

# Rejection Strategies in Handwriting Recognition Systems

**Diplomarbeit**  
der Philosophisch-naturwissenschaftlichen Fakultät  
der Universität Bern

vorgelegt von

Roman Bertolami

2004

Leiter der Arbeit:

Prof. Dr. Horst Bunke  
Institut für Informatik und  
angewandte Mathematik



# Abstract

This master thesis investigates multiple rejection strategies for offline handwritten sentence recognition. The rejection strategies are implemented as a post-processing step of a Hidden Markov Model based text recognition system, and are based on confidence measures derived from a list of additional candidate sentences produced by the recogniser. Four different reject models are presented and three different sources of candidate sentences are investigated. Experimental results on extracted sentences from the IAM database validate the effectiveness of the proposed rejection strategies.



# Acknowledgement

I would like to dedicate some words to the people who contributed to this thesis.

I owe many thanks to Prof. Dr. Horst Bunke for supervising my thesis.

For introducing me to the field of handwritten text recognition, and for supporting me with important information and interesting discussions, I am extremely grateful to Dr. Matthias Zimmermann.

Furthermore, I would like to thank Andreas Schlapbach for helping me write this thesis.

For many interesting and fruitful discussions about, but not limited to, computer science and pattern recognition, during these years of studying, I would like to thank my colleague and friend Gregor Gabriel.

Many thanks to Anita Bertolami and Shiva Grings for proof reading, and enlarging my knowledge of the English language.

Moreover, I owe thanks to my parents Dario and Beatrice Bertolami-Ender who, with their financial support, enabled me to go to university.

Finally, I would like to thank Andrea Gäumann for her love and support during all these years.



# Contents

<b>1</b>	<b>Introduction</b>	<b>15</b>
1.1	Motivation . . . . .	15
1.2	Problem Statement and Goal . . . . .	16
1.3	Contribution . . . . .	16
1.4	Structure of the Thesis . . . . .	17
<b>2</b>	<b>Handwriting Recognition and Rejection</b>	<b>19</b>
2.1	Offline Handwritten Text Recognition . . . . .	19
2.2	Recognition System . . . . .	21
2.2.1	Text Line Normalization . . . . .	22
2.2.2	Feature Extraction . . . . .	23
2.2.3	Recognition using Hidden Markov Models . . . . .	23
2.3	Rejection Strategies . . . . .	23
2.3.1	Rejection System Architecture . . . . .	24
2.3.2	Bayes' Decision Theory . . . . .	25
2.3.3	Confidence Measure . . . . .	25
2.4	System Overview . . . . .	26
2.5	Sentence Comparison . . . . .	27
2.5.1	Aligning one Sentence Against Another . . . . .	27
2.5.2	Comparison of two sentences . . . . .	28
2.5.3	Handling more than Two Sentences . . . . .	29
2.6	Related Work . . . . .	30
2.6.1	Offline Handwriting Recognition . . . . .	31
2.6.2	Online Handwriting Recognition . . . . .	31
2.6.3	Speech Recognition . . . . .	32

<b>3</b>	<b>Reject Models</b>	<b>35</b>
3.1	Model 0: Single Alternative . . . . .	35
3.1.1	Confidence Measure . . . . .	36
3.1.2	Example . . . . .	36
3.2	Model 1: Multiple Alternatives . . . . .	37
3.2.1	Confidence Measure . . . . .	37
3.2.2	Example . . . . .	38
3.3	Model 2: Considering the Current Word . . . . .	40
3.3.1	Confidence Measure . . . . .	40
3.3.2	Example . . . . .	41
3.4	Model 3: Multi-Layer Perceptron . . . . .	43
3.4.1	MLP Architecture . . . . .	44
3.4.2	Feature Vector Acquisition . . . . .	45
3.4.3	Combining Multiple Multi-Layer Perceptrons . . . . .	46
3.4.4	Confidence Measure . . . . .	47
<b>4</b>	<b>Alternative Sentences Generation</b>	<b>49</b>
4.1	Recognition Lattice . . . . .	50
4.2	<i>N</i> -Best List Extraction . . . . .	50
4.3	Language Model Variation . . . . .	52
4.3.1	Language Model Integration . . . . .	53
4.3.2	Variation of GSF and WIP . . . . .	54
4.3.3	Example . . . . .	55
<b>5</b>	<b>Experiments and Results</b>	<b>57</b>
5.1	Evaluation Methodology . . . . .	57
5.1.1	Confusion Matrix . . . . .	57
5.1.2	Performance Measures . . . . .	58
5.1.3	Statistical Background . . . . .	59
5.1.4	ROC Curve Plot . . . . .	59
5.1.5	Error-Reject Plot . . . . .	60
5.2	Experimental Setup . . . . .	61
5.2.1	Offline Handwriting Recognition System . . . . .	62
5.2.2	Database . . . . .	62
5.2.3	Optimizing the Language Model Variations . . . . .	63



<i>CONTENTS</i>	9
5.2.4 Multi-Layer Perceptron Optimization . . . . .	64
5.2.5 Training and Smoothing . . . . .	65
5.3 Test Set Results . . . . .	69
5.3.1 Model 0 Results . . . . .	70
5.3.2 Model 1 Results . . . . .	71
5.3.3 Model 2 Results . . . . .	73
5.3.4 Model 3 Results . . . . .	75
5.3.5 N-Best Lists . . . . .	77
5.3.6 Variation of GSF . . . . .	77
5.3.7 Variation of GSF and WIP . . . . .	79
5.4 Discussion . . . . .	79
5.4.1 Reject Models . . . . .	79
5.4.2 Alternative Candidate Sentence Sources . . . . .	80
5.4.3 General Remarks . . . . .	81
<b>6 Conclusion and Outlook</b>	<b>83</b>
6.1 Conclusion . . . . .	83
6.2 Outlook . . . . .	84
<b>Bibliography</b>	<b>89</b>



# List of Figures

2.1	Example of handwritten text. . . . .	20
2.2	Example of different writing styles. . . . .	22
2.3	Handwriting recognition system overview. . . . .	26
2.4	Rejection procedure overview. . . . .	27
2.5	Alignment examples. . . . .	28
2.6	Sentence comparison example. . . . .	29
2.7	Alignment example with six sentences. . . . .	30
3.1	Rejection with a single alternative candidate sentence. . . . .	36
3.2	Counting the number of times $n$ , a hypothesised word occurs in alternative candidate sentences. . . . .	39
3.3	Example of a Multi-Layer Perceptron. . . . .	45
3.4	Example of a feature vector acquisition. . . . .	45
3.5	Cross validation process. . . . .	46
4.1	Example of a (pruned) recognition lattice. . . . .	51
4.2	Example of an $n$ -best list extraction. . . . .	52
4.3	Candidate sentences based on language model variation. . . . .	56
5.1	Confusion matrix. . . . .	58
5.2	Decision landscape. . . . .	60
5.3	Receiver-Operating-Characteristic (ROC) curve example. . . . .	61
5.4	Error-reject plot example. . . . .	62
5.5	Multi-Layer Perceptron validation. . . . .	65
5.6	Relative frequencies $p(c n)$ obtained from the training set. . . . .	66
5.7	Probabilities $p(n c)$ estimated on the training set. . . . .	67
5.8	Distribution of training samples. . . . .	68

5.9	ROC Curve Plot of Model 0. . . . .	70
5.10	Error-Reject Plot of Model 0. . . . .	71
5.11	ROC Curve Plot of Model 1. . . . .	72
5.12	Error-Reject Plot of Model 1. . . . .	73
5.13	ROC Curve Plot of Model 2. . . . .	74
5.14	Error-Reject Plot of Model 2. . . . .	74
5.15	ROC Curve Plot of Model 3. . . . .	76
5.16	Error-Reject Plot of Model 3. . . . .	76
5.17	ROC Curve Plot of Model 1, Model 2, and Model 3 with alternative candidate sentences based on $n$ -best list extraction. . . . .	77
5.18	ROC Curve Plot of Model 1, Model 2, and Model 3 with alternative candidate sentences based on GSF variation. . . . .	78
5.19	ROC curve plot of Model 1, Model 2, and Model 3 with alternative candidate sentences based on GSF and WIP variation. . . . .	78

# List of Tables

3.1	Estimated probabilities $p(c n)$ . . . . .	39
3.2	Example Model 1: Resulting confidence measures $\rho_1$ . . . . .	40
3.3	Trained probability $p(c w)$ of a word $w$ of being correctly. . . . .	42
3.4	Estimated probabilities $p(c n)$ . . . . .	42
3.5	Probabilities $p(n c)$ of being in class $n$ , given the correctness of the recognition. . . . .	42
3.6	Calculated confidence measures for every word $w$ . . . . .	43
5.1	Values $\alpha_i$ used as GSF in language model variation. . . . .	63
5.2	Values $\alpha_i$ and $\beta_i$ used as GSF and WIP in language model variation. . . . .	64
5.3	Extract of the frequencies $p(\text{correct} w)$ computed on the training set. . . . .	68



# Chapter 1

## Introduction

### 1.1 Motivation

The goal of a handwriting recognition system is to process handwritten text electronically, transcribing it with the highest possible recognition rate, and to achieve a similar accuracy as humans do.

The domain of handwriting recognition is divided into two fields, offline and online recognition. In online recognition the movement of the pen is tracked, and movement information is recorded during the writing process. In general, this makes online recognition a less difficult task than offline recognition, where only the image of the handwritten text is available and processed. In this master thesis the task of offline handwriting recognition is considered.

Industrial applications using offline handwritten text recognition are mainly found in the specific field of address reading and bank cheque processing. In the future, possible applications of unconstrained text recognition could be the recognition of personal notes or automatic transcription of large handwritten archives.

For many years, research has been conducted on the topic of offline handwriting recognition. In the case of isolated numerals or digits, high recognition rates are achieved today. But as the complexity of the problem increases, as for example the recognition of words or numeral strings, the recognition rates decrease significantly. One main problem of recognising entire words is segmenting the words into their individual characters, or, at a higher level, considering the recognition of whole sentences, the segmentation of the sentence into words. The problem is usually solved by using a Hidden Markov Model (HMM) recogniser which implicitly segments the words or sentences into its components during the recognition process.

Writer independent recognition of general handwritten text is still considered

a very difficult problem. Depending on the experimental setup, word recognition rates between 50% and 80% are reported in the literature (Perraud et al., 2003; Vinciarelli et al., 2003; Zimmermann and Bunke, 2004).

For many applications such low recognition rates are not acceptable. If a complete automation of the transcription process is not required, rejection strategies may be used to reject certain parts of the handwritten text to achieve the required level of accuracies on the remaining parts.

The rejection of input (for example letters, words, sentences) is typically based on a confidence measure. If the confidence measure exceeds a specific threshold, the recognition result is accepted. Otherwise, it is rejected.

In the literature a large number of confidence measures are proposed depending on the application and the nature of the underlying recogniser. For offline handwriting recognition research most of these confidence measures are derived from the scores of the  $n$ -best list, which is produced by the recogniser. In contrast to offline handwriting recognition, confidence measures based on the integration of a statistical language model are frequently used in the field of continuous speech recognition. In this master thesis similar confidence measures are investigated for the first time in offline handwriting recognition.

## 1.2 Problem Statement and Goal

In this thesis the problem of word rejection in writer independent offline recognition of general handwritten text is addressed. The proposed rejection strategies are based on confidence measures derived from multiple alternative candidate sentences.

The aim of this thesis is to investigate the ability of alternative candidate sentences to reject individual words. The impact of several reject models, as well as different strategies to generate multiple alternative candidate sentences, are examined.

## 1.3 Contribution

This master thesis contributes to the field of offline handwriting recognition by proposing new rejection strategies for general text recognition. The confidence measures of these rejection strategies are based on alternative candidate sentences extracted from recognition lattices.

Different strategies of producing multiple candidate sentences are investigated, as the quality of these sentences has an essential impact on the performance of the reject model. Some of these strategies are based on the integration of a



statistical language model. The integration of variations of the proportion of the language model into the reject models has only been investigated in the domain of speech recognition, but not in handwriting recognition.

Experiments have been conducted to investigate the performance of the different reject models and alternative candidate sentence generation strategies.

## 1.4 Structure of the Thesis

The remaining chapters of this master thesis are organized as follows:

In Chapter 2 the handwritten text recognition system as well as objectives and backgrounds of rejection are discussed. A general system overview is given to identify the main parts of the recognition system. Furthermore, related literature in on- and offline handwriting and speech recognition is presented.

Four reject models based on alternative candidate sentences are presented in Chapter 3. For each of the reject models, its confidence measure is explained and an example is provided to illustrate the behaviour of the reject model.

Chapter 4 starts with an introduction to recognition lattices. After that different strategies to produce alternative candidate sentences are presented. The quality of the alternative candidate sentences is a key aspect for the proposed reject models.

Conducted experiments and achieved results are presented in Chapter 5. Performance measures are introduced in the evaluation methodology section. The experimental setup is explained, and test set results show the performance of the introduced reject models and alternative candidate sentences generation strategies.

The main conclusions of this thesis are drawn and possible future research is discussed in Chapter 6.



## Chapter 2

# Handwriting Recognition and Rejection

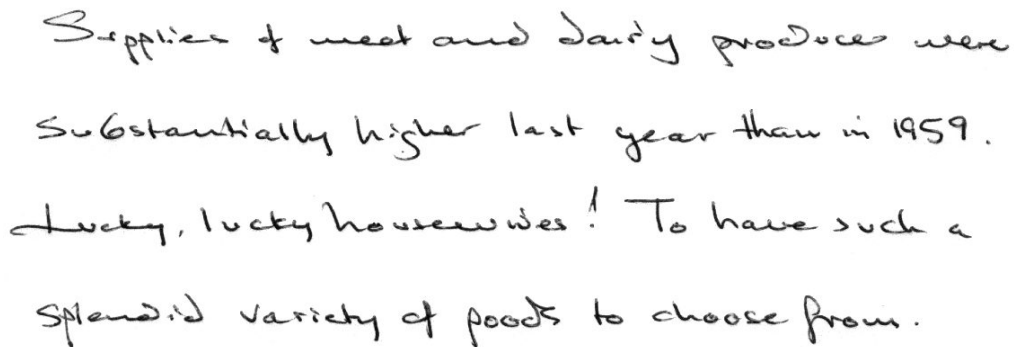
In this master thesis an unconstrained handwritten sentence recognition system that includes the ability to reject individual words is presented. From an image of a handwritten text the recogniser produces a network of possible recognitions which is used by the post-processor to determine the confidence measure. Based on the confidence measure, rejections are made to classify incorrect words.

The rest of the chapter is organized as follows: First in Section 2.1 an introduction to offline handwritten text recognition is given. Section 2.2 describes the recognition system used in this master thesis, and Section 2.3 discusses the basics of rejection. A short system overview is provided in Section 2.4. Section 2.5 explains the way how multiple candidate sentences are handled. Finally, related work in the fields of on- and offline handwriting and continuous speech recognition are presented in Section 2.6

### 2.1 Offline Handwritten Text Recognition

The most general case of handwriting recognition is offline handwritten text recognition. In contrast to the recognition of isolated characters or single words, more effort has to be made for segmentation, as the number of words in a line or sentence is not known in advance.

An offline handwriting recognition system tries to find a correct transcription for a document written by hand. The document can contain an isolated character or digit, a single word, or some general text. The document is usually available as a greyscale image scanned at a relatively low resolution. Figure 2.1 presents an example of a handwritten document.



Supplies of meat and dairy produce were  
substantially higher last year than in 1959.  
Lucky, lucky housewives! To have such a  
splendid variety of foods to choose from.

**Figure 2.1:** Example of handwritten text.

Offline handwritten text recognition is more general than online handwriting recognition, where the user is forced to use a specific writing instrument and a specific writing pad. No such constrain is necessary for offline handwriting recognition. The user can select a writing instrument (for example a quill) and a writing pad (for example paper) just as he wants, as long as the document shows some contrast.

The recognition system used in this master thesis is writer independent. Writer independent recognition systems are optimized to the highest possible generalization regarding the number of writing styles, as well as the used writing instruments. No handwritten samples of the writers in the test set are available for the training and the validation of the system. In many applications, writer independent recognition is a requirement. In an address reading system for example, it is obviously not feasible to collect training samples from every customer.

Recognition of general text means finding the correct sequence of words for a given image of handwritten text. The correct number of words in a line or sentence is not known in advance. The unknown segmentation of the text into isolated words leads to additional types of errors. The problem is similar to the segmentation of a word image into individual characters. The result of the sentence or line recogniser may contain too many, or too few words, depending on the segmentation of the image into its individual words. Inserting too many words into the recognition result is called over-segmentation. The term under-segmentation is used if too few words are present in the result of the recognition process.

In specific application domains like bank cheque or address reading, task-specific knowledge is available to facilitate the recognition task. In the case of bank cheque recognition, the legal and the courtesy amount of the cheque must be equivalent. Address reading systems have a reduced lexicon con-

taining only zip codes, city names, and street names. Additionally, relations between zip codes and city names, and between street names and cities can be used to improve the performance of the system.

In contrast to bank cheque or address reading, task-specific knowledge is reduced to statistical language models for the case of general text recognition. Most often so called  $n$ -gram models are used to integrate the information about the language (Marti and Bunke, 2001; Perraud et al., 2003).  $N$ -gram models are based on the observed frequency of adjacent words. Although  $n$ -gram models are a very raw approximation of the natural language, their integration into the recognition process can significantly improve the performance of a handwriting recognition system.

The integration of grammar-based syntax analysis is presented by Zimmermann (2003). The result of the Hidden Markov Model (HMM) based recogniser is an  $n$ -best sentences list, containing the  $n$  most probable interpretations of the text image. Each of these sentences is afterwards analyzed with help of a stochastic context-free grammar, which represents the language model. The actual recognition is determined by combining the weighted scores from the HMM based recogniser and the syntax analysis.

Applications of offline handwriting recognition technology are mainly found in address reading systems and automatic bank cheque processing. An address reading system can be used for automatic, machine-aided mail sorting, while bank cheque recognition enables the automatic processing of financial transactions. The recognition of historical documents is a third potential application of offline handwriting technology. Large handwritten archives could be processed and automatically transcribed with offline handwriting recognition of general text technology.

## 2.2 Recognition System

The HMM based handwritten text recognition system used in this master thesis corresponds to the recogniser described by Zimmermann et al. (2003). It is an enhanced version of the recognition system developed by Marti and Bunke (2001). The enhanced recogniser of Zimmermann et al. (2003) is capable of handling complete sentences, which may consist of several lines of handwritten text. Improvements are made in the language model integration as well as in the modeling of the characters.

After text line normalization and extraction of feature vector sequences during the pre-processing phase, features are extracted from the normalized text images. Then the actual recognition is performed by Viterbi decoding (Viterbi, 1967), supported by a word bigram statistical language model. The Baum-

Charles obliged with "April Serenade". This week it appears,  
 The Government should settle this argument with two words to  
 on new techniques - and on the universities to  
 Vaughan will burst on to the London Palladium stage  
 think hard and long before his next jump into the  
 I don't think he will storm the charts with this one, but it's a good start.  
 He double-crosses the five pals with whom he lives,

**Figure 2.2:** Example of different writing styles.

Welch algorithm (Rabiner, 1989) is used for the training of the character HMMs.

