Using ASR methods for OCR

Ashish Arora[1*], Chun Chieh Chang[1*], Babak Rekabdar[4], Daniel Povey[1,2], David Etter[2], Desh Raj[1], Hossein Hadian[3], Jan Trmal[1], Paola Garcia[1], Shinji Watanabe[1], Vimal Manohar[1,2], Yiwen Shao[1], Sanjeev Khudanpur[1,2]
[1]Center for Language and Speech Processing,
[2]Human Language Technology Center of Excellence,
Johns Hopkins University, Baltimore, USA
[3]Department of Computer Engineering, Sharif University of Technology, Iran
[4]School of Mathematics, Statistics and Computer Sciences, College of Science, University of Tehran, Iran
{ashish.arora.88888, chang.jonathan.1995, dpovey, etterd, r.desh26, hn.hadian, jtrmal, leibny, vimal.manohar91, sywcs007wow}@gmail.com
babak.rekabdar@ut.ac.ir, shinjiw@ieee.org, khudanpur@jhu.edu

***Abstract*—Hybrid deep neural network hidden Markov models (DNN-HMM) have achieved impressive results on large vocabulary continuous speech recognition (LVCSR) tasks. However, the recent approaches using DNN-HMM models are not explored much for text recognition. Inspired by the current work in automatic speech recognition (ASR) and machine translation, we present an open vocabulary sub-word text recognition system. The sub-word lexicon and sub-word language model (LM) helps in overcoming the challenge of recognizing out of vocabulary (OOV) words, and a time delay neural network (TDNN) and convolution neural network (CNN) based DNN-HMM optical model (OM) efficiently models the sequence dependency in the line image. We present results on 12 datasets with training data varying from 6k lines to 600k lines. The system is built for 8 languages, i.e., English, French, Arabic, Chinese, Farsi, Tamil, Russian, and Korean. We report competitive results on several commonly used handwritten and printed text datasets.**

***Index Terms*—OCR, ASR, open vocabulary, lattice-free MMI, BPE**

I. Introduction

Text recognition is the task of transcribing printed text or handwritten text line images. Since both text recognition and automatic speech recognition (ASR), convert a sequence of vectors into text, approaches used for transcription of text line image are also similar to speech recognition. Hybrid hidden Markov model (HMM) [1] based speech/text recognizer is one such approach that is common in both ASR and text recognition. Recently, the performance of HMM-based speech recognition systems is further boosted by using a sequence discriminative objective called lattice-free maximum mutual information (LF-MMI) [2]. In addition, although text recognition and ASR are inherently open vocabulary tasks, many text recognition and ASR systems still rely on a fixed vocabulary, thereby making it a closed vocabulary setup. A closed vocabulary setup usually faces the challenge that the word which is not present in the lexicon (OOV: out-of-vocabulary word), cannot be recognized by the system.

Hence, recognition of rare words can require a highly specific vocabulary to achieve good performance in the closed vocabulary scenario. Previous approaches addressed this issue by simultaneously hypothesizing an unknown word and identifying it as a sequence of characters. The system provides both a word probability and the probability of an unknown word [3]. Another approach is to have sub-words in lexicon and LM so that it can recognize OOV words as a sequence of sub-words [4]. Recently, sub-word/hybrid LM based approaches [5] have started gaining popularity in text recognition community. However, the focus of previous works has been to build an open vocabulary sub-word setup with limited latency.

In this work, we present a DNN-HMM-based offline text recognition system trained with LF-MMI objective function. An open-vocabulary word-based system and a sub-word based system were built and compared at different OOV rates. These setups are implemented in the open-source ASR toolkit Kaldi [6] and are made publicly available.

A. Contributions

We explored adopting state-of-the-art acoustic models used in ASR for text recognition and implemented a sub-word lexicon and sub-word LM based system. The contributions of this paper are summarized below.

- We adapt a low latency CNN-TDNN-HMM acoustic model from ASR trained with LF-MMI objective function for text recognition. The optical model (OM) is explained in Section II-A.
- We implement different data augmentations and perform language specific modifications such as decomposition and bidirectional reordering. The details about the data augmentations are mentioned in Section II-B.
- To overcome the challenge of OOV words, we implement a new sub-word based algorithm for text recognition. The open vocabulary setup is explained in Section II-C.
- We obtained state-of-the-art results on several commonly used datasets in the literature. We have open sourced the scripts for all setups.

