Bidirectional Language Model for Handwriting Recognition

Volkmar Frinken[1], Alicia Fornés[1], Josep Lladós[1], and Jean-Marc Ogier[2]

[1] Computer Vision Center, Dept. of Computer Science
Edifici O, UAB, 08193 Bellaterra, Spain
{vfrinken,afornes,josep}@cvc.uab.cat
[2] L3i Laboratory, Université de La Rochelle
Av. M. Crépeau, 17042 La Rochelle Cédex 1, France
jean-marc.ogier@univ-lr.fr

Abstract. In order to improve the results of automatically recognized handwritten text, information about the language is commonly included in the recognition process. A common approach is to represent a text line as a sequence. It is processed in one direction and the language information via n-grams is directly included in the decoding. This approach, however, only uses context on one side to estimate a word's probability. Therefore, we propose a bidirectional recognition in this paper, using distinct forward and a backward language models. By combining decoding hypotheses from both directions, we achieve a significant increase in recognition accuracy for the off-line writer independent handwriting recognition task. Both language models are of the same type and can be estimated on the same corpus. Hence, the increase in recognition accuracy comes without any additional need for training data or language modeling complexity.

Keywords: handwriting recognition; language models; neural networks.

1 Introduction

The recognition of handwritten text is a very active research field among researchers on pattern recognition [12]. Promising approaches for handwriting recognition are segmentation-free and learning-based, such as hidden Markov models (HMM) [2, 13], neural networks (NN) [6], or combinations thereof [3].

Still, the recognition of unconstrained text can not be considered a solved problem. The main reason is the difficulty in dealing with the high variability encountered in different handwriting styles. Often, a semantic understanding of the text is necessary to be able to read a text. In case of automatic recognition systems, contextual understanding is usually emulated by estimating word probabilities, such as n-grams [5, 7]. Yet, despite their simplicity and inability to capture any long-term relationships between words, n-gram approaches perform remarkably well and are still state-of-the-art.

Current handwriting recognition systems represent the text line as a sequence and perform the recognition usually in the direction of writing, i.e., left to right

for Roman scripts. This allows to directly include n-gram language model information in the decoding. In this form of language probability estimation, however, only a limited context is used to estimate the occurrence probability of a word. As a result, recognizers face the problem of error propagation. Correct word that are required in a larger context might be dropped due to pruning. Instead a wrong hypotheses propagates wrong language model information to the following words and may disturb their recognition, hence creating a form of decoding direction dependent error.

As a consequence, one-directional decoding seems to be an unnecessary restriction, especially when the input data are off-line text images. A word's probability can be estimated more robustly by considering both n-grams, the one considering the words on the left, and the one considering the words on the right side. Thus, taking also the reversed decoding direction into account could reduce the recognition error-propagation.

In this paper we propose the use of bi-directional n-grams for improving the recognition performance of unconstrained handwritten text. In order to do this, N-best lists are created for both directions separately, using a distinct forward and a backward language model. Then, these lists are combined to produce the final recognition output. Note that the system used in this paper is based on Neural Networks [6], but it could easily be extended to HMM-based approaches as well.

The rest of the paper is structured as follows. In Section 2 the proposed bidirectional language model approach is introduced and explained in detail. The experimental evaluation is presented in Section 3 and conclusions are drawn in Section 4.

2 Bidirectional Language Models

The ambiguity of different handwritten text and the huge variances in different writing styles require an integration of contextual information for an automatic transcription. The standard way of doing this is to integrate a statistical language model in the decoding process. However, language modeling using bi-grams do not capture the language sufficiently well. One option is to increase the complexity of the language model by using higher order n-grams, however, the number of distinct n-grams increases exponentially with n. Hence, even in a large training corpus, many word combinations do not occur at all or they occur with a frequency not high enough for a robust occurrence probability estimation.

Another challenge to handwriting recognition is the error propagation of a mis-recognized word. As a sequential decoding problem, common recognition methods process the text line in one direction, left-to-right or right-to-left. Hence, any mis-recognition propagates in the direction of recognition due to the language model which takes the current recognition result to estimate the next word's probability.