### 2.2.1 Text Line Normalization

Every writer has a different writing style, which contributes a lot to the complexity of the recognition task. Figure 2.2 provides an illustration of different writing styles.

The intention of the pre-processing operations is to normalize the text lines in order to reduce the variations caused by different writing styles and instruments. These operations include the following normalization procedures (see Marti and Bunke, 2001; Zimmermann et al., 2003):

*Skew Correction:* The text line has to be aligned horizontally. The normalization is performed by correcting the skew angle.

*Slant Correction:* The writing's slant is transformed into a vertical position by applying a shearing.

*Line Positioning:* The location of the upper and lower baseline is normalized. For this purpose a vertical scaling operation is performed.

*Horizontal Scaling:* The variations in the width of the handwritten text are normalized.

*Image Contrast:* The contrast of the greyscale image is normalized.

### 2.2.2 Feature Extraction

The sliding window technique is used to extract a sequence of feature vectors from the normalized text images. The width of the window is one pixel while the height is the image's height. The window is moved from the left to the right, one pixel per step, computing nine geometrical features at every window position.

The first three features contain information about the number and the distribution of the black pixels in the window. The next four features describe the position and the orientation of the upper and the lower contour in the window. Feature eight contains the number of vertical black/white transitions. The last feature contains the number of black pixels between the upper and the lower contour.

### 2.2.3 Recognition using Hidden Markov Models

Hidden Markov Models (HMMs) are widely used in the field of pattern recognition. Coming from speech recognition, HMMs have become very popular in handwriting recognition because of the implicit segmentation of a text line image into its word images.

Token passing (Young et al., 1989), an alternative formulation of the Viterbi decoding algorithm, is used in the recognition phase. The advantage of the token passing model is the ability to produce not only the most probable sentence, but also a recognition lattice which contains a network of hypothesised sentences (see Section 4.1 for more details about lattices).

A bigram language model is integrated in the decoding step. This integration of a statistical language model intends to increase the recognition rate by preferring more frequently observed word sequences to less frequently observed word sequences.

## 2.3 Rejection Strategies

Current handwritten text recognition systems are far from being perfect. Depending on the experimental setup, the systems achieve word recognition rates between 50% and 80% (Perraud et al., 2003; Vinciarelli et al., 2003; Zimmermann and Bunke, 2004). For many applications such low recognition rates are not acceptable. Unless a complete automation of the transcription process is a requirement, rejection strategies are used to reject certain parts of the

handwritten text to achieve the required level of accuracies on the remaining input.

Generally, the objective of rejections is to identify the input patterns (for example letters, words, or sentences) which are problematic and might have been recognised incorrectly (Matti et al., 2001). These patterns are rejected, while the unproblematic patterns are accepted. Rejection can also be used to direct the probably incorrectly recognised patterns into a separate classifier, which is usually called reject handler. The task of the reject handler is to reanalyze the problematic input pattern.

### 2.3.1 Rejection System Architecture

A rejection strategy can be incorporated into a recognition system at two possible stages:

*At the recognition stage :* An  $n$ -class classification problem is modelled as an  $n + 1$ -class problem, where the additional class represents the rejection.

*At the post-processing stage:* After the  $n$ -class classification is performed, a two-class recogniser is applied to decide whether to reject an input sample or not.

In this master thesis rejection strategies at the post-processing stage are investigated. These strategies have the advantage that no modification of the recognition process itself is required. The proposed rejection procedure is an independent part of the system. It can work together with different recognisers, as long as the recognition output is a recognition lattice, containing a network of the most probable interpretations of the processed image (see Section 4.1 for details).

Each of the reject models introduced in Chapter 3 of this master thesis is based on alternative candidate sentences. These candidate sentences can be seen as possible answers to the given question, which is to provide a correct transcription of the handwritten text image. The words of most probable candidate sentences are the ones to be accepted or rejected, as there is no reason to accept any words of less likely candidate sentences. For the most probable candidate sentence the term *hypothesised candidate sentence* is used in this master thesis, while the term *alternative candidate sentences* is used to describe the additional sentences.



### 2.3.2 Bayes' Decision Theory

The entire rejection problem can be considered as a classification task with two classes. The goal of the resulting two-class problem is to distinguish between "good" sets of candidates, where the hypothesis is correct, and "bad" sets of candidates, where the hypothesis is not the correct answer. In the ideal case, every "good" set of candidates is accepted, while all "bad" sets are rejected. Of course, in practice usually some "good" sets are rejected (false rejection), and some "bad" sets are accepted (false acceptance).

A minimization of these errors is provided by the Bayes classification for a two class problem, if the posterior probabilities of candidate sets as well as the different costs of errors and rejections are known (Schürmann, 1996; Gorski, 1997). Given the feature vector  $(f_1, \dots, f_m)$ , describing the set of candidates, Bayes classification determines the optimal error-reject characteristic of a decision maker by means of the posterior probabilities  $p(c|f_1, \dots, f_m)$ , where  $c \in \{correct, incorrect\}$ . Fukunaga (1993) presents a comprehensive description of the Bayes classification for the two class rejection problem.

To find the optimal error-reject characteristic the following three steps are performed:

1. The features  $(f_1, \dots, f_m)$  describing a candidate set are defined.
2. The posterior probabilities  $p(c|f_1, \dots, f_m)$  are estimated.
3. The decision rule is applied with different costs for errors and rejects.

The first step is probably the most delicate, and difficult one. According to Gorski (1997) the feature set determines the potential of the decision maker. The second step is performed using relative frequencies derived from the training set. In the third step, each cost represents one point in the resulting error-reject curve.

### 2.3.3 Confidence Measure

The rejection of input (for example letters, words, or sentences) is typically based on a *confidence measure*. If the confidence measure exceeds a specific threshold  $t$ , the recognition result is accepted, otherwise it is rejected. With the confidence measure, the features describing the set of candidates are reduced to one quantity. In the optimal case of Bayes rule, this quantity represents the posterior probability of the features, which are used to describe the candidate sets.

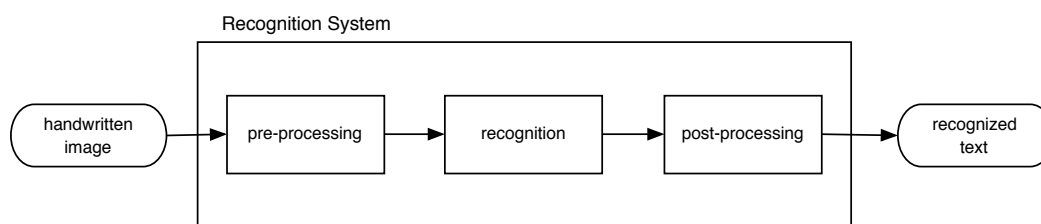
In this master thesis four different confidence measures are introduced. These confidence measures are based on alternative candidate sentences, and allow

to reject input words of the hypothesised sentence. Additionally, three different sources of candidates are investigated as the quality of the alternative candidates is considered to be an important aspect of the rejection system.

The presented confidence measures make use of estimation of the posterior probabilities as postulated by Bayes' decision theory.

## 2.4 System Overview

The offline handwriting recognition system can be divided into three major parts: pre-processing, recognition based on the Hidden Markov Model (HMM), and post-processing as illustrated in Figure 2.3. The output of the system is the transcription of the recognised text, hypothesis of the recogniser. Some of the words in this hypothesis are marked as rejected by the post-processing rejection procedure.



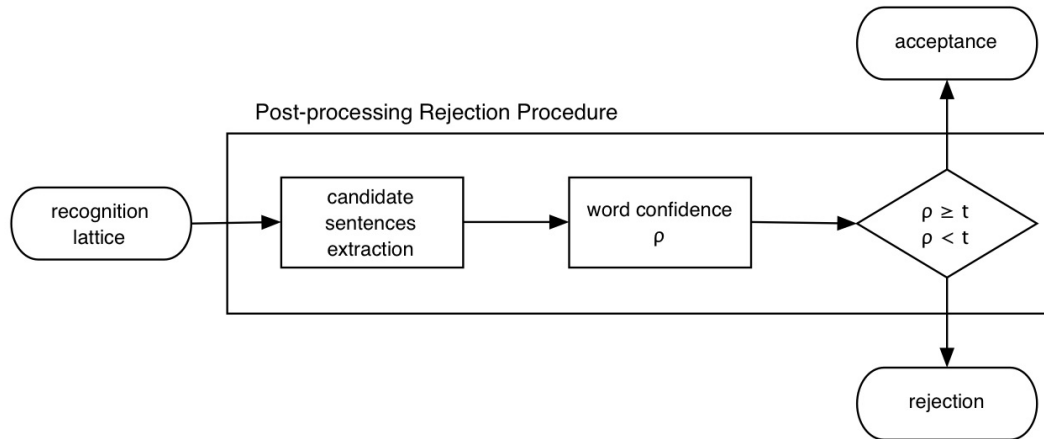
**Figure 2.3:** *Handwriting recognition system overview.*

The pre-processing part segments the text image into text lines. Sentence fragments are then extracted from the text lines. Furthermore the skew, the slant and the positions of the baselines are normalized to reduce the impact of the different writing styles. The pre-processing concludes with the extraction of feature vector sequences which are used as input data for the HMM based recogniser.

The HMM based recogniser produces a recognition lattice for every sentence, containing a network of possible transcriptions (see Section 4.1). The recognition is performed by the Viterbi decoding. For this decoding, the trained HMMs, the dictionary and the  $n$ -gram language model are required. The trained HMMs represent the different character classes, the dictionary maps the words to corresponding sequences of character HMMs, and the  $n$ -gram model, as the statistical language model, is used to prefer more probable word sequences over less probable word sequences.

Rejection is performed at the post-processing step. Multiple candidate sentences are extracted from the recognition lattice. Based on these alternative

candidate sentences, a confidence measure is computed for every word in the sentence according to a specific reject model. A word is only accepted if the confidence measure  $\rho$  is higher than a specified threshold  $t$ . Otherwise it is rejected. An illustration of the rejection procedure is provided in Figure 2.4.



**Figure 2.4:** *Rejection procedure overview.*

## 2.5 Sentence Comparison

All reject models investigated in this thesis are based on multiple candidate sentences. In order to obtain the information needed to compute the confidence measures, the candidate sentences have to be aligned appropriately. This section explains how the post-processing rejection procedure aligns and compares the multiple candidate sentences, which were extracted from a recognition lattice. The alignment is based on a dynamic string alignment procedure, which aligns two sentences, and which is extended to allow it to process more than two sentences.

### 2.5.1 Aligning one Sentence Against Another

If alternative candidate sentences are generated, then despite the fact that the sentences originate from the same image, they generally do not contain the same amount of words. That is why the sentences must be submitted to an alignment process which is based on dynamic string alignment, using the string edit distance (Wagner and Fischer, 1974).

To align the sentences, the following costs are used to calculate the string edit distance: a substitution costs 10 penalty units while insertions and deletions

a)	Mr.	Lisbon	had		escaped	10		See		figs	10		0
	Mr.	Lisbon	has	it	taped	10		for		figs	10		0
	0	0	10	7	10						10		0

c)	See		figs	10		Ben		gone		and		.	10
	See	Fig.	5	10		Ben			Germany		.	.	10
	0	7	10	10		0		7		10		0	0

**Figure 2.5:** Alignment examples.

cost 7 penalty units. Of course, leaving a word standing that coincides has no penalty<sup>1</sup>.

Four alignment examples are provided in Figure 2.5, where two sentences are aligned against each other. The upper sentence is the reference sentence against which the second sentence is aligned. The third row shows the resulting costs for each column.

In Figure 2.5 the string edit distance for Example a) is 27, as there are two substitutions (*had* → *has* and *escaped* → *taped*) and one insertion (→ *it*). In Example b), the substitution *See* → *for* arises an edit distance of 10 while an insertion (→ *Fig.*) and a substitution (*figs* → *5*) lead to a distance of 17 in Example c). The same costs arise in Example d) where a deletion (*gone* →) and a substitution (*and* → *Germany*) have to be applied to align the two sentences.

All the string edit distances are minimal for these sentences, and that is why the alignment is performed as shown.

## 2.5.2 Comparison of two sentences

After the alignment of the two sentences, every column of the alignment process falls into one of the four following categories:

*Hit (H)*: the word of the hypothesis matches exactly the word of the alternative. In Figure 2.5 the first word of Example a) illustrates such a hit.

*Substitution (S)*: the words of the two sentences differ. The last column of Example a) in Figure 2.5 gives an example of a substitution.

*Deletion (D)*: the word of the hypothesised sentence must be deleted because

<sup>1</sup>The presented costs originate from the string alignment tool HResults, which is part of the Hidden Markov Model Toolkit (HTK) (Young et al., 2002).

Mr.	Lisbon	had		escaped
Mr.	Lisbon	has	it	taped
1	1	0		0

**Figure 2.6:** Sentence comparison example.

there is no corresponding word in the alternative. In Figure 2.5 the second column of Example d) represents a deletion.

*Insertion (I):* to achieve the additional word in the alternative, a word has to be inserted into the hypothesised sentence. An insertion can be found in Example c) of Figure 2.5 in the second column.

The comparison of two sentences in the reject models is done based on these categories, and not based on the string edit distance. Hits are taken as a match (1), substitutions and deletions are taken as a mismatch (0), and insertions are ignored. Figure 2.6 shows this behaviour for the sentence aligned in Example a) of Figure 2.5.

Neglecting the insertions can be justified by the fact that the reject models presented in this master thesis are only interested in the words that are part of the hypothesised candidate sentence. Because insertions mean that the word only appears in the second sentence, they are of less interest. Ignoring the insertions has the added advantage that the alignment has exactly as many columns as the hypothesis' amount of words. This is an important feature when more than two sentences have to be aligned.

### 2.5.3 Handling more than Two Sentences

In general, comparing more than two sentences can be very difficult, because the sentences might all have a different length. The problem of aligning  $k$  sentences can be solved by filling a  $k$ -dimensional matrix of costs (Tompa, 2000). But the computational effort of this algorithm is considerably high. Wang and Jiang (1994) have proven that the problem of finding the optimal alignment for multiple sentences is *NP*-complete (Cook, 1971).

In this thesis, the alignment of multiple sentences is much more simple and efficient. Instead of aligning  $k$  sentences against each other,  $k - 1$  sentences are aligned against one sentence. This approach is possible because there is only one hypothesised sentence and  $k - 1$  alternative candidate sentences. The alignment becomes even simpler, considering that the insertions are ignored. During the alignment process, all  $k - 1$  alternative candidate sentences get pruned or expanded to the same length of the hypothesised sentence.

a)

Mr. Brown Oxford Dictionary
Mr. Dr. near Oxford Dictionary
it is , near Oxford Dictionary
Mother , near Oxford Dictionary
Mr. Dr. been Oxford Dictionary
it 's near Oxford Dictionary

b)

Mr.	Brown	Oxford	Dictionary
Mr.	Dr.	Oxford	Dictionary
it	is	Oxford	Dictionary
Mother	,	Oxford	Dictionary
Mr.	Dr.	Oxford	Dictionary
it	's	Oxford	Dictionary
2	0	5	5

**Figure 2.7:** Alignment example with six sentences.

Similar to the alignment, the comparison of  $k$  sentences is based on  $k - 1$  comparisons. Each of the alternative candidate sentences is compared to the hypothesised sentence which results in  $k - 1$  binary sequences of the same length. The result of the comparison is achieved by summing up the content of the sequences for every column. The resulting number denotes the number of words in the alternative sentences that are equal to the word in the hypothesised sentence.

Figure 2.7 provides an example, in which the six sentences are aligned and compared. Figure 2.7 a) shows the list of candidate sentences, where the first sentence is the hypothesised candidate sentences, while the other five sentences are the alternative candidate sentences. The result of the alignment is provided in Figure 2.7 b). It can be seen that several words from the alternative candidate sentences are ignored, such as for example *near* in the first alternative sentence, or the comma in the second alternative sentence. The last row of Figure 2.7 b) shows the resulting values of the comparison.

## 2.6 Related Work

In the literature, a large number of rejection strategies are proposed depending on the application and the nature of the underlying recogniser. In this section related work in offline and online handwriting recognition, and continuous speech recognition research are presented.

### 2.6.1 Offline Handwriting Recognition

In offline handwriting recognition rejection strategies for address reading (Brakensiek et al., 2003), cheque processing (Gorski, 1997), and character recognition (Pitrelli and Perrone, 2003) systems are presented.

Brakensiek et al. (2003) introduces confidence measures for an HMM based handwriting recognition system for German address reading. In order to reject isolated handwritten street and city names, four different strategies are described, based on normalized likelihoods and the estimation of posterior probabilities. For the likelihood normalization the number of frames is used. In the case of estimation of the posterior probabilities, the normalization is performed using a garbage model, a two-best recognition, and a character-based recogniser. On the complete dictionary, the best performance is given by the two-best recognition model. For detection and rejection of out-of-vocabulary-words the garbage model performs better than the two-best recognition model. Rejection strategies for cheque processing systems are presented by Gorski (1997), where an artificial neural network computes a confidence measure from a set of 10-20 features. Most features represent quantities derived from the scores of the  $n$ -best candidate list produced by the recogniser, such as for example, the log of the best score. The decision-making task is defined as a two-class recognition problem. The neural network is used to estimate the posterior probabilities of the classes.

Pitrelli and Perrone (2003) investigate several confidence measures for an offline handwritten character recognition system. The described measures of recognition confidence are recognition score, likelihood ratio, estimated posterior probability and exponentiated probability. An additional confidence measure is built by using a Multi-Layer Perceptron (MLP) to combine the previous confidence measures. Compared to the raw recognition score, simple confidence measures are able to reduce the rejections by up to 30% at the same error rate. The combination of multiple confidence measures using an MLP improved performance significantly by reducing the rejections up to 53%.

### 2.6.2 Online Handwriting Recognition

For the case of online handwriting recognition similar confidence measures, as in the case of offline handwriting recognition, are used.

Pitrelli and Perrone (2002) evaluate similar confidence measures in the field of online handwriting recognition, as Pitrelli and Perrone (2003) have investigated in offline recognition. An artificial neural network, combining different confidence measures, is used to decide when to reject isolated digits or words. In the case of isolated digits, the system requires 13% false rejection rate to

achieve a false acceptance rate lower than 10%. The word verification system obtains a false acceptance rate under 10% at a false rejection rate of 33%.

Various confidence measures for online handwriting recognition are investigated by Marukatat et al. (2002). The confidence measures are integrated in an isolated word recognition system as well as in a sentence recognition system. Four different letter-level confidence measures based on different implicit anti-models are applied. Anti-models are used to normalize the likelihood of an unknown observation sequence by calculating the ratio between the probability of the hypothesised word and its anti-model. An implicit anti-model of a hypothesised word is derived from competing hypothesis or, more generally, from other models in the system. At the isolated word level, the best performing model allows a reduction of the error rate to 5% at a rejection rate of about 30%. Rejecting about 30% of the embedded words in a sentence recognition process reduces the error rate from 30% to 10%.

