Aalto University

End-to-end speech recognition using lattice-free MMI

Hossein Hadian, Hossein Sameti, Daniel Povey, Sanjeev Khudanpur

Presented by Tamás Grósz

Outline

End-to-end speech recognition

E2E models directly transcribe speech to text without requiring predefined alignment between acoustic frames and characters

- Single model is used
- New training methods are needed

The MMI method

- MMI stands for Maximum mutual Information
- It is a sequence discriminative training criteria

The MMI method

- MMI stands for Maximum mutual Information
- It is a sequence discriminative training criteria
- The objective takes into account the whole utterance -> sequence

The MMI method

- MMI stands for Maximum mutual Information
- It is a sequence discriminative training criteria
- The objective takes into account the whole utterance -> sequence
- We use an objective function that optimizes some criteria associated with the task -> discriminative

The MMI method in details

$$F_{MMI}(\lambda) = \sum_{u \in U} log \frac{P_\lambda(O_u|H_{w_u})P(w_u)}{\sum_{\hat{w}} P_\lambda(O_u|H_{\hat{w}_u})P(\hat{w}_u)}$$

The MMI method in details

$$F_{MMI}(\lambda) = \sum_{u \in U} log \frac{\boxed{P_\lambda(O_u|H_{w_u})P(w_u)}}{\sum_{\hat{w}} P_\lambda(O_u|H_{\hat{w}_u})P(\hat{w}_u)}$$

The numerator simply calculates the probability of the correct transcription (w_u) using the model (λ).

The MMI method in details

$$F_{MMI}(\lambda) = \sum_{u \in U} log \frac{P_\lambda(O_u|H_{w_u})P(w_u)}{\sum_{\hat{w}} P_\lambda(O_u|H_{\hat{w}_u})P(\hat{w}_u)}$$

The MMI method in details

$$F_{MMI}(\lambda) = \sum_{u \in U} log \frac{P_\lambda(O_u|H_{w_u})P(w_u)}{\boxed{\sum_{\hat{w}} P_\lambda(O_u|H_{\hat{w}_u})P(\hat{w}_u)}}$$

- The denominator calculates the summed probability of all possible sequences of words.

The MMI method in details

$$F_{MMI}(\lambda) = \sum_{u \in U} log \frac{P_\lambda(O_u|H_{w_u})P(w_u)}{\boxed{\sum_{\hat{w}} P_\lambda(O_u|H_{\hat{w}_u})P(\hat{w}_u)}}$$

- The denominator calculates the summed probability of all possible sequences of words.
- It requires decoding with the model.

The MMI method in details

$$F_{MMI}(\lambda) = \sum_{u \in U} log \frac{P_\lambda(O_u|H_{w_u})P(w_u)}{\boxed{\sum_{\hat{w}} P_\lambda(O_u|H_{\hat{w}_u})P(\hat{w}_u)}}$$

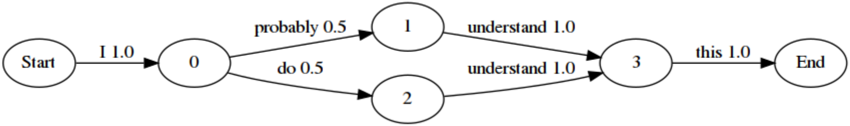
- The denominator calculates the summed probability of all possible sequences of words.
- It requires decoding with the model.
- Summing over all sequences is not practically feasible, instead:
  - N-best list (less used since it is too crude)

The MMI method in details

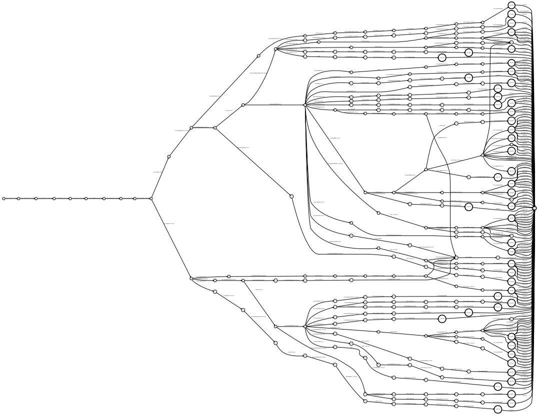
$$F_{MMI}(\lambda) = \sum_{u \in U} log \frac{P_\lambda(O_u|H_{w_u})P(w_u)}{\boxed{\sum_{\hat{w}} P_\lambda(O_u|H_{\hat{w}_u})P(\hat{w}_u)}}$$

- The denominator calculates the summed probability of all possible sequences of words.
- It requires decoding with the model.
- Summing over all sequences is not practically feasible, instead:
  - N-best list (less used since it is too crude)
  - Lattice structure

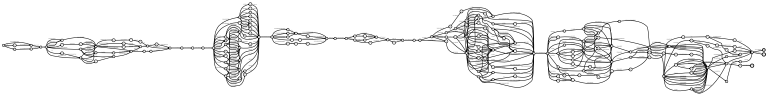
Lattice


Start
I 1.0
0
probably 0.5
do 0.5
1
2
understand 1.0
understand 1.0
3
this 1.0
End

Lattice



Lattice



MMI loss function

How do we train a DNN with the MMI method?