The rest of the paper is organized as follows. In Section II, we briefly describe our HMM-based text recognition system. In Section III, we describe the experimental setup and present

*Equal Contribution

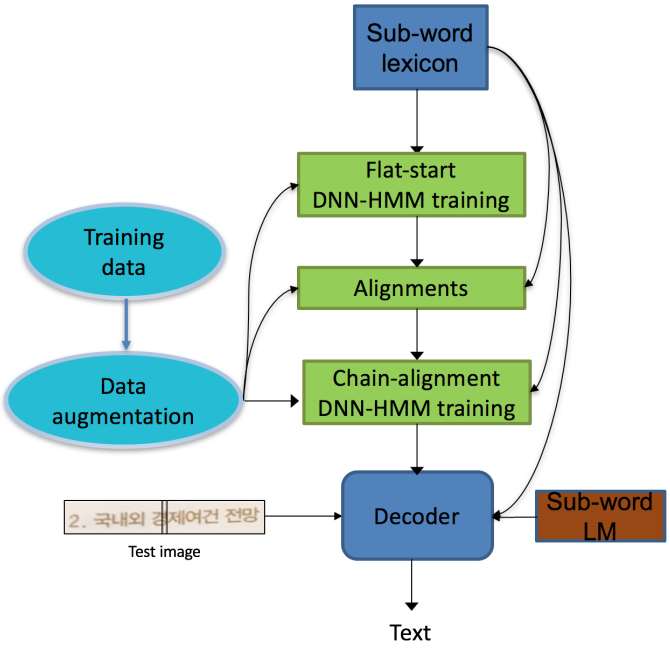

Fig. 1. **Text recognition setup with four major components: optical model (green), sub-word lexicon (top dark blue), sub-word language model (brown) and data augmentation (bottom light blue) setup**

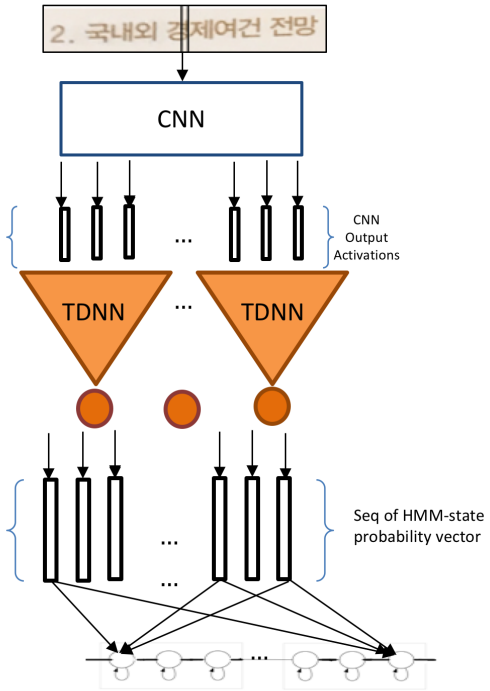

Fig. 2. **The DNN-HMM optical model. DNN is used to estimate the emission probabilities of HMM states. OM provides the likelihood of the sequence of observations for a given word sequence.**

the results in Section IV. The conclusion is presented in Section V.

II. Text Recognition with Hidden Markov Models

Our HMM-based text recognizer setup is shown in Figure 1. It has three main components: an optical model, an image augmentation setup, and a sub-word LM and sub-word lexicon. A comprehensive review of the traditional hybrid HMM can be found in [1].

A. Optical model

A DNN helps in learning a complex representation from a raw pixel feature vector and hence can work seamlessly with diverse datasets. In a DNN-HMM system, a neural network is used to estimate the emission probability for all the states of a context-dependent character-based HMM. A DNN-HMM based OM is shown in Figure 2. While building a DNN-HMM system for text recognition, we made the following considerations.

- We used a CNN-TDNN-HMM setup because a CNN can extract relevant features from the raw pixels and the following TDNN [7] can efficiently model the context on either side of the feature vector.
- We use L2-regularization, batch normalization, and scheduled dropout for faster training and to avoid over-fitting.
- Since the features extracted from line images and features extracted from an acoustic recording files have different properties, we require different values for the tolerance, frame sub-sampling factor and chunk-width hyper-parameters.

