Optimizing the Integration of a Statistical Language Model in HMM based Offline Handwritten Text Recognition

ICPR 2004 Cambridge, UK

Matthias Zimmermann and Horst Bunke

Computer Science Department

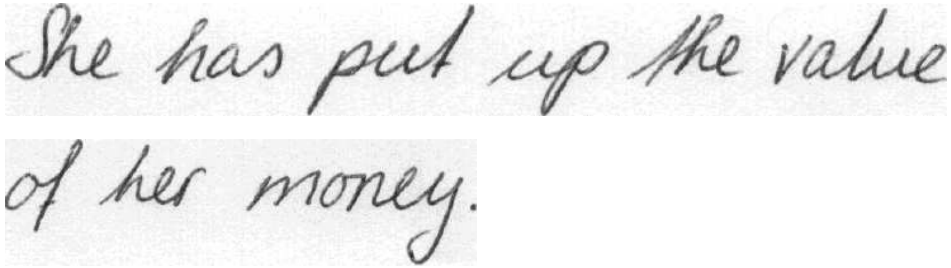
University of Bern, Switzerland

Organization

1. Introduction
2. Bigram Language Model
3. Language Model Integration
4. Grammar Scale Factor
5. Word Insertion Penalty
6. Setup, Optimization and Evaluation
7. Conclusions

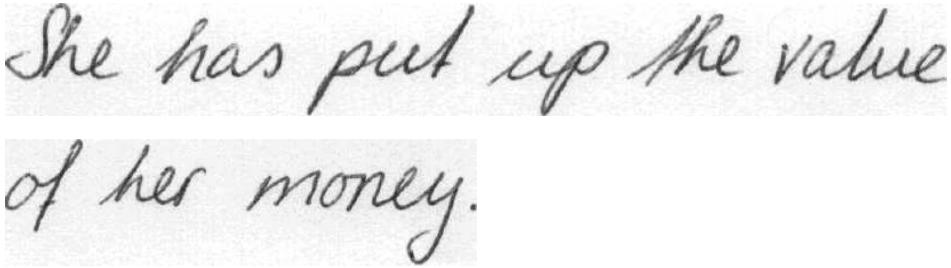
1. Introduction

Recognition of Handwritten Sentences



1. Introduction

Recognition of Handwritten Sentences

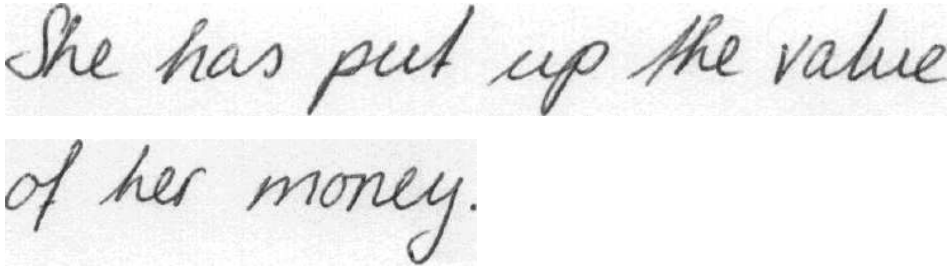


↓

Recognizer

1. Introduction

Recognition of Handwritten Sentences



↓

Recognizer

↓

She has put up the value of her money .

1. Introduction

Handwriting Recognition and Language Modeling

- recognition of handwritten text is investigated since many years

1. Introduction

Handwriting Recognition and Language Modeling

- recognition of handwritten text is investigated since many years
- the use of statistical LM is relatively new

1. Introduction

Handwriting Recognition and Language Modeling

- recognition of handwritten text is investigated since many years
- the use of statistical LM is relatively new
- the use of LM consistently improves baseline recognizer

1. Introduction

Handwriting Recognition and Language Modeling

- recognition of handwritten text is investigated since many years
- the use of statistical LM is relatively new
- the use of LM consistently improves baseline recognizer
- LM are predominantly used within the HMM framework

1. Introduction

Handwriting Recognition and Language Modeling

- recognition of handwritten text is investigated since many years
- the use of statistical LM is relatively new
- the use of LM consistently improves baseline recognizer
- LM are predominantly used within the HMM framework
- so far, the optimization of the integration is hardly addressed

