A Priori and A Posteriori Integration and Combination of Language Models in an On-line Handwritten Sentence Recognition System

Solen Quiniou, Eric Anquetil

A Priori and A Posteriori Integration and Combination of Language Models in an On-line Handwritten Sentence Recognition System

Solen Quiniou *Éric Anquetil*
IRISA - INSA
Campus de Beaulieu
35042 Rennes Cedex, France
{Solen.Quiniou, Eric.Anquetil}@irisa.fr

Abstract

This paper investigates the integration of different language models into an on-line sentence recognition system. The impact of n-gram and n-class (based on statistically and on morpho-syntactically classes) models, built on the Brown corpus, is compared in terms of word recognition rate. Furthermore, their integration in different steps of the recognition process (during it or to rescore the N-best list of proposed sentences) is considered, thus showing better performances when used the sooner. Combinations of these models are also studied, in addition to the integration in the aforementioned recognition steps. All experiments are carried out on sentences from the Brown corpus which were written by several writers.

Keywords: on-line sentence recognition, statistical language models, model combination, N-best list rescoring.

1. Introduction

With the emergence of new devices like PDA's and Tablet PC's, users are able to write larger pieces of text. Handwriting recognition systems can thus take advantage of the linguistic context of the words to improve the recognition of these words.

Language models are used to represent such knowledge and essentially come from speech recognition where n-gram models (statistical language models) are the most commonly applied [12]. These models can also be put together with other language models which are often extensions of these n-gram models (like class based, skipping or caching models) [11]. The performances of these resulting combined models depend on their component models and also on the approach used to realize that combination, as well as on the corpus used to build the models. Another point which affects the overall performance is the step of the recognition process during which these models are incorporated.

In off-line sentence recognition, several works make use of n-gram language models [17, 18, 19]. In [18], a trigram model is incorporated during the recognition process, with a weight used to balance its relative influence toward the recognition system. This model is shown to reduce the word error rate to 18.2%. In [19], a stochastic context-free grammar is used to reorder a N-best sentence list produced by a recognition system including a bigram language model. The language models are combined using a log-linear interpolation and the use of the grammar lead to a 20.6 % word error rate whereas the system with only the bigram model achieved a 20.7 % rate. This improvement is very small since the recognition system is already optimized thanks to the bigram model.

In on-line sentence recognition, [13] uses a model which combines two class-based models, one using statistical classes and the other morpho-syntactical ones. These models are combined with a simple interpolation and the resulting model is used a posteriori to rescore the N-best sentence list from the recognition system. The use of only statistical classes reduced the word error rate from 34 % to 23 % whereas the model combining both types of classes lead to a 22.5 % word error rate. Nonetheless, the impact of the language model toward the recognition system is not optimized here.

In [14], we have addressed the integration of different kinds of language models, built on a relatively small corpus, into our on-line recognition system which lead to a significant word error rate reduction (from 17.5 % to almost 9.7 % with n-gram or n-class models), using a language weight to balance the influence of this model towards the recognition system. This current work relates first experiments carried out on a larger corpus (the Brown corpus) and we will first compare the impact of language models built on the Susanne corpus to the ones built on the Brown corpus. The larger size of this corpus would also allow investigations on language models with longer size histories. We have also extended our models to combined ones and we will discuss different strategies to incorporate such combined models into the recognition system.

The remaining parts of this article are the following. The statistical language modeling is described in section 3 after the presentation of the sentence recognition problem in section 2. Then, the approach used to combine language models is related in section 4. In section 5, an overview of the recognition system is given while the experimental results are displayed in section 6. Finally, section 7 draws some conclusions.

2. Sentence recognition problem

A sentence recognition system aims at retrieving the most likely sentence $\hat{W}$ between candidate sequences $W = w_1 \dots w_n$ given a signal S (the handwritten sentence to recognize).

