ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Combining higher-order N-grams and intelligent sample selection to improve language modeling for Handwritten Text Recognition

Jafar Tanha, Jesse de Does and Katrien Depuydt *

Institute for Dutch Lexicology (INL)
Leiden, Netherlands
{jafar.tanha, jesse.dedoes, katrien.depuydt}@inl.nl

Abstract. We combine two techniques to improve the language modeling component of a Handwritten Text Recognition (HTR) system. On the one hand, we apply a previously developed intelligent sample selection approach to language model adaptation for handwritten text recognition, which exploits a combination of in-domain and out-of-domain data for construction of language models. On the other hand, we apply rescoring methods to enable more complex language modeling in HTR. It is shown that these techniques complement each other very well, and that the combination leads to a significant error reduction in a practical HTR task for historical data.

1 Introduction

An indispensable component of state-of-the-art hand-written text recognition (HTR) systems are language models [1] [2], which are necessary to guide the decoding step by ranking and constraining the possible word sequence hypotheses. Language models are usually constructed from large text corpora which – ideally – are *in-domain*, linguistically close to the language of the document collection which is being processed. However, for HTR of *historical documents*, obtaining effective models is much less straightforward: models built from the strictly in-domain data are generally unsatisfactory because not enough data can be obtained to avoid overfitting. Therefore, one can use *out-of-domain* data to improve the language model, but in order to exploit the larger pool of out-of-domain data one has to surmount two difficulties: (1) indiscriminate use of out-of-domain data may not benefit, in fact even deteriorate system performance and (2) the use of the complete out-domain data for training may increase the complexity of the system, making the decoding step almost untractable [3] [4].

The above-mentioned issues are typically dealt with by using *domain adaptation* or *language model adaptation* techniques [5, 6, 3, 4]. In this paper we consider the language model adaptation using a semi-supervised learning approach in the Co-Training [7] framework, which has been proposed by Tanha et. al. [8] for language modeling in HTR.

*The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 600707 - tranScriptorium.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Another important issue in language modeling for HTR systems is the use of higher order language models, which may substantially increase the computational cost of the decoding step of HTR system [9, 10]. Current HTR systems often rely on small-scale language models derived from the HTR training set [4, 10]. The use of bigram language models is not unreasonable in this situation. However, in order to exploit the information in the out-of-domain data to its full potential, more advanced models are essential. We enable the application of these techniques by applying re-scoring algorithms for recognition lattices (word graphs). In our experiments, we use the tranScriptorium HTR engine described in [11] on a set of digitised images of manuscripts written by the 18th and early 19th-century British philosopher Jeremy Bentham[1].Our experimental results show that the proposed methods produce a language model better matched to the in-domain data and also reduce the computational cost needed to exploit a large amount of out-domain data in the decoding step of HTR.

The rest of the paper is organised as follows. In Section 2, we briefly review the basics of language modeling in HTR, and section 3 introduces our co-training approach to language model adaptation. Section 4 gives the proposed method for using higher order N-gram model. Section 5 presents the results and Section 6 concludes the paper.

2 Hidden Markov Models and Language models in Text Recognition

In this section, we first address the role of language models in HTR. We then state the main challenge in the use of language model for HTR.

In a hand-written text line recognizer the goal is to recognise the most likely word sequence, $W = (w_1, ..., w_m)$, for a known observation sequence of text images, $X = (X_1, ..., X_l)$ given by the feature extraction process, as follows:

$$\hat{W} = \arg\max_{w} \quad p(word\ sequence|text\ image) \tag{1}$$

For simplicity, the resulting hidden markov model (HHM) is reformulated using the Bayes rule as:

$$\hat{W} = \arg\max_{w}\ p(text\ image|word\ sequence) \times p(word\ sequence) \tag{2}$$

The HMM-based recognizer used in this paper is supported by a statistical language model in the decoding step. The current HMM-based approaches to HTR systems typically utilize a statistical bigram language model during the decoding step [4, 10]. The main reason for that is the computational cost of the decoding step, which is substantially higher if trigram or higher language models are used in HTR. However, it is to be expected that trigram or higher language models will give better recognition performance. Especially in our context, where have extracted useful information from out-of-domain data, much will be lost if we cannot apply more sophisticated models.