1. Introduction

General Principle

$$\hat{W} = \underset{W}{argmax} \; p(X|W)p(W)$$

1. Introduction

General Principle

$$\hat{W} = \underset{W}{argmax}\ p(X|W)p(W)$$

$p(X|W)$ provided by HMM (optical model)

1. Introduction

General Principle

$$\hat{W} = \underset{W}{argmax}\ p(X|W)p(W)$$

$p(X\|W)$	provided by HMM (optical model)
$p(W)$	provided by the Language Model (LM)

1. Introduction

General Principle

$$\hat{W} = \underset{W}{argmax}\ p(X|W)p(W)$$

$p(X\|W)$	provided by HMM (optical model)
$p(W)$	provided by the Language Model (LM)

$p(W) =$ $p(She\ has\ put\ up\ the\ value\ of\ her\ money\ .)$

2. Bigram Language Model

Statistical LM provide probabilities for given sentences

$$p(W) = p(She\ has\ put\ up\ the\ value\ of\ her\ money\ .)$$

2. Bigram Language Model

Statistical LM provide probabilities for given sentences

$$p(W) = p(She\ has\ put\ up\ the\ value\ of\ her\ money\ .)$$

Approximation by Bigram LM

$$p(W) = p(She|\diamond)\ p(has|She)\ p(put|has)\ \ ...\ \ p(.|money)$$

2. Bigram Language Model

Statistical LM provide probabilities for given sentences

$$p(W) = p(She\ has\ put\ up\ the\ value\ of\ her\ money\ .)$$

Approximation by Bigram LM

$$p(W) = p(She|\diamond)\ p(has|She)\ p(put|has)\ \ ...\ \ p(.|money)$$

Problems of such LM

- LM only remembers the previous word

2. Bigram Language Model

Statistical LM provide probabilities for given sentences

$$p(W) = p(She\ has\ put\ up\ the\ value\ of\ her\ money\ .)$$

Approximation by Bigram LM

$$p(W) = p(She|\diamond)\ p(has|She)\ p(put|has)\ \ ...\ \ p(.|money)$$

Problems of such LM

- LM only remembers the previous word
- Smoothing needed for unseen word transitions

2. Bigram Language Model

Statistical LM provide probabilities for given sentences

$$p(W) = p(She\ has\ put\ up\ the\ value\ of\ her\ money\ .)$$

Approximation by Bigram LM

$$p(W) = p(She|\diamond)\ p(has|She)\ p(put|has)\ \ ...\ \ p(.|money)$$

Problems of such LM

- LM only remembers the previous word
- Smoothing needed for unseen word transitions
- Each additional word decreases $p(W)$

3. Language Model Integration

Direct integration of the LM in HMM framework

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \log p(w_i|w_{i-1})$$

X_i current part of feature vector sequence

3. Language Model Integration

Direct integration of the LM in HMM framework

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \log p(w_i|w_{i-1})$$

X_i	current part of feature vector sequence
w_i	associated word candidates

3. Language Model Integration

Direct integration of the LM in HMM framework

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \log p(w_i|w_{i-1})$$

X_i	current part of feature vector sequence
w_i	associated word candidates
$p(X_i\|w_i)$	HMM provides likelihoods, not probabilities

3. Language Model Integration

Direct integration of the LM in HMM framework

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \log p(w_i|w_{i-1})$$

X_i	current part of feature vector sequence
w_i	associated word candidates
$p(X_i\|w_i)$	HMM provides likelihoods, not probabilities
$p(w_i\|w_{i-1})$	LM provides probabilities
	but Bigram LM prefers short sentences

4. Grammar Scale Factor

Use of the Grammar Scale Factor (GSF)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1})$$

4. Grammar Scale Factor

Use of the Grammar Scale Factor (GSF)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1})$$

α GSF, also called linguistic weight, LM weight, ...
used to weight the influence of the LM on
the recognition result

4. Grammar Scale Factor

Use of the Grammar Scale Factor (GSF)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1})$$

