Low-Resource Speech Recognition and Keyword-Spotting

M.J.F. Gales, K.M. Knill and A. Ragni

15 September 2017

Low Resource Speech Processing

- Low-resource can refer to various elements:
  - available acoustic model training data
  - available audio transcriptions
  - available lexicon (phonetic lexicon)
  - available language model training data
  - available language processing resources (parsers/PoS tagger)

- Highlighted described in context of the Babel Programme
  - ran from March 2012 to November 2016
  - see web-page for CUED references
    `http://mi.eng.cam.ac.uk/~mjfg/BABEL/index.html`

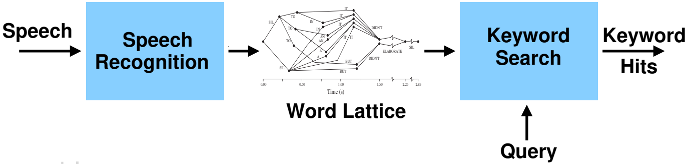
IARPA Babel Program

"The Babel Program will develop agile and robust speech recognition technology that can be rapidly applied to any human language in order to provide effective search capability for analysts to efficiently process massive amounts of real-world recorded speech."

Babel Program BAA

Task: Key Word (Phrase) Spotting

- Specified task is KWS - query terms can be words or phrases

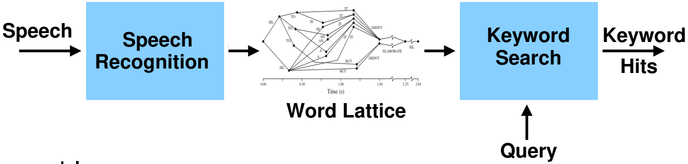


- Key problems are:
  - ASR systems with very limited training data available
  - ASR systems for highly diverse languages
  - KWS systems with high out-of-vocabulary query terms
  - KWS for low accuracy ASR systems

This talk focuses on ASR

Task: Key Word (Phrase) Spotting

- Specified task is KWS - query terms can be words or phrases



- Key problems are:
  - ASR systems with very limited training data available
  - ASR systems for highly diverse languages
  - KWS systems with high out-of-vocabulary query terms
  - KWS for low accuracy ASR systems

This talk focuses on ASR

IARPA Babel Program Specifications

- Language Packs
  - Conversational/scripted telephone data (plus other channels)
  - Full: 60-80 hours transcribed speech
  - Limited: 10 hours transcribed speech
  - Very Limited: 3 hours transcribed speech
  - additional untranscribed audio data available
  - 10 hour Development and Evaluation sets
  - Lexicon covering training vocabulary
  - X-SAMPA phone set
- Increasing number of development languages: 4/5/6/7
  - total: 25 languages (inc. surprise languages, Pashto repeated)
- Surprise Language evaluation
  - decreasing development time - final phase 1 week
  - 80 hours of data to transcribe/KWS - 1 week

IARPA Babel Program Primary Evaluations

- Base Period (BP): > 0.3 MTWV
  - Full Language Pack (FLP), 60-80 hours of transcribed data
- Option Period 1 (OP1): > 0.3 MTWV
  - Limited Language Pack (LLP), 10 hours of transcribed data
- Option Period 2 (OP2): > 0.3 MTWV
  - Very Limited Language Pack (VLLP), 3 hours transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - multi-language (ML) data allowed from BP and OP1
- Option Period 3 (OP3): > 0.6 MTWV, $< 50\%$ WER
  - Full Language Pack (FLP), 40-60 hours of transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - ML data allowed from BP/OP1/OP2/OP3+non-Babel

IARPA Babel Program Primary Evaluations

- Base Period (BP): > 0.3 MTWV
  - Full Language Pack (FLP), 60-80 hours of transcribed data
- Option Period 1 (OP1): > 0.3 MTWV
  - Limited Language Pack (LLP), 10 hours of transcribed data
- Option Period 2 (OP2): > 0.3 MTWV
  - Very Limited Language Pack (VLLP), 3 hours transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - multi-language (ML) data allowed from BP and OP1
- Option Period 3 (OP3): > 0.6 MTWV, $< 50\%$ WER
  - Full Language Pack (FLP), 40-60 hours of transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - ML data allowed from BP/OP1/OP2/OP3+non-Babel

IARPA Babel Program Primary Evaluations

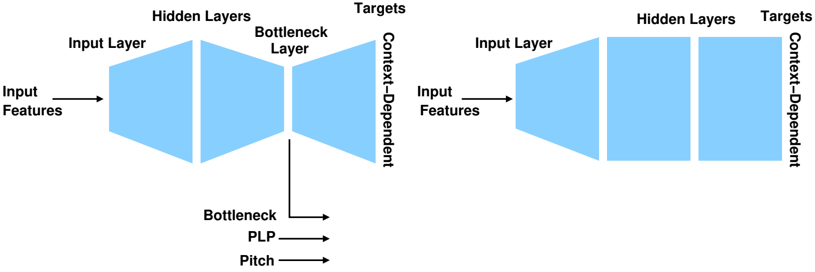
- Base Period (BP): > 0.3 MTWV
  - Full Language Pack (FLP), 60-80 hours of transcribed data
- Option Period 1 (OP1): > 0.3 MTWV
  - Limited Language Pack (LLP), 10 hours of transcribed data
- Option Period 2 (OP2): > 0.3 MTWV
  - Very Limited Language Pack (VLLP), 3 hours transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - multi-language (ML) data allowed from BP and OP1
- Option Period 3 (OP3): > 0.6 MTWV, $< 50\%$ WER
  - Full Language Pack (FLP), 40-60 hours of transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - ML data allowed from BP/OP1/OP2/OP3+non-Babel

