Unsupervised Training of Discriminative Neural Network Lexicon Models

Jiahui Geng

jgeng@cs.rwth-aachen.de

February 9, 2018

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

Outline

Introduction

Literature

Baseline Framework

- Model
- Training Criterion
- Decoding

Neural Network Lexicon Models

- Generative Model
- Discriminative Model
- Training Criterion
  - Maximum Likelihood
  - Distance-based Criterion
- Conclusion and Outlook

Introduction

Why unsupervised learning?

- **Overcome the lack of reference translations**
- **Rich monolingual sentence resource**

Why neural network?

- **Easy integration of (possibly unlimited) source side contexts**
- **Implicit smoothing for rare words**
- **Effective representation via hidden layers**
- **Flexible model capacity: cover large vocabularies with low memory requirement**

Literature

[Tran & Bisk^{+} 16] Unsupervised Neural Hidden Markov Models, Workshop on Structured Prediction for NLP 2016

- Modeling HMM transmission and emission for generative direction in POS tagging

[Bourlard & Morgan 94] Connectionist Speech Recognition: A Hybrid Approach

- Integration of NN acoustic models into a HMM, in discriminative direction

[Graves et al. 12] Supervised sequence labelling with recurrent neural networks

- Combine LSTM with HMM models to form a hybrid sequence labelling system, HMM to model the sequence, neural network for localised classification

Problem Definition

Assumption

- 1:1 alignment between source and target words
- No reordering problem

Evaluation: token-level error rate

$$\text{Error rate} = \frac{\sum_{n=1}^{N} [\hat{c}_n \neq r_n]}{N}$$

Notation

- $\hat{c}_1^N$ = Translation output
- r_1^N = Reference output

Baseline Framework

Joint probability

$$p(c_1^N, x_1^N) = \prod_{n=1}^{N} p(c_n | c_{n-m+1}^{n-1}) p(x_n | c_n)$$

- m-gram target LM and a word-to-word lexicon model
- Resemble the m-th order hidden Markov model (HMM)
- LM pre-trained

Lexicon model: count-based table

$$p(x|c) = \theta_{x|c}$$

$$\forall c \sum_{x} \theta_{x|c} = 1$$

Baseline Framework

Training: Maximum likelihood

$$
\begin{aligned}
&\operatorname*{argmax}_{\theta} \left\{ p(x_1^N; \theta) \right\} \\
= &\operatorname*{argmax}_{\theta} \left\{ \sum_{c_1^N} p(x_1^N, c_1^N; \theta) \right\}
\end{aligned}
$$

Use EM algorithm as the iterative method:

$$
Q(\hat{\theta}, \theta) = \sum_{c_1^N} p(c_1^N | x_1^N; \theta) \cdot \log\, p(c_1^N, x_1^N; \hat{\theta})
$$

Decoding: Maximizing position-wise sum of marginal posterior

$$
\hat{c}_1^N = \operatorname*{argmax}_{c_1^N} \left\{ \sum_{n=1}^{N} p_n(c | x_1^N) \right\}
$$

Corpus Statistics

Spelling correction

- Recover natural strings from their corrupted versions

Train & Test	Running Words	64k
(Lexicon)	Vocabulary	27
Train	Running Words	275M
(LM)	Vocabulary	27

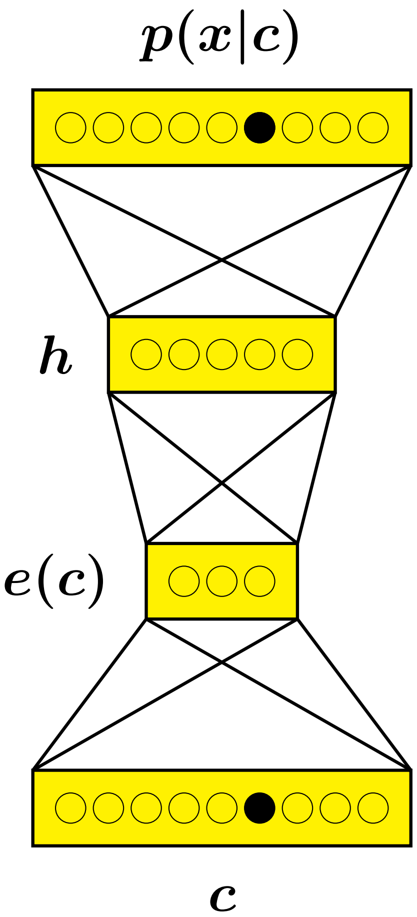
Eutrans

- Spanish $\rightarrow$ English task consists of common conversation scripts in hotels