### 2.6.3 Speech Recognition

Additional confidence measures, based on the integration of a statistical language model, are used in the field of continuous speech recognition. The integration of the language model in the recognition process can be controlled by two factors: the *Grammar Scale Factor (GSF)* which weights the impact of the statistical language model against the acoustic recognition of the utterance, and the *Word Insertion Penalty (WIP)* which controls the segmentation of the recogniser (see Section 4.3 for details).

In continuous speech recognition it has been observed that those words of the hypothesised sentence that are very sensitive to variations of the integration of the statistical language model are frequently recognised incorrectly. Such words are therefore to be rejected.

Sanchis et al. (2000) investigate the use of the GSF to classify incorrect words in a speech recognition system. The GSF is varied to generate additional candidate sentences, from which the confidence measures are derived. Two models, Model 1 and Model 2, based on acoustic stability are presented, analogous to Model 1 described in Section 3.2, and Model 2 described in Section 3.3. The study additionally investigates the reduction of computational costs of the reject models. The presented experiments show that models with reduced computational costs provide approximately the same results as the ordinary models. Model 2, which respects the currently processed word, performs significantly better than Model 1.

Not only the GSF, but also the WIP is varied by Zeppenfeld et al. (1997) in the field of conversational telephone speech recognition. Multiple candidate sentences derived from GSF and WIP variations are used to determine the



confidence measure. With the presented technique the error rate is reduced from over 50% down to 38%.

San-Segundo et al. (2000) considers five different word-level confidence measures based on the three features, language model back-off sequence, language model score, and phonetic length of recognised words. The first three confidence measures are these features themselves, while the fourth confidence measure is a combination of the three features based on a decision tree model. The fifth confidence measure combines the features using a Multi-Layer Perceptron (MLP). The MLP combination outperforms the other confidence measures and is able to detect over 43% of the incorrectly recognised words at a false rejection rate of 5%.



# Chapter 3

## Reject Models

The reject models investigated in this thesis are based on confidence measures derived from a list of candidate sentences. In addition to the recogniser's top ranked output in form of a hypothesised sentence  $W = (w_1, \dots, w_n)$ , the list contains  $K$  alternative sentences  $(\hat{W}_1, \dots, \hat{W}_K)$  produced by the recognition process.

Compared to confidence measures that are computed during the recognition itself the approach presented in this thesis is completely different. It has the advantage of being relatively independent of the underlying recogniser. The only requirement is the existence of multiple alternative candidate sentences.

In this chapter four different reject models are introduced. For each model the confidence measure is presented and an example is provided to illustrate the behaviour of the rejection.

Section 3.1 presents a confidence measure based on a single alternative candidate sentence. In the confidence measure introduced in Section 3.2 not only one, but multiple candidate sentences are considered. In Section 3.3 a more precise confidence measure based on multiple candidate sentence is presented. Finally, Section 3.4 introduces a confidence measure determined by a Multi-Layer Perceptron.

### 3.1 Model 0: Single Alternative

The most evident way to reject a word of the hypothesised sentence based on alternatives, is to produce a single alternative sentence and to reject all words of the hypothesised sentence that are not present in the aligned alternative sentence. Even if this idea looks trivial, it leads to a very efficient reject model because no training is needed and only one alternative per sentence has to be computed. Therefore, rejections can be performed very fast.

Transcription:	Mr.	Lisbon	has	it	taped
Hypothesis:	Mr.	Lisbon	had		escaped
Alternative:	Mr.	Lisbon	has	it	taped

**Figure 3.1:** Rejection with a single alternative candidate sentence.

### 3.1.1 Confidence Measure

After aligning the two sentences as described in Section 2.5, the confidence measure  $\rho_0$  of this model is defined for every word of the hypothesised sentence in the alignment as follows:

$$\rho_0 = \begin{cases} 1 & : \text{ same word in alternative} \\ 0 & : \text{ otherwise} \end{cases} \quad (3.1)$$

This approach is quite easy to implement, but it also depends heavily on the available alternative. Using a 2-best list for this model is not suitable because, as the candidate sentences from an  $n$ -best list always differ, the rejection procedure rejects at least one word in every sentence. Above all, for recognisers providing good sentences, this might lead to many false rejections.

It is for these reasons that in this master thesis the alternative is generated by varying the proportion of the language model as described later in Section 4.3. It can be expected that small variations produce a few rejections only, while large variations result in high rejection rates. Obviously, no training is needed for this model, but the way in which the alternative sentence is generated has to be determined and optimized experimentally.

### 3.1.2 Example

An example of this simple rejection strategy is given in Figure 3.1. The first line shows the transcription which is the desired result of the recognition process. The second line provides the actual result of the recogniser, while the third line shows the alternative sentence.

It can be seen that the hypothesis differs from the alternative sentence at the words *had* and *escaped*. The reject model therefore rejects these two words, and retains the equal ones *Mr.* and *Lisbon*. In the example given in Figure 3.1, this strategy leads to a perfect result: the wrongly recognised words *had* and *escaped* are identified as incorrect and rejected, while the correctly recognised words *Mr.* and *Lisbon* are accepted.

Notice that the fourth column of the alignment with the word *it* in the transcription and the alternative sentence is ignored, because no corresponding word

is available in the hypothesised sentence.

## 3.2 Model 1: Multiple Alternatives

The second proposed reject model is an extension of Model 0. Instead of considering just a single alternative sentence, the confidence measure of Model 1 requires a list of  $K$  alternative candidate sentences ( $\hat{W}_1, \dots, \hat{W}_K$ ), in addition to the recogniser's top ranked sentence  $W$ . These sentences have to be aligned against the top ranked sentence (the hypothesis) as described in Section 2.5.

The confidence measure of Model 1 is derived from the domain of continuous speech recognition, where it has been applied by Sanchis et al. (2000).

### 3.2.1 Confidence Measure

The probability of a word  $w$  of the hypothesised sentence being recognised correctly can be defined as  $p(c|n, w)$ , where  $c \in \{0, 1\}$  (0 stands for incorrect and 1 for correct) and  $n = 0, \dots, K$  represents the number of times a word  $w$  is observed in the  $K$  alternative candidate sentences.

To reduce computational complexity, and because the training set is not large enough to estimate  $p(c|n, w)$  for every  $n$  and every word  $w$ , it is assumed that the probability of being recognised correctly is independent of the current word  $w$ . This assumption results in the approximation  $p(c|n) \simeq p(c|n, w)$ . The probability  $p(c|n)$  is then used as confidence measure  $\rho_1$  for this reject model.

$$\rho_1 = p(c|n) \quad (3.2)$$

During the training phase, the quantity  $p(c|n)$  is estimated for every  $n = 0, \dots, K$ , using the relative frequencies obtained from the training set as presented in Equation 3.3:

$$p(\text{correct}|n) \simeq \frac{x_n}{x_n + y_n} \quad p(\text{incorrect}|n) \simeq \frac{y_n}{x_n + y_n} \quad (3.3)$$

The quantity  $x_n$  counts the number of correctly recognised words, in which  $n$  is the number of times a hypothesised word appears in the alternative candidate sentences. The quantity  $y_n$  is used to count the cases of the words which are not correctly recognised.

This approach is still quite simple as it basically implies summing up the words with a hit in the aligned alternative candidate sentences, and looking for the value of  $p(c|n)$  in the precomputed probability function.

The following two engraving simplifications possibly decrease the performance of the rejection procedure:

- The assumption that the probability of a correct recognition is independent of the currently processed word is strong, because there might be words that are easy to recognise correctly, while others are not.

The experiments with the reject model described later in Section 3.3 show that the probability of being recognised correctly differs considerably from word to word. While, for example, the word *of* is recognised correctly with a probability of more than 92%, the probability for the word *or* lies under 23%.

- Just summing up the identical words in the alternatives leads to information loss. All alternatives have the same weight on the resulting one-dimensional feature vector  $n$ . This seems reasonable as long as all the sources of the different alternatives are of more or less the same quality and reliability, but in general this condition is not true.

A good illustration of the problem is given by considering that the alternative candidate sentences are acquired by extracting an  $n$ -best list as described in Section 4.2. Here it is obvious that the reliability of the first sentence differs from the last sentence of this list. Nevertheless, these sentences have the same impact on the confidence measure  $\rho_1$ .

### 3.2.2 Example

An example of the reject model described above is given in Figure 3.2, with the transcribed sentence “*Mr. Lisbon has it taped*”. The hypothesised sentence “*Mr. Lisbon had escaped*” proposed by the recogniser is shown on the second line. Furthermore  $K = 5$  alternative candidate sentences ( $\hat{W}_1, \dots, \hat{W}_5$ ) are generated. The last line shows  $n$ , the number of times a word of the hypothesis is observed in the alternative sentences.

In this example the words *had* and *escaped* are the words that were recognised incorrectly and that should be rejected by the rejection process, while *Mr.* and *Lisbon* should be accepted. The missing word *it* in column four is ignored as it is not part of the hypothesised sentence.

Table 3.1 shows the probabilities  $p(c|n)$ , which have been estimated during the training phase. This means that the probability of being correct if none of the alternative candidate sentences contain a matching word is 0.0625, while, for example, the probability of a word being recognised correctly when  $n = 4$  is about 58%. It can be seen, as expected, that the probability of a correct recognition raises with an increasing number  $n$  of matching words in the alternatives.

Transcription:	Mr.	Lisbon	has	it	taped
Hypothesis $\hat{W}$ :	Mr.	Lisbon	had		escaped
Alternative $\hat{W}_1$ :	Mr.	Lisbon	has	it	taped
Alternative $\hat{W}_2$ :	Mr.	Lisbon	has	it	taped
Alternative $\hat{W}_3$ :	Mr.	Lisbon	had		escaped
Alternative $\hat{W}_4$ :	Mr.	Lisbon	had		escaped
Alternative $\hat{W}_5$ :	Mr.	Lisbon	had		escaped
$n$ :	5	5	3		3

**Figure 3.2:** Counting the number of times  $n$ , a hypothesised word occurs in alternative candidate sentences.

$n$	$p(\text{correct} n)$
0	0.0625
1	0.1630
2	0.2832
3	0.3973
4	0.5872
5	0.9157

**Table 3.1:** Estimated probabilities  $p(c|n)$ .

$w$	$\rho_1$
Mr.	0.9157
Lisbon	0.9157
had	0.3973
escaped	0.3973

**Table 3.2:** Example Model 1: Resulting confidence measures  $\rho_1$ .

With regard to the estimated probabilities provided in Table 3.1, and the values  $n$  on the last line of Figure 3.2, the confidence measures  $\rho_1$  for each word of the hypothesised sentence  $W$  are computed. The resulting values are shown in Table 3.2.

If the threshold for rejection is set to 0.5, which means that the system only accepts words with  $\rho_1 > 0.5$ , the desired effect is obtained, which is to reject *had* and *escaped*, as well as to accept *Mr.* and *Lisbon*.

### 3.3 Model 2: Considering the Current Word

Model 2 addresses the earlier mentioned problem that some words are more likely to be recognised correctly than others. For its confidence measure  $\rho_2$ , Model 2 takes into account the currently processed word  $w$ , instead of assuming that the recognition result is independent of word  $w$  as was supposed by Model 1 in Section 3.2.

By considering the currently processed word, additional information is integrated into the calculation of the confidence measure. This added information could lead to a superior performance of the rejection procedure of Model 2 over the rejection procedure of Model 1.

The model introduced in this section is equivalent to Model 2 introduced by Sanchis et al. (2000) in the field of speech recognition. Sanchis et al. (2000) show that Model 2 can clearly outperform the rejection strategy of Model 1.

#### 3.3.1 Confidence Measure

Similar to Model 1 in Section 3.2, the confidence measure of Model 2 is an approximation of  $p(c|n, w)$ , but in contrast, the currently processed word  $w$  is no longer ignored but incorporated into the confidence measure.

Using Bayes' rule,  $p(c|n, w)$  can be expressed as follows:

$$p(c|n, w) = \frac{p(n|c, w) \cdot p(c|w)}{\sum_{x=0,1} p(n|x, w) \cdot p(x|w)} \quad (3.4)$$



As the system is not able to calculate  $p(n|c, w)$ , an approximation is necessary by assuming that  $p(n|c, w) \simeq p(n|c)$ , meaning that the probability of being in class  $n$ , given  $c \in \{0, 1\}$  and the word  $w$ , is independent of  $w$ . The resulting term is shown in Equation 3.5

$$\frac{p(n|c, w) \cdot p(c|w)}{\sum_{x=0,1} p(n|x, w) \cdot p(x|w)} \simeq \frac{p(n|c) \cdot p(c|w)}{\sum_{x=0,1} p(n|x) \cdot p(x|w)} \quad (3.5)$$

By means of the approximation presented in Equation 3.5 the confidence measure  $\rho_2$  is defined as follows:

$$\rho_2 = \frac{p(n|c) \cdot p(c|w)}{\sum_{x=0,1} p(n|x) \cdot p(x|w)} \quad (3.6)$$

In this model both  $p(n|c)$  and  $p(c|w)$  have to be estimated during the training phase. From the training set the quantities  $p(n|c)$  are estimated using the relative frequencies obtained by counting the number of times a hypothesised word that is correct  $x_n$  (incorrect  $y_n$ ) has  $n$  hits in the alternative candidate sentences.

$$p(n|correct) \simeq \frac{x_n}{x_0 + \dots + x_K} \quad p(n|incorrect) \simeq \frac{y_n}{y_0 + \dots + y_K} \quad (3.7)$$

The quantities  $p(c|w)$  are estimated using the relative frequencies determined by counting the number of times a word  $w$  has been recognised correctly ( $w_1$ ) and incorrectly ( $w_0$ ) respectively.

$$p(correct|w) \simeq \frac{w_1}{w_0 + w_1} \quad p(incorrect|w) \simeq \frac{w_0}{w_0 + w_1} \quad (3.8)$$

If there are no or not enough training samples of one word  $w$ , this approach is not feasible because  $p(c|w)$  is not available or can not be determined with adequate care. If, for example, a word  $w$  appears only once in the training corpus, the estimation of  $p(c|w)$  is 1, if this one sample has been correct, and 0 otherwise. For this reason a minimal amount of samples for one word has to be available in the training corpus. If these samples are not available, the confidence measure  $\rho_1$  of Model 1 (see Section 3.2) is used instead of  $\rho_2$  to approximate  $p(c|n, w)$ .

### 3.3.2 Example

In this example the same sentence “*Mr. Lisbon has it taped*” as in the previous one (Figure 3.2) is discussed. Instead of  $\rho_1$  of Model 1, the confidence measure  $\rho_2$  of Model 2 is used, which respects the particularity of the currently processed word  $w$ .

The estimated values for  $p(c|w)$  obtained from the training set are shown in Table 3.3. It can be seen that for *Lisbon* and *escaped* the system does not provide any result because the training corpus does not contain these words, or it does not contain enough of these words to estimate the probabilities. This means that for these words the confidence measure  $\rho_2$  is not practicable, and  $\rho_1$  must be used instead.

Word $w$ :	$p(1 w)$	$p(0 w)$
Mr.	0.5416	0.4584
Lisbon	-	-
had	0.7916	0.2084
escaped	-	-

**Table 3.3:** Trained probability  $p(c|w)$  of a word  $w$  of being correctly.

For the words *Lisbon* and *escaped* the confidence measure  $\rho_1$  is used and therefore  $p(c|n)$  must be estimated as well. The required values for  $n = 3$  and  $n = 5$  are shown in Table 3.4 which is an extract of Table 3.1.

$n$	$p(\text{correct} n)$
3	0.3973
5	0.9157

**Table 3.4:** Estimated probabilities  $p(c|n)$ .

The estimated probabilities for  $p(n|c)$  are shown in Table 3.5. If a sentence is correct, it is more likely that the alternative candidate sentences match, which is expressed in a higher value of  $n$ .

Correctness	$n = 0$	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
correct	0.0021	0.0145	0.0602	0.0949	0.1406	0.6877
incorrect	0.0551	0.1512	0.2924	0.2401	0.1512	0.11

**Table 3.5:** Probabilities  $p(n|c)$  of being in class  $n$ , given the correctness of the recognition.

Next, the confidence measure for every word  $w$  of the hypothesised sentence is calculated using the estimated probabilities. The results of these computations are displayed in Figure 3.6. For the words *Mr.* and *had* the confidence measure  $\rho_2$  can be used while  $\rho_1$  is required to compute the confidence measure of the words *Lisbon* and *escaped*.

If the threshold for rejection is set to about 0.7, the procedure provides the desired results. It can be assumed that, with the consideration of  $p(c|w)$ , the confidence measure becomes more precise compared to  $p(c|n)$ .

Word $w$ :		n	Computation	Result
Mr.	$\rho_2$	5	$\frac{0.6877 \cdot 0.5416}{0.11 \cdot 0.4584 + 0.6877 \cdot 0.5416}$	0.8808
Lisbon	$\rho_1$	5		0.9157
had	$\rho_2$	3	$\frac{0.0949 \cdot 0.7916}{0.2924 \cdot 0.2084 + 0.0949 \cdot 0.7916}$	0.6002
escaped	$\rho_1$	3		0.3973

**Table 3.6:** Calculated confidence measures for every word  $w$ .

### 3.4 Model 3: Multi-Layer Perceptron

Model 3 employs a *Multi-Layer Perceptron (MLP)*, using feature vectors derived from multiple alternative candidate sentences, to determine a confidence measure.

An MLP is an artificial neural network consisting of multiple layers of computational neurons connected in a feedforward way. Each neuron in a layer has directed connections to every neuron of the subsequent layer. MLPs are usually trained using the back-propagation-of-error algorithm (Rojas, 1996).

The entire post-processing rejection procedure can be conceived as a two-class classification problem. Based on a feature vector, provided by  $K$  alternative candidate sentences, the system must decide whether to accept a word, or to reject it. This problem can be solved sub-symbolically with a MLP using one layer of hidden neurons (Pao, 1993).

Successful experiments using an MLP to calculate confidence measures have been presented by Pitrelli and Perrone (2002, 2003). In these studies the MLP based confidence measures perform better than the other investigated confidence measures.

The confidence measure based on an MLP is expected to address the previously mentioned problem of Model 1 in which some alternative candidate sentence sources are of higher quality than others. The sources which produce better sentences should have a larger impact on the confidence measure than sources of weaker quality.

Additionally, the MLP is able to consider relations between different sources of alternative candidate sentences, as usually these source are not independent. It is possible for example that if neither the second sentence, nor the last sentence of the  $n$ -best list match the word in the hypothesis, the probability of an

incorrect recognition is higher than if the third and fourth sentence mismatch the hypothesis.

The additional information content that is considered in the confidence measure of Model 3 is expected to lead to a superior performance over Model 1.

### 3.4.1 MLP Architecture

Two possible architectures can be considered to solve the two-class problem. In both architectures the MLP is entirely connected and consists of  $n$  input channels and  $m$  internal neurons. The proposed architectures differ only in the number of output channels.