Classically, we have:

$$\hat{W} = \arg\max_{W} p(W|S). \quad (1)$$

Since the probabilities $p(W|S)$ of equation 1 are small, their decimal logarithms are used instead and these probabilities can be decomposed as follow:

$$\log[p(W|S)] = \log[p(S|W)] + \gamma \log[p(W)] \quad (2)$$

where $p(S|W)$ is the a posteriori probability of the signal S for the given sentence W and is estimated by the recognition system often based on HMM's (we call this term *graphical model*), $p(W)$ is the a priori probability of the sequence W, often given by a statistical *language model* and γ (also called *Grammar Scale Factor*) is introduced to balance the influence of the language model against the graphical model.

Since our recognition system is non-probabilistic (see section 5), equation 1 is replaced by:

$$\hat{W} = \arg\max_{W} score(W|S) \quad (3)$$

with

$$score(W|S) = score(S|W) + \gamma \log[p(W)] \quad (4)$$

where $score(S|W)$ is given by our word recognition system and whose values have the same order of magnitude than log-probabilities.

3. Statistical language modeling

Statistical language modeling aims at capturing regularities of a language by use of statistical inference on a corpus of that language [12]. The a priori probability of a n words sentence $W = w_1^n = w_1 \dots w_n$ is thus given by:

$$p(W) = \prod_{i=1}^{n} p(w_i|h_i) \quad (5)$$

where $h_i = w_1 \dots w_{i-1}$ is called *history* of word i.

The main problem with equation 5 is the high number of histories leading to a tremendous number of probabilities to estimate. Furthermore, most of these probabilities occur too few times to be estimated reliably. A solution to issue this problem is to merge histories in equivalence classes:

$$p(W) = \prod_{i=1}^{n} p(w_i|h_i) = \prod_{i=1}^{n} p(w_i|\Phi_i(h_i)) \quad (6)$$

where $\Phi_i(h_i)$ assigns to history h_i its equivalence class.

There are several techniques to define $\Phi_i(h_i)$, the simplest one being n-gram language models.

3.1. N-gram language models

N-gram language models merge histories ending with the same n-1 words, into equivalence classes:

$$p(W) = \prod_{i=1}^{n} p(w_i|w_{i-n+1}^{i-1}). \quad (7)$$

The probability $p(w_i|w_{i-n+1}^{i-1})$ given by equation 7 is the relative frequency of the sequence w_{i-n+1}^{i} in a corpus:

$$p(w_i|w_{i-n+1}^{i-1}) = \frac{N(w_{i-n+1}^{i})}{N(w_{i-n+1}^{i-1})} \quad (8)$$

where $N(.)$ stands for the number of occurrences of a certain event. One issue of this approach is that the model fits to the training corpus and probabilities of non-occurring n-grams (i.e. sequences of n words) are estimated to zero. Moreover, the longest the size of the history is, the more data are needed because of the large number of probabilities to estimate. Thus, larger corpuses are needed as the size of histories is increased as well as techniques to take into account the probabilities of unseen n-grams.

One solution to solve this latter problem is called *smoothing*. It first reduces probabilities of n-grams occurring in the corpus, then redistributes this mass of probabilities among n-grams never encountered. Among different smoothing techniques we chose the Kneser-Ney modified interpolated method, shown in [8] to be very efficient. Nonetheless one limit of this approach is that non-zero probabilities will be assigned to n-grams impossible from a linguistic point of view.

3.2. N-class language models

N-class models merge words into classes. There are two main approaches to define those word classes: based on statistical criteria or on predefined classes.

3.2.1. Statistical classes

In that case, classes are created by merging words which share the same context. Each word thus belongs to exactly one class and its probability is based on its class and on those of the previous words:

$$p(w_i|w_{i-n+1}^{i-1}) = p(w_i|c_i)\, p(c_i|c_{i-n+1}^{i-1}) \quad (9)$$

where $p(w_i|c_i)$ is the probability of the word w_i in its class c_i and $p(c_i|c_{i-n+1}^{i-1})$ is the probability of the class c_i to occur given the history of classes c_{i-n+1}^{i-1}.

