OPEN VOCABULARY HANDWRITING RECOGNITION USING COMBINED WORD-LEVEL AND CHARACTER-LEVEL LANGUAGE MODELS

Michał Kozielski[1], *David Rybach*[1], *Stefan Hahn*[1], *Ralf Schlüter*[1], *Hermann Ney*[1,2]

[1] Human Language Technology and Pattern Recognition
Computer Science Department, RWTH Aachen University, Aachen, Germany

[2]Spoken Language Processing Group, LIMSI CNRS, Paris, France

{kozielski,rybach,hahn,schlueter,ney}@cs.rwth-aachen.de

ABSTRACT

In this paper, we present a unified search strategy for open vocabulary handwriting recognition using weighted finite state transducers. Additionally to a standard word-level language model we introduce a separate n-gram character-level language model for out-of-vocabulary word detection and recognition. The probabilities assigned by those two models are combined into one Bayes decision rule. We evaluate the proposed method on the IAM database of English handwriting. An improvement from 22.2% word error rate to 17.3% is achieved comparing to the closed-vocabulary scenario and the best published result.

Index Terms— open vocabulary recognition, handwriting recognition, character-based language models

1. INTRODUCTION

In speech and handwriting recognition, the open vocabulary scenario assumes that the training data covers only a part of the entire word space. This condition leads to the encounter of unknown words, called OOV words. The problem has been already extensively studied mainly in the former domain. Most of the current approaches to solve it rely on so called sub-lexical methods.

Bisani [1] approaches the OOV problem by using a hybrid lexicon consisting of word fragments and full words. After decoding, the fragments are concatenated greedily to form words. Vertanen [2] uses one mixed language model (LM) with OOV words replaced by graphone sequences. Shaik [3] decomposes words into different sub-word units, for example morphemes and syllables. Those approaches posses however some drawbacks. If the words are decomposed into some smaller units, the important lexical constraint is missing. In the mixed models that include both words and sub-words, the LM is polluted as the contexts of those two types of units interfere with each other. Additionally, it is difficult to tune the ratio of sub-words transcribed during recognition if the OOV rate between training and development sets differs.

Another kind of approach to the problem is OOV detection using confidence scores. Burget [4] tries to detect OOV words by using combined posteriors from strongly- and weakly-constrained systems. Lin [5] proposes to compare word and phone lattices to specify candidate OOVs. Hazen [6] introduces a generic OOV model to allow for any phoneme sequence during recognition.

Some approaches have been also made in the domain of handwriting recognition. Brakensiek [7] addresses the problem with character bi-grams and tri-grams. The recognition performance is however worse compared to the closed-vocabulary scenario. Bazzi [8] compares the recognition results of word-based and character-based system with a hybrid system where the character LM has been augmented with a word constraint.

We propose a combined probabilistic approach to the open vocabulary handwriting recognition. We recognize OOV words on character level by hypothesizing them as sequences of characters. A separate character-level LM is created and combined into the word-level LM. The combined model enforces the word constraint in a probabilistic manner and the mutual interaction between those two models is easy to tune using system parameters. A similar approach has been successful in the domain of speech recognition [9].

2. PROPOSED METHOD

2.1. Theoretical background

The goal of the recognition task is to find the word sequence $w_1^N := w_1, ..., w_N$ that best explains the observation sequence $x_1^T := x_1, ..., w_T$. This is accomplished by maximizing the posterior probability $p(w_1^N|x_1^T)$ with an unknown number of words N. The posterior probability is modelled by the Bayes' decision rule:

$$\hat{w}_1^N = \arg\max_{w_1^N}\{q^\gamma(w_1^N)p(x_1^T|w_1^N)\} \quad (1)$$

Two stochastic models represent the probabilities appearing in this equation. The acoustic (or visual) model $p(x_1^T|w_1^N)$ outputs a sequence of characters given a sequence of visual observations. The language model assigns a prior probability to the word sequence:

$$q(w_1^N) = \prod_{i=1}^{N} q(w_i|w_{i-n+1}^{i-1}) = \prod_{i=1}^{N} q(w_i|h_i) \qquad (2)$$

We compute the probability $q(w_i|h_i)$ for every word in the word space given an n-gram context $h_i = w_{i-n+1}, ..., w_{i-1}$ of $n-1$ previous words. The parameter γ is a scaling exponent of the language model, called LM scale.