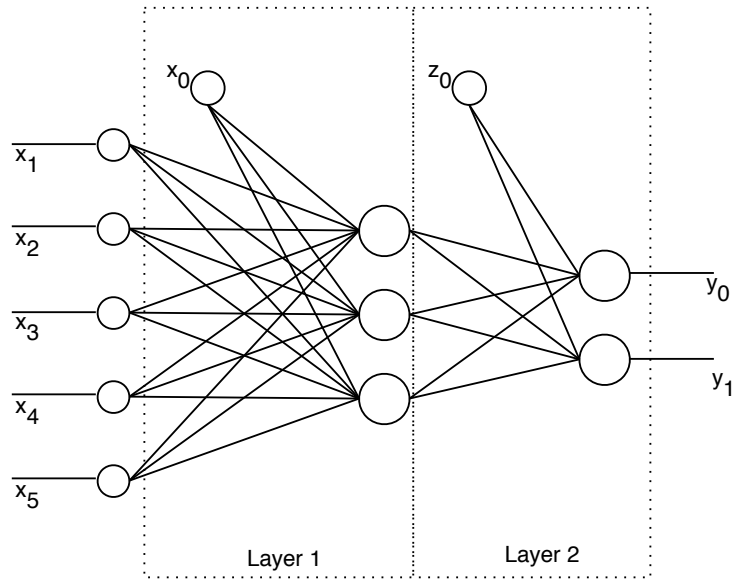
The first possible architecture is an MLP with a single output channel  $y_1$ . This output channel estimates the probability  $p(\text{correct}|x_1, \dots, x_n)$  of being recognised correctly given the feature vector  $(x_1, \dots, x_n)$ . The estimation of a false recognition  $p(\text{incorrect}|x_1, \dots, x_n)$  is given by  $1 - y_1$ .

The advantage of this architecture is that less weights have to be determined and therefore less training samples are required. The drawback is the training and the validation of the MLP. For training with the back-propagation algorithm, a threshold  $t$  has to be defined for which rejections should be made if  $y_1 < t$ . The optimal value of threshold  $t$  is hardly to be defined in a general rejection system.

It is because of this problem that an MLP architecture with two output channels  $y_0$  and  $y_1$  is preferred in this master thesis. The value of  $y_0$  represents the score for rejection, while  $y_1$  represents the acceptance score. During the training phase a sample with feature vector  $(x_1, \dots, x_n)$  is regarded as rejected if  $y_0 > y_1$ , otherwise it is regarded as accepted. During the testing phase  $y_1$  is used as a confidence measure, as it indicates the score for a given feature vector  $(x_1, \dots, x_n)$  of being recognised correctly.

An example of the proposed MLP architecture is given in Figure 3.3. The fully connected MLP with five input neurons  $(x_1, \dots, x_5)$  and four hidden neurons has two output neurons. The nodes  $x_0$  and  $z_0$  are used to adjust the biases of the nodes in layer 1 and layer 2 respectively. The resulting output of the MLP is stored in the vector  $(y_0, y_1)$ . The value of  $y_0$  stands for the rejection score, while the value of  $y_1$  represents the score for acceptance of a word with feature vector  $(x_1, \dots, x_5)$ . The quantity  $y_1$  is used as a confidence measure in the rejection process.

To train the weights of the MLP, an sufficiently large training set has to be available. If the MLP consists of  $n$  input channels,  $m$  hidden neurons, and 2 output classes,  $nm + 3m + 2$  weights need to be trained. In the above mentioned example, where  $n = 5$  and  $m = 3$ , 26 weights must be quantified.



**Figure 3.3:** Example of a Multi-Layer Perceptron.

	Mr.		Brown		Oxford		Dictionary	
$\hat{W}_1$	Mr.	1	Dr.	0	Oxford	1	Dictionary	1
$\hat{W}_2$	it	0	is	0	Oxford	1	Dictionary	1
$\hat{W}_3$	Mother	0	,	0	Oxford	1	Dictionary	1
$\hat{W}_4$	Mr.	1	Dr.	0	Oxford	1	Dictionary	1
$\hat{W}_5$	it	0	's	0	Oxford	1	Dictionary	1

**Figure 3.4:** Example of a feature vector acquisition.

### 3.4.2 Feature Vector Acquisition

The feature vectors  $(x_1, \dots, x_n)$ , which are used as input data for the MLP, are acquired from multiple candidate sentences after the sentence alignment. Instead of summing up the matching words, as in Model 1 introduced in Section 3.2, every alternative candidate sentence  $\hat{W}_i$  contributes one element  $x_i$  to the feature vector.  $x_i$  is 1 if the word of  $\hat{W}_i$  matches the word in the hypothesised sentence, and 0 otherwise. When  $K$  alternative candidate sentences are available, this strategy leads to a binary vector  $(x_1, \dots, x_K)$  of length  $K$  for every word in the hypothesised candidate sentence.

The example of Figure 3.4 shows how a feature vector is derived from multiple alternative candidate sentences. Every word of the hypothesis is tested against the corresponding words in the alternative candidate sentences  $\{\hat{W}_1 \dots \hat{W}_5\}$ ,

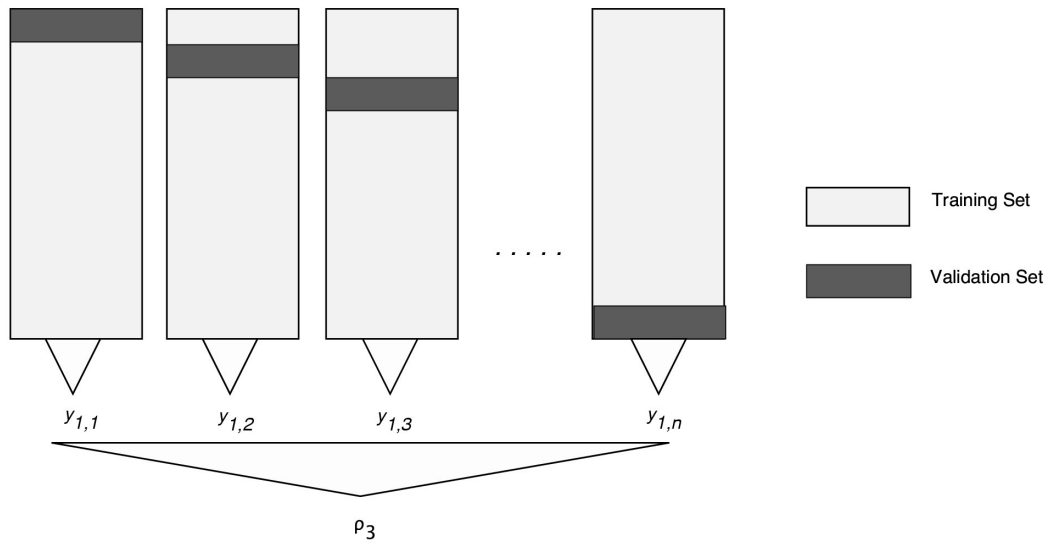


Figure 3.5: Cross validation process.

and if these words match,  $x_i$  is set to 1, else it is set to 0. The resulting feature vectors  $(x_1, \dots, x_5)$  for the hypothesised words are as follows: *Mr.*: (1,0,0,1,0), *Brown*: (0,0,0,0,0), *Oxford*: (1,1,1,1,1), and *Dictionary*: (1,1,1,1,1).

### 3.4.3 Combining Multiple Multi-Layer Perceptrons

For the three preceding reject models, no validation set is required because there were no additional parameters to optimize. This is different to the MLP based confidence measure, where a validation set is needed, because during several iterations the MLP is trained and the effect of the training is validated for each iteration.

But in the sentence database used in this thesis, only two sets are available, and therefore the set, which is used for training in the preceding models, is split up in a training and in a validation set.

To reduce the dependency on the selection of the training and validation set, a cross validation is performed (Kohavi, 1995). The original data set  $M$  is divided into  $n$  mutual exclusive subsets  $M_1, \dots, M_n$ , with the property  $M = \bigcup_{i=1, \dots, n} M_i$ . Then,  $n$  separate MLPs are constructed, each of them using one set  $M_i$  as validation set, and the remaining sets for training. An illustration of the cross validation process is shown in Figure 3.5.

### 3.4.4 Confidence Measure

The confidence measure  $\rho_3$  of Model 3 is a combination of the confidence measures  $y_{1,i}$  ( $i = 1, \dots, n$ ) of the  $n$  MLPs as it can be seen in Figure 3.5. The following combination schemes have been investigated in this thesis: mean value ( $\sum_{i=1 \dots n} \frac{y_{1,i}}{n}$ ), minimum value ( $\min(y_{1,1}, \dots, y_{1,n})$ ) and maximum value ( $\max(y_{1,1}, \dots, y_{1,n})$ ). By means of validation on the entire training set, calculating the mean value has outperformed the minimum and the maximum value. Therefore, the confidence measure  $\rho_3$  for the MLP based reject model is

$$\rho_3 = \sum_{i=1, \dots, n} \frac{y_{1,i}}{n} \quad (3.9)$$

where  $y_{1,i}$  is the score of being accepted from MLP  $i$ , and  $n$  is the number of MLPs in the combination process.





## Chapter 4

# Alternative Sentences Generation

In this chapter different strategies to generate multiple alternative candidate sentences are presented. The candidate sentences originate from a recognition lattice, which is the result of the recogniser, and which contains a network of possible interpretations of a given handwritten text image.

The quality of the alternative candidate sentences is a key aspect for a good performance of the reject models introduced in Chapter 3. At best, an alternative sentence distinguishes itself from the hypothesised sentence exactly at the position where the words are recognised incorrectly. Of course, in practice, this is rarely the case, as alternatives sometimes differ in words that are correct or coincide with words that are incorrectly chosen by the recogniser. Nonetheless, the alternative candidate sentences must be chosen thoughtfully because they have an essential impact on the performance of the rejection system.

The candidate sentences can be immediately produced during the recognition process but, in this thesis, a different strategy has been implemented: instead of directly generating alternative candidate sentences, the recogniser builds a form of intermediate data structure, a so called *Recognition Lattice*, from which the candidate sentences can be extracted. The main advantage of this strategy is a significantly improved performance when multiple experiments are conducted with the same handwritten text image.

This chapter is structured as follows: In Section 4.1 recognition lattices are introduced. Next,  $n$ -best lists are presented in Section 4.2 as a possible source of alternative candidate sentences. Section 4.3 discusses possibilities of language model variations to obtain alternative sentences.

## 4.1 Recognition Lattice

Recognition lattices are able to store different hypotheses from the output of the handwriting recognition system in a finite state network. A recognition lattice is a data structure which represents the most promising subspace of recognitions investigated by the Viterbi decoding step. The lattices are produced by the token passing algorithm (Young et al., 1989), which is an extension of the Viterbi decoding step, and stored in the HTK Standard Lattice Format (Young et al., 2002).

A recognition lattice consists of a set of nodes and a set of links, where every node represents an end of a word, while the links represent the transitions between word ends. This means, a link stands for a hypothesised word between two positions in the image. Every node is labelled with the position in the image. The links are labelled with a word hypothesis, an optical score, and a language model score. Additionally, every link has to remember its start and end node. A lattice must have exactly one start node with no incoming links, and one end node without any outgoing links.

An example of a lattice produced by the recogniser after analyzing the displayed handwritten sentence is shown in Figure 4.1. To increase clarity and readability the lattice has been pruned from originally 201 nodes to 28 nodes. Despite the pruning, many possible alternative ways remain to walk through the lattice.

## 4.2 *N*-Best List Extraction

A common strategy to produce multiple candidate sentences during the recognition process is to extract an *n*-best list. This list contains the *n* highest ranked and therefore most promising interpretations for a given image of a handwritten sentence.

Extracting *n*-best lists is an easy, fast and standard way to generate multiple candidate sentences in offline handwritten text recognition systems.

In *n*-best lists every sentence is different from the  $n - 1$  other sentences. This property is quite important in the context of rejecting, based on alternative candidate sentences. In many applications it is an advantage to obtain different sentences, but in this rejection application it could be a handicap. The differences between the alternatives can lead to unnecessary rejections, especially if the entire sentence has been correctly recognised by the recognition process.

An example of such an *n*-best list extraction process is provided in Figure 4.2. The original handwritten sentence can be seen in Part a) of the figure. The handwriting recognition system uses this image as input data and creates a

Mr. Lisbon has it taped.

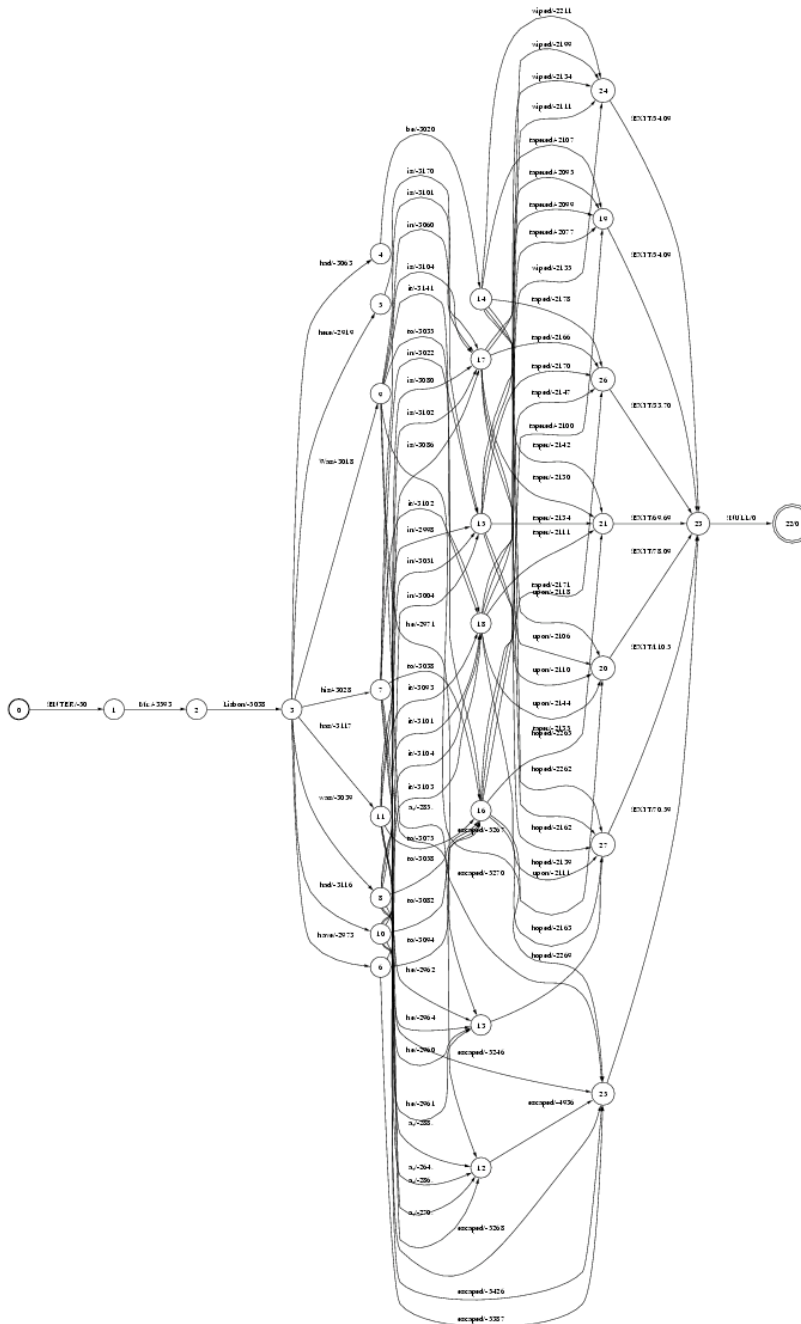


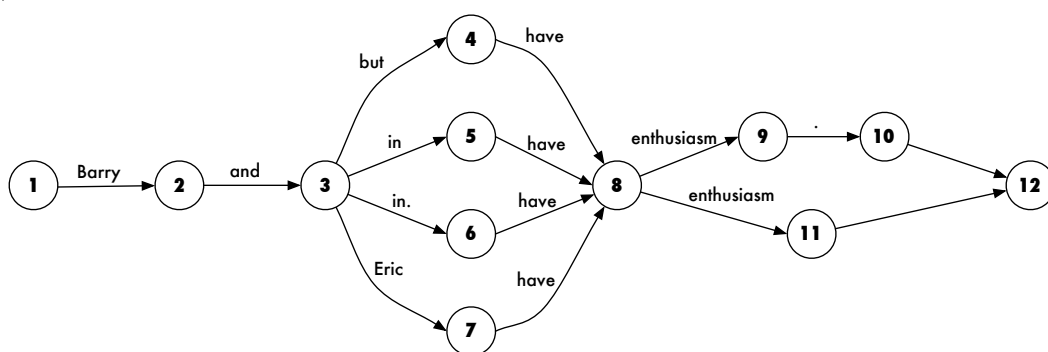
Figure 4.1: Example of a (pruned) recognition lattice.

recognition lattice containing a network of hypothesis. A pruned and simplified version of this lattice is shown in Part b). Part c) of the figure shows the six most probable sentences extracted from the lattice.

a)

Barry and Eric have enthusiasm.

b)



c)

$n$	
1	Barry and Eric have enthusiasm .
2	Barry and in have enthusiasm .
3	Barry and Eric have enthusiasm
4	Barry and but have enthusiasm .
5	Barry and in have enthusiasm
6	Barry and in. have enthusiasm .

Figure 4.2: Example of an  $n$ -best list extraction.

### 4.3 Language Model Variation

The integration of a statistical language model into HMM based recognition systems for offline handwritten text can be controlled by two factors. The *Grammar Scale Factor* (GSF), which weights the impact of the statistical language model against the optical recognition of the sentence, and the *Word Insertion Penalty* (WIP), which controls the segmentation rate of the recogniser.

By varying these two factors multiple sentences can be extracted, which are then used as alternative candidate sentences.

Integrations of language model variations in the post-processing rejection procedure have been successfully investigated in the field of continuous speech recognition (Sanchis et al., 2000; Zeppenfeld et al., 1997).

### 4.3.1 Language Model Integration

The goal of a handwritten text recognition system with an integrated statistical language model is to find the most probable word sequence  $\hat{W} = (w_1, \dots, w_n)$  for a given observation sequence  $X = (X_1, \dots, X_m)$  (Zimmermann and Bunke, 2004):

$$\hat{W} = \underset{W}{\operatorname{argmax}} p(W|X). \quad (4.1)$$

As an HMM based classifier does not compute  $p(W|X)$  but  $p(X|W)$ , Equation (4.1) can be transformed using Bayes' rule and rewritten as:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \frac{p(X|W)p(W)}{p(X)}. \quad (4.2)$$

The probability  $p(X)$  is constant for a given observation sequence  $X$ . Therefore, the denominator of Equation (4.2) can be neglected, and this results in the following simplified equation:

$$\hat{W} = \underset{W}{\operatorname{argmax}} p(X|W)p(W). \quad (4.3)$$

Because the sentence  $W$  with the maximum score is sought, taking the logarithm does not change the result but simplifies the computation, which results in:

$$\hat{W} = \underset{W}{\operatorname{argmax}} \log p(X|W) + \log p(W). \quad (4.4)$$

Following Equation (4.3) and (4.4) the result of the HMM classification system, the optical model  $p(X|W)$ , needs to be combined with the statistical language model. This statistical language model is represented by  $p(W)$ , the probability of the proposed sentence  $W$ .

The probability  $p(W)$  is often computed by so called *N-Gram Models* (Rosenfeld, 2000), where the probability of a word is dependent on the  $n - 1$  preceding words.