To create these classes, we use the incremental version of the Brown algorithm [5].

3.2.2. Predefined classes

Here, the classes correspond to defined categories which are often the grammatical nature of words (i.e. Part-Of-Speech or *POS* tags). The main difference with the previous approach is that each word can belong to several classes since the grammatical nature of a word depends on its context. The probability of a word is thus based on its

classes and on those of the previous words as well, leading to an extension of equation 9:

$$p(w_i|w_{i-n+1}^{i-1}) = \sum_{c_i \in C_i} p(w_i|c_i)\, p(c_i|c_{i-n+1}^{i-1}) \qquad (10)$$

where c_i is one class of word w_i among its class set C_i. This sum is performed efficiently using the forward-backward algorithm [2].

When POS tags are considered as predefined classes, a tagged corpus is needed, where each word is given with its class (among all its possible ones) according to its context. For our experiments, we use the tagged version of the Brown corpus [10].

4. Language models combination

The interest in combining language models is to take advantage of their specificities and thus to outperform the best of them [11]. Since we work on log-probabilities, one method to combine the models is the log-linear interpolation [4, 19]. Furthermore, this method allows a better synthesis of the models w.r.t. the simple interpolation, especially when the models have different orders of magnitude. Another difference with the simple interpolation is that the combined value is no longer a probability and has to be normalized over all words of the vocabulary. Nonetheless, this normalization is usually omitted in the field of speech recognition (because of its small effect) and since our recognition system is not probabilistic (which means that we don't need to combine real probability with the recognition system score), we will discard it too. This combination can be viewed as a weighted sum of the language models log-probabilities:

$$\log\left[p_{combin}(w_i|w_{i-n+1}^{i-1})\right] \approx \sum_{m=1}^{M} \lambda_m \log\left[p_m(w_i|w_{i-n+1}^{i-1})\right] \qquad (11)$$

where all λ_m values are tuned on a validation set to optimize either the perplexity or the word recognition rate. In the experiments, we settle the values on a validation set, w.r.t. the word recognition rate, in order to optimize them for our recognition system (see section 6).

After explaining statistical language modeling and models combination, we present our sentence recognition system. We will especially focus on the integration and on the combination of language models into it.

5. Overview of the handwritten sentence recognition system

Our on-line sentence recognition system presented in figure 1 extends our word recognition system RESIFMot [6] which we first describes.

5.1. Word recognition system

This system is based on an analytic approach. Words are segmented according to different hypotheses of letter allographs which are organized in a segmentation graph.

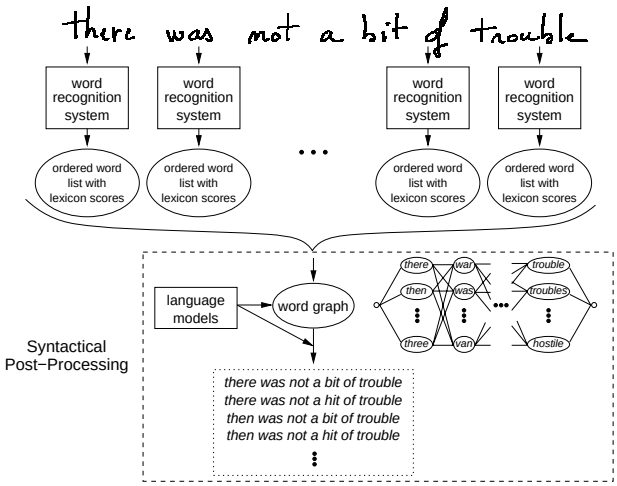

Figure 1. Sentence recognition system.

An adapted version of our character recognition system RESIFCar [1] is used to validate the correct segmentation hypotheses and to produce each allograph hypothesis. An ordered list of character strings is then produced by the exploration of this graph. These strings are ranked according to a score combining the adequation measure between the allographs and the letter models, the spatial coherence between each pair of consecutive allographs and statistical information on character n-grams.