α	GSF, also called linguistic weight, LM weight, ... used to weight the influence of the LM on the recognition result
$\alpha = 0$	LM switched off

4. Grammar Scale Factor

Use of the Grammar Scale Factor (GSF)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1})$$

α	GSF, also called linguistic weight, LM weight, ... used to weight the influence of the LM on the recognition result
$\alpha = 0$	LM switched off
$\alpha = 1$	classical integration of the LM

4. Grammar Scale Factor

Use of the Grammar Scale Factor (GSF)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1})$$

α	GSF, also called linguistic weight, LM weight, ... used to weight the influence of the LM on the recognition result
$\alpha = 0$	LM switched off
$\alpha = 1$	classical integration of the LM
$\alpha > 1$	increasing influence of the LM

5. Word Insertion Penalty

Use of the Word Insertion Penalty (WIP)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1}) + \beta$$

5. Word Insertion Penalty

Use of the Word Insertion Penalty (WIP)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1}) + \beta$$

β WIP (negative, see below)
helps to control insertion and deletion rate
of words in recognition result

5. Word Insertion Penalty

Use of the Word Insertion Penalty (WIP)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1}) + \beta$$

β	WIP (negative, see below) helps to control insertion and deletion rate of words in recognition result
$\beta = 0$	classical integration

5. Word Insertion Penalty

Use of the Word Insertion Penalty (WIP)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1}) + \beta$$

β	WIP (negative, see below) helps to control insertion and deletion rate of words in recognition result
$\beta = 0$	classical integration
$\beta > 0$	increases the # of words in recognition result

5. Word Insertion Penalty

Use of the Word Insertion Penalty (WIP)

$$\phi_i = \phi_{i-1} + \log p(X_i|w_i) + \alpha \ \log p(w_i|w_{i-1}) + \beta$$

β	WIP (negative, see below) helps to control insertion and deletion rate of words in recognition result
$\beta = 0$	classical integration
$\beta > 0$	increases the # of words in recognition result
$\beta < 0$	decreases the # of words (penalty per word in recognition result)

6. Experimental Setup

The Recognizer

6. Experimental Setup

The Recognizer

- text line normalization (slant, writing regions, contrast)

6. Experimental Setup

The Recognizer

- text line normalization (slant, writing regions, contrast)
- feature extraction (sliding window)

6. Experimental Setup

The Recognizer

- text line normalization (slant, writing regions, contrast)
- feature extraction (sliding window)
- HMM for each character
  - linear topology
  - variable character length
  - multi Gaussian

6. Experimental Setup

The Recognizer

- text line normalization (slant, writing regions, contrast)
- feature extraction (sliding window)
- HMM for each character
  - linear topology
  - variable character length
  - multi Gaussian
- Baum-Welch training

6. Experimental Setup

The Recognizer

- text line normalization (slant, writing regions, contrast)
- feature extraction (sliding window)
- HMM for each character
  - linear topology
  - variable character length
  - multi Gaussian
- Baum-Welch training
- Viterbi decoding

6. Experimental Setup

Assumptions

- recognition of isolated sentences
- closed lexicon: no new words in test set

6. Experimental Setup

Assumptions

- recognition of isolated sentences
- closed lexicon: no new words in test set

Training of Character HMM

- IAM database
- 5,799 lines of handwritten text (33,993 words)
- written by 448 persons

6. Experimental Setup

Assumptions

- recognition of isolated sentences
- closed lexicon: no new words in test set

Training of Character HMM

- IAM database
- 5,799 lines of handwritten text (33,993 words)
- written by 448 persons

Training of Bigram LM

- LOB corpus (tagged)
- 50’000 sentences (excluding test/validation set)