[1]Images and transcriptions have been produced in the *Transcribe Bentham* project [12], http://www.ucl.ac.uk/transcribe-bentham

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

2.1 Word graphs

Based on the above formulation, instead of just returning the best scoring hypothesis, recognition lattices (word graphs) can be produced during the decoding step, where a recognition lattice is a data structure that represents different hypotheses of a hand-written text recogniser in a finite state network. The lattice typically represents the most promising subspace of recognition results produced by the decoding step.

3 Semi-Supervised Co-training

Co-training [7, 13] is one of the widely used semi-supervised learning methods [8] in practical domains. In co-training, two classifiers based on two views of data or different learning algorithms are trained in parallel and then unlabeled data are classified by the classifiers. Unlabeled instances that are labeled with high confidence by one classifier are used as training data for the other. This is repeated until none of the classifiers changes.

In order to be able to use the co-training framework for domain adaptation, we need to exploit a set of *in-domain* resources $\mathcal{B}$ and a set of *out-of-domain* resources $\mathcal{E}$. Without loss of generality, we assume a partitioning of the in-domain data $\mathcal{B}$ in two subsets $\mathcal{B}_0$ and $\mathcal{B}_1$ such that $|\mathcal{B}_0| < |\mathcal{B}_1| \ll |\mathcal{E}|$. The goal here is to find a subset $\mathcal{E}_1$ of out-of-domain $\mathcal{E}$. In the setting of our HTR experiments, $\mathcal{B}_0$ consists of the HTR training and test data and $\mathcal{B}_1$ is an in-domain set consisting of the available transcriptions of the collection.

3.1 The Disagreement-based Co-training algorithm

In this study we use the Disagree-Co algorithm [4], which gradually exploits a set of informative data from the out-of-domain data, using a disagreement-based approach. We start by training two language models LM_j (j=0,1) on $\mathcal{B}_0$ and $\mathcal{B}_1$. We consider these two language models as classifiers in the co-training framework. We then apply the trained models on the $\mathcal{E}$ collection and evaluate and rank the resources by means of a scoring criterion, which is a function of perplexity and number of Out Of Vocabularies (OOVs). The used algorithm then selects an informative subset $\mathcal{S}$ of high-confidence resources from the $\mathcal{E}$ collection for each language model. Next, Disagree-Co adds to the training material of the second model a set of resources which are in the high-confidence set for the first model, but not in the high-confidence set of the second model, and vice versa. After this, the training process is repeated until the stopping condition is reached. The pseudo-code of the Disagree-Co algorithm is presented in Algorithm 1.

4 Using higher order N-gram models

Figure 1 gives an overview of the proposed method to deploy the power of higher-order models in an efficient way. We first train the Hand-written Text Recog-

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8.
Available from http://www.i6doc.com/en/.

Algorithm 1 Disagree-Co

$conf \leftarrow 0$; // threshold is a pre-defined threshold for confidence measure;
$t \leftarrow 0$; // $max - iterations$ is the number of iterations;
While ($t < max - iterations$ and $conf < threshold$) do Begin
- Build LM_0 and LM_1 from $\mathcal{B}_0$ and $\mathcal{B}_0$;
- For each R_i in $\mathcal{E}$ do Begin
  - Evaluate $\mathcal{C}_0(R_i) \leftarrow R_i$ by LM_0;
  - Evaluate $\mathcal{C}_1(R_i) \leftarrow R_i$ by LM_1;
- Endfor
- $\mathcal{H}_0$:= a high-confidence subset of $\mathcal{E}$, selected by best values w.r.t. $\mathcal{C}_0$;
- $\mathcal{H}_1$:= a high-confidence subset of $\mathcal{E}$, selected by best values w.r.t. $\mathcal{C}_1$;
- $S_0 \leftarrow \mathcal{H}_0 - \mathcal{H}_1$; $S_1 \leftarrow \mathcal{H}_1 - \mathcal{H}_0$; $\mathcal{B}_0 \leftarrow \mathcal{B}_0 \bigcup S_1$; $\mathcal{B}_1 \leftarrow \mathcal{B}_1 \bigcup S_0$;
- $\mathcal{E} \leftarrow \mathcal{E} - (S_0 \bigcup S_1)$; $t \leftarrow t + 1$;