To address both issues, the challenge to estimate sophisticated language models on sparse data as well as the problem of error propagation, we propose in this

paper to decode the text from both directions and combine the results. Forward and backward decoding require to different language models which can still be estimated on the same corpus and the combination can successfully increase the recognition accuracy.

The proposed approach is a step towards holistic language models to better capture syntactic and semantic information. While such models have been proposed for speech recognition in a sophisticated way [14], our approach does not increase the language modeling and hence the computational complexity.

2.1 Contribution

From a mathematical point of view, continuous handwriting recognition systems map a text image to a sequence of words $w_1^S = w_1 w_2 \dots w_S$. This is done by using both, an observation model ϑ that assigns a probability value to a character sequence according to the observed image and a language model LM that assigns a probability value to a given character sequence according to the language at hand. The character sequence that maximizes the combined score is then selected as the final output.

In this paper we focus on the language model probability score which can be factorized as

$$
\begin{aligned}
p(w_1^S) &= p(w_1) \cdot p(w_2|w_1) \cdots p(w_S|w_1^{S-1}) && (1)\\
&= p(w_1) \prod_{i=2}^{S} p(w_i|w_1^{i-1}) && (2)\\
&= p(w_S) \cdot p(w_{S-1}|w_S) \cdots p(w_1|w_2^S) && (3)\\
&= p(w_S) \prod_{i=1}^{S} p(w_i|w_{i+1}^S) \ . && (4)
\end{aligned}
$$

Note that we define $w_i = \varepsilon$ for $i \leq 0$ and $i > S$ to make the Equations more readable. Following from the rules of probability, it does not matter whether the LM probability is factorized such that the probability for a word w_i is conditioned on its left context w_1^{i-1} (see Eqn. 2) or its right context w_{i+1}^S. However, keeping track of the entire context is unfeasible for real word applications, hence state-of-art recognition systems use n-gram models which only take a limited number of words into account. Usually this is done in the direction of text processing, i.e., for languages that are written and recognized from left-to-right, the left hand side context of a word is considered to estimate its probability. Here, we will indicate this with $LM_{\rightarrow}$ and call it *forward LM*

$$
p_{\rightarrow}(w_1^S|LM_{\rightarrow}) = p(w_1|LM_{\rightarrow}) \prod_{i=2}^{S} p(w_i|w_{i-n+1}^{i-1}, LM_{\rightarrow}) \ . \qquad (5)
$$

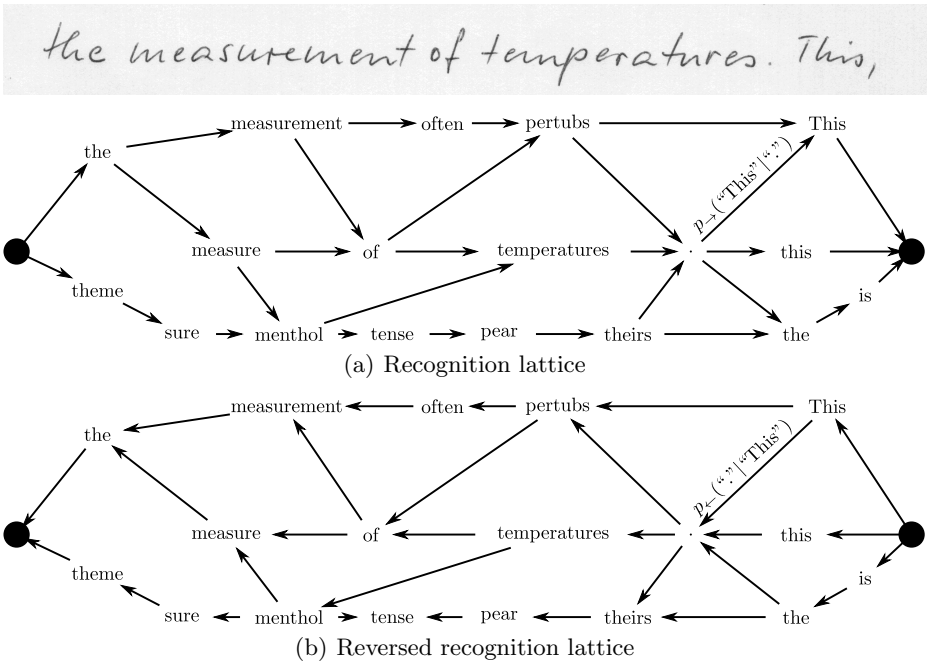


Fig. 1. In (a) the recognition lattice for the left-to-right decoding direction is given for a sample text line. In (b) the reversed lattice with the modified language model information is shown. Note that all node labels but only one edge label is shown for the sake of readability. In the left-to-right decoding, the bi-gram information that the word “This” occurs after the symbol “.” is used. In the right-to-left decoding, the corresponding edge contains the probability of the symbol “.” occurring before the word “This”.