To model the sequential dependency, we chose a TDNN over LSTM. This is because TDNNs [7] can model with low latency and low computational complexity and performed better than LSTMs in our experiments. A flat-start [8] [9] model with sequential loss function was used for training the DNN-HMM system, after which the training data was forced-aligned for the second pass CNN-TDNN-HMM model. We use a LF-MMI objective function instead of a frame-wise cross-entropy objective function as the former has been shown to give significant improvements in WER over the latter. In both word-based and sub-word based open vocabulary setups, the OM still works at the context dependent character level. To get the likelihood of a word or word sequence, the HMMs of the context dependent characters are concatenated with the help of the lexicon.

B. Data Augmentation

Our idea behind applying data augmentations is to make training data similar to the validation data. The SLAM dataset has comparatively lower resolution images than the YOMDLE dataset. Since we use YOMDLE for training and SLAM for testing, training images were randomly scaled down and then scaled up for data augmentation. Since Rimes dataset has an axis-aligned bounding box, the line images contain surrounding lines. Hence during training, bounding boxes were expanded to simulate this effect. For example, if a bounding box coordinates are (0,0), (0,100), (100,0), (100,100),

a possible expanded bounding box coordinate can be (0,0), (0,120), (120,0), (120,120). Both these bounding boxes will be resized to the same height while maintaining the aspect ratio. Similarly, deslanting and deskewing [10] was applied to the IAM line images for data augmentation. Currently, the image augmentation is part of some setups but not all.

C. Lexicon and language model

1) Open vocabulary word-based setup: An approach to building an open vocabulary system is to simultaneously hypothesize an unknown word and identify it as a sequence of characters. This approach needs two LMs, namely a word-based LM for hypothesizing the unknown word and a character-based LM for recognizing the word as a sequence of characters. This approach has been widely studied and adopted by the text recognition community. Many setups have used this unknown word decoding approach to show significant improvement in performance. A comprehensive review of this method can be found in [3].

2) Open vocabulary sub-word based setup: Implementing a sub-word open vocabulary system in Kaldi is significantly different than the implementation in other toolkits. Below, we explain the word-segmentation and sub-word implementation in Kaldi.

Byte pair encoding (BPE) [11] is used to create sub-words in a language independent, data-driven way. BPE compression is a greedy algorithm and requires a training data to learn sub-words from words. For each iteration, it greedily replaces the most frequent pair of characters with a new character symbol. Since at each iteration a new symbol is produced, the number of symbols (vocabulary size) can be controlled by fixing the number of iterations.

While building a sub-word based text recognition setup, we made the following considerations:

- We retained singletons in the lexicon to allow all possible output character sequences.
- Since it is difficult to capture the character position information in a sub-word, we turned off the character position information flag.
- We add a special space symbol in front of the sub-words to distinguish word boundaries from sub-word transitions.
- We empirically select a sub-word vocabulary size of 700 based on the system performance across diverse datasets.

BPE is applied to the training text and corpus text to convert the text at word level into text at sub-word level. A sub-word lexicon is created using this sub-word text. The decoder finds the best sub-word sequence for a given test line image. During scoring, to combine sub-words into words, we remove the space between the sub-words and the special space symbol is replaced with an actual space. In addition, since the setup uses sub-words instead of words for language modeling, it weakens the LM due to decrease in context size. Hence, a higher order 6-gram sub-word LM is used in decoding instead of commonly used 3-gram LM, which was empirically found to perform better. However, using the higher order n-gram significantly increases the decoding time. To alleviate it, decoding is performed using a pruned 6-gram LM and rescoring is performed using the unpruned 6-gram.

III. Experiments

A. System Overview

The line images are resized to a pixel height of 40 while maintaining the aspect ratio. The images are padded with white space at the start and end of the line. Our OM consists of a 6-layer CNN, a 3-layer TDNN and a softmax layer. Dilated convolutions and height sub-sampling layer are used in the network architecture. The size of the softmax layer depends on the number of context dependent characters obtained from decision tree. The decision tree is trained using the alignments obtained from training data and flat-start OM and optimizes the GMM likelihood. WFSTs are used for decoding to combine the OM with n-gram LM.