6. Experimental Setup

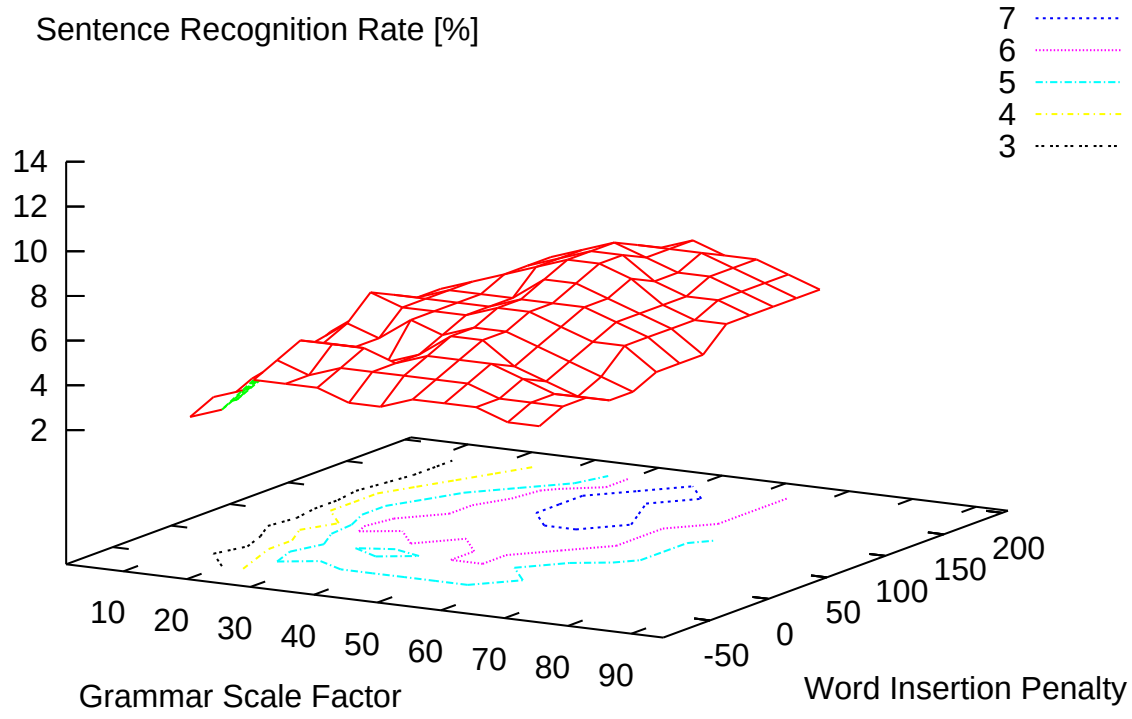
Recognition Task* Specifications

	validation	test
sentences	200	200
text lines	582	575
words	4,094	3,956
lexicon	8,827	8,821
writers	100	100

