The RWTH Large Vocabulary Arabic Handwriting Recognition System

Mahdi Hamdani[1], Patrick Doetsch[1], Michal Kozielski[1], Amr El-Desoky Mousa[1]
Hermann Ney[1,2]
[1] Human Language Technology and Pattern Recognition Group - RWTH Aachen University, Germany
[2] Spoken Language Processing Group, LIMSI CNRS, Paris, France
{surname}@cs.rwth-aachen.de

Abstract—**This paper describes the RWTH system for large vocabulary Arabic handwriting recognition. The recognizer is based on Hidden Markov Models (HMMs) with state of the art methods for visual/language modeling and decoding. The feature extraction is based on Recurrent Neural Networks (RNNs) which estimate the posterior distribution over the character labels for each observation. Discriminative training using the Minimum Phone Error (MPE) criterion is used to train the HMMs. The recognition is done with the help of n-gram Language Models (LMs) trained using in-domain text data. Unsupervised writer adaptation is also performed using the Constrained Maximum Likelihood Linear Regression (CMLLR) feature adaptation. The RWTH Arabic handwriting recognition system gave competitive results in previous handwriting recognition competitions. The used techniques allows to improve the performance of the system participating in the OpenHaRT 2013 evaluation.**

Keywords*—*RWTH Arabic Handwriting Recognition System, Hidden Markov Models, Recurrent Neural Networks

I. Introduction

Handwriting recognition is a research field with a growing complexity. The problems in the modeling and recognition of handwriting are very similar to the problems of the well developed speech recognition technology. A couple of open source systems are available and are used for handwriting recognition like the HTK Toolkit [1] and Kaldi [2].

Hidden Markov Models (HMMs) and Artificial Neural Networks (ANN) based recognition systems are the most successful recognizers in the field of Arabic handwriting recognition. The system presented in [3] had the first position in the 2009 Arabic handwriting recognition competition [4]. The classifier is a Multi-Dimensional Long Short Term Memory which is a novel type of Recurrent Neural Networks (RNNs). The sequence classification is performed using the Connectionist Temporal Classication (CTC) which is a special output layer. The RWTH Arabic handwriting recognition system had competitive results in previous competitions. The system had the second and first positions respectively in the 2010 and 2011 editions of the Arabic handwriting recognition competitions [5], [6]. The RWTH system is based on discriminatively trained HMMs combined with a Long Short Term Memory (LSTM) network for feature extraction.

Our HMM system is based on a publicly available state-of-the-art large vocabulary continuous speech recognition framework (RWTH-ASR or RASR) which has been designed for the special requirements of research applications and supports for grid-computing [7]. The RWTH handwriting recognition system is an adapted version of the RASR. Additional packages are implemented to allow feature extraction from images. Figure 1 presents an overview of the recognition system architecture.

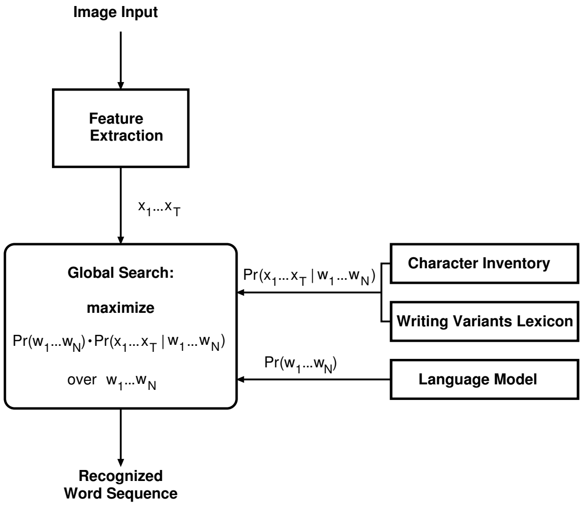

Fig. 1. RWTH OCR architecture

This paper describes the RWTH Arabic handwriting recognition system participating in the OpenHaRT 2013 evaluation [8].

