EXEMPLAR-BASED SPEECH RECOGNITION IN A RESCORING APPROACH

Georg Heigold, Google, USA

Joint work with Patrick Nguyen, Mitch Weintraub, Vincent Vanhoucke

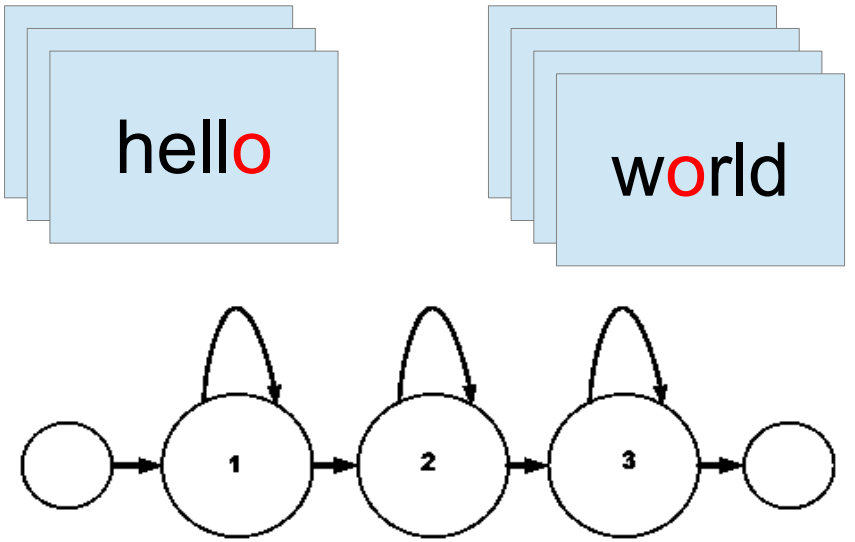
Outline

- Motivation & Objectives
- Tools: Conditional Random Fields, Dynamic Time Warping, Distributed Models, ...
- Scaling it up... & Analysis of Results
- Summary

Motivation

- Todays' speech recognition systems based on hidden Markov models (HMM)
- Potential limitation:

  “conditional frame synchronous independence”



Distribution for pooled observations

- Possible solution: HMMs with richer topology
- Here: *k*NN/non-parametric approach

Challenges

- Exemplar-based approaches require large amounts of data and computing power:
  - Store/access data: distributed memory
  - Process (all) training data: distributed computing
- Coverage ↔ context/efficiency
- Massive but noisy data

Objectives

- Investigate word templates in the domain of massive, noisy data
- Within re-scoring framework based on CRFs
- Motivated by: G. Zweig *et al.*, “Speech Recognition with Segmental Conditional Random Fields: A Summary of the JHU CLSP 2010 Summer Workshop,” in ICASSP 2011, IEEE, 2011.

Data

Voice Search

- Search by voice: “How heavy is a rhinoceros?”

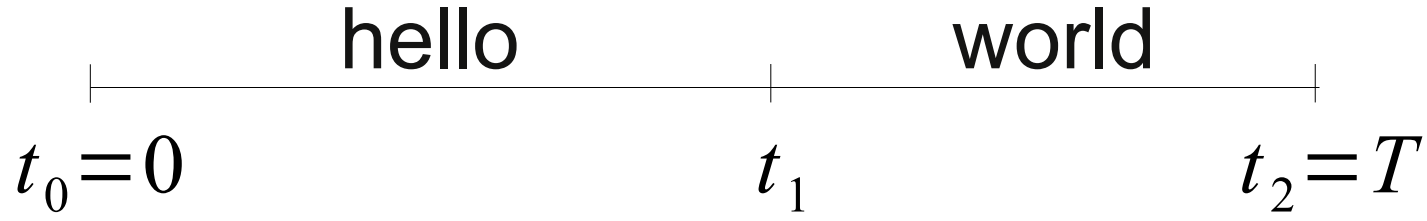
YouTube

- Audio transcriptions of videos
- Transcripts: confidence-filtered captions uploaded by users

	[h]	#Utt.	#Words	Manual transcriptions
Voice Search	3k	3.3M	11.2M	70%
YouTube	4k	4.4M	40M	0%

Hypothesis Space

- Sequence of feature vectors $X = x_1, \ldots, x_T$
- Hypothesis = sequence of words with segmentation



$$\Omega = [w_1, t_0 = 0, t_1], [w_2, t_1, t_2], \ldots, [w_N, t_{N-1}, t_N = T]$$

- Assume word-segmentations from first pass

Model

Segmental Conditional Random Field

$$p(\Omega|X) = \exp\left(\lambda \sum_n f\left([w_{n-1}, t_{n-2}, t_{n-1}]; [w_n, t_{n-1}, t_n], X\right)\right) / Z$$

- Features $f = f_1, f_2, \ldots$ (find good ones)
- Weights $\lambda = \lambda_1, \lambda_2, \ldots$ (estimate)
- Normalization constant Z
- Marginalize over segmentations (only training)

$$p(W|X) = \sum_{\Omega \in W} p(\Omega|X)$$

- G. Zweig & P. Nguyen, “From Flat Direct Models to Segmental CRF Models,” in ICASSP, IEEE, 2010.