IARPA Babel Program Primary Evaluations

- Base Period (BP): > 0.3 MTWV
  - Full Language Pack (FLP), 60-80 hours of transcribed data
- Option Period 1 (OP1): > 0.3 MTWV
  - Limited Language Pack (LLP), 10 hours of transcribed data
- Option Period 2 (OP2): > 0.3 MTWV
  - Very Limited Language Pack (VLLP), 3 hours transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - multi-language (ML) data allowed from BP and OP1
- Option Period 3 (OP3): > 0.6 MTWV, $< 50\%$ WER
  - Full Language Pack (FLP), 40-60 hours of transcribed data
  - no phonetic lexicon
  - language model harvested from the web (web-data)
  - ML data allowed from BP/OP1/OP2/OP3+non-Babel

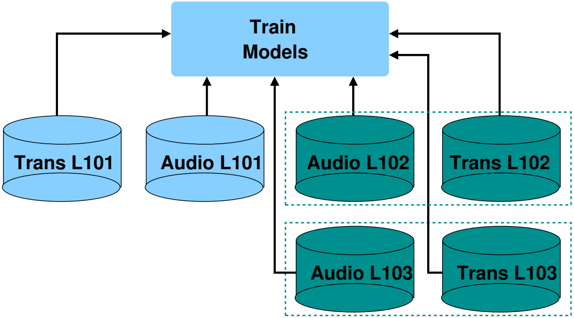
Low Resource
Speech Recognition

Use of (Deep) Neural Networks



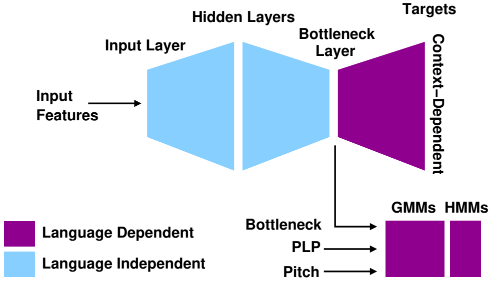
- Develop both Tandem and Hybrid system configurations
  - results are complementary (both for ASR and KWS) - see later
  - gains from techniques often apply to both set-ups
  - but systems also have different advantages
- Mixed gains from RNN/LSTM/CNN configurations
  - challenges to get KWS working well
  - BBN team got some gains in OP3

Multi-Language Framework



- Data from non-target language used to train model:
  - train complete acoustic model (see later)
  - train DNN to extract multi-language features

Multi-Language Bottleneck Features



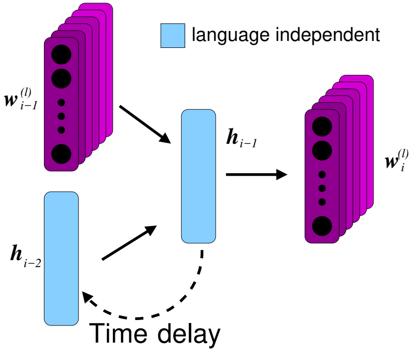
- Generate BN features from multiple languages
  - aim to make feature extractor language independent
  - language-dependent GMM used for recognition
- All layers other than output layer shared over all languages
  - output-layer language-specific - "hat-swapping"

Performance on Swahili

BottleNeck Features	TER (%)	MTWV		
		iv	oov	tot
FLP	44.6	0.5707	0.4121	0.5399
ML	41.7	0.6157	0.4733	0.5886

- Multi-Lingual (ML) BN Features trained on 11 languages
  - large gains in both ASR and KWS
- Larger gains observed as languages for BN features increases
- Other configurations possible
  - ML BN features used by all Babel teams

Multi-Language Language Models



- Current research direction
  - use ML-BN configuration but for language models
  - both input and output layers language dependent
  - far fewer parameters tied for LMs than BNs/hybrid systems

ASR: Lexicon

Phonetic Lexicons

- Most speech recognition systems use a phonetic lexicon:

A	ax
A	ey
A.	ey
A.'S	ey z
AAH	aa

- Each phone has attributes used for decision tree questions

ax	Vowel V-Back Back Short Medium Unrounded
ey	Vowel Short Dipthong Front-Start Fronting Medium Unrounded
z	Fricative Central Lenis Coronal Anterior Continuent Strident

- Phonetic lexicon generated manually
  - additional terms added using grapheme-to-phoneme (G2P) systems