The rest of this paper is organized as follows: Section II describes the Recurrent Neural Network based feature extraction method followed by the visual modeling detailed in Section III. The used vocabulary and Language Model are presented in Section IV. The decoding techniques are presented in Section V. Finally the results are presented in Section VI followed by the conclusions and future work.

II. Feature Extraction

A. Preprocessing

The first step in any pattern recognition system is the data preparation (or preprocessing) and the feature extraction. As we mentioned above the recognizer is based on HMMs (1-Dimensional), which have a limitation regarding image modeling (2-Dimensional). In fact, vertical image distortions have to be processed carefully. One way to deal with this problem is the vertical repositioning [9]. This is done by computing the center of gravity of a sliding window scanning the image from right to left (direction of writing). Afterwards the window is repositioned such that its center will be adjusted to the center of gravity.The features are pixel gray values extracted from a sliding window of size $9x30$ pixels with a maximum overlap (1 pixel shift). The 270 features are reduced to 35 using Principal Component Analysis (PCA). The used feature extraction technique is illustrated in Figure 2.

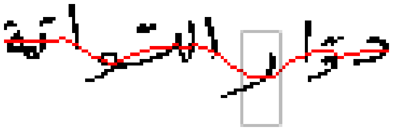

Fig. 2. Feature Extraction using repositioned sliding window

B. RNN Features

Without any preprocessing of the input images, we extract simple appearance-based image slice features x_t at every time step $t = 1, \cdots, T$ which are augmented by their temporal derivatives in horizontal direction $\Delta = x_t - x_{t-1}$.

These augmented raw slice features $X_t = [x_t, \Delta]$ together with their corresponding state alignments are then processed by a hierarchical framework originally described in [10]. Depending on the MLP hierarchy, preprocessing, and postprocessing operations, several feature sets can be generated. In order to incorporate temporal and spatial context into the features, we concatenate consecutive features in a sliding window, where the MLP outputs are later reduced by a PCA or LDA transformation (see Figure 3).

Artificial neural networks (ANNs) in a tandem HMM approach combine the discriminative parameter estimation of the ANN with the sequence modeling ability of the HMM [11]. Training the ANN requires each observation $\vec{o}_t \in \mathbb{R}^D$ at time step t in the training data to be aligned to a character label of its transcription. In order to obtain this labeling a previously trained Gaussian HMM (GHMM) applied to the training data in the forced alignment mode. Then the ANN is trained on the labeled observations. Recurrent ANN architectures (RNNs) provide a natural way to deal with contextual information over time [12]. In the presented experiments we use bidirectional Long-Short-Term-Memory (LSTM) RNNs, which lead to significant improvements in handwriting recognition [13]. The LSTM is trained in a frame-based approach with a softmax output layer using Backpropagation through time (BPTT).

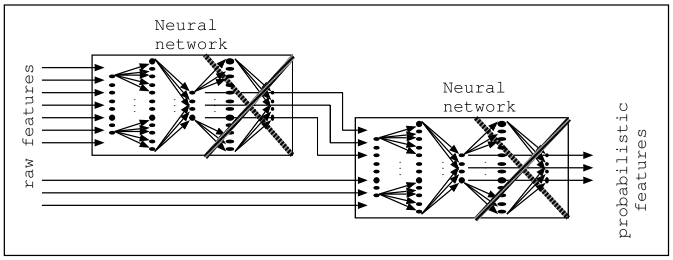

Fig. 3. Hierarchical MLP network for feature extraction

The trained LSTM it is used to calculate a posterior distribution over the character labels for each observation. In a tandem HMM approach the posterior estimates are considered as observations to train a new GHMM in order to perform the sequence modeling. See Figure 4 for an illustration.

