The FBK ASR system for Evalita 2011

R. Ronny[1], A. Shakoor, F. Brugnara, R. Gretter

FBK - Fondazione Bruno Kessler, Trento

Rome, 25-1-2012

[1] R.Ronny, now Master student ad Edinburgh, trained the system during his summer student internship at FBK.

Outline

- The processing flow in the FBK ASR system
- Brief description of some acoustic modeling techniques
- Brief description of some LM representation techniques
- Models used for Evalita evaluation
- Results

Processing in the FBK transcription system (1/2)

FONDAZIONE BRUNO KESSLER

- Segmentation. The audio is split into segments separated by silence.
- Classification. An HMM-based classifier assigns to each segment one of a number of class labels: *noise, music, male-speech, female-speech, etc...* Noise and music segments are excluded from further processing.
- Clustering. Within each class, segments are grouped into homogeneous clusters by means of a BIC-based agglomerative clustering.
- First acoustic normalization. Unsupervised CMLSN is performed with respect to a GMM, producing a specific affine transformation for each cluster.

Processing in the FBK transcription system (2/2)

FONDAZIONE BRUNO KESSLER

- Feature projection. The normalized features are projected into a lower dimensional feature space by applying an HDA linear transform.
- First recognition step. The decoder is run on the normalized and projected features using a proper Acoustic Model and a n-gram Language Model, producing the first recognition hypothesis.
- Second acoustic normalization. Supervised CMLSN is performed with respect to a set of simple triphone *target-models*, using the output of the previous step as supervision.
- Second recognition step. The decoder is run again on the resulting features using a proper Acoustic Model and a n-gram Language Model, producing the final hypothesis.

The processing flow takes advantage of a load-balancing dispatcher/collector to distribute the computational effort among several machines.

CMLLR-based acoustic normalization (1/2)

CMLLR = *Constrained Maximum Likelihood Linear Regression*[1].
CMLSN = *CMLLR-based Speaker Normalization*[2].

- Techniques meant to reduce irrelevant variability in speech data.
- Given a target Acoustic Model and a transcription of a cluster of segments, computes an affine transformation that maximizes the likelihood of transformed data with respect to the model.
- In original CMLLR, the target model coincide with the recognition model, and the procedure can be seen as a *feature-space transformation* dual form of a *model-space adaptation*.

[1] M.J.F. Gales, "Maximum likelihood linear transformations . . . " Computer Speech & Language, 1998
[2] D. Giuliani *et al.*, "Speaker normalization through constrained MLLR . . . " in Proc. ICSLP, 2004

CMLLR-based acoustic normalization (2/2)

- In CMLSN, target and recognition models are different.
- It has been observed[1] that, in supervised normalization, using simple target triphone models can be more effective.
- If the target AM is made of a single generic model, no supervision is required, and CMLSN can be effectively applied without any preliminary recognition step.

The CMLSN procedure is also applied to training data before ML training of the AM, thus producing a *normalized* AM. This can be seen as a variant of Speaker Adaptive Training.

[1] G.Stemmer *et al*, "Adaptive training using simple target models" in Proc. ICASSP 2005

HDA - Heteroscedastic Discriminant Analysis

- A dimensionality reduction technique, aiming at preserving the information of a large feature vector in a more convenient compact vector[1].
- Given a set of target classes and labeled data, applies a Maximum Likelihood criterion to estimate a linear transform that separates the feature space in a significant subspace and a nuisance subspace, that does not contain discriminant information.
- In ASR, the target classes are usually triphone HMM states.
- Its application at run-time only consists in multiplying the feature vectors by a rectangular matrix.
- Can be effectively integrated with CMLSN[2].

[1] N.Kumar *et al.*, "Heteroscedastic discriminant analysis . . . ", Speech Communication, 1998

[2] G.Stemmer *et al.*,"Integration of Heteroscedastic Linear Discriminant Analysis . . . ", Proc. ICASSP 2006

LM representation (1/2)

The FBK system employs a static representation of the LM, i.e. a large network (graph) embodying phonetic and linguistic constraints, usually 4-grams.