MMI loss function

How do we train a DNN with the MMI method?

$$\frac{\partial F_{MMI}}{\partial y_t^u} = {}^{NUM}\gamma_t^u - {}^{DEN}\gamma_t^u$$

MMI loss function

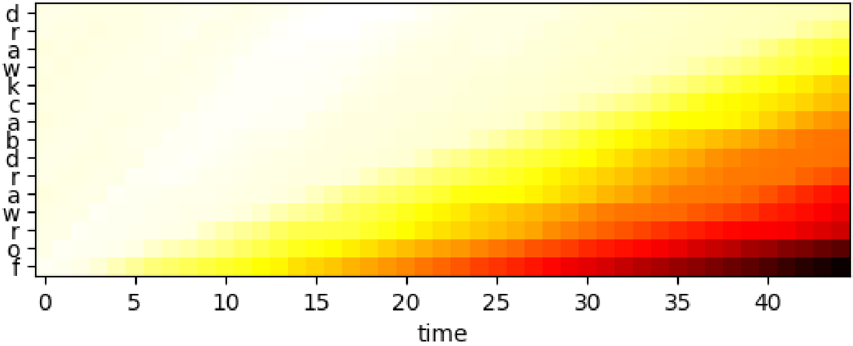
How do we train a DNN with the MMI method?

$$\frac{\partial F_{MMI}}{\partial y_t^u} = {}^{NUM}\gamma_t^u - {}^{DEN}\gamma_t^u$$

where γ is the forward-backward algorithm.

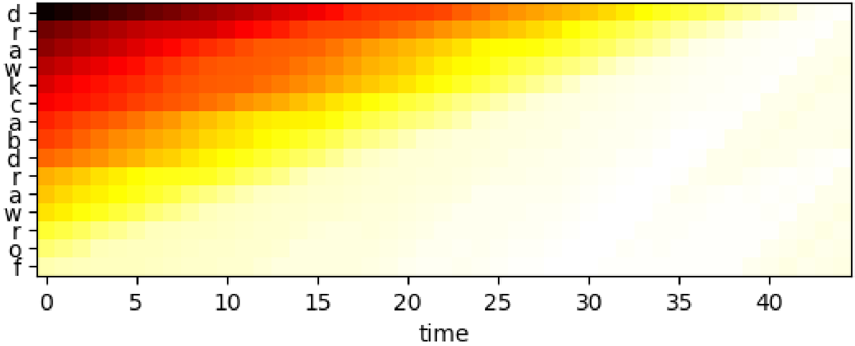
Forward-backward algorithm

- The goal is to find the alignment between the text and the audio



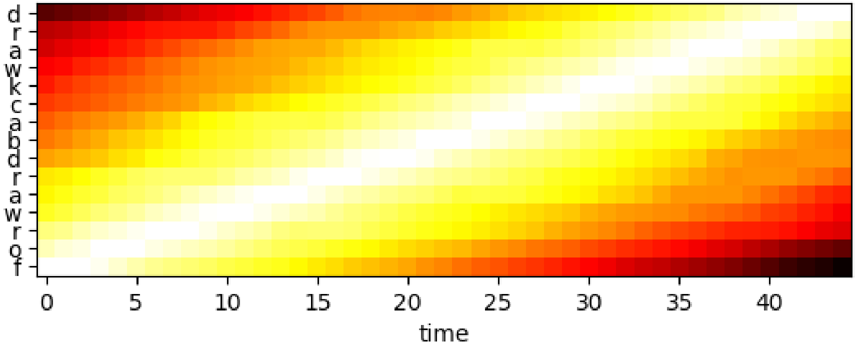
Forward-backward algorithm

- The goal is to find the alignment between the text and the audio



Forward-backward algorithm

- The goal is to find the alignment between the text and the audio



Relation to CTC

- Using the numerator is quite similiar to CTC
- The differences between CTC and MMI:

Relation to CTC

- Using the numerator is quite similiar to CTC
- The differences between CTC and MMI:
  - No decoding in CTC
  - CTC uses fixed and uniform state priors, observation priors, and transition probabilities
  - Different topology (blank label)

Lattice-free MMI

Problems:

- Requires initialization with a trained model
- Unique lattice for each utterance
- Computationally expensive

Lattice-free MMI

Problems:

- Requires initialization with a trained model
- Unique lattice for each utterance
- Computationally expensive

Solution:

- Represent the denominator as a graph
- Fit the graph in the GPU

Decoder graph

Notations

H=HMM state graph, C=context-dependency, L=the lexicon, G=the language model

Decoder graph

Notations

H=HMM state graph, C=context-dependency, L=the lexicon, G=the language model

- Traditional ASR systems use HCLG (composition)

Decoder graph

Notations

H=HMM state graph, C=context-dependency, L=the lexicon, G=the language model

- Traditional ASR systems use HCLG (composition)
- Composing a graph over all possible word sequences is not feasible