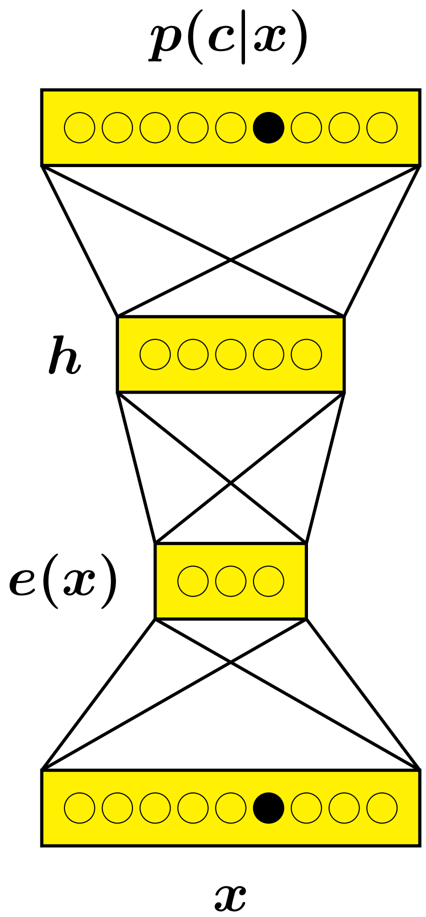
Train & Test	Running Words	85k
(Lexicon)	Vocabulary	679
Train	Running Words	4.2M
(LM)	Vocabulary	505

Neural Network Lexicon Models

Idea: replace the count-based table with ANN



Generative model



Discriminative model

Discriminative Model

Motivation

- Better performance than generative models in general
- Faster decoding because of reusing results in hypothesis expansions
- Easy integration of context words

Model

$$p(c_1^N, x_1^N) \approx \prod_{n=1}^{N} p(c_n | c_{n-m+1}^{n-1})^{\beta} \frac{p(c_n|x_n)}{p(c_n)^{\alpha}}$$

- Scaling parameters α , β
  1. Remedy for the incorrect normalization
  2. Determine the decoding performance: generate training data for M-step

Derivation of EM Algorithm

$$
\begin{aligned}
\theta^{(t+1)} &= \operatorname*{argmax}_{\theta} Q(\theta, \theta^{(t)}) \\
&= \operatorname*{argmax}_{\theta} \left\{ \sum_{c_1^N} p(c_1^N | x_1^N; \theta^{(t)}) \cdot \log p(c_1^N, x_1^N; \theta) \right\} \\
&= \operatorname*{argmax}_{\theta} \left\{ \sum_{c_1^N} p(c_1^N | x_1^N; \theta^{(t)}) \cdot \log \; p(c_1^N; \theta^{(t)})^{\beta} \prod_{n=1}^{N} \frac{p(c_n | x_n; \theta)}{p(c; \theta^{(t)})^{\alpha}} \right\} \\
&= \operatorname*{argmax}_{\theta} \left\{ \sum_{c_1^N} p(c_1^N | x_1^N; \theta^{(t)}) \cdot \sum_{n} \log \frac{p(c_n | x_n; \theta)}{p(c_n; \theta^{(t)})^{\alpha}} \right\} \\
&= \operatorname*{argmax}_{\theta} \left\{ \sum_{n} \sum_{c} p_n(c | x_1^N; \theta^{(t)}) \cdot \log p(c | x_n; \theta) \right\}
\end{aligned}
$$

EM Algorithm

E-Step

- **Calculate** $p(c;\theta^{(t)})$
- **Calculate** $p_n(c|x_1^N;\theta^{(t)})$ **with forward-backward algorithm**

M-Step

- **Calculate** $p(c|x;\theta^{(t+1)})$ **with cross-entropy as NN cost**

Prior Computation

1. From LM

- Use unigram probability
- Not updated during training

2. From lexicon model

$$
\begin{aligned}
p(c) &= \sum_x p(x, c) \\
&= \sum_x p(x) p(c|x) \\
&= \frac{1}{N} \sum_n p(c_n | x_n)
\end{aligned}
$$

- Updated from training results of each iteration

Experiments Results Analysis

Comparison between different models

Model		Prior	α	Error Rate [%]
ANN	$p(x\|c)$			17.3
	$p(c\|x)$	softmax	1	17.3
		lm	1	99.8
		lm	0.5	76.4
Table				17.4
Table(supervised)				16.9

- **Prior from the lexicon softmax output ensure the convergence of EM process**
- **Discriminative model achieves similar performance as generative model**

Why LM Prior Not Work?