Pros:

- Decoding does not assume any particular structure of LM, as soon as it can be compiled into a network.
- No run-time overhead due to dynamic access to the language model and application of the lexicon.

Cons:

- The memory required for storing the compiled network is larger than that required for storing the LM and lexicon separately.

Except for the network, other information describing the search status are allocated dinamically.

LM representation (2/2)

To reduce memory requirements, several techniques are adopted:

1. Shared-tail topology: the redundancy in the *tree-based* representation of an n-gram LM is reduced by avoiding duplication in the tree branches.
2. Network reduction: an "approximate optimization" method is applied that iteratively collapses equivalent nodes.
3. Chain merging: sequences of arcs connecting nodes with unique input and output are represented by a single "multi-arc".

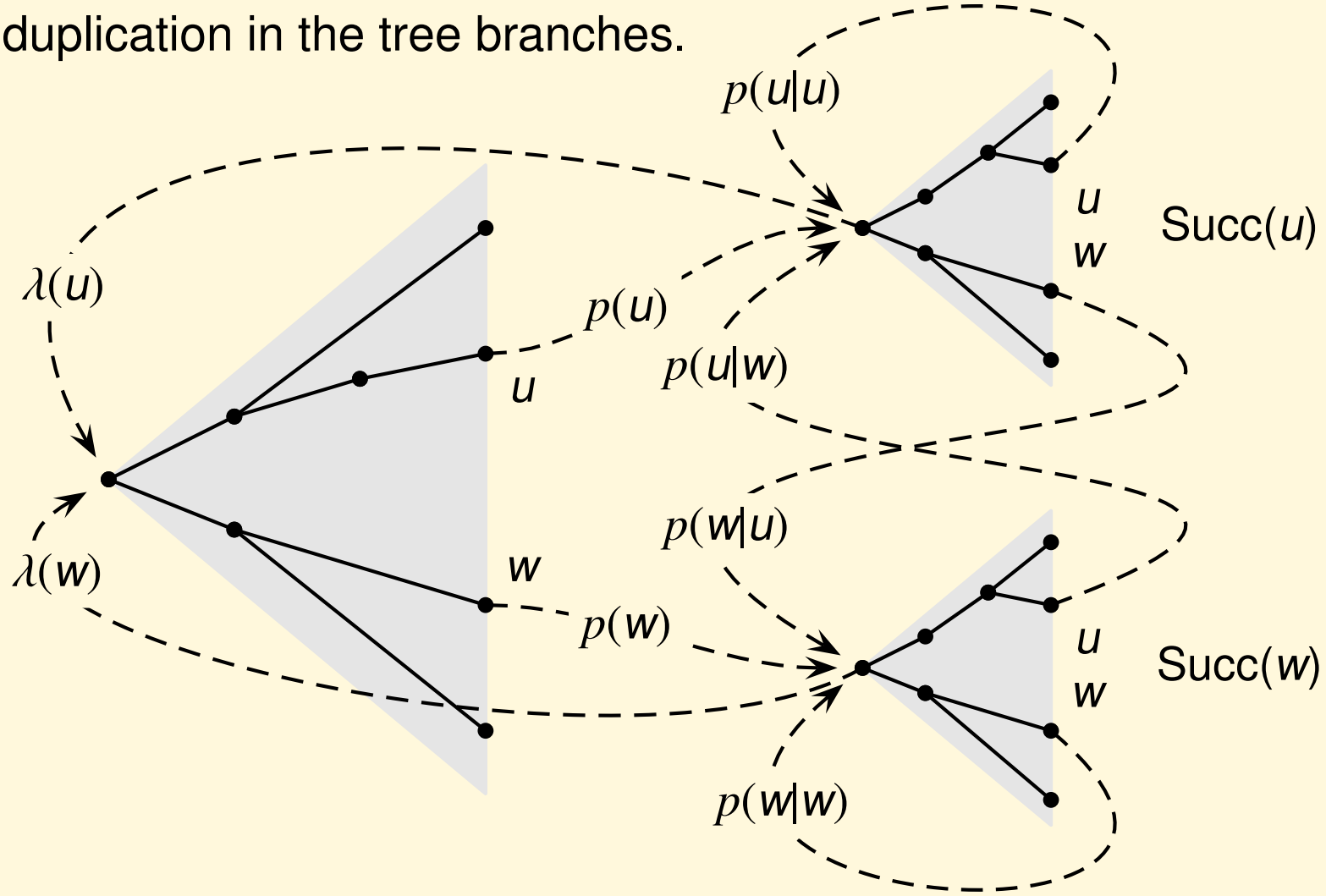
Techniques 2 and 3 are not limited to networks derived from n-grams, and tipically reduce the size by $\approx$50%.