In the typical word-level LM $p(w_1^N)$, we compute the probability only for the known words $w_i \in V$, that is those appearing in the training data's vocabulary. It is possible to incorporate the probability for an unknown word into the language model by limiting the vocabulary and substituting every word in the training set, that is not included in the vocabulary, by a special token w_{oov}. This token represents an OOV word and is treated like any other word, which means that it can appear in the context of an n-gram and that the probability $p(w_{oov}|h)$ of observing it is non-zero. However, it is not possible to transcribe such an OOV word during recognition because the w_{oov} token has no assigned sequence of characters in the lexicon, which means that the probability for it cannot be calculated by the acoustic model.

The idea to overcome this problem is to simultaneously hypothesize and recognize OOV words as sequences of separate characters. We represent the a priori knowledge of dependencies between characters by a second n-gram model of order m on character level. It is then possible to represent any word as a sequence of characters $c_1^M \in \mathcal{C}^*$ and compute the probability of it as:

$$p(c_1^M) = \delta \cdot \prod_{j=1}^{M} p(c_j|c_{j-m+1}^{j-1}) \qquad (3)$$

The parameter δ is an additive penalty. We denote the function that maps a word to a corresponding sequence of characters as $C : V \rightarrow \mathcal{C}^*$. The final language model is obtained by combining the separate word- and character-level LMs. The max function chooses the model that has a higher probability:

$$q(w_i|h_i) = max\{p(w_{oov}|h_i)p^{\alpha}(C(w_i)),\ p(w_i|h_i)\} \qquad (4)$$

The probability $p(C(w_i))$ assigned by the character-level LM is multiplied with the probability $p(w_{oov}|h_i)$ assigned by the primary word-level LM. The parameter α is a scaling exponent of the character-level LM. Both models can hypothesize any word from the word space, but the probability for an OOV word assigned by the word-level LM is zero. Whenever an OOV word is transcribed as a character sequence in the recognition process, it is retained in the context h_i as w_{oov}. For the purpose of word alignment and scoring, a character sequence is merged together into one word.

This approach clearly separates the word representation from character sequence representation in the sense that there are two separate contexts for words and characters that do not interfere with each other. It also allows for those two language models to have different orders and to be created using different discounting methods. The number of words transcribed as character sequences during recognition can be easily tuned using system parameters.

2.2. Decoding using WFST

We use a dynamic network decoder based on weighted finite-state transducers (WFST) [10], which integrates the LM dynamically as needed during recognition. The composition of the language model transducer G and the lexicon transducer L is computed on demand using composition filters for on-the-fly pushing of labels and weights [11]. The expansion from input labels of L to HMM-states is performed dynamically as well. A detailed description of the decoder can be found in [12].

Here we need to integrate two LMs: the word-based LM and a character-level LM. Both models are compiled into separate WFSTs, denoted G and G_c. In G and G_c each state q_h represents a word or character history h. An arc leaving a state q_h and labeled with a word or character w has the weight $p(w|h)$.

G contains arcs labeled with the w_{oov} token. These arcs are substituted on-demand during recognition with a copy of G_c. An w_{oov}-arc is replaced by an arc labeled with ϵ (the empty word) and weighted by the probability of the original n-gram. It leads to the start state of a copy of G_c. The final state of G_c is connected to the destination state of the replaced arc. When complex LMs are used, the fully expanded LM automaton would be too large to be kept in memory. Our implementation is based on OpenFst [13].

3. EXPERIMENTS

3.1. Database

The IAM database [14] consist of images of handwritten English text sentences, which have been built upon the LOB corpus [15]. There are 747 paragraphs of text for training, 116 for development, and 336 for evaluation. The language models have been built upon the combined LOB [15], Brown [16], and Wellington [17] corpora. We have excluded the sentences appearing in IAM development and evaluation sets for the purpose of training the language model. The training, development and evaluation text sources contain $3,338,728$; $8,633$ and $25,472$ running words, and the size of vocabulary is $101,443$; $2,396$ and $5,312$ words accordingly. The character inventory contains 77 characters plus silence and noise. We are interested in producing an open vocabulary scenario

Table 1. OOV rates on different sets with respect to a preselected vocabulary size.

Voc. size	OOV rate [%]	
	Train	Dev.
all	0	2.6
50k	1.7	3.9
20k	5.0	6.6
10k	8.9	9.6
5k	14.0	13.5
1k	28.8	25.1
100	49.1	45.8

in which OOV words exist in both training and development sets. Therefore we limit the vocabulary to a different number of most frequent words from the training set. Table 1 contains the summary of OOV rates with respect to a vocabulary size.