Output
Selected Resources from $\mathcal{E}$ collection;

nizer. It then generates the n-best hypotheses as word lattices using a bigram LM. In the mean time the N-gram ($N > 2$) language model is generated. Next, this language model is used to re-score the word lattices. The re-scored recognition lattices are then applied to evaluate the performance of HTR. Using this idea can substantially decrease the computational cost of the decoding step, because the re-scoring operation is much faster than full decoding.

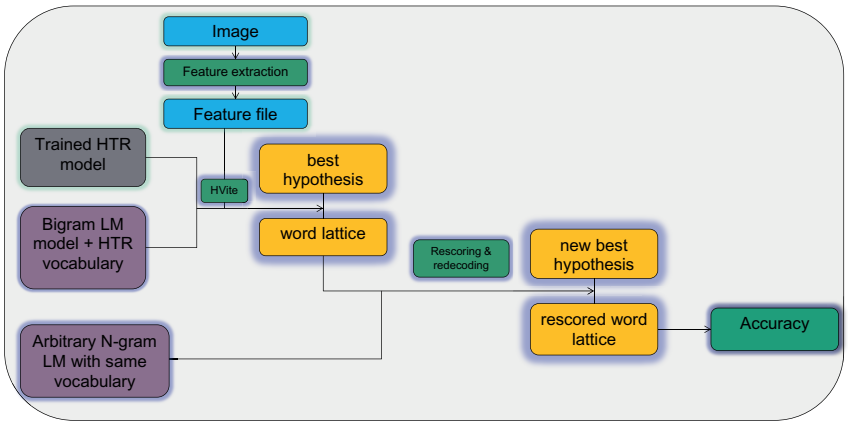

Fig. 1: The lattice rescoring approach to HTR

5 Experiment and Result

In this section we perform several experiments on linguistic resources to show the effect on the HTR system of the proposed methods for domain adaption and deployment of higher order language models. In order to evaluate the proposed methods, it is important to compare them to a strong baseline, in our case a well-tuned linear interpolation of in-domain and out-of-domain language models.

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

We make use of the English-language data processed in the tranScriptorium [11] project for the evaluation of HTR performance. This collection consists of a set of images and with ground truth transcriptions of Bentham manuscripts. Part of the ground truth transcriptions is used for language modeling, a held-out set is used for testing HTR. In addition to this, we use the corpus of all transcribed Bentham manuscripts (about 15.000 pages and 5m words), as obtained from the *Transcribe Bentham* project [12], and the public part of the ECCO (Eighteenth Century Collections Online), about 70m words.

5.1 Results

In this section we use the Bentham collection to compare the performance of the HTR system with our methods to the baseline. Table 1 shows the baseline results. For each experiment, we give the word error rate (WER) and the character error rate (CER). In the first experiment we also include the amount of OOV words. In each table the best results have been boldfaced. Table 1 shows that interpolating the language model from Bentham in-domain data with the language models from the Bentham out-of-domain and ECCO resources clearly improves the performance of the HTR system. In other words, these results emphasize that the out-of-domain data contains useful information.