Moreover, if several instances of the decoder use the same network on a machine, they can load it in shared memory.

LM representation (2/2)

To reduce memory requirements, several techniques are adopted:

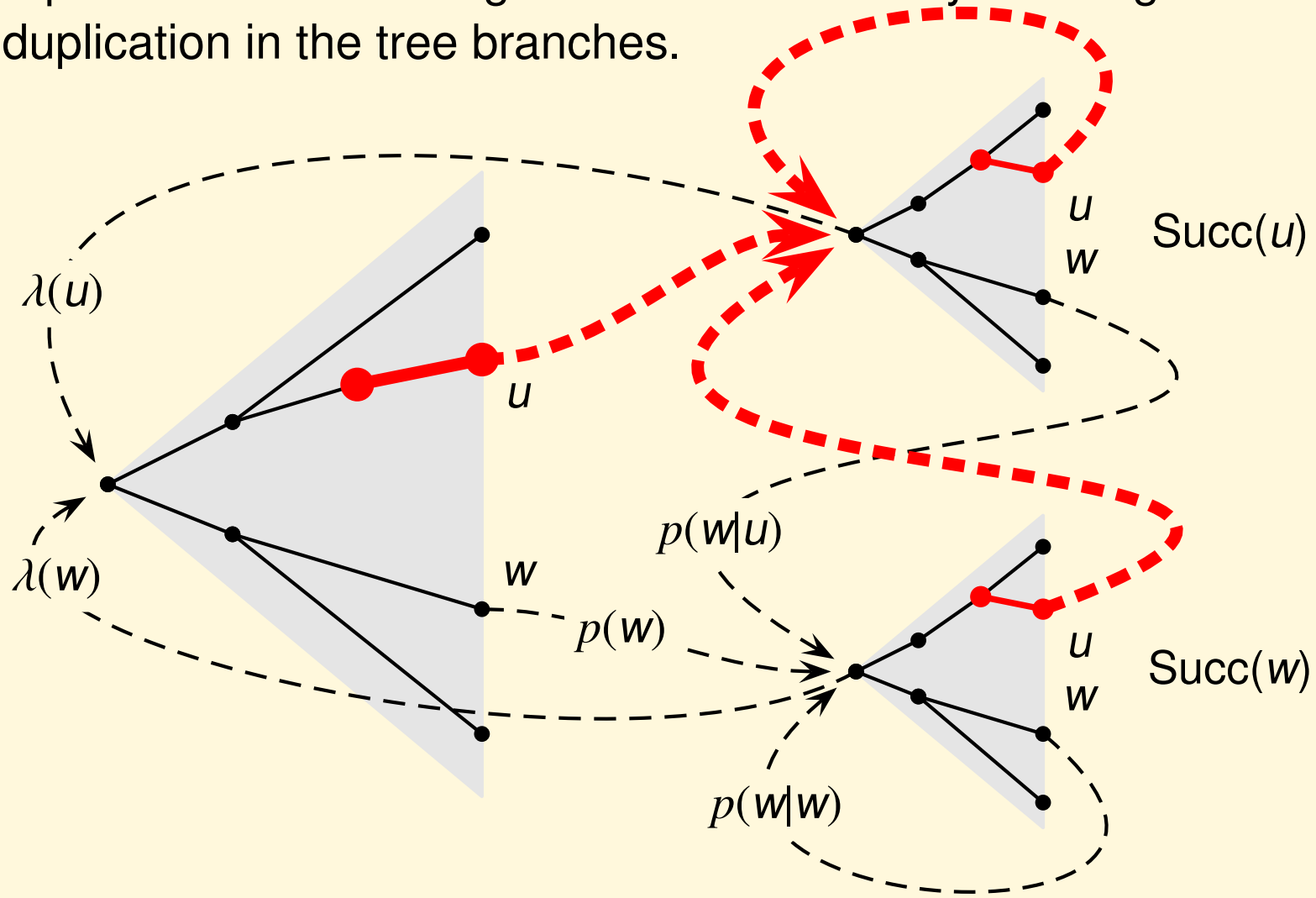
1. Shared-tail topology: the redundancy in the *tree-based* representation of an n-gram LM is reduced by avoiding duplication in the tree branches.



