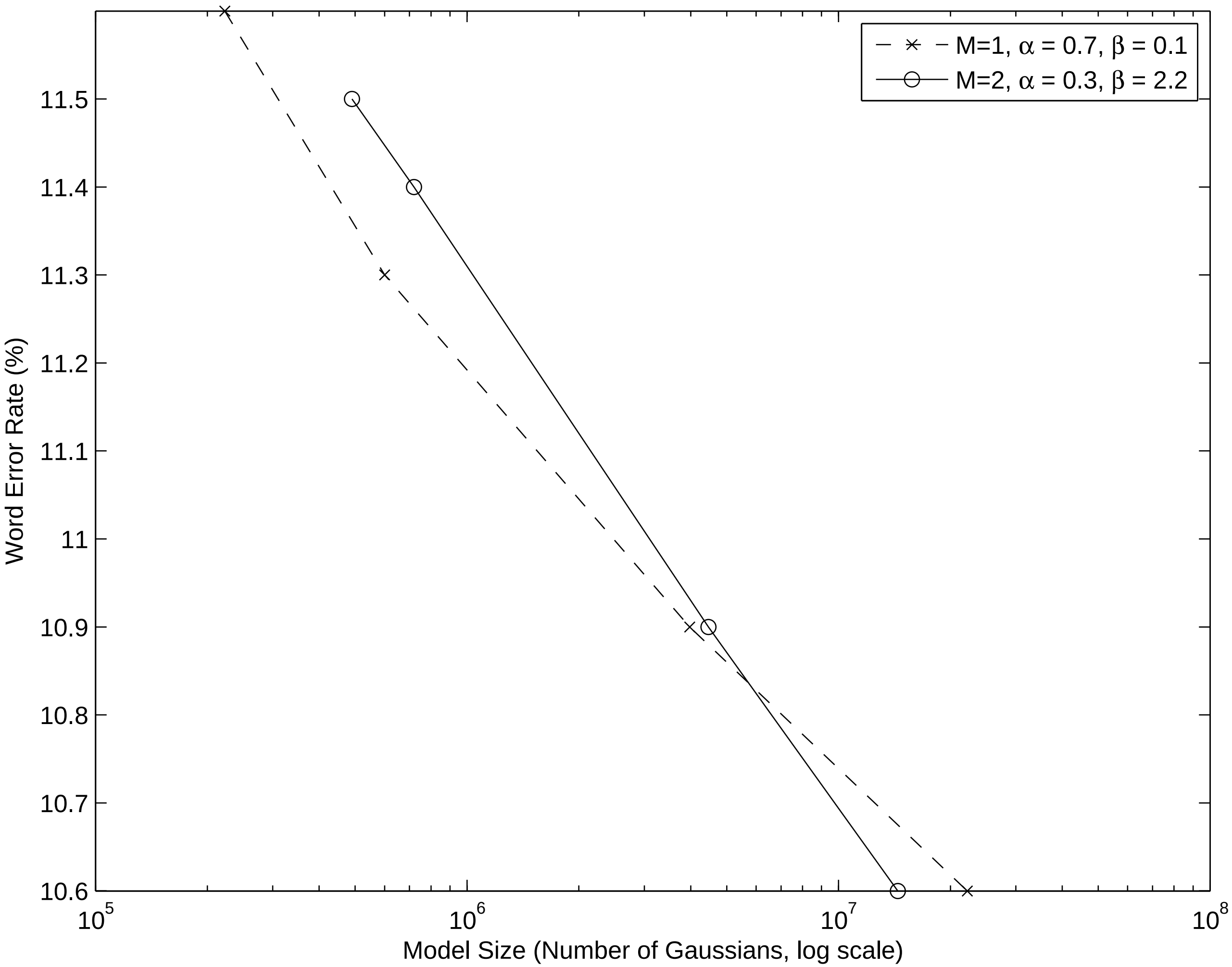
Experimental Results: Maximum Likelihood Baseline

Model	Train (hrs)	Source	WER (%)	No. Gaussians	M
ML,$\lambda = 0.6$	1k	base AM	*11.6*	327k	—
ML,$\lambda = 1.0$	1k	base AM	11.9	327k	—
BAM,$\lambda = 0.8$	1k	base AM	11.5	490k	1
BAM,$\lambda = 0.8$	1k	1% logs	11.3	600k	2
BAM,$\lambda = 0.8$	1k	1% logs	11.4	720k	1
BAM,$\lambda = 0.6$	9k	10% logs	10.9	3,975k	2
BAM,$\lambda = 0.6$	9k	10% logs	10.9	4,465k	1
BAM,$\lambda = 0.6$	87k	100% logs	10.6	22,210k	2
BAM,$\lambda = 0.6$	87k	100% logs	10.6	14,435k	1