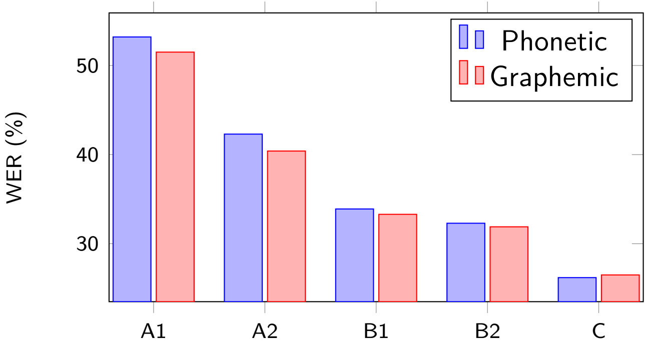
Graphemic Lexicons

- As well as manual cost other issues with phonetic lexicons
  - inconsistencies depending on the phonetician
  - sometimes transcriptions generated for particular speaker
- An alternative is to generate a graphemic lexicon

A	a^I
A.	a^I;B
A.'S	a^I;BA s^F
AAH	a^I a^M h^F

  - deterministic process - no manual/G2P system required
  - CUED system additional markers added (phonetic possible)
    - A - apostrophe following the letter
    - B - abbreviation (A., B. etc)
    - position - I (initial), M (middle), F (final)

Performance on English - Non-Native Learners



- For "beginners" graphemic systems outperform phonetic
  - as ability improves ASR performance improves
  - graphemic systems can be useful for (even) English!

Writing Systems

- English/European languages Latin script is used

What about general languages world-wide?

- There are a range of writing schemes used:
  - Pictographic - graphemes represent concepts
  - Logographic - graphemes represent words of morphemes
  - Syllabries - graphemes represent syllables
  - Segmental - form examined on the Babel project
- Segmental writing systems can be further partitioned as
  - alphabet - consonants and vowels both written
  - abugida - vowels marked as diacritics on consonants
  - abjad - only the consonants are written

Writing Systems

- English/European languages Latin script is used

What about general languages world-wide?

- There are a range of writing schemes used:
  - Pictographic - graphemes represent concepts
  - Logographic - graphemes represent words of morphemes
  - Syllabries - graphemes represent syllables
  - Segmental - form examined on the Babel project
- Segmental writing systems can be further partitioned as
  - alphabet - consonants and vowels both written
  - abugida - vowels marked as diacritics on consonants
  - abjad - only the consonants are written

Example Writing Schemes

Language	System	Script	Graphemes
Pashto	Abjad	Arabic	47
Tagalog	Alphabet	Latin	$53^{\dagger}$
Tamil	Abugida	Tamil	48
Zulu	Alphabet	Latin	$52^{\dagger}$
Kazakh	Alphabet	Cyrillic/Latin	$126^{\dagger}$
Telugu	Abugida	Telugu	60
Amharic	Abugida	Ethiopic	247
Mongolian	Alphabet	Cyrillic	$66^{\dagger}$

- Count excludes apostrophe, hyphen, punctuation ...
  - † includes capitals for Latin/Cyrillic scripts

Graphemic System Attributes

- Often no attributes associated with graphemes
  - limits decision tree questions to grapheme
  - no attributes such as voiced/unvoiced
- Interesting to examine additional attributes
  - bottom-up clustering of observed graphemes
  - make use of attributes of the unicode coding

Kazakh Lexicon

- Mixture of Cyrillic and Latin script
  - use unicode descriptors to map between forms

i	G6;D2D3D6	LATIN SMALL LETTER I
I	G6;D8D3D6	LATIN CAPITAL LETTER I
и	G6;D1D2D3	CYRILLIC SMALL LETTER I
ѝ	G6;D1D2D3D4	CYRILLIC SMALL LETTER I WITH GRAVE
й	G6;D1D2D3D5	CYRILLIC SMALL LETTER SHORT I

where the following attributes are defined

D1	CYRILLIC	D2	SMALL	D3	LETTER	D4	WITH GRAVE
D5	SHORT	D6	LATIN	D8	CAPITAL		

- Able to relate accented letters to root grapheme
  - also detect diacritics from actual graphemes

Phonetic vs Graphemic Performance

Language	Id	Script	TER (%) Phon	TER (%) Grph	TER (%) CNC
Tok Pisin	207	Latin	40.6	41.1	39.4
Kazakh	302	Cyrillic/Latin	53.5	52.7	51.5
Telugu	303	Telugu	69.1	69.5	67.5

- Comparable performance of graphemic/phonetic systems
  - graphemic/phonetic systems are complementary to one another
- Similar trend observed over all the Babel languages

ASR: Regularisation

Stimulated Systems

- Consider one layer of a standard deep neural network

$$\mathbf{h}^{(l)} = \boldsymbol{\sigma}\left(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right)$$