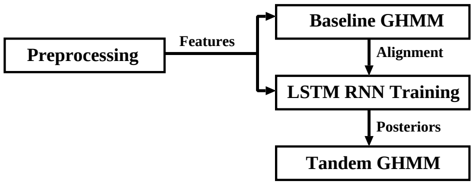

Fig. 4. The three steps of the LSTM Tandem HMM approach: An alignment obtained by a baseline HMM is used to train the LSTM. Afterwards the posterior estimates are used as observations to train the Tandem GHMM

Two combination schemes are possible for the final HMM/ANN system. In the tandem approach, the posterior probabilities are used as features to train the HMM from scratch.

$$x_t \mapsto \phi(p(s_t, w|x_1^T)) \in \mathbb{R}^n \tag{1}$$

These posterior probabilities can be used directly as state emission probability of the HMM. In this case, we speak about the hybrid approach which allows to reduce the training time.

$$p(x_t|s_t, w) \overset{!}{=} \frac{p(s_t, w|x_t)}{p(s_t, w)^\alpha} \tag{2}$$

with α: Priori scaling factor.

III. Visual Modeling

A. Model Description

In off-line handwriting recognition, we are searching for an unknown word sequence $w_1^N := w_1, \ldots, w_N$, for which the sequence of features $x_1^T := x_1, \ldots, x_T$ fits best to the trained models. We maximize the posterior probability $p(w_1^N|x_1^T)$ over all possible word sequences w_1^N with unknown number of words N. This is described by the Bayes' decision rule:

$$x_1^T \rightarrow \hat{w}_1^N(x_1^T) = \arg\max_{w_1^N} \left\{ p^\kappa(w_1^N) p(x_1^T|w_1^N) \right\} \tag{3}$$

with κ being a scaling exponent of the language model.

In this work, we use a writing variant model refinement [14] of our visual model

$$p(x_1^T|w_1^N) = \max_{v_1^N|w_1^N} \left\{ p^\alpha_{\Lambda_\mathrm{v}}(v_1^N|w_1^N) p^\beta_{\Lambda_\mathrm{e,t}}(x_1^T|v_1^N, w_1^N) \right\} \tag{4}$$

with v_1^N a sequence of unknown writing variants, α a scaling exponent of the writing variant probability depending on a parameter set Λ_v, and β a scaling exponent of the visual character model depending on a parameter set $\Lambda_\mathrm{e,t}$ for emission and transition model.

The used model is a Gaussian HMM with a Bakis topology, i.e. each state has a transition to the two next states. Each Gaussian is shared between two successive states. This property guaranty that each Gaussian is visited at least once. The used topology with 6 states is presented in Figure 5.

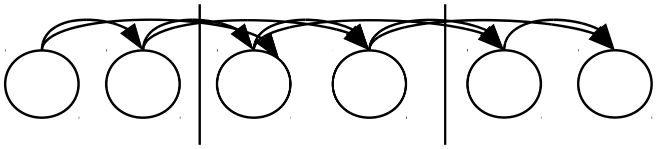

Fig. 5. Gaussian HMM Topology

The position of a character in the word as well as its context (previous and next characters) are important to define its shape. Arabic handwriting contains also diacritics and special ligatures. This basic information concerning Arabic handwriting style has to be carefully handled to build an Arabic handwriting recognition system.

Generally, an Arabic character can be written in different ways depending on its position. We can find 1 to 4 variants for each character. Basically, the Arabic handwriting contains 28 letters. If we take into account position dependency we can reach more than 100 different character forms.

B. Context Dependent Modeling

The basic idea is to model characters within their context which is now a standard approach in speech recognition systems. The "triphones" are widely used in speech recognition technology. This technique is not yet used in all systems in Arabic handwriting recognition because of the lack of sufficiently big open access databases in this field. One of the problems in the context dependent modeling is that the number of possible triphones is huge, some triphones are not seen in the training. The number of parameters to estimate is very high wich can be overcome by state tying [15].

There are multiple proposed methods for state tying in the literature, the most successful is to use cart trees. A cart tree is a binary tree, the nodes are tagged with questions and the leaves are tagged with class labels. In our implementation questions are related to the shape of the character. The mixture label at the leaf identifies the mixture model for the triphone state.