The HMM and the  $n$ -gram language model merely deliver approximations of probabilities. Therefore, two additional parameters,  $\alpha$  and  $\beta$ , are necessary to

compensate the deficiencies of the optical model and the language model.

$$\hat{W} = \underset{W}{\operatorname{argmax}} \log p(X|W) + \alpha \log p(W) + n\beta \quad (4.5)$$

In this thesis, the term *Grammar Scale Factor* (GSF)<sup>1</sup> is used for parameter  $\alpha$ , and *Word Insertion Penalty* (WIP) for the parameter  $\beta$ . The optimal values for  $\alpha$  and  $\beta$  are determined empirically because the probabilistic meaning of these two parameters is unclear, and although the two parameters are commonly utilized, very little research has addressed the meanings and systematic optimization of GSF and WIP (Takeda et al., 1998).

The GSF is used to weight the language model against the optical model produced by the HMM character models. If  $\alpha$  is set to 0 the language model is ignored, and increasing the GSF increases the impact of the language model on the most probable sentence calculation.

The WIP helps to control the insertion and deletion of words. It allows the system to balance the word insertion rate and the word deletion rate during the decoding. Multiplied with  $n$ , which is the length of sentence  $W$ , the WIP is added to the recognition score. Selecting  $\beta \leq 0$  can be used to reduce over-segmentation while choosing  $\beta \geq 0$  helps to decrease under-segmentation.

### 4.3.2 Variation of GSF and WIP

Zimmermann and Bunke (2004) investigate the systematical optimization of the integration of the two language model factors, GSF and WIP, into HMM based recognition systems for offline handwritten text. The experiments presented in this study show substantial improvements in the performance of the general text recogniser when GSF and WIP are optimized.

In the domain of continuous speech recognition, it has been shown that words with a high stability concerning the integration of a statistical language model are relatively error-free compared to words that rapidly change when this integration is varied (Zeppenfeld et al., 1997). This means that words which are observed less frequently in alternative candidate sentences, provided by language model variations, are more likely to be incorrect compared to words which appear in all or most candidate sentences.

In the domain of speech recognition, Sanchis et al. (2000) only vary the GSF in their study concerning efficient rejection strategies while Zeppenfeld et al. (1997) vary the GSF as well as the WIP. In this master thesis both approaches are investigated separately.

---

<sup>1</sup>In other works the terms linguistic weight, language weight or language model weight are used.

When the GSF is varied, while the WIP is kept constant,  $K$  different values  $\alpha_i$  ( $i = 1, \dots, K$ ) are chosen in the range  $[\alpha - x, \alpha + y]$  where  $x \in [0, \alpha]$ ,  $y \geq 0$ . The quantity  $\alpha$  represents the best performing GSF, which has been globally optimized (Zimmermann and Bunke, 2004).  $\alpha$  is the GSF of the hypothesised candidate sentence. After the selection of  $\alpha_i$  ( $i = 1, \dots, K$ ),  $\alpha$  in Equation (4.5) is substituted by  $\alpha_i$  and the resulting set of  $K$  alternative candidate sentences is given by

$$\{\hat{W}_1, \hat{W}_2, \dots, \hat{W}_K\} = \bigcup_{i=1, \dots, K} \underset{W}{\operatorname{argmax}} \log p(X|W) + \alpha_i \log p(W) + n\beta \quad (4.6)$$

The variation of GSF and WIP is an extension of Equation (4.6). Instead of choosing  $K$  values for the GSF,  $K$  parameter pairs  $(\alpha_i, \beta_i)$  ( $i = 1, \dots, K$ ) are selected where  $\alpha_i \in [\alpha - x, \alpha + y]$  and  $\beta_i \in [\beta - s, \beta + t]$ . The values of  $\alpha$  and  $\beta$  represent the globally optimized GSF and WIP of the hypothesised candidate sentence,  $x \in [0, \alpha]$  and  $y, s, t \geq 0$ . The set of alternative candidate sentences can then be expressed as follows:

$$\{\hat{W}_1, \hat{W}_2, \dots, \hat{W}_K\} = \bigcup_{i=1, \dots, K} \underset{W}{\operatorname{argmax}} \log p(X|W) + \alpha_i \log p(W) + n\beta_i \quad (4.7)$$

### 4.3.3 Example

In this example the same handwritten text image is considered as in the example shown in Figure 4.2 of Section 4.2. Multiple alternative candidate sentences for the sentence “*Barry and Eric have enthusiasm.*” are produced.

The GSF as well as the WIP are varied, the GSF in the range from 0 to 60, and the WIP in the range from -100 to 150. Assuming that  $\alpha = 30$  and  $\beta = 50$ , this leads to  $x, y = 30$ ,  $s = 150$ , and  $t = 100$ . Nine alternative candidate sentences are produced, and therefore  $K$  is set to 9. The values of  $\alpha_i$  and  $\beta_i$  are evenly distributed in the selected range. This leads to  $\alpha_{1,2,3} = 0$ ,  $\alpha_{4,5,6} = 30$ ,  $\alpha_{7,8,9} = 60$ ,  $\beta_{1,4,7} = -100$ ,  $\beta_{2,5,8} = 25$ , and  $\beta_{3,6,9} = 150$ .

The pre-computed lattice is rescored with each of the  $(\alpha_i, \beta_i)$  pairs according to Equation 4.7. The results of these rescoring processes are shown in Figure 4.3. In this example all the sentences differ from each other but generally this is not at all the case.

Figure 4.3 provides an excellent illustration of the influence of the WIP on the segmentation of the sentence. The average amount of words for  $\beta_i = -100$  is 4.33 compared to 7 when  $\beta_i = 150$ . In two cases ( $i = 6, 8$ ) increasing the WIP causes the dot at the end of the sentence to be correctly recognised.

Barry and Eric have enthusiasm.

$i$	$\alpha_i$	$\beta_i$	$\hat{W}_i$
1	0	-100	Barry arm inch we enthusiasm
2	0	25	Barry arm inch we m run rush
3	0	150	B my arm inch we m run rush :
4	30	-100	Barry and include enthusiasm
5	30	25	Barry and Eric have enthusiasm
6	30	150	Barry and Eric have enthusiasm .
7	60	-100	Barry and include enthusiasm
8	60	25	Barry and include enthusiasm .
9	60	150	Barry and in have enthusiasm .

**Figure 4.3:** Candidate sentences based on language model variation.



# Chapter 5

## Experiments and Results

In this chapter experiments conducted in order to illustrate the behaviour of the different rejection strategies are presented. In Section 5.1 measure and plots are presented which enable to quantify the performance of rejection strategies. Section 5.2 describes the experimental setup, and explains training and validation. The results of the test set runs are presented in Section 5.3 and discussed in Section 5.4.

### 5.1 Evaluation Methodology

For every recognised input word, the post-processing rejection procedure computes a confidence measure  $\rho$ , ranging in the interval  $[0,1]$ , on which the decision, whether to accept or to reject the word, is based. This confidence measure quantifies the assumed correctness of the word. The closer  $\rho$  is to 1, the more certain is the system that a word is correct. And the closer  $\rho$  is to 0, the less confidence is given to the recognition. The decision, whether to accept or to reject a word, is controlled by a threshold  $t$ . Words with a confidence measure  $\rho$  higher than or equal to  $t$  are accepted, while words with  $\rho$  below threshold  $t$  are rejected.

Confusion matrix and the statistics derived from this matrix are used to illustrate the performance of the different rejection strategies. Furthermore, Receiver-Operating-Characteristic (ROC) curve as well as error-reject plots are presented, which graphically illustrate the performance of a rejection strategy.

#### 5.1.1 Confusion Matrix

The efficiency of the rejection procedures are evaluated using a confusion matrix (Figure 5.1). A recognised word can either be a correct or a false recogni-

	Output of the recogniser:	
	Correct	False
Accept	<b>CA</b>	<b>FA</b>
Reject	<b>FR</b>	<b>CR</b>

**Figure 5.1:** *Confusion matrix.*

tion which can be accepted or rejected by the post-processing rejection procedure. The result of the post-processor therefore falls in one of the four categories:

**Correct Acceptance (CA)** A correctly recognised word is accepted by the post-processor.

**False Acceptance (FA)** A word is not recognised correctly but nonetheless it is accepted by the post-processor.

**Correct Rejection (CR)** An incorrectly recognised word is rejected by the post-processing rejection procedure.

**False Rejection (FR)** A word that is recognised correctly is rejected by the post-processor.

Obviously correct acceptances and correct rejections are the desired categories, while false acceptances and false rejections are errors and thus to be avoided.

### 5.1.2 Performance Measures

Based on the confusion matrix many different statistics quantifying the behaviour and the performance of the underlying rejection procedure can be derived. In this thesis the following measures have been used:

**Error Rate (ERR)** The error rate describes the ratio of false accepted words to all recognised words:

$$ERR = \frac{FA}{CA + FA + CR + FR} \quad (5.1)$$

**Reject Rate (REJ)** The reject rate describes the ratio of all rejected words to all recognised words:

$$REJ = \frac{CR + FR}{CA + FA + CR + FR} \quad (5.2)$$

**False Acceptance Rate (FAR)** The false acceptance describes the false acceptance on wrongly recognised words by measuring the percentage of incorrectly recognised words that are accepted:

$$FAR = \frac{FA}{FA + CR} \quad (5.3)$$

**False Rejection Rate (FRR)** The false rejection rate describes the false rejection on correctly recognised words by measuring the percentage of correctly recognised words that are rejected:

$$FRR = \frac{FR}{FR + CA} \quad (5.4)$$

### 5.1.3 Statistical Background

In statistical theory, the problem of decision making is expressed in the decision landscape (Daugman, 2000; Maltoni et al., 2003). Two distributions represent the two states of the world, the hypotheses  $H_0$  and  $H_1$ .  $H_0$  assumes that the input word has been recognised incorrectly, while  $H_1$  assumes a correct recognition. Figure 5.2 illustrates the idea of the decision landscape. The abscissa is the confidence measure produced by the post-processing reject model, while the vertical axis represents the probability density. The decision criteria whether to accept or to reject is given by the threshold  $t$ . Acceptance is on the right side of  $t$ , while rejection is on the left side of  $t$ . Different thresholds  $t$  lead to different rejection and acceptance result.

The likelihoods that the decisions are correct or not correspond to the four areas that lie under the two probability distributions of Figure 5.2 on either side of the decision criteria  $t$ . Moving the threshold  $t$  to the right or to the left will change the relative likelihood of the four outcomes. Shifting  $t$  to the right is raising correct rejection and false rejection, while shifting  $t$  to the left is increasing correct acceptance and false acceptance.

In this master thesis, 100 different values equally distributed in the interval  $[0, 1]$  have been used for threshold  $t$ . Thus, the full range of possible applications is covered. Extremely low rejection rates as well as very high rejection rates are possible.

### 5.1.4 ROC Curve Plot

As can be seen in the decision landscape (Figure 5.2), False Acceptance Rate (FAR) and False Rejection Rate (FRR) strictly trade off. Both of them are functions of the threshold  $t$ . If  $t$  is decreased and the system is more tolerant to

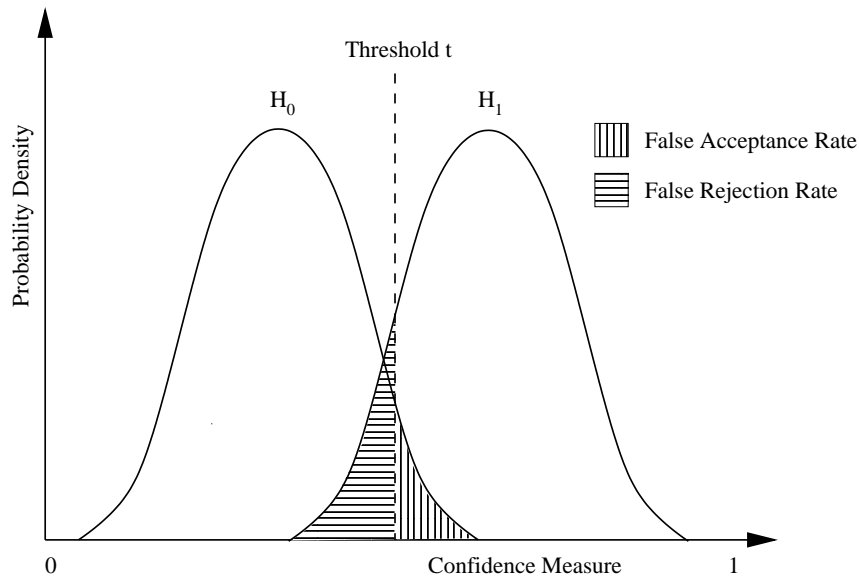


Figure 5.2: *Decision landscape.*

input variation, then FAR increases while FRR decreases; vice versa, if  $t$  is raised, then FRR increases and FAR decreases and the system becomes more strict.

To illustrate the performance of a rejection model, a *Receiver-Operating-Characteristic* (ROC) curve is constructed by plotting the FAR of Equation 5.3 against the FRR of Equation 5.4. An example of a ROC curve is given in Figure 5.3. To reach a low false acceptance rate, the rejection threshold must be raised, leading to an increased false rejection rate. In contrast, a low false rejection rate is at the cost of a higher false acceptance rate.

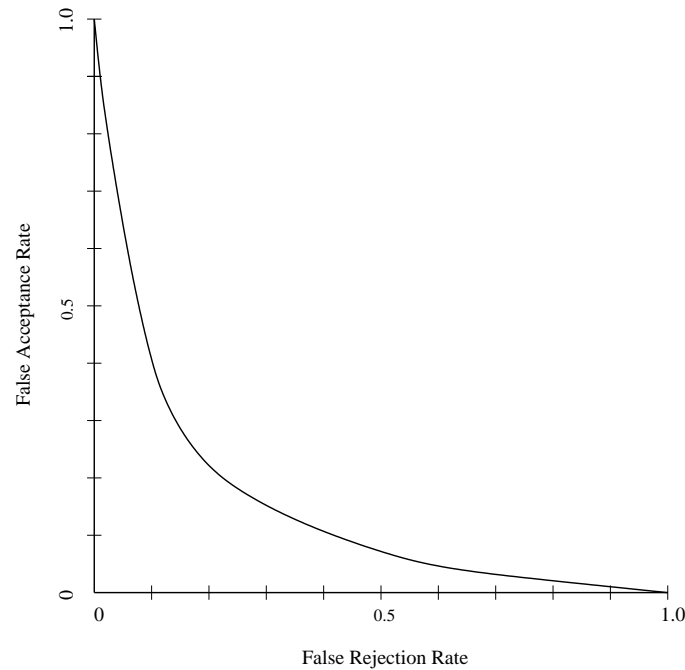
As the goal of a general rejection strategy is to reach both, a low FAR and a low FRR, a good rejection model produces a ROC curve close to the origin.

In contrast to a general rejection strategy, as it is considered in this thesis, the tolerable FAR (FRR) might be fixed in a specific application. In this case the post-processing rejection strategy is optimized on this FAR (FRR).

### 5.1.5 Error-Reject Plot

A second characteristic curve that can be derived from the aforementioned measures is the *Error-Reject Plot*. Here the Error Rate (ERR) of Equation 5.1 is plotted against the Reject Rate (REJ) of Equation 5.2. Raising the REJ reduces the ERR of the system as ERR and REJ naturally trade off.

An example of an error-reject plot is provided in Figure 5.4. As expected, in-



**Figure 5.3:** Receiver-Operating-Characteristic (ROC) curve example.

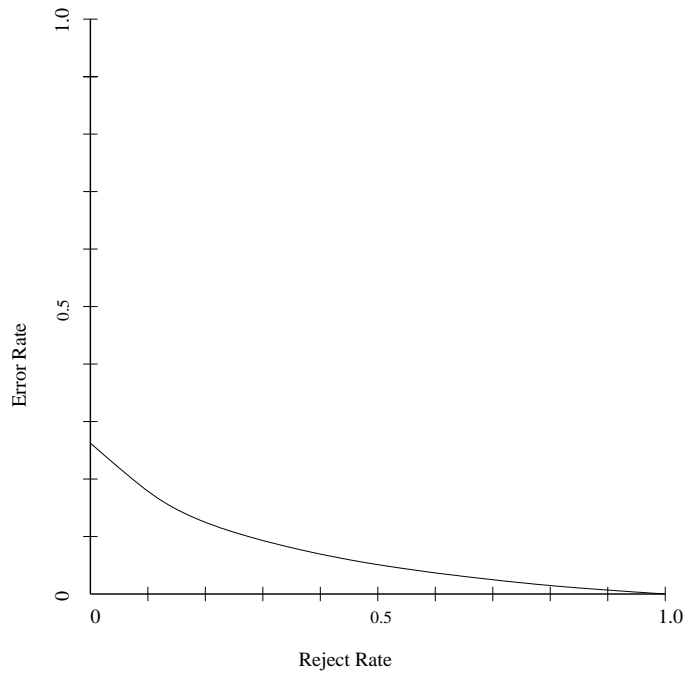
creasing the number of rejections clearly decreases the number of incorrectly accepted words. Of course, the better a rejection procedure performs, the larger is the decrease of the error rate.

In a specific application, the acceptable ERR is usually given, and the rejection procedure is trained to achieve this ERR at a REJ as low as possible.

In an error-reject plot, it can be directly perceived how many samples have to be rejected to achieve a given error rate. This makes the error-reject plot easier to read than the *Receiver-Operating-Characteristic* (ROC) curve. The latter has the advantage that reject models, which produce nearly the same curves in an error-reject plot, nevertheless generate ROC curves that can visually be distinguished. That is why the ROC curves are more appropriate for comparing different reject models while the error-reject plot is more convenient to describe the overall performance of a single reject model.

## 5.2 Experimental Setup

The handwriting recognition system and the database used for the experiments are described in this section. Furthermore, optimizations of the language model integration and of the Multi-Layer Perceptron (MLP) architecture are discussed before the training of the reject models is explained.



**Figure 5.4:** *Error-reject plot example.*

### 5.2.1 Offline Handwriting Recognition System

All experiments reported in this master thesis make use of the Hidden Markov Model (HMM) based handwritten sentence recognition system described in Section 2.2. The recognition system uses word bigram models as a statistical language model and is based on individual character models with a linear topology and multi-Gaussian output densities (see Marti and Bunke (2001); Zimmermann and Bunke (2004) for details).

Furthermore, the same experimental setup as described in Zimmermann and Bunke (2004) is used. The number of states is selected depending on the individual character, and a mixture of eight Gaussians for every state is used. The tagged LOB Corpus (Johansson et al., 1986) is used for the included bigram language model.

### 5.2.2 Database

The training and the test set each contain 200 complete English sentences. The 400 sentences, with an average length of 23.1 words, have been written by 200 individual writers, where the first 100 writers are present in the training set while the second 100 writers contributed to the test set. The lexicon has been closed over the test (training) set and included 8,819 (8,825) words. The closing

$i$	$\alpha_i$
1	0
2	0.9523
3	1.9047
4	2.8571
	...
	...
64	60

**Table 5.1:** Values  $\alpha_i$  used as GSF in language model variation.

of the lexicon over the test (training) set ensures that all words of the test set are contained in the task lexicon.