For the IAM dataset, deslanting and deskewing is applied for augmentation and also to the test set. A smaller topology of four states is used for punctuations and space and an eight state HMM topology is used for other characters. The Rimes dataset has 6.3% and 6.1% OOV rate on test and validation set for lexicon built with training text. Expanding bounding box as augmentation is applied on this, as well as a subset of Madcat Arabic and Madcat Chinese dataset. Paragraph scoring is further applied to Rimes dataset. For Madcat Chinese and YOMDLE Chinese, we also use character decomposition [12] and 16 HMM states instead of eight for non-space characters. Similar to [13], line images of Madcat Chinese dataset are also resized to a higher height of 80 to capture more information, since Chinese characters are intrinsically complex. Semi-supervised training [14] is used for YOMDLE Korean and YOMDLE Tamil, and additional synthetic data is used for YOMDLE Chinese dataset [12].

B. Databases and language modeling

We report WER results on several text recognition datasets. However, most intermediate results are presented for IAM and Rimes datasets because of their popularity in the text recognition community. Final results are reported for six commonly used handwritten recognition datasets (Bentham [21], IAM [15], Madcat Arabic [17], Rimes [16], Madcat Chinese [19], IFN-ENIT [20]) and six printed text recognition datasets (UW3 [18], YOMDLE Chinese, YOMDLE Farsi, YOMDLE Korean, YOMDLE Russian, YOMDLE Tamil). YOMDLE and SLAM [12] [22] are machine printed dataset that includes complex layouts of document images, including mobile camera images of books, newspapers, receipts, Power Point slides, social media and web pages. YOMDLE datasets were used for training and SLAM datasets were used for testing. Both datasets have around 15k to 20k line images each for Chinese, Korean, Farsi, Russian, and Tamil language. In addition, there are a large number of abbreviations, URLs, and typing errors in both datasets, which significantly increase the OOV rate. The details about the datasets are given in Table I.

For IAM dataset, LOB [23](excluding text present in IAM test and validation set), Brown [24], and Wellington [25] text

TABLE I
Total number of lines (sum of test, train, and validation split), OOV rate (calculated from lexicon built with training text), and total unique characters in different datasets

Dataset	Total lines	OOV rate	Total characters
IAM	9.8k	2.3	79
Bentham	11.5k	1.7	93
Rimes	13k	6.3	99
Madcat Arabic subset	25.3k	16.5	145
YOMDLE Tamil	25.6k	49.1	176
IFN-ENIT	26.5k	0	39
YOMDLE Chinese	27.5k	0	4310
YOMDLE Russian	29.7k	46.0	234
YOMDLE Korean	31.4k	55.3	1502
YOMDLE Farsi	33.6k	20.0	254
UW3	96.4k	2.5	89
Madcat Chinese	278k	0	2879
Madcat Arabic	740k	5.3	164

corpora and training text were used for language modeling. When all words from the corpus and training data were included in the lexicon and LM, we observed 2.32% OOV rate on test and 2.71% OOV rate on the validation set. For IAM dataset, punctuation marks were separated from words while calculating WER.

C. Effect of data augmentation and parameter tuning

Table II shows improvements from adding batch normalization, L2-regularization, and scheduled dropout in DNN-HMM optical model discussed in section II-A. We used the LF-MMI objective for all the experiments in this paper, which may sometimes lead to overfitting [2]. To overcome this challenge and to improve training of DNN, we use a combination of the three techniques. Addition of both L2-regularization and batch normalization, led to significant reduction in the WER. Further it was observed that addition of scheduled dropout led to some more improvements.

TABLE II
WER from open vocabulary sub-word setup for IAM dataset (with data-split available with dataset)

L2-regularization	Batch normalization	Schedule Dropout	WER
Y	Y	N	9.17
Y	N	Y	9.79
N	Y	Y	9.95
Y	Y	Y	**8.46**

Table III shows improvements from different data augmentations discussed in section II-B. The three different image augmentations i.e. expanding bounding box, de-slanting and de-skewing, and random scaling for respective datasets, significantly helped in improving the results by approximately 7-12% relative. For IAM dataset, we observed that simply using deslanting and deskewing as a prepossessing step for the test data also gave good performance improvements.