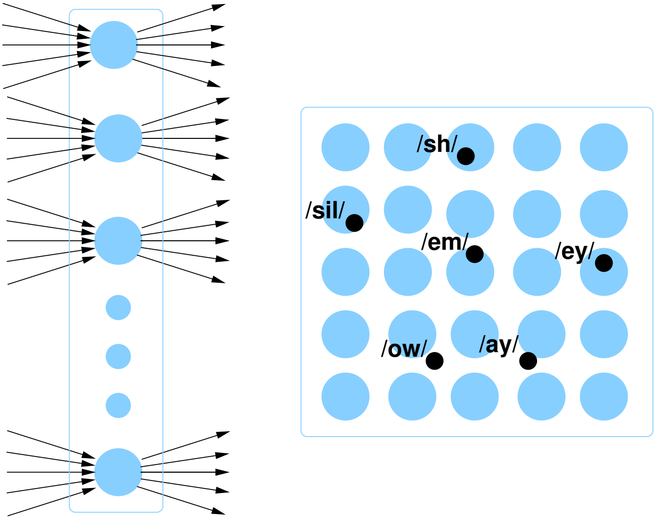
  - $\boldsymbol{\sigma}()$ - non-linear activation function
  - $\mathbf{W}^{(l)}, \mathbf{b}^{(l)}$ - network parameters for layer l

- No structure enforced on parameters
  - possible to arbitrarily order nodes (and get same result)
  - highly complicated relationship between layers

but that's kind of why we like them!

- Stimulated training: performance/interpretability balance

Stimulated Systems



Stimulated Network Training

- Introduce regularisation term into training

$$\mathcal{F}(\boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{\lambda}) + \alpha\mathcal{R}(\boldsymbol{\lambda})$$

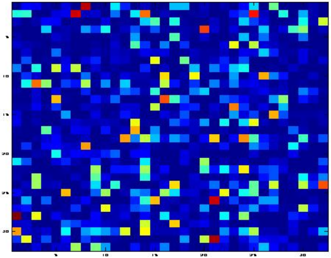
- Regularisation term $\mathcal{R}(\boldsymbol{\lambda})$ based on KL-divergence

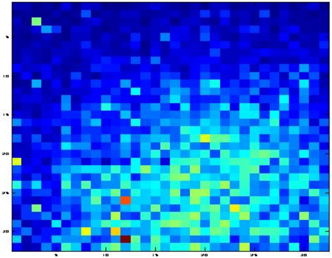
$$\mathcal{R}(\boldsymbol{\lambda}) = \sum_t \sum_l \sum_i g(\mathbf{s}_i, \hat{\mathbf{s}}_{p_t}) \log \left(\frac{g(\mathbf{s}_i, \hat{\mathbf{s}}_{p_t})}{\overline{h}_{ti}^{(l)}} \right)$$

$$g(\mathbf{s}_i, \hat{\mathbf{s}}_{p_t}) \propto \mathcal{N}(\mathbf{s}_i; \hat{\mathbf{s}}_{p_t}, \sigma^2\mathbf{I})$$

  - $\hat{\mathbf{s}}_{p_t}$ position in grid-space of active phone at time t
  - $\mathbf{s}_i$ position of node in grid-space of node i
  - $\overline{h}_{ti}^{(l)}$ (normalised) activation for node i of layer l at time t

Stimulated Training: Activation Function





Stimulated Training

Language	Id	Stimu Train	TER (%)	MTWV iv	MTWV oov	MTWV tot
Amharic	307	✗	41.1	0.6500	0.5828	0.6402
		✓	40.8	0.6619	0.5935	0.6521
Javanese	402	✗	50.9	0.4991	0.4448	0.4924
		✓	50.7	0.5024	0.4679	0.4993

- Stimulated training on hybrid system only
  - results based on combined hybrid/tandem systems
- Consistent gains (all languages) for ASR and KWS
  - enabled larger networks to be trained

ASR: Language Model

Language Model Training Data

- Concentrated on the acoustic model - LM also impacted
  - training data determines possible vocabulary for systems
  - vocabulary impacts OOV rates (both ASR/KWS)
  - quantity of data determines accuracy (and order) of LMs
- Significant quantities of data available on the web
  - Wikipedia - about 290 languages have entries
  - 1st item quantity, 2nd term “quality” measure:

English	5,056,964	911.38
Swedish	2,603,446	7.58
German	1,897,531	99.3
Cebuano	1,859,449	2.12
Dutch	1,851,256	10.86

Can we make use of web-data for language model training?