LM representation (2/2)

To reduce memory requirements, several techniques are adopted:

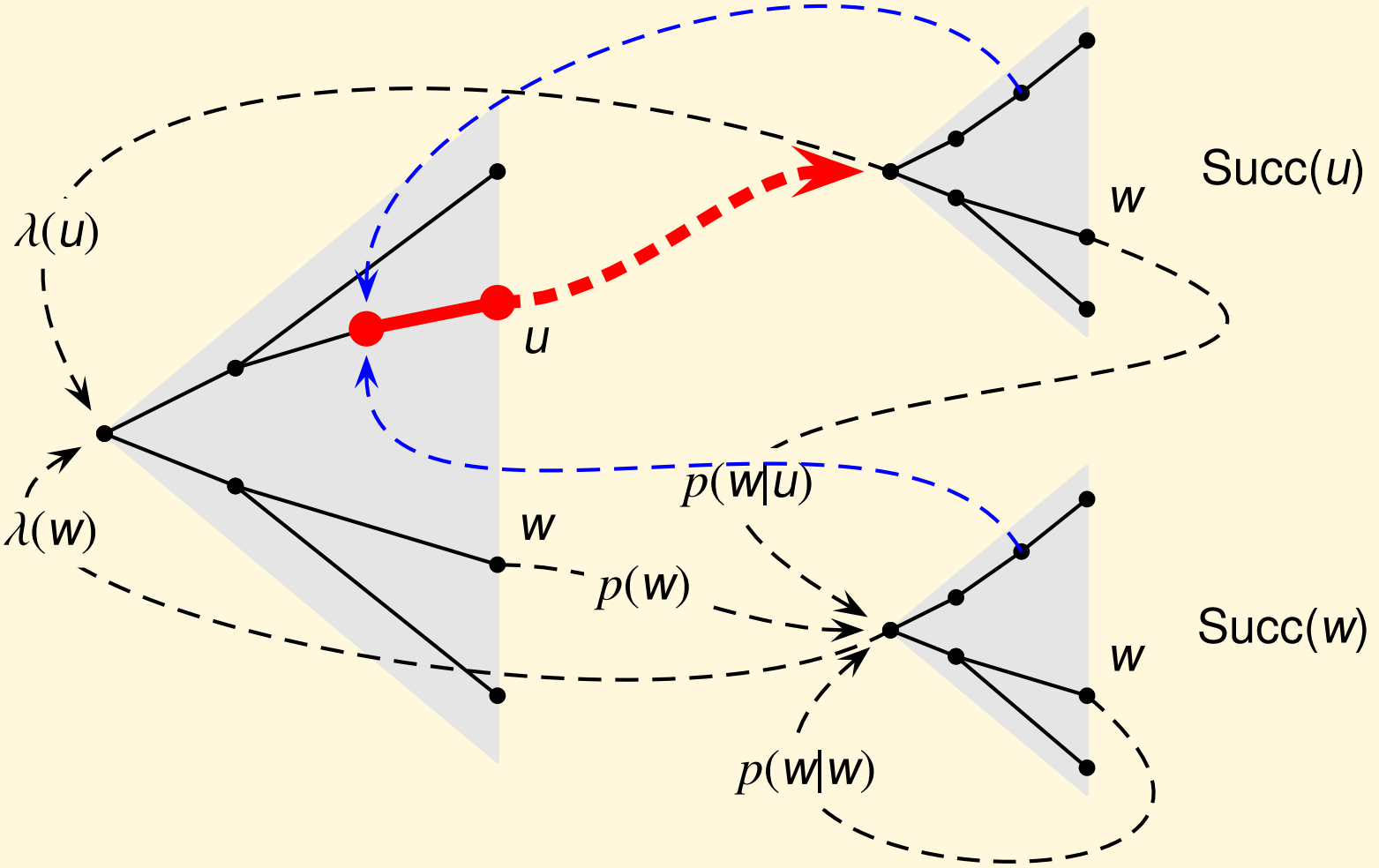
1. Shared-tail topology: the redundancy in the *tree-based* representation of an n-gram LM is reduced by avoiding duplication in the tree branches.