Training

Criterion: Conditional Maximum Likelihood

$$F(\lambda) = \log p_\lambda(W|X)$$

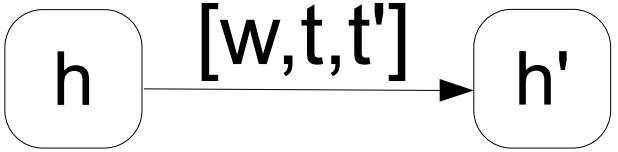
- Including *l1*-regularization $-C_1\|\lambda\|_1$ (sparsity) and *l2*-regularization $-C_2\|\lambda\|_2^2$
- Optimization problem: $max_\lambda F(\lambda)$
- Optimization by L-BFGS or Rprop
- Manual or automatic transcripts used as truth for supervised training

Rescoring

Re-scored word sequence = word sequence associated with $\hat{\Omega} = argmax_{\Omega}\, p\left(\Omega | X\right)$

Transducer-Based Representation

- Hypothesis space limited word lattice from first pass
- Features: $f\left(h;[w,t,t'],x_t^{t'}\right)$



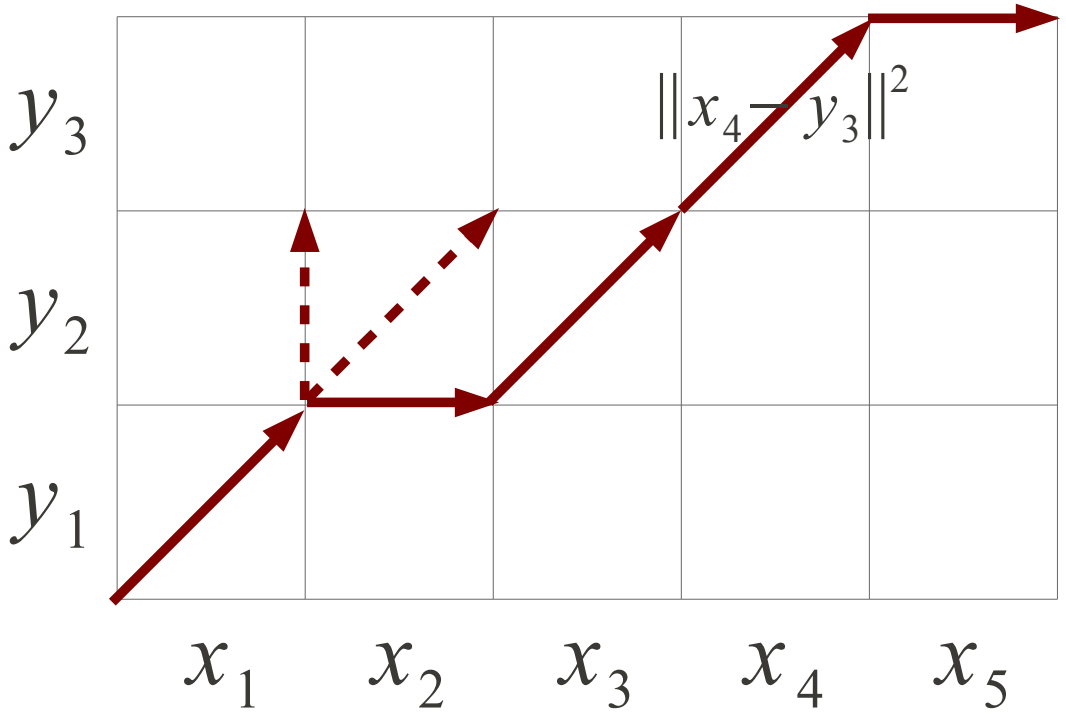
- Standard lattice-/transducer-based training algorithms can be used
- B. Hoffmeister *et al.*, “WFST Enabled Solutions to ASR Problems: Beyond HMM Decoding,” TASLP 2012.