Statistics of decoding results in spelling experiments with $p(c)$ fixed

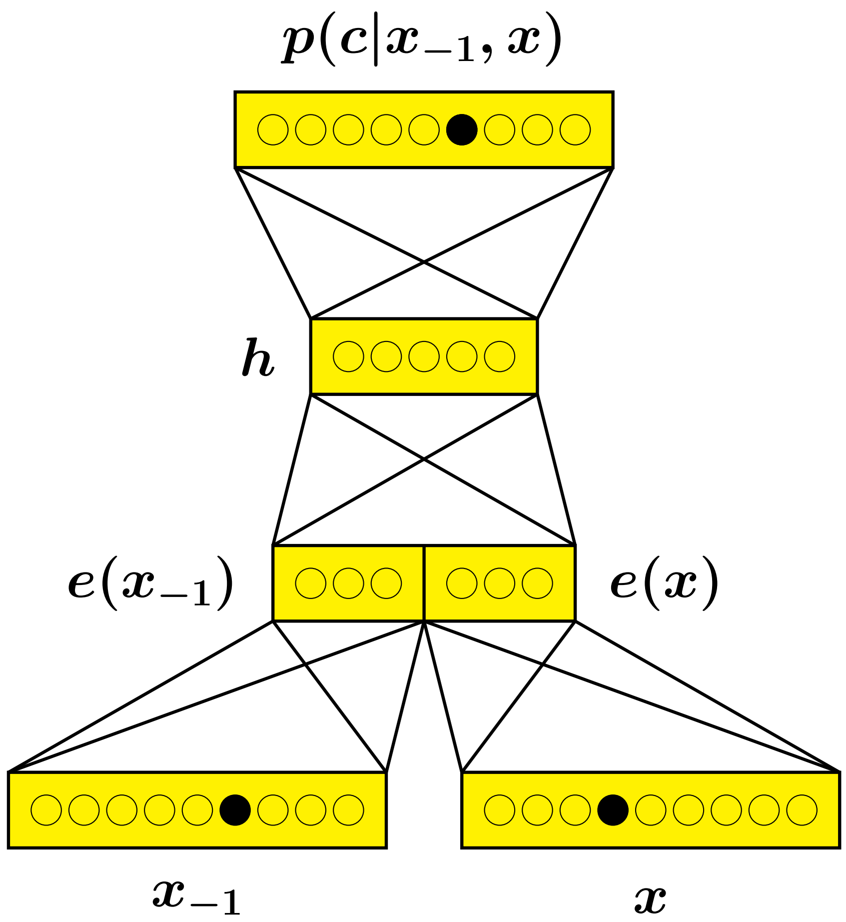
$\alpha : 1$	$e : 0.8\%$	$x : 99.2\%$	$--$	$--$	$--$	$--$	$--$
$\alpha : 0.5$	$t : 34.1\%$	$h : 19.0\%$	$: 22.7\%$	$a : 12.8\%$	$o : 2.2\%$	$e : 8.3\%$	$r : 0.9\%$
$\alpha : 0.5$	$the : 4413$		$that : 6207$		$to : 1286$		–

- Limited characters appear
- Specific words appear frequently

Analysis

- If $p(c)$fixed, maximum likelihood tends to decoding output with many repeating words with high LM scores
- If $p(c)$ not fixed, even if $p(c|x)$ favours repetitive words to make LM score high, the prior reduces the effect by dividing with a large prior probability

Context Modeling for ANN Lexicons



Discriminative direction with a predecessor context

Context Modeling for ANN Lexicons

Supervised learning results

Model	Context width	Spelling	Eutrans	Europarl
		Error rate[%]	Error rate[%]	Error rate[%]
$p(c\|x)$	0	17.0	2.56	45.8
	1	18.8	2.46	41.7
	2	19.7	2.19	38.5
	4	20.7	1.92	37.9

- Context model helps in supervised training for Eutrans and Europarl
- Context model does not help in spelling experiments for supervised training