Afterwards, a lexical post-processing step is performed to retrieve the nearest word of each string hypothesis in a dictionnary [7]. Thus, this word recognition system outputs a list of words ranked according to a *lexicon score* depending on edit operations used to transform the character string into the corresponding word.

5.2. Sentence recognition system

In the sentence recognition system, each handwritten word of the segmented input sentence is given to the prior word recognition system which outputs the corresponding ordered list of candidate words for each handwritten word. A word graph is then built using these lists where each node represents a $(n-1)$-gram w_{i-n+1}^{i-1} (and is valuated by the likelihood of word w_{i-1}) and each edge corresponds to the n-gram w_{i-n+1}^{i} (and is valuated by one or several *language model probabilities*). The Viterbi algorithm [9] is then performed to find the N-likeliest paths in the word graph (each path corresponding to a sentence), by combining graphical and linguistic information, and produce the N-best list of sentences. There are actually 3 kinds of such combination:

- all language models are directly used during the Viterbi search, in conjunction with the word likelihoods. This is an optimal use of the language models since they directly participate in the selection of the words of the recognized sentence. Nevertheless, it becomes expensive with long range history models because, as an edge represents a $(n-1)$-

gram, the total number of edges in the graph becomes tremendous;

- only the word likelihoods take part in the Viterbi algorithm to produce the N-best sentence list and then the language models are used to rescore the hypotheses of this list. This is the simplest approach to use language models and the size of the word graph remains reasonable since an edge stands for a word. The main drawback of this approach is that the improvement brought by the use of the language models is strongly dependent on the sentences of the N-best list produced by the recognition system;

- the latter approach is somewhere between the two previous ones. Indeed, some language models are used during the Viterbi search while the remaining models take part in the N-best list rescoring. Thus, this approach tries to take into account the advantages of both previous approaches. The idea is to use models with small size histories to generate the N-best list of sentences (thus models a priori included) and then to reorder them, using models with longer size histories (models a posteriori integrated). Nonetheless, the efficiency of models with longer size histories depends on the percentage of n-grams appearing in the sentences and actually estimated in the language model.

We will compare these 3 approaches in the next section as well as other experimentations, after presenting the linguistic and handwritten data we use.

6. Experiments and results

6.1. Data

The language models and the lexicon were extracted from the Brown corpus [10] with the SRILM toolkit [16]. The lexicon includes 13,748 words and the corpus contains 52,954 sentences (1,002,675 words) where 46,836 sentences (900,108 words) were actually used for the learning of the models. For the POS language models, we use the tagged version of the Brown corpus, containing 145 POS classes.

The handwritten material consists of 118 different sentences from the remaining part of the Brown corpus and were written by several writers. These sentences were segmented manually to introduce no bias due to incorrect segmentations.

The validation set includes 179 sentences (2,930 words) written by 7 writers (this set is used to tune the global language weight and the weight of each language model when a combined model is built) whereas the test set includes 260 sentences (4,137 words) written by 7 writers different from the ones of the validation set.

6.2. Experiments

Compared to our previous work [14], we stand here in a real context for the experiments. Indeed, a validation set is used to tune the optimal value of the global language weight γ (see equation 4) as well as the values of the λ_m (see equation 11) used to combine langage models with log-linear interpolation (all values are taken from 0 to 1, with a 0.1 step). Results are then given on the test set, using these optimal values.

We first compare n-gram and n-class models built on the larger corpus and integrated a priori on the recognition system. Then, we present the a priori integration of a model combining statistical and morpho-syntactical class based models. Finally, we focus on longer range language models and their a posteriori use to rescore N-best sentence list, produced by the recognition process using the best a priori language model from the first experiments.