Language Model Training Data

- Babel project using conversational telephone speech
  - Wikipedia not a perfect match!
- A number of issues need to be considered
  - sources of data to use
  - ensure match to target language (language identification)
  - select data that matches target domain
  - tidying data
- Once sources found - build language model component(s)
  - interpolate (linear/log-linear) with matched source
  - interpolation weights often small - Swahili VLLP
    VLLP-LM 0.885, TED 0.015, Blogs 0.008, General 0.0926

BABEL OP3 Web Collection

Language	Id	LM	Data (K) words	Data (K) vocab	FLP Weight	OOV (%) ASR	OOV (%) KWS
Pashto	104	FLP	535	14.4	—	1.96	11.38
		Web	104624	376.3	0.981	0.68	3.05
Amharic	307	FLP	388	35.0	—	9.80	15.42
		Web	13911	223.6	0.976	5.67	9.16
Georgian	404	FLP	406	34.3	—	8.16	14.93
		Web	137041	278.6	0.911	3.02	5.22

- Quantity of web-data available highly dependent on language
  - interpolation weight ("match") of web data 0.089 to 0.019
  - remember need for rapid deployment

Efficient:

Model Training

Keyword Spotting

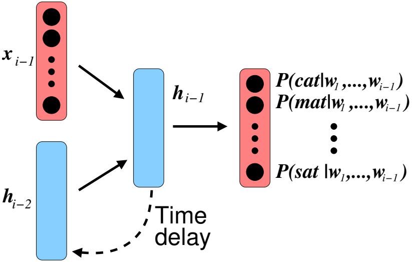
System Combination

Efficient Model Building

- Rapid/efficient system development important in Babel
  - handle any language
  - rapid development of surprise language: 1 week!
  - large amounts of evaluation data ($\approx$ 80 hours)
- "Plug and Play" scripts developed (all sites)
  - standardised language pack distributions
  - common system set-up for all languages
- Various "bottlenecks" needed to be addressed
  - state-of-the-art systems
  - rich lattices (large quantities of data)
  - system combination (best performance)

Efficiency: RNNLMs

RNN Language Models



- Recurrent neural networks model complete word history

$$P(\boldsymbol{\omega}_{1:L}) \approx \prod_{i=1}^{L} P(\omega_i|\omega_{i-1}, \tilde{\boldsymbol{h}}_{i-2}) \approx \prod_{i=1}^{L} P(\omega_i|\tilde{\boldsymbol{h}}_{i-1})$$

- Issues that need to be addressed: training & decoding

Cross-Entropy Training Criteria

- Standard training criterion for word sequence $\boldsymbol{\omega}_{1:L} = \omega_1, \ldots, \omega_L$

$$\mathcal{F}_{\mathsf{ce}} = -\frac{1}{L}\sum_{i=1}^{L} \log\left(P(\omega_i|\tilde{\boldsymbol{h}}_{i-1})\right)$$

  - GPU training makes this reasonable BUT
- Compute cost for softmax normalisation term $Z(\tilde{\boldsymbol{h}}_{i-1})$

$$P(\omega_i|\tilde{\boldsymbol{h}}_{i-1}) = \frac{1}{Z(\tilde{\boldsymbol{h}}_{i-1})}\exp\left(\boldsymbol{w}^{\mathsf{T}}_{f(\omega_i)}\tilde{\boldsymbol{h}}_{i-1}\right)$$

  - required as unobserved sequence (contrast acoustic model)
  - scales with vocabulary size and training data

Alternative Training Criteria

- Variance Regularisation: eliminate decoding normalisation

$$\mathcal{F}_{\tt vr} = \mathcal{F}_{\tt ce} + \frac{\gamma}{2}\frac{1}{L}\sum_{i=1}^{L}\left(\log(Z(\tilde{\boldsymbol{h}}_{i-1})) - \overline{\log(Z)}\right)^2$$

  - $\overline{\log(Z)}$ average (log) history normalisation
  - all normalisation terms tend to be the same
- Noise Contrastive Estimation: efficient decoding and training

$$\mathcal{F}_{\tt nce} = -\frac{1}{L}\sum_{i=1}^{L}\left(\log(P(y_i = {\tt T}|\omega_i, \tilde{\boldsymbol{h}}_{i-1}) + \sum_{j=1}^{k}\log(P(y_i = {\tt F}|\hat{\omega}_{ij}, \tilde{\boldsymbol{h}}_{i-1})\right)$$

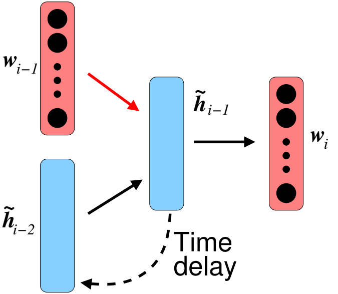
  - $\hat{\omega}_{ij}$ competing samples for ω_i - often sample from uni-gram LM

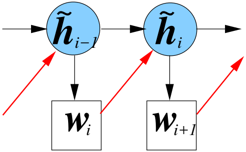
Impact of RNN LM (Pashto)

LM Data	RNN Crit Trn	RNN Crit F-T	Time (hrs) Train	Time (hrs) Rescore	TER (%)
FLP	—		—		44.1
FLP+Web	—		—		43.8
	CE	CE	125.0	23.0	42.8
	NCE	VR	10.7	2.0	43.0

- Gains from web-data for N-gram
  - larger gains from RNNLM
  - modified training reduced training time >5 days to $<1/2$ day
- BUT KWS requires large lattices to handle high WERs ...
  - interacts badly with the RNNLM ....