Context Modeling for ANN Lexicons

$$
\begin{aligned}
\theta^{(t+1)} &= \operatorname*{argmax}_{\theta} Q(\theta, \theta^{(t)}) \\
&= \operatorname*{argmax}_{\theta} \left\{ \sum_{c_1^N} p(c_1^N | x_1^N; \theta^{(t)}) \cdot \log p(c_1^N, x_1^N; \theta) \right\} \\
&= \operatorname*{argmax}_{\theta} \left\{ \sum_{c_1^N} p(c_1^N | x_1^N; \theta^{(t)}) \cdot \sum_{n} \log \frac{p(c_n | x_n, x_{n-1}; \theta)}{p(c_n; \theta^{(t)})^{\alpha}} \right\} \\
&= \operatorname*{argmax}_{\theta} \left\{ \sum_{n} \sum_{c} p_n(c | x_1^N; \theta^{(t)}) \cdot \log p(c | x_n, x_{n-1}; \theta) \right\}
\end{aligned}
$$

- **Still uses maximum likelihood as training criterion**
- **Add source-side context in neural lexicon model**

Context Model Results

Unsupervised training results for Eutrans

Model		α	β	Error Rate [%]
ANN	$p(x\|c)$			28.3
	$p(c\|x)$	0.1	0.5	25.4
	$p(c\|x, x_{-1})$	0.05	0.1	28.1
	$p(c\|x, x_{-1}, , x_{+1})$	0.05	0.1	29.1
	$p(c\|x_1^N)$	0.05	0.5	29.0
Table				29.3
Table (supervised)				2.2

- More context = worse performance

Analysis

Statistics of decoding results in unsupervised training (Eutrans)

context width	’is there’	’by credit card’	’key to room’
0	0	194	1661
1	0	1570	1789
2	201	1570	4475

- **More fixed collocations, which come from LM, appear when adding context**
- **Maximum likelihood criterion is not suitable for context model**

Distance-based Criterion

$$\underset{\{p(c|x)\}}{\operatorname{argmin}} \left\{ \sum_{c_1^N} \frac{p(c_1^N)}{\prod_n p_n(c_n)} - \prod_n p_n(c_n|x_n) \right\}^2$$

- **Unfolding the formula we get:**

$$\sum_{c_1^N} \frac{(p^2(c_1^N))}{\prod_n p_n^2(c_n)} - 2 \cdot \underbrace{\sum_{c_1^N} p(c_1^N) \cdot \prod_n \frac{p_n(c_n|x_n)}{p_n(c_n)}}_{\text{maximum likelihood}} + \sum_{c_1^N} \prod_n p_n^2(c_n|x_n)$$

- **Assuming $p(c)$ and $p(c_1^N)$ fixed:**
  $\Rightarrow$ minimization of distance = maximum likelihood + quadratic absolute normalization:

$$\sum_c p_n^2(c|x_1^N) = 1.0$$

Quadratic Softmax

Motivation

- Different convergence process, may faster
- Easy to implement

Method

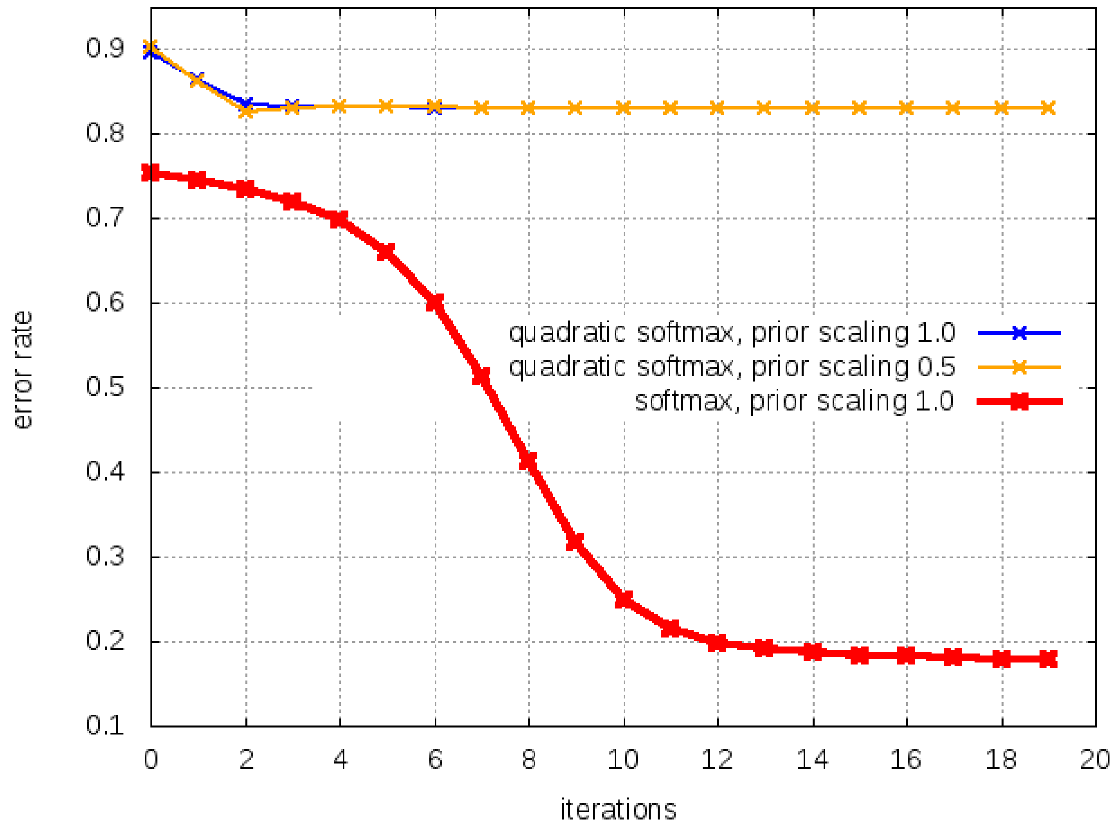
$$y = [y_1, ...y_c, ...y_C]$$

$$\text{Softmax} \Rightarrow \frac{e^{y_c}}{\sum_c e^{y_c}}$$

$$\text{Quadratic softmax} \Rightarrow \sqrt{\frac{e^{y_c}}{\sum_c e^{y_c}}} = \frac{e^{\frac{y_c}{2}}}{\sqrt{\sum_c e^{y_c}}}$$

Quadratic Softmax: Results

Use quadratic softmax for spelling correction



- ▶ **No difference when varying prior scaling parameter**
- ▶ **Quadratic softmax does not properly converge**

Another Distance Interpretation

$$\underset{\{p(c|x)\}}{\operatorname{argmin}} \left\{ \sum_{c_1^N} (p(c_1^N) - \prod_n \frac{p(c_n|x_n)}{p_n(c_n)})^2 \right\}$$

- **Unfolding the formula, we get:**

$$\sum_{c_1^N} p^2(c_1^N) - 2 \cdot \underbrace{\sum_{c_1^N} p(c_1^N) \cdot \prod_n \frac{p_n(c_n|x_n)}{p_n(c_n)}}_{\text{maximum likelihood}} + \sum_{c_1^N} \prod_n \frac{p_n^2(c_n|x_n)}{p_n^2(c_n)}$$

$\Rightarrow$ **minimization of distance = maximum likelihood + prior softmax normalization:**

$$\sum_c \frac{p_n^2(c|x_n)}{p_n^2(c)} = 1.0$$

Prior Softmax

Motivation

- Softmax output layer takes given prior into account
- Lexicon may learn to adapt to the prior