Obviously, every text image can also be recognized in the reversed direction, requiring different n-grams, indicated here with $LM_{\leftarrow}$ (*backward LM*)

$$p_{\leftarrow}(w_1^S|LM_{\leftarrow}) = p(w_S|LM_{\leftarrow}) \prod_{i=1}^{S-1} p(w_i|w_{i+1}^{i+n-1}, LM_{\leftarrow}) \ . \tag{6}$$

Although the Equations (2) and (4) are a factorization of the same probability, their n-gram simplification in Equations (5) and (6) is expected to produce different results. Yet, both can be estimated on the same corpus. Thus, we propose in this paper to exploit this fact. We show that a significant improvement of the recognition rate can be achieved by combining the recognition output of the two systems using the forward LM and the backward LM of the same n-gram order.

2.2 Approach

We propose to generate two different N-best lists of recognition hypotheses, one generated by a left-to-right and one generated by a right-to-left decoding. Afterwards, these N-best lists can be combined to generate a new output.

A straightforward way to build both lists is to use a recognizer for handwritten text that produces a recognition lattice, such as HMMs or BLSTM Neural Networks in conjunction with a Token Passing algorithm. A recognition lattice (see Fig. 1), is a directed graph with node and edge labels and constitutes a comprehensive way of storing various decoding paths. From this, N-best lists can easily be generated by searching the most likely paths across the lattice using A^*-search. The exact specifications, what information is stored in the nodes and labels may vary, but usually a node represents a word and an edge indicates a transition between two words. In our approach, nodes are labeled with the position where the word ends. Edges are labeled with two probability scores, the bi-gram transition probability between the word at the starting node and the word at the ending node and the observation probability of the word at the ending node. From this, we generate the N-best lists of the forward direction.

Next, we reverse the directions of the edges and adjust the bi-gram probabilities. That is, an edge $e = (u, v) \in V \times V$ from node u to v labeled with $p_{\rightarrow}(v|u, LM)$ and $p_{obs}(v)$ is changed into an edge $e = (v, u)$ with labeling $p_{\leftarrow}(u|v, LM)$ and $p_{obs}(v)$. A path in the new lattice now represents a decoding using the reversed bi-gram language model and an N-best list is also generated. Note that the word ordering of the hypotheses in this list is in reversed order and needs to be changed back.

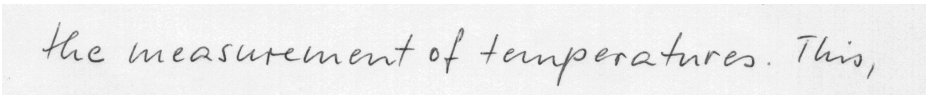
To make use of higher order N-grams, the bi-gram word transition probabilities on the edges are ignored. Instead, an A^*-search on the lattices is done using an external language model file to generate the forward and backward N-best lists.

Finally, the two N-best lists can be combined using a generalized *recognizer output voting error reduction* (ROVER) scheme [4, 16]. In this system, the N-best output word strings are first aligned and then combined in a weighted voting scheme. The weights of the word hypotheses in the combination are based on their posterior probabilities which are estimated from the N-best lists of the recognizers. The combination was done using the SRILM toolkit [15]. The toolkit allows the use of different weight parameters, which were optimized on the validation set.

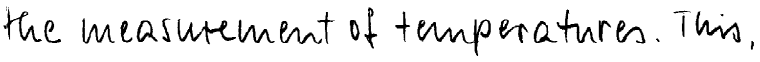
3 Experimental Evaluation

3.1 Setup

For the experiments, we have used the IAM off-line database [11], which contains forms of unconstrained handwritten English text. The database is composed of 1,539 pages (13,353 text lines, 115,320 words) written by 657 writers. In our experiments we have followed the benchmark defined by the authors, which



(a) The original text line image.