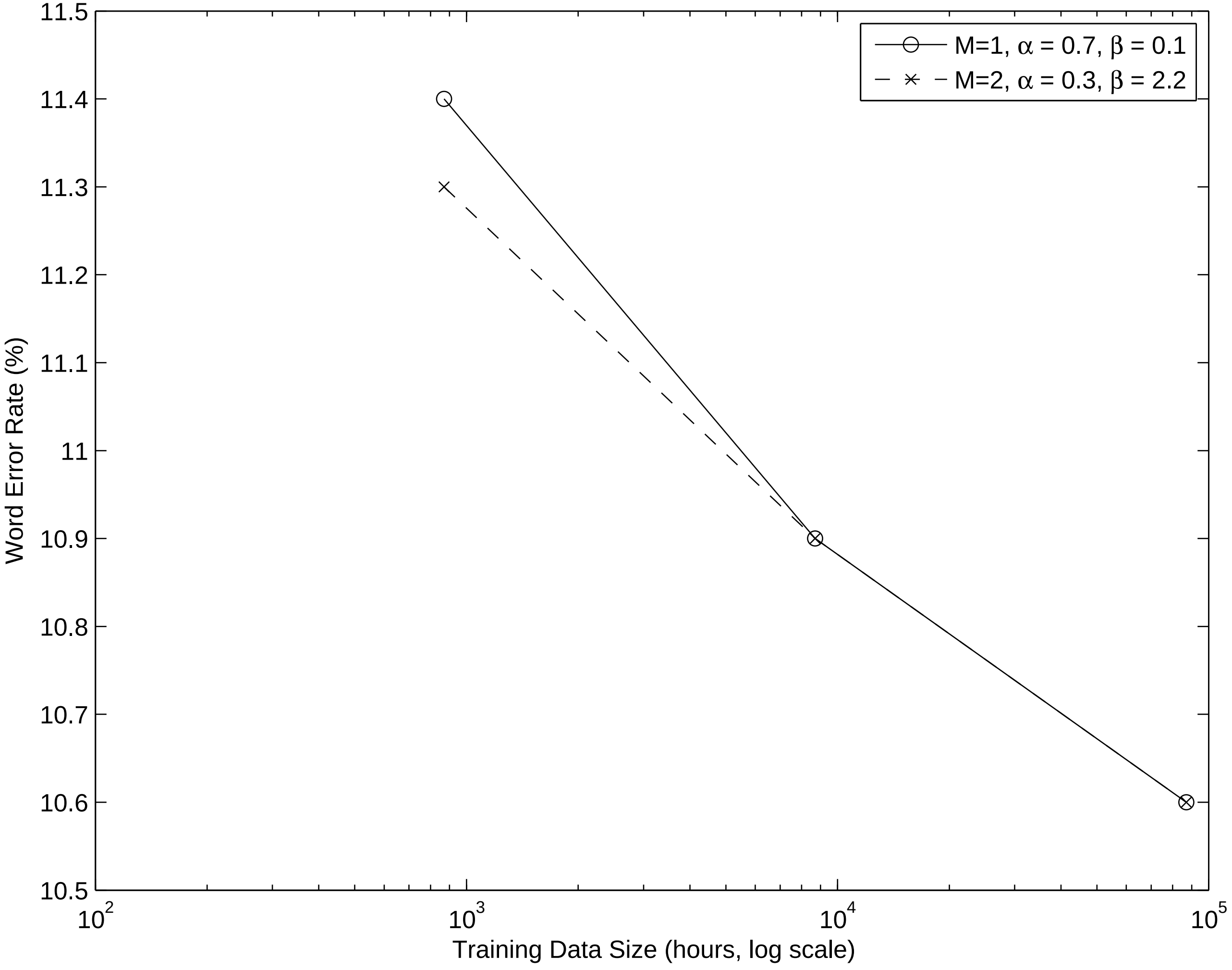
- BAM steadily improves with more data, and model
- phonetic context does not really help beyond triphones
- 1.3% (11% rel) WER reduction on ML baseline

Google

Experimental Results: WER with Model Size



Experimental Results: WER with Data Size



Experimental Results: bMMI Baseline

Model	Train (hrs)	Source	WER (%)	No. Gaussians	M
bMMI, $\lambda = 0.6$	1k	base AM	*9.7*	327k	—
bMMI, $\lambda = 1.0$	1k	base AM	9.8	327k	—
BAM, $\lambda = 0.8$	87k	100% logs	9.2	40,360k	3

- 0.6% (6% rel) WER reduction on tougher 9.8% bMMI baseline

Google

Experimental Results: M-phone Hit Ratios

10-best Hypotheses for Test Data for BAM Using $M = 3$ (7-phones) Trained on the Filtered Logs Data (87 000 hours)

left, right context size	0	1	2	3
0	1.1%	0.1%	0.2%	4.3%
1	0.1%	26.0%	0.9%	3.4%
2	0.7%	0.9%	27.7%	2.2%
3	3.8%	2.9%	2.0%	23.6%

- For large amounts of data, DT clustering of triphone states is not needed

Google

Experimental Results: Validation Setup

- train on the dev set with $N_{\min} = 1$
- test on the subset of the dev set with 0% WER at 10-best; 80% utterances; 1st pass AM: 7.6% WER
- use only BAM AM score, very small LM weight.

Context type	M	WER, (%)
CI phones	1	4.5
CI phones	5	1.5
+ word boundary	1	1.8
+ word boundary	5	0.6

- triphones do not overtrain

BAM: Conclusions and Future Work

- distributed acoustic modeling is promising for improving ASR
- expanding phonetic context is not really productive, whereas more Gaussians do help

Future work:

- bring to the new world of (D)NN-AM
- discriminative training
- wish: steeper learning rate as we add more training data

Parting Thoughts on ASR Core Technology

Current state:

- automatic speech recognition is incredibly complex
- problem is fundamentally unsolved
- data availability and computing have changed significantly: 2-3 orders of magnitude more of each

Challenges and Directions:

- re-visit (simplify!) modeling choices made on corpora of modest size
- multi-linguality built-in from start
- better modeling: feature extraction, acoustic, pronunciation, and language modeling

ASR Success Story: Google Search by Voice

<u>What contributed to success</u>:

- DNN acoustic models
- clearly set user expectation by existing text app
- excellent language model built from query stream
- clean speech:
  - users are motivated to articulate clearly
  - app phones do high quality speech capture
  - speech tranferred error free to ASR server over IP