The objective is to tie the states which are similar. Decision trees are binary trees in which the nodes are tagged with questions and the leaves are tagged with class labels. The questions concern the data to be classified using the tree. The questions are generally predefined using prior knowledge about the data. There are standard questions used in speech recognition systems based on phonetic properties (e.g. " Is the left context a vowel? "). The phonetic classes are predefined in the system.

In this work, visual classes are defined using shape properties. Some characters in Arabic language are very similar like ـر and ـز, د and ذ, etc. The lexicon characters are divided into classes, some examples are presented in Table I.

C. Discriminative Training

The Minimum Phone Error (MPE) criterion is defined as the (regularized) posterior risk based on the error function $E(V, W)$, which is probably the training criterion of choice in Large Vocabulary Continuous Speech Recognition (LVCSR). For MPE, the loss function to be minimized is described by:

$$L^{(\mathrm{MPE})}[p_\Lambda(X_r, \cdot), W_r] = \sum_{W \in \cdot} E(W, W_r) \frac{p_\Lambda(X_r, W_r)^\gamma}{\sum\limits_{V} p_\Lambda(X_r, V)^\gamma}, \quad (5)$$

which is based on the error function $E(V, W)$ like for example the approximate phone error [16]. In OCR, a phoneme unit usually corresponds to a character if words are modeled by character sequences.

TABLE I. EXAMPLES OF VISUAL CLASSES

Types	Examples of Characters	Images
Small Ascenders	ـذ, ـشـ, ـسـ	
Descenders	ـز, ـر, ـو	
Occlusions	ـفـ, ة, ـهـ	

IV. Language Modeling

Based on the available training data for constrained and unconstrained tasks, n-gram language models (LMs) were estimated using [17], smoothed by the Modified Kneser-Ney method.

The LM training text is first of all normalized using the following preprocessing steps. Indian digits, which are widely used in Arabic text, are mapped to Arabic (the lexicon contains only Arabic digits). The numbers are reversed including optional decimal points and then the digits of the numbers are separated by spaces. Punctuation and special characters are separated from the words. These steps allow to reduce noisy text and to have a better distribution of the probabilities.

The recognition of out-of-vocabulary (OOV) words is possible using the method described in [18]. The words of the text corpus are decomposed using the MADA+TOKAN toolkit [19]. The M most frequent full-words are left and the decomposed form of the remaining text is used. The final vocabulary is defined by selecting again the N most frequent words. The selected vocabulary contains new elements which are prefixes, suffixes and also new words. The prefixes and suffixes are tagged with a special marker ("+"). The recognition of unknown words is possible by the combination of the different vocabulary elements. In case of a prefix/suffix recognition, the marker is removed and the successive sub-words are combined.

V. Decoding

A. Decoder

The used decoder is based on the history conditioned lexical tree (HCLT) search [20]. HCLT search is a one-pass dynamic programming algorithm which uses a pre-compiled lexical prefix tree as representation of the pronunciation dictionary. The search space is constructed dynamically by integrating parts of the LM as needed during search. The decoder can deal with huge vocabularies and complex language models in a memory efficient way.

B. Writer Adaptation

During recognition, in a first pass, we estimate in an unsupervised way the writer clustering. This step is done using the Bayesian Information Criterion (BIC) stopping condition [21]. The clusters are supposed to be the unknown writers or their writing styles. In the second pass, we use these clusters for a writer dependent estimation of the Constrained Maximum Likelihood Linear Regression (CMLLR) based feature adaptation. CMLLR consists of normalizing the features by the use of a maximum likelihood estimated affine transform.

VI. Results

We present in this section the results of the RWTH system on two tasks. The first part will concern the results of the RWTH system on the Arabic handwriting recognition competition. After that, the results of the system on the OpenHaRT data are presented.

The following feature extraction parameters are applied for both systems. A scaling to 30 pixels height was performed keeping the aspect ratio. Then, the vertical repositioning method was applied and the features were reduced by PCA to 35 components using a sliding window of size 9. A 12-state baseline GHMM with six separate Gaussian mixture Models was trained on the features and used to generate the alignment for the RNN training.