D. Effect of the OOV rate

We evaluate our word-based open vocabulary setup (Section II-C1) by measuring the improvement between open vocabulary and closed vocabulary scenario for different OOV rates for

TABLE III
WER from open vocabulary sub-word based system without augmentation (WER1) and with augmentation (WER2) for different augmentations for different datasets

Dataset	Augmentation	WER1	WER2
YOMDLE Korean	random scaling	28.9	**27.0**
IAM	de-slant and de-skew	12.3	**11.0**
Rimes	expanding bounding box	9.0	**8.3**

IAM dataset. The experiment is conducted with IAM official split and without text normalization for LM. Table IV shows the WERs for the open vocabulary setup and the baseline closed vocabulary setup for different lexicon size and hence at different OOV rates for the IAM dataset. For a vocabulary size $|\mathcal{V}|$, we used top $|\mathcal{V}|$ most frequent words from the LM text and remaining words are mapped to <unk> symbol. A 3-gram word-based LM was built using an open-source pocoLM toolkit and a 4-gram character based LM was built using the lexicon words. We can see that decreasing OOV rate, the performance of both setups improved consistently, as did the relative improvement between closed and open vocabulary setups. However, the performance gap between closed vocabulary and open vocabulary systems reduced significantly for smaller OOV rates.

TABLE IV
WER from closed vocabulary word-based system (WER1)and open-vocabulary word-based system (WER2) for IAM dataset (with data-split available with dataset)

Lexicon size	OOV rate	WER1	WER2
50k	4.5	12.1	**9.9**
100k	3.1	10.5	**9.4**
150k	2.7	10.0	**9.2**
200k	2.3	9.7	**8.9**

E. Effect of the order of n-gram

As briefly discussed in the Section II-C2, for an open vocabulary sub-word setup, using a standard 3-gram LM weakens the model. It is due to the fact that a word can be formed by combining approximately 1–6 sub-words, which reduces the context for a 3-gram sub-word LM to 1–2 words. To alleviate this problem, a higher order n-gram sub-word LM is used. Table V shows the WERs for the open vocabulary sub-word setup at different n-gram order with the same language modeling text. For Madcat Arabic and Rimes dataset, only training text is used for LM training whereas for IAM dataset a corpus text as mentioned in III-B is also used.

TABLE V
WER from open-vocabulary sub-word system at different n-gram order for different datasets and different corpus size (number of words)

Dataset	corpus size	3-gram	4-gram	5-gram	6-gram
IAM	7.6M	10.01	8.65	8.52	**8.46**
Rimes	0.26M	9.56	9.36	9.29	**7.66**
Madcat Arabic	8.6M	9.56	8.33	8.15	**8.05**

IV. Final results

A. Comparison with word based setup

Table VI shows the comparison between the word-based setup and the sub-word based setup for different datasets. Both the setups have similar DNN structure, and the same corpus text and augmentations are used for both. Due to time constraints, we use a closed vocabulary word-based setup for comparison in the case of Madcat Arabic and Madcat Arabic subsets, whereas for IAM and Rimes an open vocabulary word-based setup is built. In addition to the details mentioned in Section III-D, a 3-gram word LM is built on the training corpus and a 4-gram character LM built with the lexicon is used for Rimes. We see that increasing OOV rate the relative difference in performance between the two systems increases, which demonstrates the usefulness of our open vocabulary text recognition system.

TABLE VI
Comparison between our word-based system (WER1) and BPE-based system (WER2)

Dataset	OOV rate	WER1	WER2
IAM	2.3	8.9	**8.4**
Madcat Arabic	5.3	10.8	**8.0**
RIMES	6.3	9.3	**7.6**
Madcat Arabic subset	16.5	24.2	**13.7**

B. Comparison with published results

Table VII shows comparison of our system with published results for different datasets. To make the results comparable with other groups, following modifications were made before reporting the results. For IAM, we followed a similar data split as [26], which has 6,482, 976, and 2,915 samples for train, validation, and test. Same refernece tokenization and hypothesis normalization (separating a subset of punctuation marks (; ! , : ") from word and joining a subset of contraction ('s,'t,'ll,'m,'ve,'re,'d) to word) were used. For LM training, we used the IAM training transcripts, LOB (excluding the text present in IAM test and validation set), Brown and Wellington text corpora. Tokenization similar to hypothesis normalization are applied to language model text. For Bentham dataset [27], punctuation marks were separated from words while calculating WER. For Rimes, a similar procedure as in [28] was followed and 10% of the total training samples were used for validation purposes to get a split of 10,203, 1,130, and 778 for train, validation, and test, respectively. Paragraph level error rates were computed. Similar to [13] and [28], no extra corpus text was used for Madcat Arabic and Rimes dataset. Furthermore, following the method in [13], we normalized certain diacritics while computing the WER and CER. We used the same splits for all databases except Madcat Arabic (and have made it publicly available) because of unavailability of original data splits. For the Madcat Chinese and UW3 datasets, results from character based model and open vocabulary word-based model are reported, respectively. Madcat Chinese is a relatively new dataset and do not have published results. For Madcat Chinese, results are compared with a system built with ESPnet [29]. Since IFN-ENIT does not have OOV words, results from a closed vocabulary word-based setup are reported. For UW3 and Madcat Chinese, CER is reported instead of WER.