6.2.1. Comparison between a priori integrated language models

Here, we compare different language models, in terms of number of parameters (i.e. the number of probabilities in the model with also the number of correspondances between words and classes, in the case of class-based models) as well as of word recognition rate. Two kinds of word recognition rates are computed: the first one w.r.t. the total number of words of the sentences to recognize in the test set and the second one w.r.t. only words actually present in the candidate word lists from the word recognition system (see figure 1). Indeed, in some of the word lists given by the word recognition system, the correct word to recognize doesn't appear which makes it impossible to recognize. This measure better represents the impact of the language model in the word recognition system. These results are given in table 1.

Table 1. Word recognition rates for n-gram and n-class language models.

Model	Word rec. rate	Pres. word rec. rate	Nb. of param.
Baseline	83.53 %	87.43 %	-
Bigram	91.72 %	96.00 %	414,640
Trigram	92.30. %	96.64. %	479,093
Biclass (1,500)	91.79 %	96.07 %	203,116
Triclass (1,500)	92.45 %	96.10 %	271,594
Biclass (100)	90.89 %	95.13 %	23,908
Triclass (100)	91.11 %	95.35 %	112,747
Biclass (POS)	90.60 %	94.83 %	49,812
Triclass (POS)	90.31 %	94.55 %	71,938

The rise in the word recognition rate is the most important with the trigram model (a 53.24 % word error rate reduction over all the words and 73.26 % over only the present words). Although the increase with the bigram model is slightly under the one achieved with the trigram model (a word error rate decrease of 49.73 % and of 68.18 % over only present words), this model will be preferred because of its smaller number of parameters. The fact that the improvement with the trigram model is small w.r.t. the bigram model can be due to the small repre-

sentation of trigrams from the test set within the language model (11.30 % of them, compared to 31.04 % of the bigrams from the test set and effectively estimated by the bigram model) and the fact that trigrams appearing only once in the learning corpus are discarded.

Now, concerning the class-based models, statistical classes perform better than morpho-syntactical ones (which was shown in [11, 13]). The biclass model with 100 statistical classes achieves better performances than the biclass one with 145 POS classes although the first model has twice as less parameters as the latter one. The same conclusions can be drawn for the corresponding triclass models but the triclass model with POS classes owns less parameters than the one with 100 statistical classes.

Finally, the biclass and triclass models with 1,500 statistical classes achieve word recognition rates slightly above the ones of both bigram and trigram models. This can be due to a better representation of n-classes from the test set and actually in the corresponding n-class model. In fact, there are 34.00 % of such biclasses and 13.58 % of such triclasses. Moreover, these n-class models include twice as less parameters as the corresponding n-gram ones. Thus, the biclass model with 1,500 statistical classes performs the best tradeoff between performance and compactness and allows a 50.15 % decrease of the word error rate and a 68.42 % decrease of the word error rate over actually present words.

These experimentations on language models built on a large corpus confirm the conclusions drawn in [14] with models built on a smaller corpus.

6.2.2. A priori combination of language models

Because the information used to create the statistical and POS classes are from different nature, we want to combine the two kinds of biclass models and to a priori integrate them i.e. during the recognition process. We compare the performance of this composite model towards the components models as well as w.r.t. an a posteriori integration of this composite model or of its components (the composite model is created using equation 11). The a posteriori integration of a language model uses this model to rescore the N-best list of sentence hypotheses given by the recognition system. Here, N is set to 100.

Table 2. Word recognition rate for a priori and a posteriori combined language models.

Model	A priori integration	A posteriori integration
Biclass (1,500) & biclass (POS)	91.81 %	88.47 %
Biclass (1,500)	91.79 %	88.47 %
Biclass (POS)	90.60 %	87.65 %

Table 2 gives the word recognition rate for the different integration of composite or component models. Concerning the a priori integration of the combined model, the achieved rate is almost the one obtained with only the biclass model with 1,500 statistical classes. Furthermore, the values which weight the impact of the models in the combined one are 0.9 for the statistical class based model and 0.1 for the morpho-syntactical class based one. This explains the small impact of the latter model.