Table 1: The results of the baseline methods for HTR system

Method	WER %	CER %	OOV %	Size of model
Initial model using only Batch 1 training set	23.89	9.9	9.44	1-gram=1894 2-gram=6641
Inter-InOut-Dic-InOut	19.43	8.3	-	1-gram=12966 2-gram=795029
Inter-InOutECCO-Dic-InOutECCO	**19.36**	**8.3**	5.4	1-gram=64416 2-gram=5811657

In the second experiment we improve on this setup in two ways: (1) we first apply the Disagree-Co algorithm for domain adaptation and use the resulting language model for interpolation, and (2) we then build a higher N-gram language model from the resulting resources of Disagree-Co and evaluate them. As shown in table 2, different N-gram models (N=3,4 and 5) have been evaluated. The best result is produced by 5-grams.

Table 2: The results of the Adapted LM for HTR system

Method	WER %	CER %	N-gram model
In-domain + Bigram LM	23.89	9.9	1-gram=7686 2-gram=32638
Adapted Bigram LM	18.83	8.1	1-gram=59547 2-gram=202736
Adapted Trigram LM	16.02	7.3	1-gram=134877 2-gram=1879559 3-gram=6418944
Adapted 4-gram LM	16.09	7.4	1-gram=134877 2-gram=1879559 3-gram=6418944 4-gram=11456738
Adapted 5-gram LM	**15.62**	**7.2**	1-gram=134877 2-gram=1879559 3-gram=6418944 4-gram=11456738 5-gram=12987653

ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015, i6doc.com publ., ISBN 978-287587014-8. Available from http://www.i6doc.com/en/.

6 Conclusion

We have studied and tested several ways in which well-tuned approaches to language modeling can improve hand-written text recognition results, when the resulting language models are deployed in the tranScriptorium HTR system. Our methods for the combination of an intelligent sample selection approach to exploition of out-of-domain training data with a rescoering approach to the deployment of the higher N-gram models obtained from this data, have been shown to yield significant improvement in HTR results.

References

[1] Thomas Plötz and Gernot A Fink. Markov models for offline handwriting recognition: a survey. *IJDAR*, 12(4):269–298, 2009.

[2] Salvador Espana-Boquera, Maria Jose Castro-Bleda, Jorge Gorbe-Moya, and Francisco Zamora-Martinez. Improving offline handwritten text recognition with hybrid hmm/ann models. *IEEE Transactions on PAMI*, 33(4):767–779, 2011.

[3] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *Proceeding of Conference on EMNLP*, pages 355–362, 2011.

[4] J. Tanha, J. de Does, and K. Depuydt. An intelligent sample selection approach to language model adaptation for hand-written text recognition. *the ICFHR conference, 2014*.

[5] George Foster, Cyril Goutte, and Roland Kuhn. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Conference on EMNLP, 2010*.

[6] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *ACL*, volume 2007, page 22, 2007.

[7] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *ICML*, pages 92–100. ACM, 1998.

[8] J Tanha. Ensemble approaches to semi-supervised learning. *Ph.D thesis, Informatics Institute, University of Amsterdam*, 2013.

[9] U-V Marti and Horst Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *J.Pattern Recognition and AI*, 15(01):65–90, 2001.

[10] Matthias Zimmermann and Horst Bunke. Optimizing the integration of a statistical language model in hmm based offline handwritten text recognition. In *ICPR*, volume 2, pages 541–544, 2004.

[11] Joan Andreu Sánchez, Günter Mühlberger, Basilis Gatos, Philip Schofield, Katrien Depuydt, Richard M Davis, Enrique Vidal, and Jesse de Does. transcriptorium: a european project on handwritten text recognition. In *Proceedings of the 2013 ACM symposium on Document engineering*, pages 227–228. ACM, 2013.

[12] Martin Moyle, Justin Tonra, and Valerie Wallace. Manuscript transcription by crowdsourcing: Transcribe bentham. *LIBER Quarterly*, 20(3), 2011.

[13] J. Tanha, M. van Someren, and H. Afsarmanesh. Disagreement-based co-training. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 803–810, Nov 2011.