Voigtlaender et al. [26] used a deep multidimensional long short term memory network (MDLSTM) with connectionist temporal classification (CTC) loss function. The system was trained end to end in an open vocabulary recognition setup with weighted finite state transducer. Puigcerver et al. [28] used a convolution neural network for feature extraction and LSTMs for modeling context on either side of the feature vector. A CTC objective function and WFST based decoding was performed to get final results in a closed vocabulary scenario. Rawls et al. [13] used a system built with CNN, LSTM, and CTC objective function. No language model was used and greedy decoding was performed on the output of the neural network. Our results, without using extra corpus text for LM, do not match [13], but on including extra corpus text show an improvement in WER over [13]. For IFN-ENIT, although [6] performs better than our system, they use a full CNN-based system for isolated word recognition, which is difficult to adopt for more general line recognition system. The major advantage in our approach over other approaches is that using a sub-word lexicon and sub-word language model ensures that we do not have OOV words and our sub-word LM is as strong as a word-based LM.

TABLE VII
Performance comparison of our DNN-HMM text recognition setup for different amount of training data

Dataset	Training Size	Best WER	Our WER
IAM	6.4K	**9.3** [26]	10.0
Bentham	9.1k	8.6 [27]	**8.0**
RIMES	10.2k	9.0 [28]	**7.6**
YOMDLE Korean	13.9k	16	**12.5**
YOMDLE Tamil	14.1k	12.4	**8.9**
YOMDLE Farsi	16.2k	**11.5**	12.7
YOMDLE Russian	17.9k	**7.5**	**7.4**
IFN-ENIT	19.7K	**5.9** [30]	7.6
UW3	91.8k	0.6 [31]	**0.1**
Madcat Chinese	185k	6.2	**4.4**
Madcat Arabic	600k	**5.9** [13]	8.0

V. Conclusion

In this paper, we presented an open vocabulary sub-word based text recognition system. We adopted the hybrid deep neural network HMM (DNN-HMM) acoustic model used in ASR for text recognition, and implemented a new sub-word based algorithm for lexicon and language modeling. We showed that modifications made for optical modeling, data augmentation, sub-word lexicon and sub-word language model help in significantly improving the performance of our system. Finally, we achieved an approximate 10% relative improvement which is consistent across majority of datasets.

REFERENCES

[1] Wang ZR, Du J, Wang WC, Zhai JF, Hu JS. A comprehensive study of hybrid neural network hidden Markov model for offline handwritten Chinese text recognition. International Journal on Document Analysis and Recognition (IJDAR).2018.

[2] Daniel Povey et al. Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In: Interspeech (2016), pp. 27512755.

[3] Micha Kozielski et al. Open vocabulary handwriting recognition using combined word-level and character-level language models. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on (2013), pp. 82578261

[4] Smit P, Virpioja S, Kurimo M. Improved subword modeling for WFST-based speech recognition. InINTERSPEECH 201718t Annual Conference of the International Speech Communication Association 2017 Aug 23.

[5] Meng Cai et al. An open vocabulary OCR system with hybrid word subword language models. In: Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on. Vol. 1. IEEE. 2017, pp. 519524.

[6] Daniel Povey et al. The Kaldi speech recognition toolkit. In: (2011).

[7] Peddinti V, Povey D, Khudanpur S. A time delay neural network architecture for efficient modeling of long temporal contexts. In Sixteenth Annual Conference of the International Speech Communication Association 2015.

[8] Hossein Hadian et al. End-to-end speech recognition using lattice-free MMI. In: Proc. Interspeech 2018 (2018), pp. 1216.