Constraint

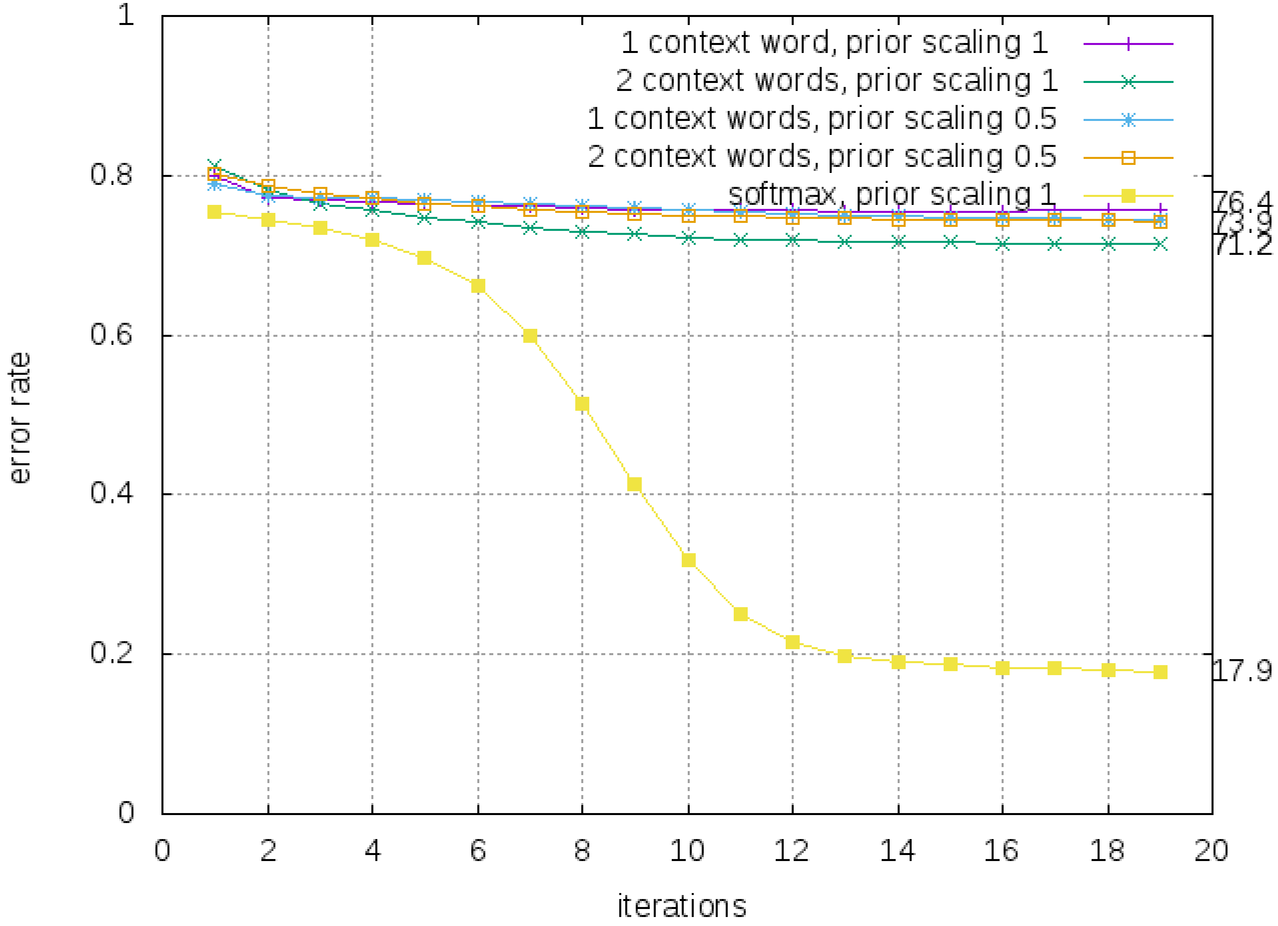
$$\sum_c \frac{p_n^2(c|x_n)}{p_n^2(c)} = 1.0$$

Implementation

Prior softmax $\Rightarrow \sqrt{\frac{e^{y_c} \cdot p^2(c)}{\sum_c e^{y_c}}} = \frac{e^{\frac{y_c}{2}}}{\sqrt{\sum_c e^{y_c}}} \cdot p(c)$

Prior Softmax Experiments

Use prior softmax for spelling correction



- **Seems to converge for this training process but the performance not satisfying**

Outlook

Conclusion

- Similar performance for both generative and discriminative models w/o context
- Worse performance when in introducing contextual information
- Modified output layer not satisfying

Outlook

- Modify the training criterion

Thank you for your attention

Jiahui Geng

jgeng@cs.rwth-aachen.de

http://www.hltpr.rwth-aachen.de/

References

[Bourlard & Morgan 94] H. Bourlard, N. Morgan: *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[Doetsch & Kozielski$^+$ 14] P. Doetsch, M. Kozielski, H. Ney: Fast and robust training of recurrent neural networks for offline handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pp. 279–284. IEEE, 2014.

[Graves & Mohamed$^+$ 13] A. Graves, A.r. Mohamed, G. Hinton: Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649. IEEE, 2013.

[Graves et al. 12] A. Graves et al.: *Supervised sequence labelling with recurrent neural networks*, Vol. 385. Springer, 2012.

[Mohamed & Dahl$^+$ 12] A.r. Mohamed, G.E. Dahl, G. Hinton: Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 14–22, 2012.

[Tran & Bisk+ 16] K. Tran, Y. Bisk, A. Vaswani, D. Marcu, K. Knight: Unsupervised Neural Hidden Markov Models. *arXiv preprint arXiv:1609.09007*, Vol., 2016.

[Zeyer & Doetsch+ 17] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, H. Ney: A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pp. 2462–2466. IEEE, 2017.