Now, compared to the a posteriori integration, the a priori integration performs in a really better way since the word recognition rate for each model is almost 3 % above the one with this a posteriori integration. This highlights the fact that the language model has to be used as soon as possible during the recognition process. Indeed, when it is used a posteriori, i.e. to rescore the N-best list of sentences, its performances strongly depend on the sentences produced by the recognition system.

6.2.3. A posteriori integration of long range models

Until now, we use language models with a one or two word history (i.e. bigram and trigram kind of models) which may be inadequate to model longer dependencies. A simple solution to issue this is to increase the history length. Nonetheless, an a priori incorporation of such models during the word graph exploration is computationally expensive. As was shown in section 5.2, one solution would be to use these long history models a posteriori in conjunction with an a priori integrated model which participates in the selection of the N-best sentences. Thus, the probability of a sentence can be either a combination of the probability given by the a priori model and of the one from the a posteriori model or only the probability from the a posteriori model (leading to a combined word recognition rate and a word recognition rate, respectively).

Table 3. Word recognition rates and n-gram coverage for longer history language models.

Model	Comb. word rec. rate	Word rec. rate	n-gram coverage
4-gram	91.79 %	91.96 %	1.24 %
5-gram	91.79 %	91.98 %	0.23 %
4-class (1,500)	91.16 %	91.79 %	1.46 %
5-class (1,500)	91.14 %	91.81 %	0.27 %
4-class (100)	91.50 %	90.94 %	7.55 %
5-class (100)	91.45 %	90.97 %	1.49 %
4-class (POS)	90.94 %	89.66 %	11.79 %
5-class (POS)	90.51 %	89.05 %	6.05 %

Table 3 gives the word recognition rates for n-gram and n-class language models with a three or four words history used a posteriori to rescore the 100-best list of sentences produced by a sentence recognition system including a biclass model based on 1,500 statistical classes. This table also includes the coverage of n-grams of the considered order i.e. the percentage of n-grams of such order appearing in the test set and actually estimated by each language model. For example, only 1.24 % of the 4-grams in the test set are actually in the 4-gram language model.

The combined word recognition rate is better than the word recognition rate for models that are less accurate

than the a priori language model (it's the oppposite for more accurate models). Indeed, the 4-class and 5-class models with 100 statistical classes and with POS classes has less classes than the biclass model with 1,500 classes leading to less accurate language models. The combined word recognition rate achieved by these latter n-class models is above the one with only the corresponding biclass or triclass model but is under the one with only the a priori language model. In the same way, the word recognition rate obtained by the more accurate models (i.e. 4-class and 5-class models with 1,500 classes and 4-gram and 5-gram models) are above the rate achieved with the a priori model and also to the one for the corresponding bigram or biclass model but under the one with the corresponding trigram or triclass model. These results can be explained by the small coverage of the corresponding n-grams or n-classes from the test set and actually estimated by the corresponding language models.

These experiments show that long range order models are useful only if there is enough data to estimate them and if the n-grams within the models actually appear in the test set. To issue this, the language models should be built on larger corpuses and the test set should include more different sentences. Indeed, there are only 118 different sentences in our current test set, which reduces the number of different n-grams especially of higher order. The number of considered sentences could also be questioned.

7. Conclusion

This paper investigated the integration of different kinds of statistical language models at different steps of the recognition process. These models were built on a relatively large corpus and we see that n-class models based on statistically built classes achieves the best tradeoff between word error rate decrease and number of parameters (i.e. compactness). The same conclusions have already been drawn from our previous work on models estimated on a smaller corpus. We also addressed combinations of small history models which lead to no significant improvements. Finally, we saw that the use of the language models only for the N-best sentence list reordering is not optimal and that the language model had to be integrated as soon as possible. Thus, we studied the use of long history models for this N-best list rescoring task in conjunction with a small history model included in the recognition step. The success of this integration was shown to be strongly dependent on the percentage of n-grams of the test set actually estimated by the different language models. Since this percentage was very small, the performances were not improved.