[9] Hossein Hadian et al. Flat-Start Single-Stage Discriminatively Trained HMM-Based Models for ASR. In: IEEE/ACM Transactions on Audio, Speech, and Language Processing 26.11 (2018), pp. 19491961.

[10] Bertolami R, Uchida S, Zimmermann M, Bunke H. Non-uniform slant correction for handwritten text line recognition. InDocument Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on 2007 Sep 23 (Vol. 1, pp. 18-22). IEEE.

[11] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In: arXiv preprint arXiv:1508.07909 (2015).

[12] Optical Character Recognition with Chinese and Korean Character Decomposition (submitted to ICDAR 2019)

[13] Rawls S, Cao H, Mathai J, Natarajan P. How To Efficiently Increase Resolution in Neural OCR Models. In2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR) 2018 Mar 12 (pp. 140-144). IEEE.

[14] Manohar V, Hadian H, Povey D, Khudanpur S. Semi-supervised training of acoustic models using lattice-free MMI. In2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018 Apr 15 (pp. 4844-4848). IEEE.

[15] U.-V. Marti and H. Bunke, The iam-database: an english sentence database for offline handwriting recognition, International Journal on Document Analysis and Recognition, vol. 5, no. 1, pp. 3946, 2002

[16] E. Augustin, M. Carre , E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Preteux, Rimes evaluation campaign for handwritten mail processing, in International Workshop on Frontiers in Handwriting Recognition (IWFHR06), 2006, pp. 231235.

[17] S. Strassel, Linguistic resources for arabic handwriting recognition, in Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt, 2009.

[18] I.Phillips,Users reference manual for the uw english / technical document image database iii, UW-III English/Technical Document Image Database Manual, 1996.

[19] Z. Song, S. Ismael, S. Grimes, D. S. Doermann, and S. Strassel, Linguistic resources for handwriting recognition and translation evaluation. in LREC, 2012, pp. 39513955.

[20] M. Pechwitz, S. S. Maddouri, V. Margner, N. Ellouze, H. Amiri, et al. Ifn/enit-database of handwritten arabic words. Citeseer

[21] Snchez JA, Mhlberger G, Gatos B, Schofield P, Depuydt K, Davis RM, Vidal E, De Does J. tranScriptorium: a european project on handwritten text recognition. InProceedings of the 2013 ACM symposium on Document engineering 2013 Sep 10 (pp. 227-228). ACM.

[22] A Synthetic Recipe for OCR (submitted to ICDAR 2019)

[23] S.Johansson, E.Atwell, R.Garside, and G.Leech, The Tagged LOB Corpus: Users Manual, Norwegian Computing Centre for the Humanities, 1986.

[24] W. Francis and H. Kucera, Brown corpus manual, manual of information to accompany a standard corpus of present-day edited american english, Tech. Rep., 1979.

[25] Bauer L. Manual of information to accompany the Wellington corpus of written New Zealand English. Wellington: Department of Linguistics, Victoria University of Wellington; 1993.

[26] Paul Voigtlaender, Patrick Doetsch, and Hermann Ney. Handwriting recognition with large multidimensional long short-term memory recurrent neural networks. In: Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on. IEEE. 2016, pp. 228233.

[27] Bluche T, Ney H, Kermorvant C. The LIMSI handwriting recognition system for the HTRtS 2014 contest. InDocument Analysis and Recognition (ICDAR), 2015 13th International Conference on 2015 Aug 23 (pp. 86-90). IEEE.

[28] Joan Puigcerver. Are Multidimensional Recurrent Layers Really Necessary for Handwritten Text Recognition? In: Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on. Vol. 1. IEEE. 2017, pp. 6772.

[29] Kim S, Hori T, Watanabe S. Joint CTC-attention based end-to-end speech recognition using multi-task learning. InAcoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on 2017 Mar 5 (pp. 4835-4839). IEEE.

[30] Arik Poznanski and Lior Wolf. Cnn-n-gram for handwriting word recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, pp. 23052314.

[31] Breuel TM, Ul-Hasan A, Al-Azawi MA, Shafait F. High-performance OCR for printed English and Fraktur using LSTM networks. InDocument Analysis and Recognition (ICDAR), 2013 12th International Conference on 2013 Aug 25 (pp. 683-687). IEEE.