Features: An Example

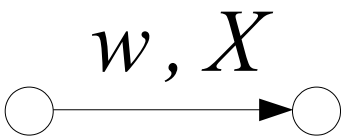
- Acoustic and language model scores from first-pass GMM/HMM (two features / weights)
- Why should we use them?
  - “Guaranteed” baseline performance at no additional cost
  - Backoff for words with little or no data
  - Add complementary but imperfect information without building full, stand-alone system

Dynamic Time Warping (DTW)

- "*k*-nearest neighbors for speech recognition"
- Metric: DTW distance $DTW(X, Y)$
- DTW distance: Euclidean distance between two sequences of vectors $X = x_1, \ldots, x_T, Y = y_1, \ldots, y_S$
- Use dynamic programming
- Literature: Dirk Van Compernolle, *etc.*



“1 feature / word”

- Hypothesis 

, templates Y
- $kNN_v(X)$: *k*-nearest templates to X associated with word v

$$f_v(w, X) = \frac{\delta(v, w)}{|kNN_v(X)|} \sum_{Y \in kNN_v(X)} DTW(X, Y)$$

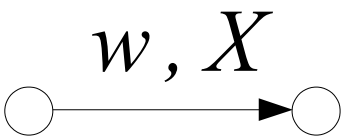
average distance between *X* and *k*-nearest templates *Y*

- One feature and weight per word, one active feature per word hypothesis

Templates

- Templates: instances of feature vector sequences representing a word
- Here: PLPs including HDA (and CMLLR)
- Extract from training data using forced alignment
- Ignore templates not in lattice or silence
- Imperfect because:
  - Incorrect word boundaries: 10-20%
  - Incorrect word labeling: 10-20%
  - Worse for short words like 'a', 'the',...

“1 feature / template”

- Hypothesis 

, templates Y, scaling factor β

$$f_Y(w, X) = \exp(-\beta\, DTW(X, Y))$$

- Reduce complexity by considering word-dependent subsets of templates, *e.g.,* templates assigned to w
- One feature / weight per template
- Non-linearity needed for arbitrary, non-quadratic decision boundaries

“1 feature / template”

- Properties:
  - Doesn't assume correct labeling of templates
  - Learn relevance/complementarity of each template
  - Is sparse representation
- Similar to SVMs with Gaussian kernel, in particular if using margin-based MMI

“1 feature / word” *vs.* “1 feature / template”

Features	WER [%]	
	Voice Search	YouTube
AMLM	14.7	57.0
+ “1 feature / word”	14.3	56.7
+ “1 feature / template”	14.1	55.9

Adding More Context

- (Hopefully) better modeling by relaxing frame independence assumption
- More structured search space → more efficient search
- So far: acoustic unit = context
- Context may be: + preceding word, + left/right phones, + speaker information, *etc.*
- But: number of contexts ↔ coverage

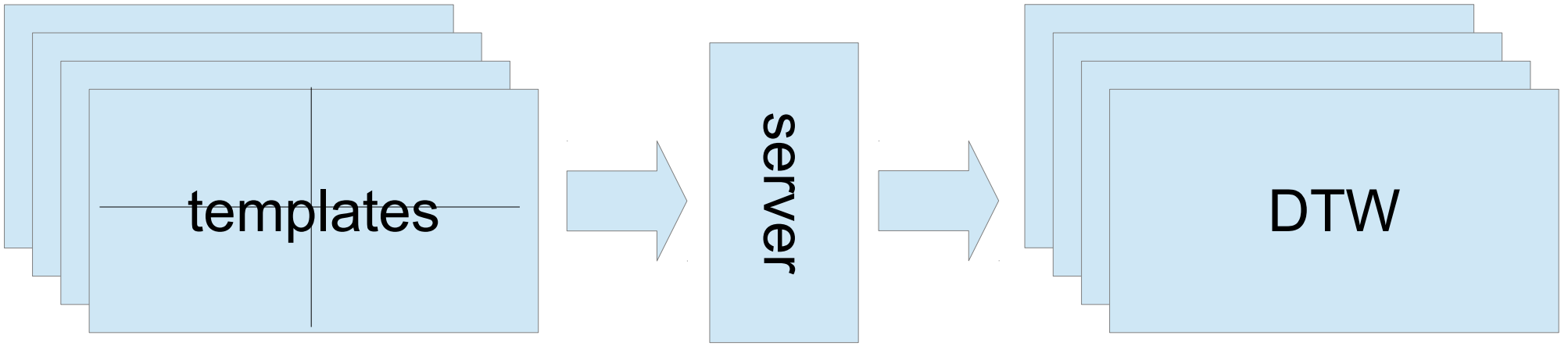
Bigram Word Templates (YouTube)

- More templates don't help and are inefficient
- Short filler words with little context dominate
  'the', 'to', 'and', 'a', 'of', 'that', 'is', 'in', 'it' make up 30% of words
- Consider word template in context of preceding word

Features	Context	WER [%]
AMLM	N/A	57.0
+ "1 feature / word"	unigram	55.9
	bigram	55.0

- Gain from bigram discriminative LM: ~0.2%

Distributed Templates / DTW



- T. Brants *et al.*, “Large Language Models in Machine Translation.”