3.2. System overview

For preprocessing, we correct the slant of gray-scale images with a median of angle values estimated by three different deslanting algorithms [18][19][20]. We segment the images with a sliding window of a constant shift and width, and a height equal to the size of the original image. A horizontal cosine window is applied to each frame to smooth the image on its borders. Each frame is normalized to a size of 8×32 pixels using 1st- and 2nd-order moments [21]. We then take gray-scale values of all pixels and reduce their number to 20 components using PCA transformation. The final feature vector of size 24 is augmented by original moment values [21]. The system is writer adaptive using CMLLR for feature transformation [22]. For classification we use an HMM model with 6 segments per character. A segment consists of 2 states with the same emission distribution. The model is trained with the Viterbi algorithm using maximum likelihood (ML) as training criterion. Emission distributions are trained with Gaussian mixtures with a total of $\sim$30k densities. The parameters have been optimized experimentally on the development set.

3.3. Language model

As word-level language model we use standard 3-gram model with modified Kneser-Ney discounting build upon the training text source containing one sentence per line. Because we recognize whole paragraphs of text, which contain multiple sentences, the language model has to be able to hypothesize the sentence boundary. For most of the experiments we use a vocabulary composed of the 10k most frequent words which is associated with an OOV rate of a reasonable size. On the other hand the OOV rate is sufficiently large to observe the results of applying the method described in this paper. The character-level language model is build upon a list of words,

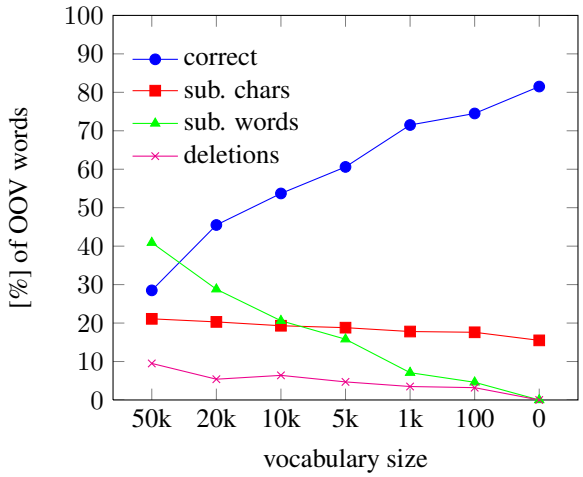

Fig. 1. Distribution of recognition errors on OOV words from the development set with respect to a vocabulary size.

Table 2. Comparison of results between closed and open vocabulary scenarios on the development set.

Voc.	OOV [%]	Closed voc. [%]		Open voc. [%]	
		WER	CER	WER	CER
50k	3.9	14.8	5.9	12.2	4.8
20k	6.6	18.7	8.2	12.4	5.1
10k	9.6	22.0	9.0	12.8	5.1
5k	13.5	-	-	13.0	5.2
1k	25.1	-	-	14.3	5.6
100	45.8	-	-	18.6	7.0
0	100.0	-	-	21.0	7.7

one word per line, split into separate characters. The Witten-Bell method is used for discounting. We cannot use the standard modified Kneser-Ney method because of lack of singletons in the training data. We discuss different approaches to creating candidate word list in section 3.4.

3.4. Experimental results

We evaluate our method by measuring the improvement between the closed and open vocabulary scenarios. Error rates and statistics are calculated using the Levenshtein alignment between reference and hypothesis. The character sequences are merged together into a word prior to aligning. The decoder can hypothesise the word boundary between two consecutive character sequences.

Table 2 shows the recognition results for different vocabulary sizes. The absolute difference between results in the open vocabulary scenario is considerably smaller than in the closed vocabulary. Without a vocabulary the system still maintains a very good word error rate of 21.0%. That means that the character-level LM can learn correctly the more fre-

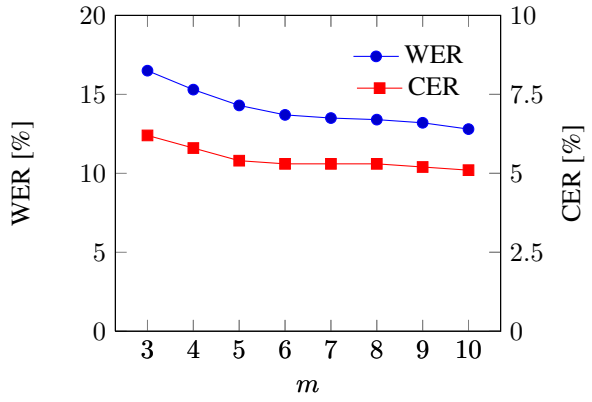

Fig. 2. Error rates with respect to an order of the character-level LM on the development set. The vocabulary size is 10k.

quent words and can substitute to some degree the lexical constraint enforced by the word-level LM.