(b) The normalized text line.

Fig. 2. The text line preprocessing.

consists in 6,161 lines in the training set, and 920 lines in the validation set, and 2,781 lines in the test set.

First of all, each text line has been binarized and normalized in order to cope with different handwriting styles. The normalization consists in correcting the skew and slant, and normalizing the size and width of the text. The result of the text line normalization process can be seen in Fig. 2. Once the text lines are normalized, a sliding window moving from left to right over the text image. At each column of width one pixel, the following nine features are extracted: the 0th, 1st and 2nd moment of the black pixels' distribution within the window, the position of the top-most and bottom-most black pixel, the inclination of the top and bottom contour of the word at the actual window position, the number of vertical black/white transitions, and the average gray scale value between the top-most and bottom-most black pixel. For a more detailed description of the normalization and feature extraction, we refer to [10].

As a recognizer, a bidirectional LSTM neural network (BLSTM NN) is used, i.e., the sequence of feature vectors is fed into the network from both directions, left-to-right and right-to-left. The output layer consists of one node for each possible character. By normalizing the output activations, the result is a matrix of posterior probabilities for each letter and each position. Given that matrix and a bi-gram language model, a token passing algorithm can be used to generate the recognition lattices. For details about BLSTM networks and the CTC token passing algorithm, we refer to [6].

Both the forward and the backward N-grams with $N = 2, 3, 4$ are estimated on the union of the Brown and Wellington corpus [1, 8, 9] as well as the part of the LOB corpus not used in the validation or testing. The total amount of text is 3.34M words in 162.6K sentences.

We chose the dictionary to be the 20,000 most frequent English words. Since we consider the open vocabulary recognition task, some words in the training, validation, and test set do not occur in the dictionary and can not be recognized. This imposes an upper bound to the word recognition rate of 93.74%.

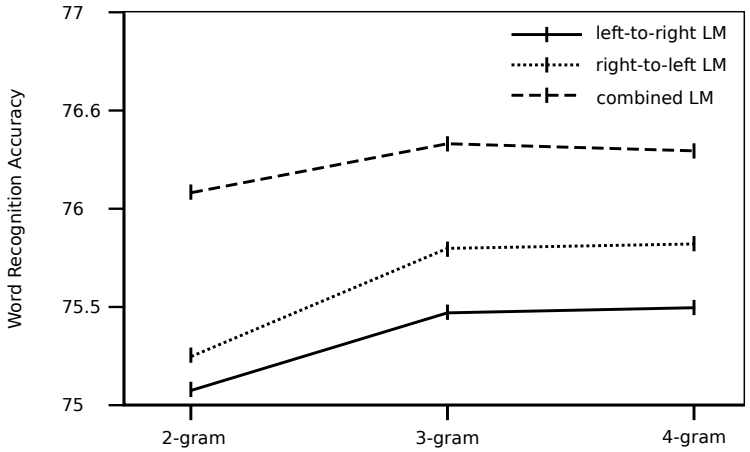


Fig. 3. Word level recognition accuracies of the different systems.

3.2 Results

In Fig. 3, the impact of the bidirectional language model on the handwriting recognition task can be seen. The solid line indicate the standard left-to-right language model and it can be seen that the recognition accuracy increases from 75.08% using a 2-gram LM, up to 75.47% (3-grams) and 75.50% (4-grams). The results using bi-grams are comparable to the ones found in [6, 3].

Using a right-to-left language model, the recognition rates are consistently higher by reaching 75.25% (2-grams), 75.80% (3-grams), and 75.82% (4-grams). The lack of significant increase when switching from a 3-gram to the 4-gram LM can be explained by the size of the language corpus. Obviously the limit of the generalization capability is reached.

The proposed, combined language model, however, achieves a significant increase by combining the left-to-right and right-to-left models. With bi-gram model, a recognition accuracy of 76.08% is reached, outperforming even the 4-gram recognition with an unidirectional model. The performance using the 3-gram combined model is 76.33% and slightly better than using the 4-gram (76.29%) combined model. However, this difference is not statistically significant, while all increases from the uni-directional models to the proposed bi-directional model is statistically significant at $\alpha = 0.05$ for every N.