The sentences originate from the segmented IAM database (Marti and Bunke, 1999; Zimmermann and Bunke, 2002), which has been created at the Institute of Computer Science and Applied Mathematics at the University of Berne to build, train and test offline handwriting recognition systems for general English texts. The entire database consists of about 1,500 forms of handwritten text, including more than 10,000 handwritten text lines written by over 500 writers.

### 5.2.3 Optimizing the Language Model Variations

As explained in Section 4.3, the global optimal values of the Grammar Scale Factor (GSF)  $\alpha$  and the Word Insertion Penalty (WIP)  $\beta$  have to be determined experimentally, because there is no exact mathematical model of GSF and WIP. Zimmermann and Bunke (2004) have systematically optimized the language model integration for this experimental setup. They show in their study, that  $\alpha = 30$  and  $\beta = 50$  maximize the recognition rate. Because the same experimental setup is used in this master thesis, the same values for  $\alpha$  and  $\beta$  are used.

The number of alternative candidate sentence  $K$  is set to 64 and the GSF is varied in the range from 0 to 60, while the WIP is varied in the range from -100 to 150. These values have been shown to perform well in the preliminary studies. In terms of Subsection 4.3.2, this means that  $x = y = 30$ , while  $s = 150$  and  $t = 100$ . The variations of the GSF and the WIP are equally distributed.

Table 5.1 shows the different values for  $\alpha_i$  in the case where only the GSF is varied. For example, if  $i = 35$  the resulting value for  $\alpha_{35}$  is 32.3809.

The values of  $\alpha_i$  and  $\beta_i$  used when both the GSF and the WIP are varied are shown in Table 5.2. If, for example,  $i = 52$ , the resulting values are  $\alpha_{52} = 51.4285$  and  $\beta_{52} = 7.1428$ .

$i$	$\alpha_i$	$i$	$\beta_i$
1-8	0	1,9,...,57	-100
9-16	8.5714	2,10,...,58	-64.2857
17-24	17.1428	3,11,...,59	-28.5714
25-32	25.7142	4,12,...,60	7.1428
33-40	34.2857	5,13,...,61	42.8571
41-48	42.8571	6,14,...,62	78.5714
49-56	51.4285	7,15,...,63	114.2857
57-64	60	8,16,...,64	150

**Table 5.2:** Values  $\alpha_i$  and  $\beta_i$  used as GSF and WIP in language model variation.

### 5.2.4 Multi-Layer Perceptron Optimization

To conduct rejection experiments with Model 3 as introduced in Section 3.4, the number of internal neurons  $m$  of the MLPs has to be determined. A traditional value for  $m$  is half of the sum of input and output neurons.

If  $K = 64$  is the number of input neurons, and two output neurons (accept and reject score) are present, then 33 internal neurons are needed. With these 33 internal neurons, 2213 weights have to be trained.

But as there are only about 4150 word samples available in the training set, less than 2 training samples would be available per weight for training.

To avoid under-training, two strategies are considered:

*Reduction of internal neurons:* Not an MLP with 33 internal neurons, but MLPs with 20, 10 and 2 internal neurons are evaluated.

*Reduction of input channels:* Instead of 64 input channels, only 16 input channels are used. These smaller MLPs are evaluated with 9, 5 and 2 internal neurons.

On the training set, both strategies are validated for the different candidate sentence generation strategies. The results of this validation is provided in Table 5.5.  $K$  is the number of input channels, while  $m$  is the number of hidden neurons. The values are FAR values sampled from an ROC curve where FRR is 0.1, ..., 0.9. The best value of each column is highlighted with bold font.

If the alternative candidate sentences originate from language model variation (GSF and GSF & WIP) the best performing MLPs are the one with 64 input channels and 20 internal neurons. By extracting the candidate sentences from  $n$ -best lists, the MLPs with 64 input and 10 internal neurons outperform the other ones. In any case, the reduction of input channels does not provide the desired effect at all.



		GSF & WIP								
$K$	$m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
64	20	<b>0.22571</b>	<b>0.12518</b>	<b>0.07711</b>	<b>0.04760</b>	<b>0.02902</b>	<b>0.01941</b>	<b>0.01160</b>	<b>0.00544</b>	<b>0.00160</b>
64	10	0.27727	0.15306	0.10014	0.06904	0.04655	0.02288	0.01393	0.00779	0.00216
64	2	0.31471	0.18978	0.12316	0.08370	0.05970	0.04048	0.02319	0.01546	0.00773
16	9	0.33554	0.20380	0.13474	0.09356	0.05900	0.03791	0.02875	0.01083	0.00464
16	5	0.33573	0.20613	0.13481	0.09092	0.06110	0.03879	0.02188	0.01136	0.00406
16	2	0.34505	0.21441	0.14369	0.09689	0.06699	0.04640	0.03465	0.02310	0.01155

		GSF								
$K$	$m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
64	20	0.43446	<b>0.20371</b>	<b>0.13620</b>	<b>0.08990</b>	<b>0.06320</b>	<b>0.04119</b>	<b>0.02189</b>	<b>0.00939</b>	<b>0.00272</b>
64	10	<b>0.43011</b>	0.20824	0.14025	0.09311	0.06570	0.04225	0.02224	0.01137	0.00346
64	2	0.43469	0.22433	0.15562	0.09337	0.06795	0.04497	0.02751	0.01217	0.00609
16	9	0.43933	0.21424	0.14706	0.09961	0.07117	0.04384	0.02264	0.01219	0.00501
16	5	0.43933	0.21488	0.15007	0.09712	0.07023	0.04430	0.02322	0.01209	0.00500
16	2	0.48847	0.21471	0.15202	0.09737	0.06746	0.04424	0.02458	0.01283	0.00642

		NBEST								
$K$	$m$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
64	20	0.69722	0.40195	0.10800	0.05915	0.04259	0.02533	0.01529	0.00775	0.00246
64	10	<b>0.68747</b>	<b>0.39718</b>	<b>0.09943</b>	<b>0.04387</b>	<b>0.02538</b>	<b>0.01554</b>	<b>0.00830</b>	<b>0.00545</b>	<b>0.00075</b>
64	2	0.71831	0.43661	0.18885	0.09237	0.06376	0.04398	0.02635	0.01742	0.00871
16	9	0.80232	0.60465	0.40697	0.20930	0.08794	0.04791	0.02429	0.01085	0.00401
16	5	0.80246	0.60493	0.40739	0.20986	0.08491	0.04476	0.02520	0.00933	0.00421
16	2	0.80267	0.60535	0.40802	0.21070	0.10398	0.06091	0.04018	0.01781	0.00880

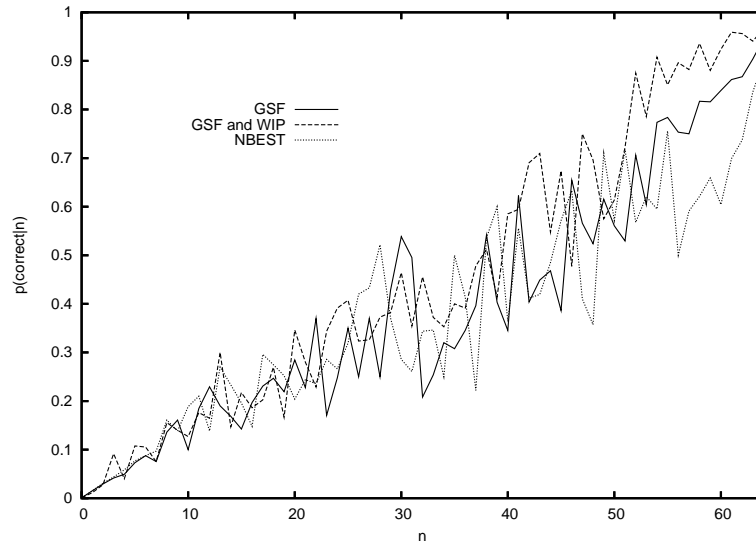
Figure 5.5: Multi-Layer Perceptron validation.

Therefore, confidence measure  $\rho_3$  uses the following amount of input channels and internal neurons: 64 input channels with 20 internal neurons for the language model variation based rejection strategy, and 64 input channels with 10 internal neurons if the candidate sentences originate from  $n$ -best lists.

### 5.2.5 Training and Smoothing

The quantities  $p(c|n)$ ,  $p(c|w)$  and  $p(n|c)$  have to be estimated using relative frequencies obtained from the training set. These quantities are estimated separately for the different alternative candidate sentences extraction strategies. Additionally, multiple MLPs have to be trained for Model 3 using the well-known back-propagation algorithm.

The probabilities  $p(c|n)$  are estimated for every  $n = 0, \dots, 64$  as described in Section 3.2. For each of the alternative candidate sentence extraction strategies, the resulting frequencies of the training are shown in Figure 5.6. A linear trend can be determined, but the curves also contain large jumps, as for example the  $n$ -best curve, where  $p(\text{correct}|n = 28) = 52\%$  while  $p(\text{correct}|n = 29) = 37\%$ . Additionally, it can be seen that the variation of GSF and WIP reaches a probability  $p(\text{correct}|n)$  over 80% for every  $n > 54$ , while  $n$ -best list extraction



**Figure 5.6:** Relative frequencies  $p(c|n)$  obtained from the training set.

achieves  $p(\text{correct}|n)$  over 80% only if  $n = 63$  or  $n = 64$ .

Examples of the estimated probabilities  $p(c|w)$  are provided in Table 5.3. Only words with more than ten input samples in the training set are considered in the estimation of  $p(c|w)$ . In absolute values this means that of the 1457 different words appearing in the training set, the quantities  $p(c|w)$  are only available for 50 words. However, these 50 different words represent more than 50% of the words in the training set. Furthermore, large differences in the probabilities of different words are present. While  $p(\text{correct}|\text{the})$  is higher than 90%, the same probabilities for the words *her* and *I* lie slightly below 39 %.

For every alternative sentences extraction strategy the estimated probabilities  $p(n|c)$  are shown in Figure 5.7. The first plot shows  $p(n|\text{correct})$ , the probability of a correctly recognised sample of being in class  $n$ , and the second plot shows  $p(n|\text{incorrect})$ , the probability of a wrongly recognised sample of being in class  $n$ .

Estimation of probabilities using relative frequencies implies that enough training samples are available for every quantity to be estimated. But as the training set is never arbitrarily large, it may occur that too few samples are available for an adequate estimation of the probabilities. In this case the estimation results are smoothed.

In this master thesis smoothing is applied to the values of  $p(c|n)$ , because the

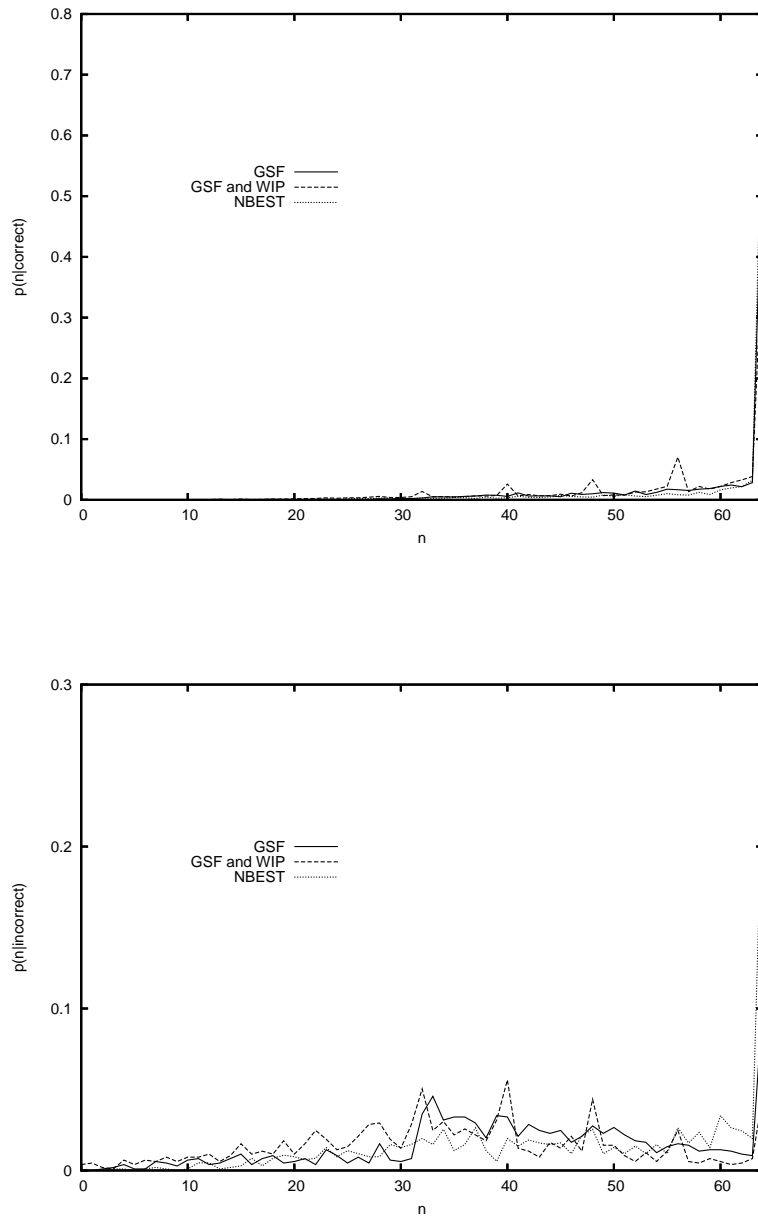
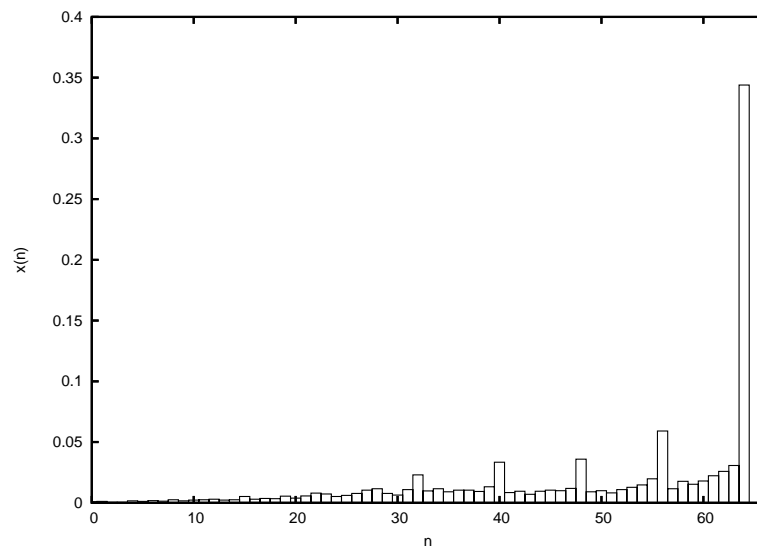


Figure 5.7: Probabilities  $p(n|c)$  estimated on the training set.

$w$	$p(\text{correct} w)$
It	0.8
as	0.6364
but	0.6875
on	0.7778
not	0.875
the	0.9039
her	0.3889
for	0.7813
an	0.6
I	0.3846

**Table 5.3:** Extract of the frequencies  $p(\text{correct}|w)$  computed on the training set.



**Figure 5.8:** Distribution of training samples. The value  $x(n)$  stands for the relative amount of samples in class  $n$ .

distribution of training samples on the different values of  $n$  are quite variable. This fact is illustrated in Figure 5.8 where  $x(n)$  represents the relative amount of samples in one class. This means that for any  $n$ ,  $x(n)$  multiplied with the total number of training samples are the number of training samples  $y(n)$  available to estimate  $p(c|n)$ . Figure 5.8 shows that  $x(n)$  differs significantly for various  $n$ . Almost 35% of the training sample are in one single class ( $n = 64$ ), while very few training samples are available for  $n < 15$ .

For the estimated probabilities  $p(c|n)$ , smoothing is performed by moving the values towards the straight line  $\frac{n}{K}$ , where  $K$  is the number of alternative candidate sentences, if not enough training samples  $y(n)$  are available to estimate  $p(c|n)$ . The strength of the smoothing is controlled by the threshold  $\tau$ , which decides if and how much the calculated frequency  $\hat{p}(c|n)$  is smoothed. Equation 5.5 provides a mathematical representation of the smoothing process.

$$p(c|n) \approx \begin{cases} \hat{p}(c|n) & : y(n) > \tau \\ \frac{y(n)}{\tau} \cdot \hat{p}(c|n) + \frac{(\tau - y(n))}{\tau} \cdot \frac{n}{K} & : y(n) \leq \tau \end{cases} \quad (5.5)$$

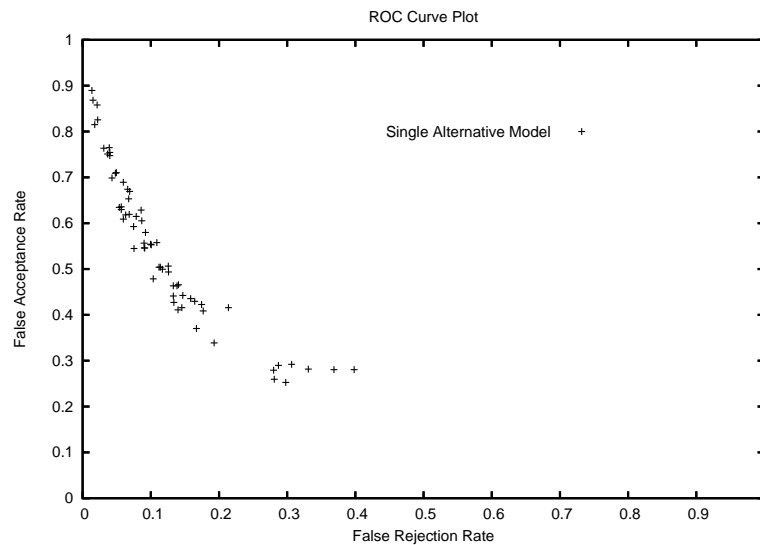
In the experiments performed within the scope of this master thesis, smoothing is applied with  $\tau = 20$ , which has not been experimentally optimized, but seems to be a reasonable choice. If less than 20 samples of the training set are available for estimating one value  $p(c|n)$ , smoothing is applied, and the value is moved towards  $\frac{n}{K}$  depending on the available number of samples. If more than 20 samples are available, the obtained relative frequency  $\hat{p}(c|n)$  is used as an estimation for  $p(c|n)$ .

### 5.3 Test Set Results

In this section the performance of the four reject models combined with the three strategies of alternative candidate sentences generation is presented. To increase readability, plots are shown and described for every model and every sentence generation strategy, instead of showing a curve for each experiment in one plot, which would lead to a huge confusing plot with more than ten curves.

First every reject model is evaluated separately, showing the impact of the origin of the candidate sentences on the performance of the different reject models. In a second part Model 1, Model 2, and Model 3 are compared for each of the three alternative candidate sentences generation strategies, which illustrates the influence of the chosen confidence measure on the performance of the rejection system.

The outputs of the handwriting recognition system are recognition lattices which are the same for every test run. The rejection experiments described



**Figure 5.9:** ROC Curve Plot of Model 0.

in this section generate candidate sentences from these lattices, apply one of the reject models, and measure the performance of the rejection procedure.