Figure 1 shows the distribution of certain recognition errors on the OOV words from reference. The percentage of substitutions made by the character-level LM remains almost constant. This suggests an upper boundary of the recognition performance of that model and indeed the number of correct matches approaches 80% with no vocabulary at all. The percentage of substitutions made by the word-level LM decreases, but remains constant when measured as an absolute number of words. Those errors are generated mainly by a fixed list of compound words, which are transcribed as multiple in-vocabulary words.

Figure 2 shows the influence of the order of character-level LM on the recognition performance. We obtain the best results for a 10-gram LM (12.8% WER and 5.1% CER).

Table 3 shows the results for different approaches of choosing words to train the character-level LM. The word list is constructed by extracting all words from the training set or only the OOV words. Additionally the words can be weighted or not by their frequency counts. We obtain the best results if we choose only the OOV words with their frequency counts.

The parameters $\alpha = 0.4$, $\gamma = 20$ and $-\log \delta = 1$ have been optimized experimentally on the development set.

3.5. Comparison with the state-of-the-art

Table 4 shows the comparison of the results on the IAM database. To make the results comparable with other groups we use a vocabulary composed of the 20k most frequent words. We achieve a word error rate of 17.3% on the evaluation set in the open vocabulary scenario. The word error rate of 22.2% obtained in the closed vocabulary scenario serves as a baseline. It is also the best closed vocabulary result published so far. The results in the lower part of the table are the best results reported so far for IAM. España-Boquera [23] used neural networks to perform particular preprocessing steps. Toselli [24] developed an HMM-based system using gray-scale and gradient features. Graves [25] used an LSTM recurrent neural network with a CTC output layer. Finally, Bertolami [26] applied a voting strategy to several HMM models.

Table 3. Comparison of results on the development set between different approaches of creating training data for the character-level LM.

Voc.	IV words	OOV words	WER [%]	CER [%]
10k	with freq.	with freq.	12.8	5.2
10k	w/o freq.	w/o freq.	13.3	5.3
10k	-	with freq.	12.8	5.1
10k	-	w/o freq.	13.4	5.3
1k	with freq.	with freq.	15.0	5.9
1k	w/o freq.	w/o freq.	17.5	6.6
1k	-	with freq.	14.3	5.6
1k	-	w/o freq.	17.5	6.6

Table 4. Comparison with results reported by other groups on the development and evaluation sets.

Systems	Voc.	WER [%]		CER [%]	
		Dev.	Eval	Dev.	Eval
open vocabulary	20k	12.4	17.3	5.1	8.2
closed vocabulary	20k	18.7	22.2	8.2	11.1
España et al. [23]	50k	19.0	22.4	-	9.8
Toselli et al. [24]	9k	-	25.8	-	-
Graves et al. [25]	20k	-	25.9	-	18.2
Bertolami et al. [26]	20k	26.8	32.8	-	-

4. CONCLUSIONS

We have shown that the use of two combined language models significantly improves the recognition performance on an open vocabulary handwriting recognition task. The character-level LM can learn correctly the more frequent words and can substitute to some degree the lexical constraint enforced by the word-level LM. We demonstrated that the character-level LM should be constructed using the OOVs extracted from the training set. We also showed that higher-order character-level LMs perform better than their lower-order equivalents. On the IAM database our open vocabulary HMM-based system is the best system published so far. It achieves the performance of 17.3% word error rate on the evaluation set which is a 4.9% WER absolute improvement over the same system in the closed vocabulary scenario.

Acknowledgments. This work was partially supported by a Google Research Award and by the Quaero Programme, 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.

5. REFERENCES

[1] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," in *Interspeech*, Lisbon, Portugal, Sept. 2005, pp. 725–728.

[2] K. Vertanen, "Combining open vocabulary recognition and word confusion networks," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, Apr. 2008, pp. 4325–4328.

[3] M. Ali Basha Shaik, Amr El-Desoky Mousa, Ralf Schlüter, and Hermann Ney, "Hybrid language models using mixed types of sub-lexical units for open vocabulary german LVCSR," in *Interspeech*, Florence, Italy, Aug. 2011, pp. 1441–1444.