The first contest is the Arabic handwriting recognition competition which is using the IfN/ENIT dataset [6]. This dataset contains 32492 images of Arabic handwritten words (Tunisian town/village names). The database is divided in 5 sets (a-e) with an equitable distribution in the number of examples.

The neural networks are trained using the RNNlib toolkit [22]. The LSTM consists of two hidden layers with 100 and 200 nodes respectively resulting in about $785k$ weights. Convergence was detected on a separate validation set containing 20% of the training data. A tandem GHMM with the same topology as the baseline GHMM was trained on the 121 posterior estimates of the LSTM, which were reduced by PCA to 72 components.

Table II presents the results of the RWTH system at the ICDAR 2011 competition. The proposed system is ameliorated by the repositioning technique described in Section II. The results show that developed recognizer outperforms the state of the art systems.

The second dataset is provided by the MADCAT[1] (Multilingual Automatic Document Classification Analysis and Translation) program within the context of the OpenHaRT 2013 evaluation. The data consists of more than $40k$ handwritten pages with text chosen from web forums and newspapers. Table III gives statistics detailing the used data.

Table IV presents the results of the RWTH system on the OpenHaRT constrained task. The sub-lexical approach described in [18] allows the improvement of the baseline system.

TABLE II. Results of the RWTH handwriting recognition system on the Arabic handwriting recognition competition (IfN/ENIT database)

System	WER [%]	CER [%]
GHMM, MLP Tandem (ICDAR'11)	**5.9**	**4.7**
GHMM, MLP Hybrid (ICDAR'11)	10.3	8.1
GHMM	13.11	10.6
+ Repo.	6.4	4.6
GHMM, LSTM Tandem	7.2	5.6
+ Repo., [9]	**4.8**	**3.7**
BHMM, UPV, [9]	6.2	-
MD-LSTM, TUM, [3]	6.6	-

TABLE III. OpenHaRT dataset statistics

	Train set	Dev set
# of pages	42,148	470
# of paragraphs	182,879	1,832
# of words	4,361,056	48,832
# of characters	23,324,011	266,121

TABLE IV. Results of the RWTH handwriting recognition system on the OpenHaRT constrained task

System	Vocabulary size	WER [%]	CER [%]
Baseline	100k	27.4	10.9
Sub-lexical approach	94k	**26.8**	**10.1**

The LSTM based feature extraction is using a network of three hidden layers with 50, 100 and 200 nodes respectively resulting in about $920k$ weights. Convergence was detected on a separate validation set containing 10% of the training data. A tandem GHMM with the same topology as the baseline GHMM was trained on the 229 posterior estimates. It's important to mention that the tandem system was trained on the activations of the first hidden layer in both directions. The 100 dimensional vector was extracted and reduced to 20 components by PCA.

Additional text data is used for the LM training collected from newspapers and web-forums. The LDC Arabic Gigaword Second and Third Edition are also included in the training data.

The results on the unconstrained task of the OpenHaRT evaluation are presented in Table V. The LSTM features allows to improve the system performance with 6% absolute. Discriminative training gives also 3% of absolute improvement.

TABLE V. Results of the RWTH handwriting recognition system on the OpenHaRT unconstrained task

System	WER [%]	CER [%]
GHMM CI	33.2	15.4
GHMM CD	25.9	10.1
+RNN Features	19.9	5.9
+MPE	17.0	4.5

VII. Conclusions and Future Work

We presented in this paper the RWTH Arabic handwriting recognition system. The recognizer is based on context dependent HMMs with different state of the art methods used for training and decoding. Feature extraction is performed using Long Short Term Memory (LSTM) RNNs. Minimum Phone Error (MPE) based discriminative training is applied

[1] http://www.itl.nist.gov/iad/mig/tests/madcat/index.html

to improve the system accuracy. Modeling takes into account the Arabic language specificities by designing shape based questions for state tying. The system gave competitive results in previous international Arabic handwriting recognition competitions. The used techniques allows the amelioration of the baseline system using the OpenHaRT dataset.