LM representation (2/2)

To reduce memory requirements, several techniques are adopted:

1. Shared-tail topology: the redundancy in the *tree-based* representation of an n-gram LM is reduced by avoiding duplication in the tree branches.



LM representation (2/2)

To reduce memory requirements, several techniques are adopted:

1. Shared-tail topology: the redundancy in the *tree-based* representation of an n-gram LM is reduced by avoiding duplication in the tree branches.
2. Network reduction: an "approximate optimization" method is applied that iteratively collapses equivalent nodes.
3. Chain merging: sequences of arcs connecting nodes with unique input and output are represented by a single "multi-arc".

Techniques 2 and 3 are not limited to networks derived from n-grams, and tipically reduce the size by $\approx$50%.

Moreover, if several instances of the decoder use the same network on a machine, they can load it in shared memory.

Models for Evalita (1/2)

Acoustic Features

- 13 MFCC + 1^{st},2^{nd},3^{rd} derivatives
  $\rightarrow$ 52 features, reduced to 39 by HDA after normalization.
- Segment-based average normalization

Acoustic Model(s)

- Cross-word triphone HMMs: $\approx$8700 models with $\approx$6700 tied states, and $\approx$37000 Gaussians. Average mixture length is 94.
- State tying based on Phonetic Decision Tree.
- Gaussian tying based on Phonetically Tied Mixtures.
- Different models (of approximately same size) for first and second recognition steps.
- A single GMM with 1024 components on the 52-dimensional feature space for unsupervised normalization.
- A set of simple tied-states triphone models (one Gaussian per state) for supervised normalization.

Models for Evalita (2/2)

FONDAZIONE BRUNO KESSLER

Language Model

- 4-gram Language Models estimated with *Kneser-Ney* smoothing on the given corpus of ≈32M words.
- LM size: 67K words, 3.37M + 2.34M + 2.33M n-grams.
- Network size after integration with lexicon:
  6.2M nodes, 6.3M labeled arcs, 10.4M empty arcs.
- For the *contrained transcription* task, the baseline LM was adapted to the 63K words of the session report by means of *mixture adaptation*.
- The same procedure was followed for generating the automatic transcription of the training data.

Results

- Results show that, on the given task, reasonable performance can be achieved with the provided data:

Task	WER (%)	
	official (lms=7)	fixed (lms=10)
Transcription	8.4	7.5
Constrained Transcription	7.2	6.1

- Run time is approximately 3×real-time on a single CPU.
- A mistake was made in the official submission: a default LM scale was used, not appropriate for the given models. After modifying this single parameter, performance changed as shown in the table.