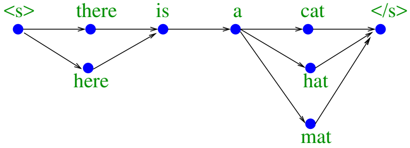
ASR Decoding with RNNLMs



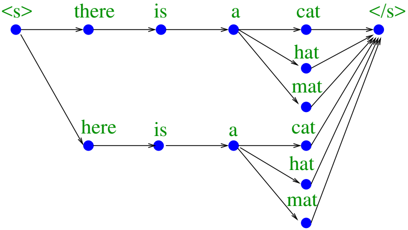


- ASR decoding LM score depends on previous hypothesis
  - history vector depends on "unobserved" word sequence
  - predictions depends on complete previous path
- Possible to use for ASR (or even use N-best lists)
  - impractical to use for lattices (and lattice generation)

ASR Decoding with RNNLMs



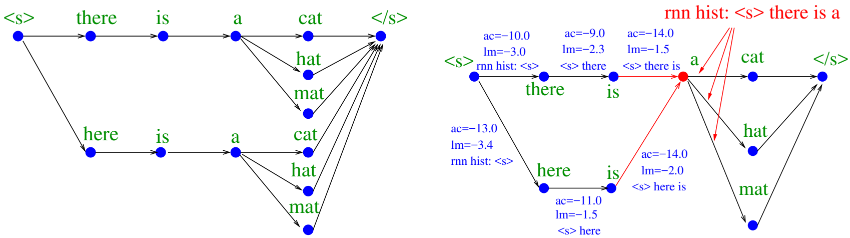
Lattice



Prefix Tree

- Consider word-lattice on the left
  - becomes prefix tree (right) using complete history
  - significant increase in number of paths

N-Gram History Approximation



Prefix Tree

N-Gram Approximation

- Use exact RNN LM value but
  - merge paths based on N-gram history
  - can also use history vector distance merging

Impact of RNN LM (Pashto)

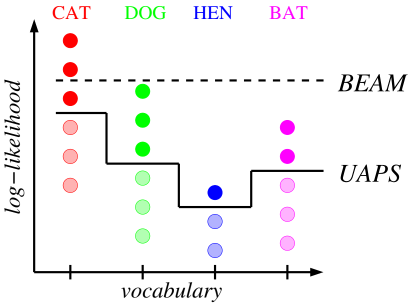
LM Data	RNN Crit Trn	RNN Crit F-T	TER (%)	MTWV iv	MTWV oov	MTWV tot
FLP	—		44.1	0.4808	0.2412	0.4541
FLP+Web	—		43.8	0.4828	0.4083	0.4750
	CE	CE	42.8	0.4975	0.4048	0.4871
	NCE	VR	43.0	0.4975	0.3953	0.4862

- Large gains for KWS than ASR from web-data
  - reduces the keyword OOV rate
- Efficient training does not impact performance

Efficiency: KWS

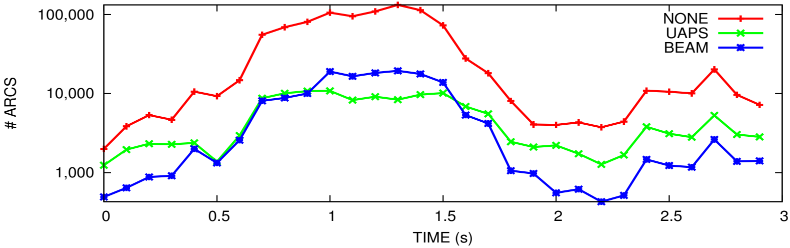
Unique-Arcs-Per-Second Pruning

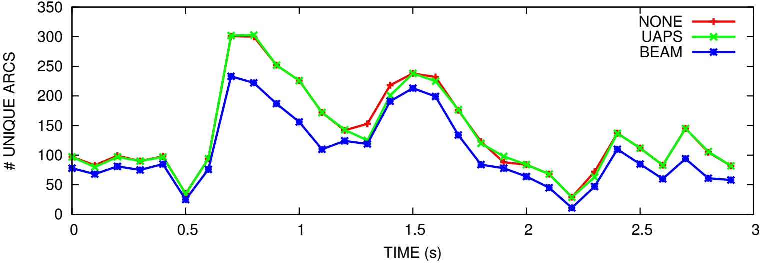
- Need compact lattices to ensure speed of KWS
  - need diverse lattices to ensure performance of KWS
  - alternative to CN-KWS and quantised-time lattices



- Modify pruning to maintain distribution over unique arcs
  - (currently) implemented as lattice post-processing stage

Unique-Arcs-Per-Second Pruning - Impact





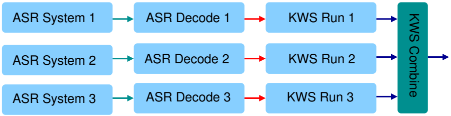
Unique-Arcs-Per-Second Pruning - Impact

Language	Id	Arcs/Sec	
		Decode	UAPS
Mongolian	401	88,479	17,623
Javanese	402	41,880	11,109