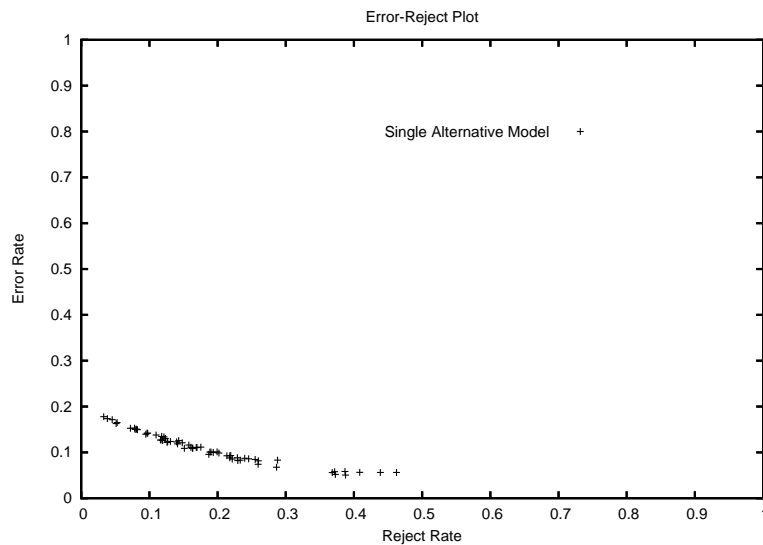
### 5.3.1 Model 0 Results

For the single alternative model (see Section 3.1 for details), 64 pairs  $(\alpha_i, \beta_i)$  are determined by varying the GSF and the WIP. With each of these pairs the recognition lattices are rescored and 64 sentences are generated out of every sentence image. Each of the 64 sentences is used as a single alternative to the hypothesised sentence, and the confidence measure  $\rho_0$  (see Equation 3.1) is computed 64 times to obtain different levels of rejection strictness.

Figure 5.9 shows the performance of Model 0 in terms of false acceptance and false rejection. The 64 values plotted in the Receiver-Operating-Characteristic (ROC) curve correspond to the 64 pairs  $(\alpha_i, \beta_i)$  used to generate the single alternative sentence.

As expected, raising the FRR decreases the FAR. It can also be seen that most of the values are in the FRR range from 0% to 20%, while there are no values for a FRR higher than 50%.

Model 0 achieves a FRR of 20% at 33.8% false acceptance. A FRR below 10% can only be obtained at a FAR of 48%. In this experiment no FAR under 20%



**Figure 5.10:** *Error-Reject Plot of Model 0.*

could be achieved because the data density for higher FRRs is too small.

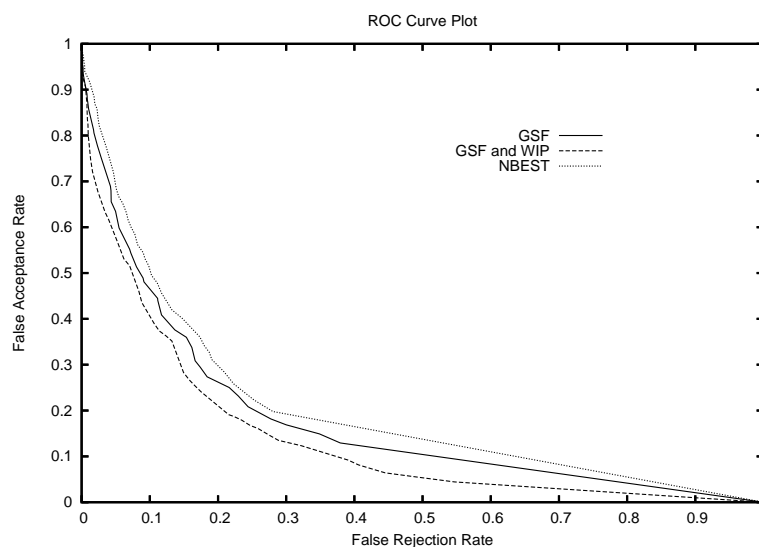
An error-reject plot, illustrating the performance of Model 0, is given in Figure 5.10. Again, the values in the error-reject plot correspond to the 64 pairs  $(\alpha_i, \beta_i)$  which were used for rescoring the lattice and generating the single alternative sentence.

Similar to the absence of FRR values over 50% in the ROC curve plot, no data is available for reject rates over 50% in the error-reject plot, and most values fall below 30%.

To keep the error rate under 10%, Model 0 needs to reject 18.7% of the words. The lowest error rate achieved with this experimental setup is 5%, at a reject rate of 38.7%.

### 5.3.2 Model 1 Results

The confidence measure  $\rho_1$  of Model 1 (see Equation 3.2) which is based on multiple alternatives has been described in Section 3.2. For this confidence measure 64 alternative candidate sentences are generated from the recognition lattices by means of GSF variation, GSF and WIP variation, and  $n$ -best list extraction.



**Figure 5.11:** ROC Curve Plot of Model 1.

For each of the alternative sentences extraction strategies,  $p(c|n)$  is estimated on the training set. The result of these estimations have been presented previously in Figure 5.6.

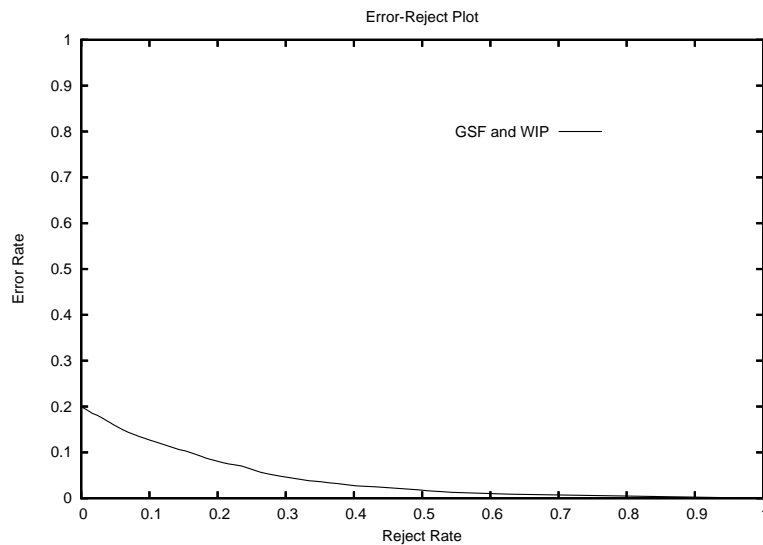
A comparison of the three alternative sentences extraction strategies is provided in Figure 5.11. The ROC curve plot shows the performance of GSF variation, GSF and WIP variation, and  $n$ -best list extraction integrated into the confidence measure  $\rho_1$  of Model 1.

Producing the alternative candidate sentences via GSF and WIP variation clearly outperforms the other strategies for any FAR. It reaches a FAR under 20%, while keeping the FRR below 21%. In comparison, varying only the GSF achieves a FAR under 20% at a FRR of 25.5%. Even worse is the use of an  $n$ -best list as source of alternative candidate sentences. This method requires a FRR of 28% to obtain a false acceptance rate below 20%.

Figure 5.12 shows the overall performance of Model 1, with alternatives based on GSF and WIP variations (the best performing strategy for Model 1) in an error-reject plot. If an error rate of 10% is tolerable, the reject rate lies slightly below 16%. To achieve an error rate under 5%, 28.5% of the input has to be rejected.

The error rate of this recognition system without any rejections lies at 19.9%. By rejecting 10% of the input words, this error rate can be reduced from over





**Figure 5.12:** *Error-Reject Plot of Model 1.*

36% to 12.6%. A reduction of 59.8% is achieved with the rejection of 20% of the input.

### 5.3.3 Model 2 Results

Similar to the experiments conducted with Model 1, 64 alternative candidate sentences produced by GSF variation, GSF and WIP variation as well as  $n$ -best list extraction, have been tested with  $\rho_2$  from Equation 3.6, which is the confidence measure of Model 2 (see Section 3.3 for details about Model 2). In contrast to Model 1, Model 2 takes into account which word  $w$  is currently being processed.

The probabilities  $p(c|w)$  and  $p(n|c)$  are estimated during the training phase. Furthermore,  $p(c|n)$  is estimated, since  $p(c|w)$  is possibly not available for every word  $w$  in the test set, and in this case, Model 1 with its confidence measure  $\rho_1$  is used instead of Model 2. The result of the training phase has been presented previously in Figure 5.6 ( $p(c|n)$ ), Figure 5.7 ( $p(n|c)$ ), and Table 5.3 ( $p(c|w)$ )

The ROC curve plot of Model 2 with the different alternative candidate sentences generation strategies is shown in Figure 5.13. The variation of GSF and WIP yields the best results, and the variation of the GSF alone clearly out-

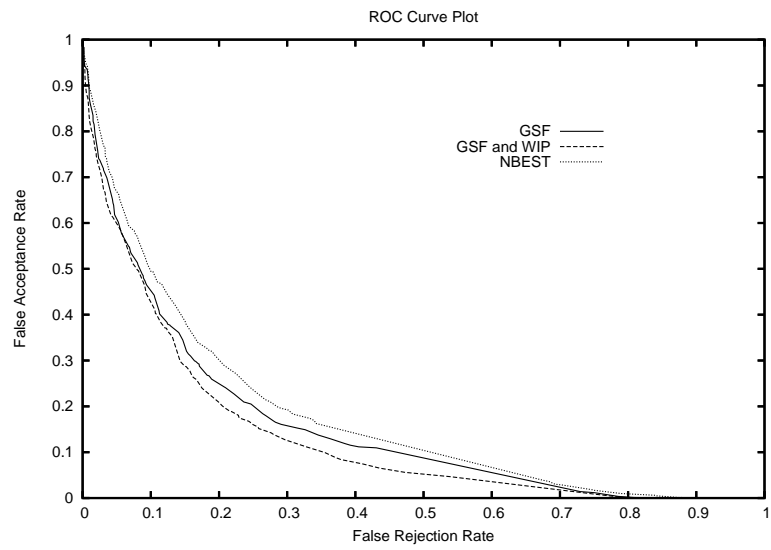


Figure 5.13: ROC Curve Plot of Model 2.

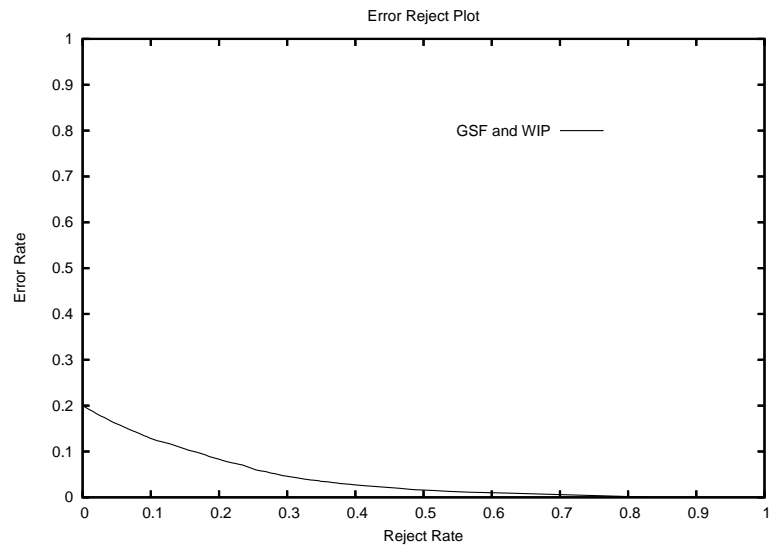


Figure 5.14: Error-Reject Plot of Model 2.

performs the  $n$ -best list extraction. For a FAR under 20%, the GSF and WIP variation requires 20.6% FRR, GSF variation needs 25.1% FRR, while  $n$ -best list extraction requires 28.5% FRR.

The error-reject plot of Figure 5.14 shows the performance of Model 2 using GSF and WIP variation to extract multiple candidate sentences. To obtain an error rate of 10%, 16.3% of the input words have to be rejected. For an error rate of 5%, the required rejection rate is 28%.

Compared to the original error rate of 19.9% of the recognition system, Model 2 achieves an error reduction of 35.5% by rejecting 10% of the words. A rejection rate of 20% of the input words reduces the error rate by 58.2% down to 8.3%.

### 5.3.4 Model 3 Results

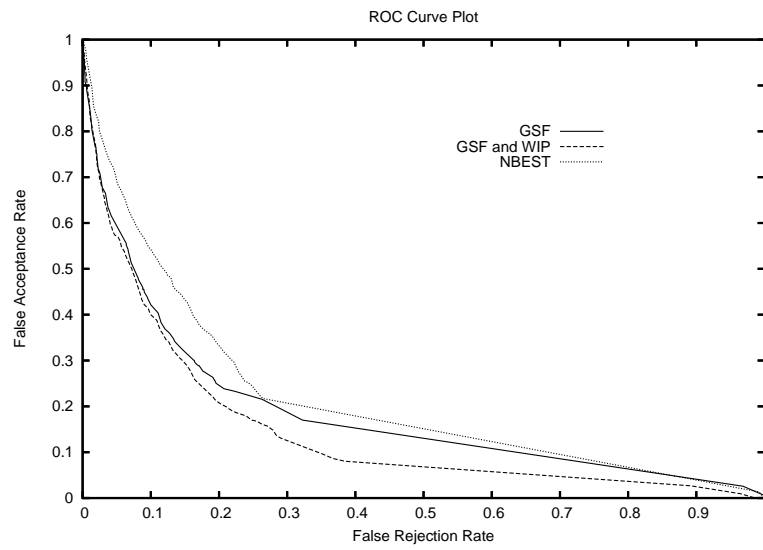
For each of the three alternative candidate sentence extraction strategies, ten MLPs are trained in the cross validation process of Model 3 described in Section 3.4. From the training set, which contains 200 sentences, 180 sentences are used to train an MLP, while 20 sentences are used for validation. Training is done with the well-known back-propagation algorithm (Rojas, 1996), which is a standard algorithm for training a supervised neural network.

From the ten MLPs the confidence measure  $\rho_3$  is computed as introduced in Equation 3.9. The number of input channels and internal neurons is set to the optimized values, which are 64 input channels for all experiments and 20 internal neurons for the experiments with language model variations (GSF variation, GSF and WIP variation) and 10 internal neurons when the candidate sentences origin from  $n$ -best lists (see Subsection 5.2.4).

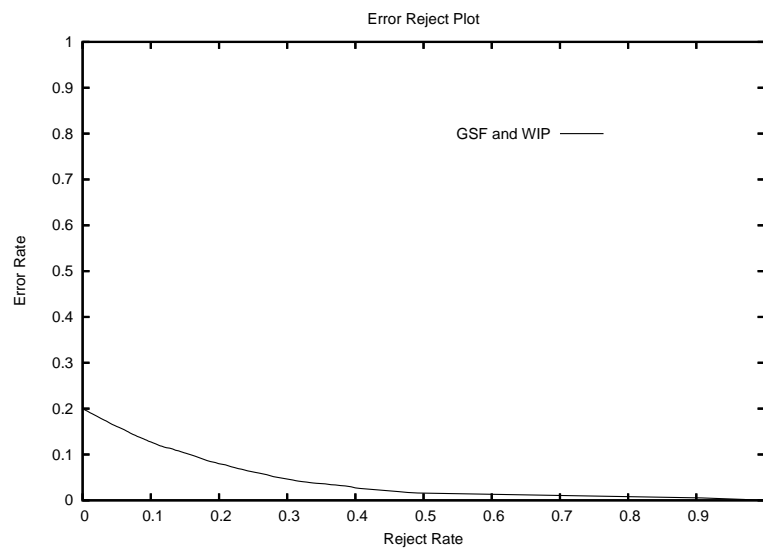
The resulting ROC curve plot is shown in Figure 5.15. The variation of GSF and WIP clearly outperforms the other strategies of alternative candidate sentences generation. The worst performance is shown by extracting  $n$ -best lists as alternative candidate sentences. With a FRR under 21%, the variation of GSF and WIP enables Model 3 to reach a FAR below 20%, where varying only the GSF requires 29.4% FRR, and extracting  $n$ -best list needs even 32.6% FRR.

Figure 5.16 shows the overall performance of the best strategy for Model 3, which is GSF and WIP variation based alternative candidate sentences generation, in an error-reject plot. Model 3 achieves an error rate under 10%, at a reject rate of 15.7%. For an even lower error rate of 5%, a reject rate of 28.6% is reached.

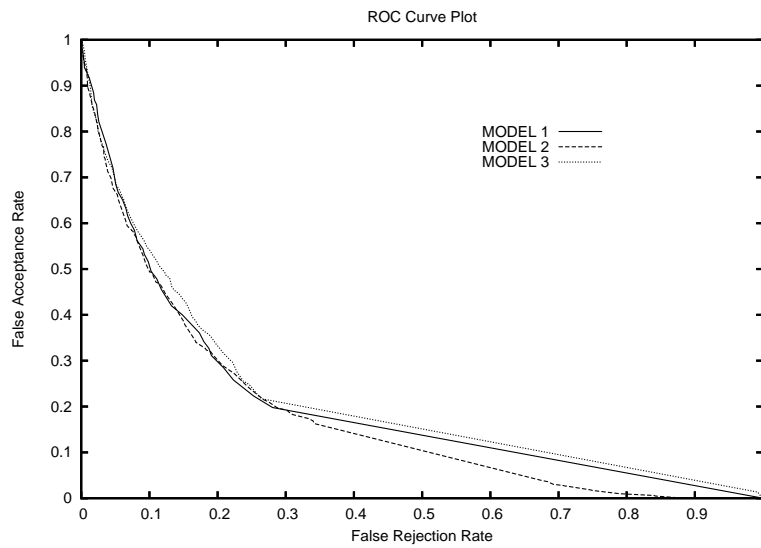
Model 3 allows to reduce the original error rate of the recognition system by 36% to 12.7% at a reject rate of 10%. At a reject rate of 20%, the error rate drops by 59.8% to 8%.



**Figure 5.15:** ROC Curve Plot of Model 3.



**Figure 5.16:** Error-Reject Plot of Model 3.



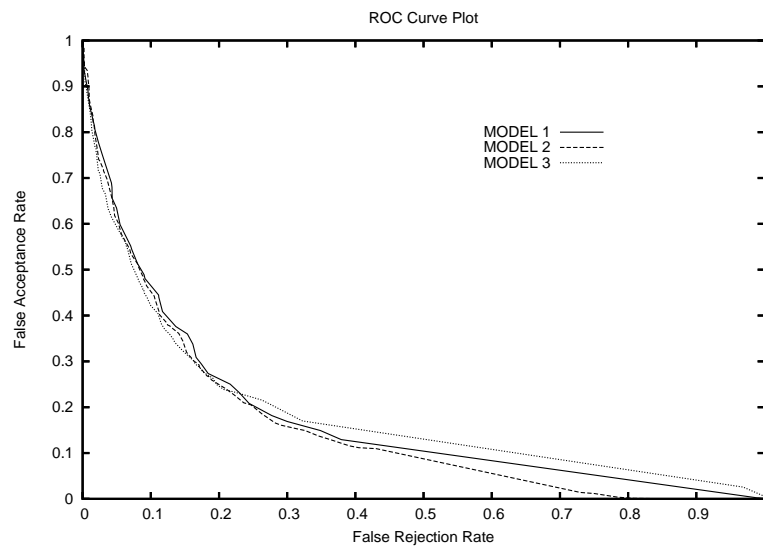
**Figure 5.17:** ROC Curve Plot of Model 1, Model 2, and Model 3 with alternative candidate sentences based on  $n$ -best list extraction.