Decoder graph

Notations

H=HMM state graph, C=context-dependency, L=the lexicon, G=the language model

- Traditional ASR systems use HCLG (composition)
- Composing a graph over all possible word sequences is not feasible
- Phone-level LM, P instead of G

Decoder graph

Notations

H=HMM state graph, C=context-dependency, L=the lexicon, G=the language model

- Traditional ASR systems use HCLG (composition)
- Composing a graph over all possible word sequences is not feasible
- Phone-level LM, P instead of G-> no need for L

Decoder graph

Notations

H=HMM state graph, C=context-dependency, L=the lexicon, G=the language model

- Traditional ASR systems use HCLG (composition)
- Composing a graph over all possible word sequences is not feasible
- Phone-level LM, P instead of G-> no need for L
- LF-MMI uses HCP

Decoder graph

Minimalization is needed to reduce the size of the denominator graph

Decoder graph

Minimalization is needed to reduce the size of the denominator graph

1. Push the weights
2. Minimize the graph
3. Reverse the arcs and swap initial and final states

Decoder graph

Minimalization is needed to reduce the size of the denominator graph

1. Push the weights
2. Minimize the graph
3. Reverse the arcs and swap initial and final states

Additional trick: chunks of 1-1.5 seconds are used instead of the entire utterance (alignment is needed)

End-to-end version

- In LF-MMI tied bi-phone or triphone HMM states are used -> alignments needed

End-to-end version

- In LF-MMI tied bi-phone or triphone HMM states are used -> alignments needed
- E2E solution: monophones or full bi-phones

End-to-end version

- In LF-MMI tied bi-phone or triphone HMM states are used -> alignments needed
- E2E solution: monophones or full bi-phones
- Phone language model for the denominator graph is estimated using the training transcriptions

End-to-end version

- In LF-MMI tied bi-phone or triphone HMM states are used -> alignments needed
- E2E solution: monophones or full bi-phones
- Phone language model for the denominator graph is estimated using the training transcriptions
- Composite HMM (with self-loops) as the numerator graph
    - No prior alignment
    - No restriction on the self-loops

Tree-free full bi-phone

Separate HMM model for each and every possible pair of phonemes.

Tree-free full bi-phone

Separate HMM model for each and every possible pair of phonemes.

- The tree is not pruned at all -> no need for alignments
- Some bi-phones never occurs in the training data -> the network learns to ignore them.

Results

Table 5: *Comparison of WER for character-based end-to-end LF-MMI (EE-LF-MMI) and related methods on WSJ.*

Method	Parameters	Lexicon	LM	WER
Phone CTC [4]	–	Y	Word NG	7.3
Attention [35]	6.6M	Y	Word NG	6.7
EE-LF-MMI	8.2M	Y	Word NG	**4.1**
EE-LF-MMI no-SP	8.2M	Y	Word NG	5.3
EE-LF-MMI	8.2M	N	Char NG	5.4

Results

Method	Params	Lex.	LM	SW	CH	Tot†
CTC [32]	50M	N	Char NG	13.8	21.8	17.8
Attention* [33]	100M	N	N	8.6	17.8	13.2
RNN-T* [33]	120M	N	N	**8.5**	**16.4**	**12.5**
EE-LF-MMI	26M	N	Char NG	12.1	21.7	16.9
EE-LF-MMI	26M	N	Char RNN	12.0	21.9	17.0
CTC [32]	50M	Y	Word NG	11.3	18.7	15.0
RNN-T* [33]	120M	Y	Word NG	8.1	17.5	12.8
EE-LF-MMI	26M	Y	Word NG	9.3	18.6	14.0
EE-LF-MMI no-SP	26M	Y	Word NG	9.7	19.0	14.4
CTC [32]	50M	Y	Word RNN	10.2	17.7	14.0
EE-LF-MMI	26M	Y	Word RNN	8.0	17.6	12.8
Phone CTC [34]	–	Y	Word NG	10.2	16.5	13.3
Phone EE-LF-MMI	26M	Y	Word NG	8.6	15.5	12.0
Phone EE-LF-MMI	26M	Y	Word RNN	**7.5**	**14.6**	**11.0**

* These use data augmentation by adding background noise.
† The total eval2000 WER for CTC and Attention is the average of SW and CH (as it is not reported).

Summary

- MMI is a sequence-discriminative loss function
- The LF version tries to reduce the space and time complexity
- E2E LF-MMI requires a lot of modifications
  - Biphones
  - Composite HMM (numerator graph)
  - Phone language model

References

- Hadian, H., Sameti, H., Povey, D., Khudanpur, S. (2018) End-to-end Speech Recognition Using Lattice-free MMI. Proc. Interspeech 2018, 12-16, DOI: 10.21437/Interspeech.2018-1423.
- Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., Khudanpur, S. (2016) Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. Proc. Interspeech 2016, 2751-2755.
- On lattice free MMI and Chain models in Kaldi, https://desh2608.github.io/2019-05-21-chain/