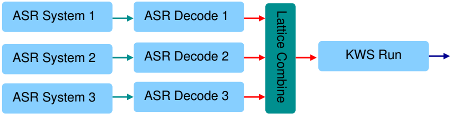
- Dramatic reduction in lattice size
  - for some languages an order of magnitude
- No degradation in performance - significantly faster
  - far richer lattices could be used for evaluation
- Approach can be applied at lattice generation stage

Efficiency: Combination

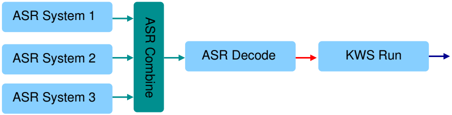
KWS System Combination Architectures



Posting-List Combination



Lattice Combination



ASR System Combination

Combination Approaches

- ASR system combination
  - minimum Bayes' risk (confusion network) combination

$$\hat{\omega} = \arg\min_{\omega} \left\{ \sum_{\overline{\omega}} \left(\sum_{m=1}^{M} P(\overline{\omega}|\boldsymbol{x}_{1:T}; \mathcal{M}^{(m)}) \mathcal{L}(\omega, \overline{\omega}) \right) \right\}$$

    multiple decode - posting-list merging/lattice combination
  - joint decoding

$$\log\left(p(\boldsymbol{x}_t|\mathrm{s})\right) \propto \sum_{m=1}^{M} \log\left(p(\boldsymbol{x}_t|\mathrm{s}; \mathcal{M}^{(m)})\right)$$

    single decode - single KWS run
- KWS posting-list merging ... see paper references

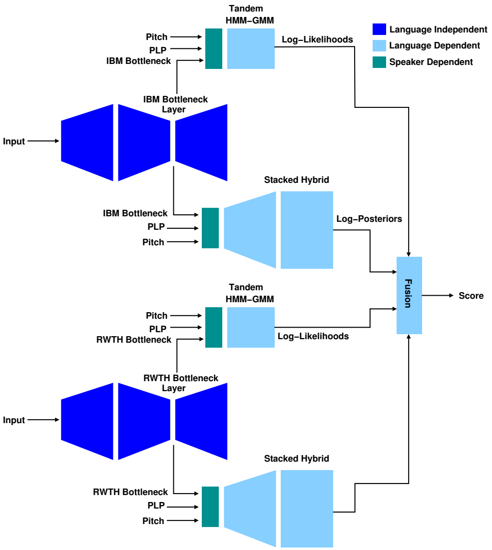
System Combination (Georgian)

System	BN Features	TER (%)	MTWV iv	MTWV oov	MTWV tot
HI (IBM)	Hybrid	40.1	0.7178	0.7254	0.7198
HA (Aachen)	Hybrid	40.0	0.7129	0.7221	0.7152
HI⊕HA	Joint	38.1	0.7390	0.7413	0.7398
HI⊗HA	Merge	37.9	0.7379	0.7542	0.7409

- Significant gains from system combination (ASR/KWS)
  - small performance differences joint/merge
  - joint decoding significantly more efficient
- Evaluation used both styles of system combination

Evaluation System

OP3 4-Way Joint Decoding



- 28 language BN features
  - A28+: fine-tuned RWTH
  - I28: IBM
- 4-way Joint (A28+⊕I28):
  1. IBM-BN Hybrid-SAT
  2. IBM-BN Tandem-SAT
  3. RWTH-BN Hybrid-SAT
  4. RWTH-BN Tandem-SAT
- Multiple models built
  - semi-supervised training
  - enriched lexicon
- Multiple LMs built

Surprise Language System Description

- Enriched lexicon based language specific peculiarities (LSP)
  - document describing general attribues of language
- Used morphological decomposition (Morfessor)

J1	4-way, 45×45 nodes, word RNNLM, LSP lexicon
J2	4-way, 45×45 nodes, word RNNLM
J3	4-way, semi-supervised, 45×45 nodes, word RNNLM, LSP lexicon
M3	4-way, semi-supervised, 45×45 nodes, morph RNNLM, LSP lexicon

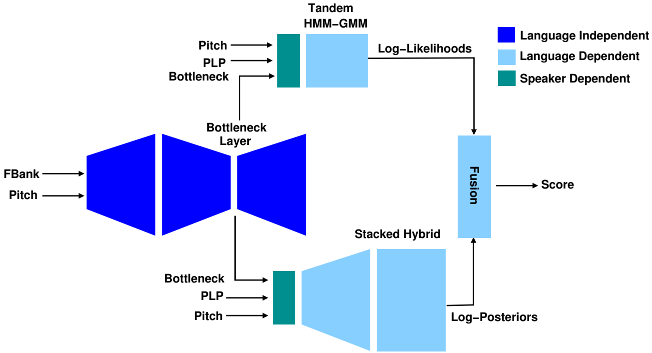
Surprise Language System Performance

System	TER (%)	STO iv	STO oov	STO tot	KST tot
J1†	36.7	0.7379	0.7389	0.7383	0.7409
J2†	37.1	0.7381	0.7194	0.7357	0.7389
J3‡	36.5	0.7431	0.7242	0.7407	0.7461
M3	—	0.6820	0.7197	0.6882	—
J3⊗M3†	—	0.7430	0.7555	0.7452	
J3⊗J2	36.0	0.7481	0.7440	0.7479	
J3⊗J1⊗J2	36.1	0.7473	0.7521	0.7487	
J3⊗J2⊗M3	—	0.7481	0.7676	0.7514	