### 5.3.5 N-Best Lists

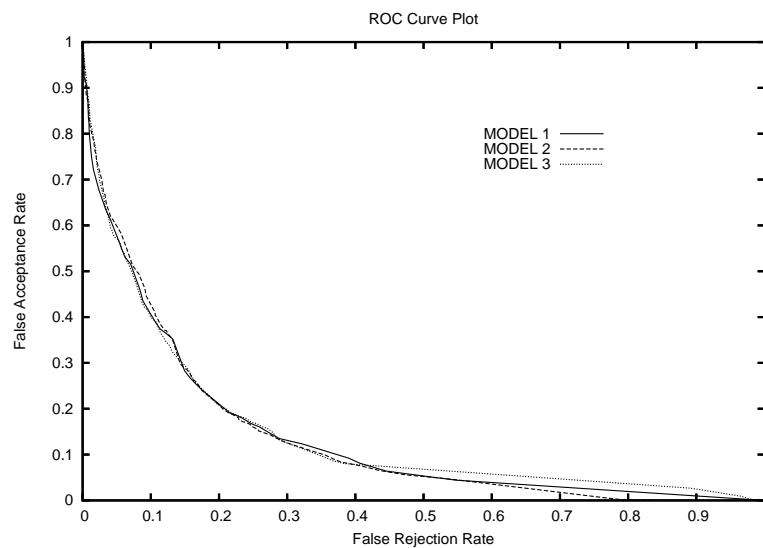
The ROC curve plot of Figure 5.17 shows the performance of the confidence measures  $\rho_1, \rho_2$ , and  $\rho_3$  of Model 1, Model 2, and Model 3 with alternative candidate sentences generated by  $n$ -best list extraction. It can be seen that no model outperforms the other for every value of FRR, especially Model 1 and Model 2 are competing. For a FAR below 20% Model 2 outperforms Model 1, while, for example, for a FAR between 20% and 30% Model 1 leads to better results. Slightly worse is the performance of Model 3, the model which uses MLPs to calculate the confidence measure.

### 5.3.6 Variation of GSF

Figure 5.18 illustrates the performance of the confidence measure of the different reject models with alternative candidate sentences based on GSF variation. Model 3 delivers the best results for a FAR between 30% and 100%, while Model 2 outperforms the other models for a FAR lower than 30%.



**Figure 5.18:** ROC Curve Plot of Model 1, Model 2, and Model 3 with alternative candidate sentences based on GSF variation.



**Figure 5.19:** ROC curve plot of Model 1, Model 2, and Model 3 with alternative candidate sentences based on GSF and WIP variation.

### 5.3.7 Variation of GSF and WIP

The curves in the ROC curve plot of Figure 5.19 show the performance of Model 1, Model 2 and Model 3 with alternative candidate sentences based on GSF and WIP variation. The results of the three reject models are quite similar. Model 2 slightly outperforms Model 1 and Model 3 for small FARs, and there exist local maxima where one model delivers better results than the others. For Model 1 this is the case when a FRR of only 1% is required. Model 3 obtains better results compared to the other models for a FRR of about 12%.

## 5.4 Discussion

In this discussion section several aspects of the rejection strategies and their performance in the experiments are presented. Comparisons of the investigated reject models and sentence generation strategies are made and conclusions are drawn.

### 5.4.1 Reject Models

The simple confidence measure  $\rho_0$  of Model 0 seems not to be a real competitor to the more complex confidence measures  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$ . Especially in the ROC curve Model 0 cannot compete with any of the other reject models. To achieve a FRR of 20% Model 0 requires a FAR of 33.8%, while the other reject models require less than 21% FAR to reach the same FRR. The confidence model  $\rho_0$  depends too strongly on the quality of one single alternative sentence. In general, one single source of sentences seems to be incapable of providing the demanded quality. In contrast, 64 sources are able to compensate the incapacity of single sources and therefore to provide more balanced confidence measures. The additional effort of considering multiple candidate sentences and estimating the posterior probabilities during the training phase leads to the desired effect of substantially superior rejection performance.

Model 2 is an extension of Model 1. Both corresponding confidence measures  $\rho_1$  and  $\rho_2$  deliver approximations of  $p(c|n, w)$ , but  $\rho_2$  of Model 2 respects the currently processed word  $w$  and therefore Model 2 is expected to deliver results that are more precise. This additional exactness can be noticed in a slightly better performance for any sentence generation strategy and most values of FRR. Nevertheless, Model 1 outperforms Model 2 for some FRR values in the ROC plot of the Figures 5.17 and 5.19, and the difference between Model 1 and Model 2 are rather small.

A possible reason for the small differences between Model 1 and Model 2 is

that for only about 50% of the input words the posterior probability  $p(c|w)$  can be estimated using relative frequencies from the training corpus. For the other words, not enough training samples exist in the training corpus and  $\rho_1$  is used as confidence measure instead of  $\rho_2$ . This means that for about half of the input, Model 2 makes exactly the same rejection decisions as Model 1 because the same confidence measure is used.

Another point at issue is the number of training samples that must be present to estimate  $p(c|w)$ . In the experiments reported in this master thesis, this number is set to 20 meaning that at least 20 samples of a word exist in the training corpus to estimate  $p(c|w)$ . But this number might be too small and the resulting estimations too imprecise, with the possible effect that  $\rho_2$  is a less accurate approximation of  $p(c|n, w)$  than  $\rho_1$ .

Similar to Model 2, Model 3 is an extension of Model 1. Not the currently processed word is additionally considered, but the source of the alternative sentence. Model 3 uses the feature vector  $(x_1, \dots, x_n)$  to describe the result of the sentence comparison and to compute the confidence measure  $\rho_3$ , instead of summing up the number of matching words, as it is performed by Model 1.

Despite the added information, Model 3 does not clearly outperform Model 1. Two possible reasons can explain the similar performance of Model 1 and Model 3. First, it could be possible that the sources of alternative candidate sentence provide relatively independent results and that the quality of these sources is quite similar in terms of rejection. Few additional information is acquired by treating the sources separately. A second reason could be that the MLPs, used to calculate the confidence measure  $\rho_3$ , are under-trained, because the training corpus is too small.

## 5.4.2 Alternative Candidate Sentence Sources

In contrast to previously published works in the domain of handwriting recognition, the rejection strategies investigated in this master thesis are based on the fact that a statistical language model supports the recognition process. So far, such rejection strategies have only been addressed in the domain of continuous speech recognition.

For any confidence measure investigated in this thesis the language model variations clearly outperform the  $n$ -best list extraction. According to Zeppenfeld et al. (1997), an advantage of the language model based alternative candidate sentences over the  $n$ -best list approach is that, for very stable input sentences, the sentences based on language model variations could potentially all have the same transcription. This leads to a high confidence measure as it is desired for stable sentences. The confidence measure based on an  $n$ -best list is limited since some words must change in every sentence of the  $n$ -best list.



The fact that every sentence of an  $n$ -best list is different, is considerably obstructive for relative short sentences, which are entirely correctly recognised. For example, if there are only five words in a sentence, the 64 alternative candidate sentences based on  $n$ -best list lead to a relative low confidence for most words of the sentence.

In the research in the domain of speech recognition it is shown that regions of high acoustic stability are relatively error-free (Zeppenfeld et al., 1997; Sanchis et al., 2000). As the language variation strategies outperform the  $n$ -best list extraction, the same observation can be made in the field of offline handwritten text recognition. Words with a high stability concerning language model variations are relatively error-free, while words that are less frequently observed in the alternative sentences based on language model variation are more often recognised incorrectly.

The additional variation of the Word Insertion Penalty (WIP) and not only the variation of the Grammar Scale Factor (GSF), leads to superior rejection results for each reject model. These results support the work of Zimmermann and Bunke (2004) which shows that a good performing integration of the language model is dependent on both factors, the GSF and the WIP. A systematic optimization of the choice of  $(\alpha_i, \beta_i)$  of Equation 4.7, which are used to rescore the lattices, could lead to additional improvements of the performance.

### 5.4.3 General Remarks

In the experiments conducted in this master thesis, the source of alternative candidate sentences has a substantially higher impact on the performance of the rejection system than any of the considered confidence measures. While the confidence measures  $\rho_1, \rho_2, \rho_3$  perform nearly the same for a given sentence extraction strategy, the variation of GSF and WIP delivers substantially better results than GSF variation, which in itself outperforms  $n$ -best list extraction.

The quality of the alternative sentences is the key for these rejection strategies based on alternative candidate sentences. A “good” alternative sentence differs from the hypothesised sentence at the points where the words were recognised incorrectly, while it matches at the points where the words were correctly recognised. Seemingly the investigated strategies of alternative sentence generation deliver sentences of quite different quality.

The results of the different confidence measures  $\rho_1, \rho_2, \rho_3$  are quite similar. This similarity can have multiple reasons. From the theoretical point of view,  $\rho_2$  and  $\rho_3$  could perform better than  $\rho_1$ , because the information that is used to compute the confidence measure is increased from  $\rho_1$  to  $\rho_2$ , and from  $\rho_1$  to  $\rho_3$ . As the amount of the considered information increases, more training effort must be made. Additional probabilities have to be estimated in Model 2, or

weights must be determined in Model 3. But if more quantities have to be set with the same training data, the estimations of the quantities can get less accurate, and under these circumstances under-training can occur.

A second reason for the similarity of the performance of  $\rho_1, \rho_2$ , and  $\rho_3$  is that  $\rho_1$  already performs quite well. To top the results of the confidence measure of Model 1 is challenging. The additional information used in Model 2 and Model 3 appears to be incapable of significantly improving the performance.

# Chapter 6

## Conclusion and Outlook

In this last chapter, main conclusions of this master thesis are drawn and possible future work is discussed.

### 6.1 Conclusion

The main goal of this master thesis was the evaluation of multiple rejection strategies with confidence measures derived from alternative candidate sentences in the domain of recognition of general handwritten text.

A post-processing rejection procedure has been proposed. This procedure generates alternative candidate sentences and computes a confidence measure for every word based on features extracted from the candidate sentences. The confidence measure is used to decide whether to accept or to reject the word.

Four different confidence measures based on alternative candidate sentences have been presented as reject models. The simplest model derives its confidence measure from a single alternative sentence. Confidence measures derived on the number of times a hypothesised word appears in the multiple candidate sentences are used in two models. The most sophisticated model uses a Multi-Layer Perceptron to determine its confidence measure.

Additionally three different sources of alternative candidate sentences were investigated. The first strategy to obtain candidate sentences were  $n$ -best lists, containing the  $n$ -most probable sentences for a given image of handwritten text. The second and the third strategies were based on language model variations. The second strategy varied the Grammar Scale Factor (GSF) to get multiple candidate sentences, while the third strategy varied the GSF as well as the Word Insertion Penalty (WIP).

The experiments and their results showed the different impact of confidence measures and the sentence generation strategies on the rejection task. The

performance of the different confidence measures for a given sentence generation strategy were quite similar, except for the model with a single alternative, which performed clearly worse. On the other hand, for the sentence extraction strategies, the language model variations, especially varying the GSF and WIP, substantially outperformed the  $n$ -best list sources.

The best performing rejection strategy was capable to achieve a false acceptance rate of 20% at a false rejection rate of only 20.6%. In terms of error-reject, the presented system allowed to obtain an error rate of 10% if 16.3% of the input words are rejected.

## 6.2 Outlook

In this section some further issues and experiments are presented which may be considered in future studies.

The experiments with the reject models presented in this master thesis were performed with a training set of 200 sentences. The estimations of the probabilities were sometimes done with a minimal amount of training samples. Experiments with a much larger set should be considered in order to obtain more stable estimations of the posterior probabilities. Additionally, the architecture of the MLPs of Model 3 could be adapted, and more internal neurons could be inserted, as there is more training data available to train the weights.

The number  $K$  of alternative candidate sentences as well as the values of GSF  $\alpha_i$  and WIP  $\beta_i$ , which were used for rescoring the lattice, were not systematically optimized. Since the quality of the alternative sentences is a key aspect of the proposed rejection strategies, a systematic optimization of the language model variations could improve the quality of the alternative candidate sentences, thereby improving the performance of the rejection procedure.

The combination with features derived from other sources is an additional promising issue. The features could be derived from the recognition score or from characteristics of  $n$ -best lists and build additional confidence measures. The additional confidence measures could be combined with the confidence measures derived from alternative candidate sentences. This combination could be done by mean value, or, if some weighting is required, by a neural network.

The comparison of the candidate sentences with the hypothesised sentences as described in Section 2.5 considers hits as a match while deletions and substitutions are taken as a mismatch. Insertions are ignored entirely. A more sophisticated comparison strategy would probably improve the performance of the rejection procedure. Including the insertions into the comparison would enable the system to detect missing words in the hypothesised sentences.

The presented rejection system is contrary to the more traditional approaches of rejection which are usually used in offline handwriting rejection. The approach introduced in this thesis makes use of the statistical language model and is based on multiple alternative candidate sentences. An experimental comparison with the more traditional rejection approaches would be interesting to estimate the improvements and illustrate the advantages and disadvantage of the presented strategy.



# Bibliography

- A. Brakensiek, J. Rottland, and G. Rigoll. Confidence measures for an address reading system. In *7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland*, volume 1, pages 294–298, 2003.
- Stephen A. Cook. The complexity of theorem-proving procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing, Shakers Heights, Ohio, May 3, 4, 5, 1971*, pages 151–158, 1971.
- J. Daugman. Biometric decision landscapes. In *Technical Report No. TR482*. University of Cambridge Computer Laboratory, 2000.
- K. Fukunaga. Statistical pattern recognition. In *Handbook of Pattern Recognition and Computer Vision*, chapter 1.2, pages 33–60. Eds. C. H. Chen, L. F. Pau, and P. S. P. Wang, World Scientific Publishing Co. Pte. Ltd, Singapore, 1993.
- Nikolai Gorski. Optimizing error-reject trade off in recognition systems. In *4th International Conference Document Analysis and Recognition (ICDAR '97) Volume I and Volume II*, page 1092ff, 1997.
- S. Johansson, E. Atwell, R. Garside, and G. Leech. *The Tagged LOB Corpus, Users's Manual*. Norwegian Computing Center for the Humanities, Bergen, Norway, 1986.
- Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145, 1995.
- D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer Professional Computing, New York, 2003.
- U.-V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *5th Int. Conf. on Document Analysis and Recognition 99, Bangalore, India*, pages 705–708, 1999.
- U.-V. Marti and H. 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.

- S. Marukatat, T. Artieres, and P. Gallinari. Rejection measures for handwriting sentence recognition. In *8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 24–29, Niagra-on-the-Lake, Canada, 2002.
- A. Matti, J. Laaksonen, E. Oja, and J. Kangas. Rejection methods for an adaptive committee classifier. In *Sixth International Conference on Document Analysis and Recognition (ICDAR)*, Seattle, Washington, 2001.
- Yoh-Han Pao. *Handbook of Pattern Recognition and Computer Vision*, chapter Neural Net Computing for Pattern Recognition, pages 125–162. Eds. C. H. Chen, L. F. Pau and P. S. P. Wang, World Scientific Publishing Co. Pte. Ltd, Singapore, 1993.
- F. Perraud, C. Viard-Gaudin, E. Morin, and P.-M. Lallican. N-gram and n-class models for on line handwriting recognition. In *7th Int. Conf. on Document Analysis and Recognition, Edinburgh, Scotland*, volume 2, pages 1053–1057, 2003.
- J.F. Pitrelli and M. P. Perrone. Confidence modeling for verification post-processing for handwriting recognition. In *8th Int. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Niagra-on-the-Lake, Canada, 2002.
- J.F. Pitrelli and M. P. Perrone. Confidence-scoring post-processing for off-line handwritten-character recognition verification. In *Seventh International Conference on Document Analysis and Recognition (ICDAR) Volume I, Edinburgh, Scotland*, page 278ff, 2003.
- L. Rabiner. A tutorial on hidden Markov models and selected application in speech recognition. In *Proc. of the IEEE*, Vol. 77 No. 2, 1989.
- Raul Rojas. *Neural Networks - A Systematik Introduction*. Springer-Verlag, Berlin, New-York, 1996.
- R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proc. of the IEEE*, 88:1270–1278, 2000.
- R. San-Segundo, B. Pellom, W. Ward, and J.M. Pardo. Confidence measures for dialogue management in the CU communicator system, June 2000.
- Alberto Sanchis, Víctor Jimenez, and Enrique Vidal. Efficient use of the grammar scale factor to classify incorrect words in speech recognition verification. In *International Conference on Pattern Recognition ICPR-2000*, volume 3, pages 278–281, Barcelona, Spain, September 2000.
- J. Schürmann. *Pattern Classification —A Unified View of Statistical and Neural Approaches*. Wiley-Intercience, John Wiley & Sons Inc, New York, 1996.



- K. Takeda, A. Ogawa, and F. Itakura. Estimation entropy of a language from optimal word insertion penalty. In *The 5th International Conference on Spoken Language Processing*, Sydney Australia, 1998.
- Martin Tompa. Lecture notes on biological sequence analysis. Technical report, Department of Computer Science and Engineering, University of Washington, Seattle, Washington, U.S.A, 2000.
- A. Vinciarelli, S. Bengio, and H. Bunke. Offline recognition of unconstrained handwritten texts using HMM and statistical language models. IDIAP-RR 03-22, Dalle Molle Institute for Perceptual Artificial Intelligence, 2003.
- A.J. Viterbi. Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.
- R. Wagner and M. Fischer. The string-to-string correction problem. *Journal of the ACM*, 21(1):168–173, 1974.
- L. Wang and T. Jiang. On the complexity of multiple sequence alignment. In *Journal of Computational Biology* 1(4), pages 337–348, 1994.
- S. Young, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.2)*. Cambridge University Engineering Department, 2002.
- S. J. Young, N. H. Russell, and J. H. S. Thornton. Token passing: A simple conceptual model for connected speech recognition systems. CUED technical report F INFENG/TR38, Cambridge University, 1989.
- T. Zeppenfeld, M. Finke, K. Ries, M. Westphal, and A. Waibel. Recognition of conversational telephone speech using the janus speech engine. In *Proc. ICASSP '97*, pages 1815–1818, Munich, Germany, 1997.
- M. Zimmermann. *Offline Handwriting Recognition and Grammar based Syntax Analysis*. PhD thesis, University of Bern, Switzerland, 2003.
- M. Zimmermann and H. Bunke. Automatic segmentation of the IAM off-line handwritten English text database. In *16th Int. Conf. on Pattern Recognition*, volume 4, pages 35–39, Quebec, Canada, 2002.
- M. Zimmermann and H. Bunke. Optimizing the integration of a statistical language model in HMM based offline handwriting text recognition. In *submitted*, 2004.
- M. Zimmermann, J-C. Chappelier, and H. Bunke. Offline grammar-based recognition of handwritten sentences. 2003.