Future works will include the use of a larger number of different handwritten sentences in the test set as well as the estimation of the language models on even larger corpuses. This would lead to a larger amount of n-grams appearing in the test set and being effectively present in the language model, thus allowing significantly better results with higher order models. We will also investigate the use of more grammatical or structural models for the N-best list rescoring which enable better representations of long linguistic dependencies [3, 19]. Furthermore, it might be interesting to align the hypotheses of the N-best list into a word transition network [15] which is more compact and also allows the generation of new sentences with the words of this hypotheses. In addition, we will study the optimal number of sentences to take into account in the N-best list (i.e. the value of N).

References

[1] E. Anquetil and H. Bouchereau, "Integration of an On-line Handwriting Recognition System in a Smart Phone Device", *16th ICPR*, 2002, pp 192–195.

[2] L. Baum, "An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Processes", *Inequalities*, 3:1–8, 1972.

[3] R. Beutler, T. Kaufmann and B. Pfister, "Using Rule-Based Knowledge to Improve LVCSR", *30th ICASSP*, 2005, pp 829–832.

[4] S. Broman and M.Kurimo, "Methods for Combining Language Models in Speech Recognition", *9th Eurospeech*, 2005, pp 1317–1320.

[5] P. Brown, V. D. Pietra, P. de Souza and J. Lai, "Class-Based N-Gram Models of Natural Language", *Computational Linguistics*, 18(4):467–479, 1992.

[6] S. Carbonnel and E. Anquetil, "Lexical Post-Processing Optimization for Handwritten Word Recognition", *7th ICDAR*, 2003, pp 477–481.

[7] S. Carbonnel and E. Anquetil, "Lexicon Organization and String Edit Distance Learning for Lexical Post-Processing in Handwriting Recognition", *9th IWFHR*, 2004, pp 462–467.

[8] S. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", Technical Report TR-10-98, Harvard University, 1998.

[9] G. Forney, "The Viterbi Algorithm", *Proceedings of the IEEE*, 61(3):268–278, 1973.

[10] W. Francis and H. Kucera, "Brown Corpus Manual", Brown University, 1979.

[11] J. Goodman, "A Bit of Progress in Language Modeling", Technical Report MSR-TR-2001-72, Microsoft Research, 2001.

[12] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, MIT Press, 1999.

[13] F. Perraud, C. Viard-Gaudin, E. Morin and P.-M. Lallican, "Statistical Language Models for On-Line Handwriting Recognition", *IEICE Transactions on Information and Systems*, E88-D(8):1807–1814, 2005.

[14] S. Quiniou, E. Anquetil and S. Carbonnel, "Statistical Language Models for On-line Handwritten Sentence Recognition", *8th ICDAR*, 2005, pp 516–520.

[15] H. Schwenk and J.-L. Gauvain, "Combining Multiple Speech Recognizers using Voting and Language Model Information", *6th ICSLP*, 2000, pp 915–918.

[16] A. Stolcke, "SRILM - An Extensible Language Modeling Toolkit", *7th ICSLP*, 2002, pp 901–904.

[17] A. Vinciarelli, S. Bengio and H. Bunke, "Offline Recognition of Unconstrained Handwritten Texts using HMMs and Statistical Language Models", *IEEE Transactions on PAMI*, 26(6):709–720, 2004.

[18] M. Zimmermann and H. Bunke, "N-Gram Language Models for Offline Handwritten Text Recognition", *9th IWFHR*, 2004, pp 203–208.

[19] M. Zimmermann, J.-C. Chappelier and H. Bunke, "Offline Grammar-Based Recognition of Handwritten Sentences", *IEEE Transactions on PAMI*, 28(5):818–821, 2006.