- † indicates systems supplied to IBM for combination
- ‡ indicates the single system submission

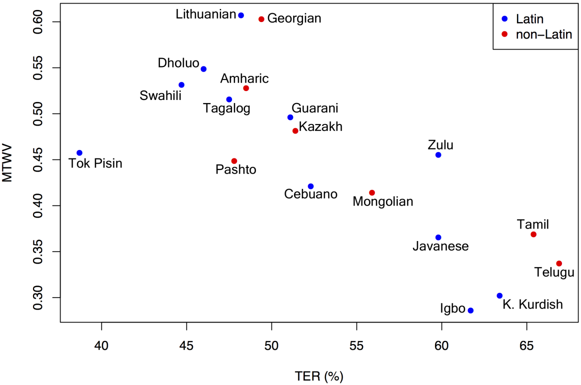
Performance Analysis

Performance Analysis (OP2 Configuration)



- Framework used for OP2 evaluation
  - combines (stacked) Hybrid-SAT and Tandem-SAT systems
  - supervision from Hybrid-SI system

Summary plot MTWV vs TER for FLP (OP-2)



- Examined performance with a consistent configuration

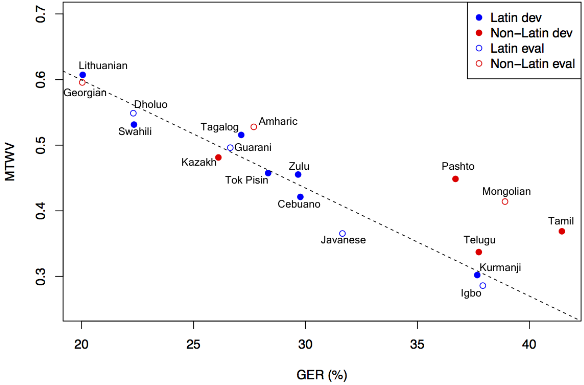
Language Variability

- Performance range: $0.3 \rightarrow 0.6$ MTWV, $< 40\% \rightarrow > 65\%$
  - correlation between Word (Token) Error Rate and MTWV
- Range of factors may impact performance:
  - recording conditions (telephone network)
  - morphological complexity of language (vocabulary size)
  - syntactic complexity of language (impact of language model)
  - grapheme to phoneme relationship
  - "confusability" of words
  - nature of the keywords being used
  - accuracy of transcriptions

Interested in what is important (and predict)

- So we tried many things ... many didn't correlate

Graphemic Error Rate for Prediction



- Graphemic Error Rate (GER) correlated well
  - basic (PLP/GMM/ML) ASR on training data (fast,simple)
  - handles many aspects of impact factors

Performance Prediction

Language	Id	Script	%TER		MTWV	
			pred	obs	pred	obs
Dholuo	403	Latin	45.4	46.0	0.561	0.549
Guarani	305	Latin	49.5	51.1	0.490	0.496
Igbo	306	Latin	60.2	61.7	0.304	0.286
Javanese	402	Latin	54.2	59.8	0.408	0.362
Amharic	307	Ethiopic	50.5	48.5	0.473	0.528
Mongolian	401	Cyrillic	61.1	55.9	0.288	0.414
Georgian	404	Mkhedruli	43.3	49.2	0.599	0.596

- Not bad - even for non-Latin languages
  - BUT still had to build a basic system ...

Conclusions

Conclusions

- “Plug and Play” systems built for 25 diverse languages
  - graphemic lexicons worked well for all languages
- Multi-language acoustic models important
  - either bottleneck features, or “complete” models
- Predicting difficulty of a language challenging
  - need more languages to draw conclusions
- Babel programme data a wonderful resource

Acknowledgements

This work was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.
This work made use of data provided by IARPA [1].
The authors would like to thank the contributions of the members of the CUED Babel team during the project, and all the members of the LORELEI team, in particular the IBM and RWTH Aachen Babel teams.

[1] The following data was used in the FLP configuration: IARPA-babel106-v0.2f, IARPA-babel202b-v1.0d, IARPA-babel204b-v1.1b, IARPA-babel205b-v1.0a, IARPA-babel206b-v0.1d, IARPA-babel207b-v1.0a, IARPA-babel301b-v1.0b, IARPA-babel302b-v1.0a, IARPA-babel303b-v1.0a, IARPA-babel304b-v1.0b, IARPA-babel104b-v0.4bY, IARPA-babel306b-v2.0c, IARPA- babel401b-v2.0b, IARPA-babel402b-v1.0b, IARPA-babel403b-v1.0b, IARPA-babel404b-v1.0a.