Scalability

	#Templates [M]	Audio [h]	Memory [GB]
Phone	0.5	30	1
Triphone	25	1,500	45
Word	10	1,000	30
Word / bigram	20	2,000	60
Debugging	20	2,000	500

- Computation time and WER decrease from top to bottom

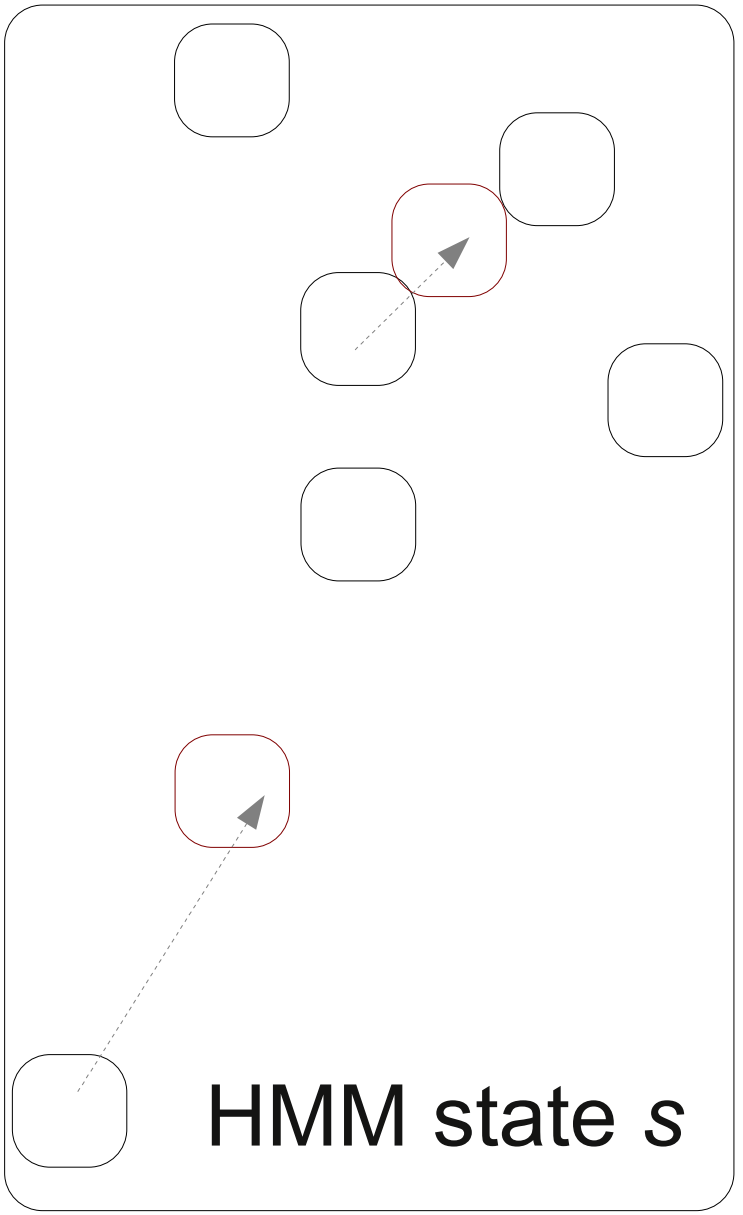
Sparsity

- Impose sparsity by *l1*-regularization (*cf.* template selection)
- Active word templates similar to support vectors in SVMs
- Inactive templates don't need to be processed in decoding

Active templates	Standalone	With AMLM
Voice Search	>90%	<1%
YouTube	>90%	1%

Data Sharpening

- Standard method for outlier detection, smoothing
- Replace original vector *x* aligned with some HMM state by average over k-nearest feature vectors aligned to same HMM state
- But: breaks long-span acoustic context if on frame-level



Data Sharpening (YouTube)

Setup	WER [%] Data sharpening	
	No	Yes
*k*NN, with oracle[1]	26.1	20.4
*k*NN, all[2]	62.4	59.5
AMLM + word templates[3]	56.4	55.9
AMLM + bigram word templates[3]	56.3	55.0

[1] Classification limited to reference word with hypothesis in lattice

[2] Ditto but including all reference words

[3] Re-scoring on top of first-pass

DTW *vs.* HMM Scores

- Replace DTW by HMM scores for check
- Voice Search, triphone templates

	AMLM	+ HMM scores	+ DTW scores
WER [%]	14.7	14.2	14.0

- Similar results in: G. Heigold *et al.*, “A flat direct model for speech recognition,” ICASSP 2009.

Summary

- Experiments for large-scale, exemplar-based speech recognition

Up to 20 M word templates = 2,000 h waveforms = 60 GB data

- Additional context helps, data sharpening also helps...
- Only small fraction (say, 1%) of all templates needed → efficient decoding
- Modest gains: hard but realistic data conditions? unsupervised training? estimation?