4 Conclusion

The recognition of unconstrained handwritten text is still considered an open problem mainly due to the high variability in the handwriting styles. Since state-of-the-art handwriting recognition systems decode a text line sequentially, the contextual information used for solving ambiguities is only taken from one side

of a word. To increase the robustness of estimating a word's language model probability one the one hand and to reduce the effect of error-propagation of mis-recognized words, we propose bidirectional language models. In considering contextual information from both sides of a word, our approach may be seen as a step towards full sentence language models that capture the meaning of a text holistically.

The experimental results obtained with bidirectional n-grams have shown a significant improvement over current state-of-the-art approaches. The improvement has been achieved without increasing the amount of training data, language corpus, or the complexity of the language model.

Thus, we can conclude that bidirectional language models are promising approaches. Therefore, further work could be focused on investigating holistic whole sentence analysis with bidirectional grammars and context-free grammars.

Acknowledgements

The authors thank Alex Graves for kindly providing us with the BLSTM Neural Network source code and Oriol Ramos Terrades for insightful discussions. This work has been partially supported by the European projects FP7-PEOPLE-2008-IAPP and ERC-2010-AdG-20100407-269796, the Spanish projects TIN2011-24631, TIN2009-14633-C03-03, Consolider-Ingenio 2010 (CSD2007-00018), 2010 CONE3 00029, and the mobility research grant 10 BE-1 00020.

References

1. Laurie Bauer. Manual of Information to Accompany The Wellington Corpus of Written New Zealand English. Technical report, Department of Linguistics, Victoria University, Wellington, New Zealand, 1993.
2. H. Bunke, S. Bengio, and A. Vinciarelli. Offline Recognition of Unconstrained Handwritten Texts using HMMs and Statistical Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):709–720, 2004.
3. S. Espana-Boquera, M.J. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martínez. Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4):767–779, 2011.
4. Jonathan Fiscus. A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). In *Workshop on Automatic Speech Recognition and Understanding*, pages 347–354. IEEE, December 1997.
5. Joshua T. Goodman. A Bit of Progress in Language Modeling - Extended Version. Technical Report MSR-TR-2001-72, Microsoft Research, One Microsoft Way Redmond, WA 98052, 8 2001.
6. A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, 2009.
7. F. Jelinek. *Mathematical Foundations of Speech and Language Processing*, volume 138, chapter Stochastic Analysis of Structured Language Modeling, pages 37–71. Springer-Verlag, 2004.

8. Stig Johansson, Eric Atwell, Roger Garside, and Geoffrey Leech. The tagged lob corpus: Users' manual. Technical report, The Norwegian Computing Centre for the Humanities, 1986.
9. H. Kucera and W. N. Francis. *Manual of Information to accompany A Standard Corpus of Present-Day Edited American English, for use with Digital Computers.* Brown University, Department of Linguistics, Providence, Rhode Island, 1964. Revised 1971. Revised and amplified 1979.
10. Urs-Victor Marti and Horst Bunke. Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.
11. U.V. Marti and H. Bunke. The iam-database: An English Sentence Database for Offline Handwriting Recognition. *Int'l Journal on Document Analysis and Recognition*, 5(1):39–46, 2002.
12. R. Plamondon and S.N. Srihari. Online and Off-Line Handwriting Recognition: A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.
13. Thomas Plötz and Gernot A. Fink. Markov Models for Offline Handwriting Recognition: A Survey. *Int'l Journal on Document Analysis and Recognition*, 12(4):269–298, 2009.
14. Ronald Rosenfeld, Stanley F. Chen, and Xiaojin Zhu. Whole-Sentence Exponential Language Models: A Vehicle for Linguistic-Statistical Integration. *Computers, Speech and Language*, 15:55–73, 2001.
15. A. Stolcke. SRILM: An Extensible Language Modeling Toolkit. pages 901–904, 2002.
16. Andreas Stolke, Yochai König, and Mitchel Weintraub. Explicit Word Error Minimization in N-Best List Rescoring. In *EUROSPEECH*, pages 163–166, 1997.