[4] L. Burget, P. Schwarz, P. Matejka, M. Hannemann, A. Rastrow, C. White, S. Khudanpur, H. Hermansky, and J. Cernocky, "Combination of strongly and weakly constrained recognizers for reliable detection of OOVs," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, Apr. 2008, pp. 4081–4084.

[5] Hui Lin, J. Bilmes, D. Vergyri, and K. Kirchhoff, "OOV detection by joint word/phone lattice alignment," in *Automatic Speech Recognition Understanding, 2007. ASRU. IEEE Workshop on*, Dec. 2007, pp. 478–483.

[6] T.J. Hazen and I. Bazzi, "A comparison and combination of methods for OOV word detection and word confidence scoring," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, 2001, vol. 1, pp. 397–400.

[7] A. Brakensiek, J. Rottland, and G. Rigoll, "Handwritten address recognition with open vocabulary using character n-grams," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, 2002, pp. 357–362.

[8] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open-vocabulary OCR system for english and arabic," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 6, pp. 495–504, June 1999.

[9] M. A Basha Shaik, D. Rybach, S. Hahn, R. Schlüter, and H. Ney, "Hierarchical hybrid language models for open vocabulary continuous speech recognition using WFST," in *Workshop on Statistical and Perceptual Audition*, Portland, OR, USA, Sept. 2012, pp. 46–51.

[10] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Speech recognition with weighted finite-state transducers," in *Handbook of Speech Processing*, Jacob Benesty, M. Sondhi, and Yiteng Huang, Eds., chapter 28, pp. 559–582. Springer, 2008.

[11] Cyril Allauzen, Michael Riley, and Johan Schalkwyk, "Filters for efficient composition of weighted finite-state transducers," in *Proceedings of the 15th international conference on Implementation and application of automata*, 2011, pp. 28–38.

[12] David Rybach, Ralf Schlüter, and Hermann Ney, "A comparative analysis of dynamic network decoding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, May 2011, pp. 5184–5187.

[13] Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri, "OpenFst: a general and efficient weighted finite-state transducer library," in *Proceedings of the 12th international conference on Implementation and application of automata*, 2007, pp. 11–23.

[14] 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, pp. 39–46, Nov. 2002.

[15] S. Johansson, E. Atwell, R. Garside, and G. Leech, *The Tagged LOB Corpus: User's Manual*, Norwegian Computing Centre for the Humanities, 1986.

[16] 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.

[17] L. Bauer, "Manual of information to accompany the wellington corpus of written new zealand english," Tech. Rep., 1993.

[18] Moisés Pastor, Alejandro Toselli, and Enrique Vidal, "Projection profile based algorithm for slant removal," in *Image Analysis and Recognition*, vol. 3212, pp. 183–190. 2004.

[19] Alessandro Vinciarelli and Juergen Luettin, "A new normalization technique for cursive handwritten words," *Pattern Recognition Letters*, vol. 22, no. 9, pp. 1043–1050, 2001.

[20] Moisés Pastor i Gadea, Alejandro H. Toselli, Verónica Romero, and Enrique Vidal, "Improving handwritten off-line text slant correction," in *Procc. of The Sixth IASTED international Conference on Visualization, Imaging, and Image Processing (VIIP 06)*, 2006.

[21] Michał Kozielski, Jens Forster, and Hermann Ney, "Moment-based image normalization for handwritten text recognition," in *International Conference on Frontiers in Handwriting Recognition*, Bari, Italy, Sept. 2012, pp. 256–261.

[22] Philippe Dreuw, David Rybach, Christian Gollan, and Hermann Ney, "Writer adaptive training and writing variant model refinement for offline arabic handwriting recognition," in *International Conference on Document Analysis and Recognition*, Barcelona, Spain, July 2009, pp. 21–25.

[23] S. España-Boquera, M.J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 4, pp. 767–779, Apr. 2011.

[24] Alejandro H. Toselli, Verónica Romero, Moisés Pastor i Gadea, and Enrique Vidal, "Multimodal interactive transcription of text images," *Pattern Recognition*, vol. 43, no. 5, pp. 1814–1825, 2010.

[25] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 855–868, May 2009.

[26] Roman Bertolami and Horst Bunke, "Hidden markov model-based ensemble methods for offline handwritten text line recognition," *Pattern Recognition*, vol. 41, no. 11, pp. 3452–3460, Nov. 2008.