The used preprocessing (window repositioning) and features (pixel values) in this work are very simple. The next step in this direction is to include a more complete feature extraction with advanced preprocessing like underline removal and image normalization. Character based language models can be used to deal with the out of vocabulary words. Connectionist-Temporal-Classification (CTC) layers can be used in the training and compared with the tandem combination approach used in this system.

Acknowledgment

This work was partially supported by a Google Research Award and by the Quaero Program, funded by OSEO, French State agency for innovation. H. Ney was partially supported by a senior chair award from DIGITEO, a French research cluster in Ile-de-France.

References

[1] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.4*. Cambridge, UK: Cambridge University Engineering Department, 2006.

[2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[3] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. MIT Press, 2009, pp. 545–552.

[4] H. El Abed and V. Märgner, "ICDAR 2009-Arabic handwriting recognition competition," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 14, no. 1, pp. 3–13. [Online]. Available: http://dx.doi.org/10.1007/s10032-010-0117-5

[5] V. Märgner and H. El Abed, "ICFHR 2010 - Arabic handwriting recognition competition," in *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2010, pp. 709–714.

[6] ——, "ICDAR 2011 - Arabic handwriting recognition competition," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2011, pp. 1444–1448.

[7] D. Rybach, S. Hahn, P. Lehnen, D. Nolden, M. Sundermeyer, Z. Tüske, S. Wiesler, R. Schlüter, and H. Ney, "Rasr - the rwth aachen university open source speech recognition toolkit," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Hawaii, USA, Dec. 2011.

[8] A. Tong, M. Przybocki, V. Maergner, and H. E. Abed, "Nist 2013 open handwriting recognition and translation (openhart'13) evaluation," in *Proceedings of the NIST 2013 Open Handwriting and Recognition Workshop*, Washington, USA, Aug 2013, in press.

[9] P. Doetsch, M. Hamdani, A. Giménez, J. Andrés-Ferrer, A. Juan, and H. Ney, "Comparison of bernoulli and gaussian hmms using a vertical repositioning technique for off-line handwriting recognition," Bari, Italy, Sep. 2012, pp. 3–7.

[10] F. Valente, J. Vepa, C. Plahl, C. Gollan, H. Hermansky, and R. Schlüter, "Hierarchical neural networks feature extraction for lvcsr system," Antwerp, Belgium, Aug. 2007, pp. 42–45.

[11] H. Hermansky, D. Ellis, and S. Sharma, "Tandem connectionist feature stream extraction for conventional HMM systems," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2000, pp. 1635–1638.

[12] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[13] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007.

[14] P. Dreuw, D. Rybach, C. Gollan, and H. Ney, "Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, Barcelona, Spain, Jul. 2009.

[15] K. Beulen, E. Bransch, and H. Ney, "State-tying for context dependent phoneme models," in *European Conference on Speech Communication and Technology*, vol. 3, Rhodes, Greece, Sep. 1997, pp. 1179–1182.

[16] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, Cambridge, England, 2004.

[17] A. Stolcke, "SRILM - an extensible language modeling toolkit," vol. 2, Denver, Colorado, USA, Sep. 2002, pp. 901 – 904.

[18] M. Hamdani, A. El-Desoky, and H. Ney, "Open vocabulary Arabic handwriting recognition using morphological decomposition," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, Washington DC, USA, Aug. 2013.

[19] O. R. Nizar Habash and R. Roth, "Mada+tokan: A toolkit for Arabic tokenization, diacritization, morphological disambiguation, pos tagging, stemming and lemmatization," in *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, K. Choukri and B. Maegaard, Eds. Cairo, Egypt: The MEDAR Consortium, April 2009.

[20] H. Ney and S. Ortmanns, "Progress in dynamic programming search for lvcsr," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1224–1240, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1109/5.880081

[21] M. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230898900432

[22] A. Graves, "Rnnlib: A recurrent neural network library for sequence learning problems," http://sourceforge.net/projects/rnnl/.