* writer independent: different writers in training, validation and test set

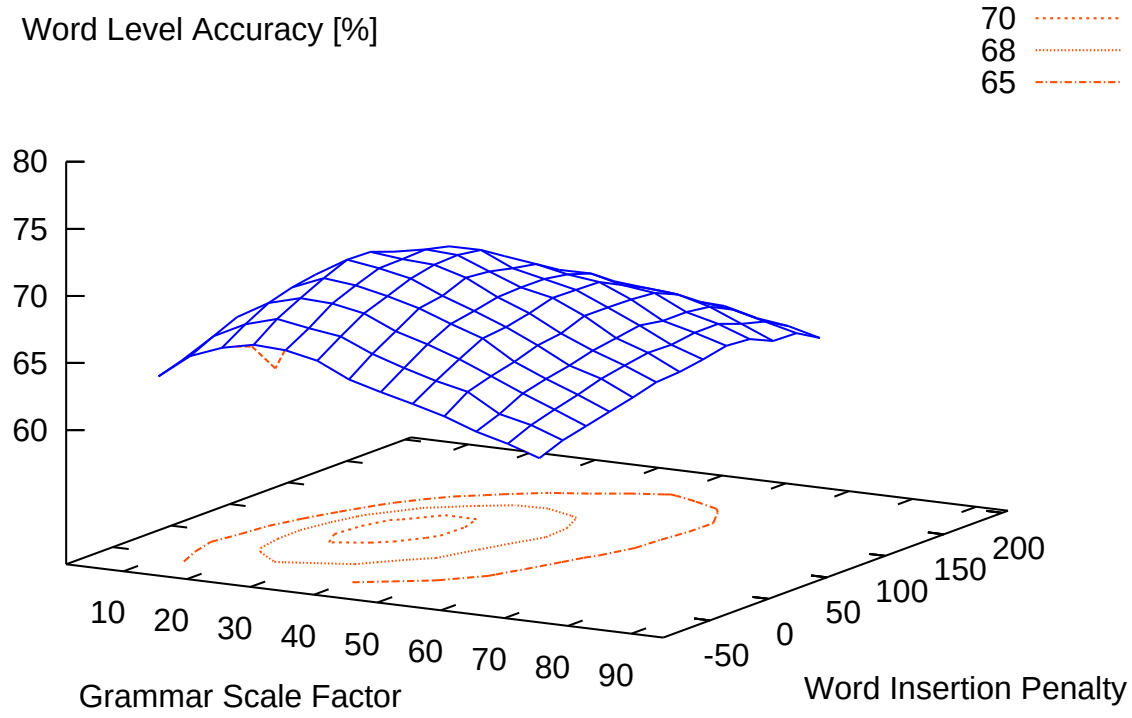
6. System Evaluation

Optimizing the Sentence Recognition Rate



6. System Evaluation

Optimizing the Word Level Accuracy (H-I)/N



6. System Evaluation

Test Set Results

α	β	Sentence Rec.	Word Level Acc.	
0	0	0.5%	49.1%	no LM

6. System Evaluation

Test Set Results

α	β	Sentence Rec.	Word Level Acc.	
0	0	0.5%	49.1%	no LM
1	0	0.5%	52.2%	baseline integration

6. System Evaluation

Test Set Results

α	β	Sentence Rec.	Word Level Acc.	
0	0	0.5%	49.1%	no LM
1	0	0.5%	52.2%	baseline integration
30	50	10.5%	76.3%	optimized integration

7. Conclusions

Main Conclusions

- the benefit of the GSF and the WIP in the domain of offline handwritten text recognition has been shown in extensive experiments on realistic data

7. Conclusions

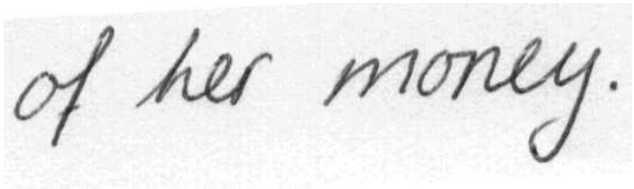
Main Conclusions

- the benefit of the GSF and the WIP in the domain of offline handwritten text recognition has been shown in extensive experiments on realistic data
- we strongly recommend the use of the GSF and the WIP to optimize the integration of statistical LM

Thank You

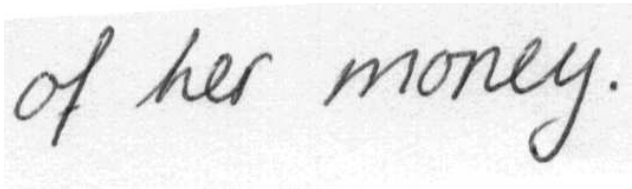
Appendix The Recognizer

Text Line Normalization

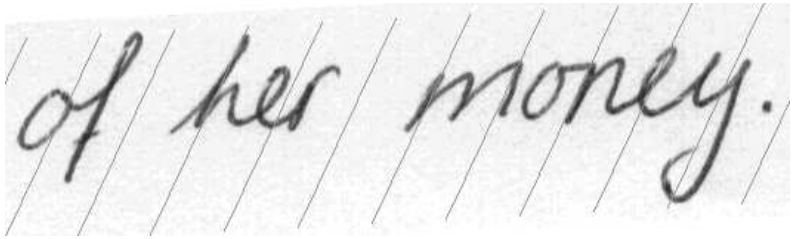


Appendix The Recognizer

Text Line Normalization

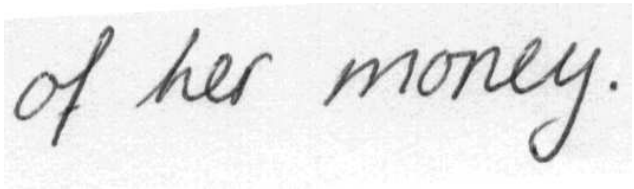


slant estimation

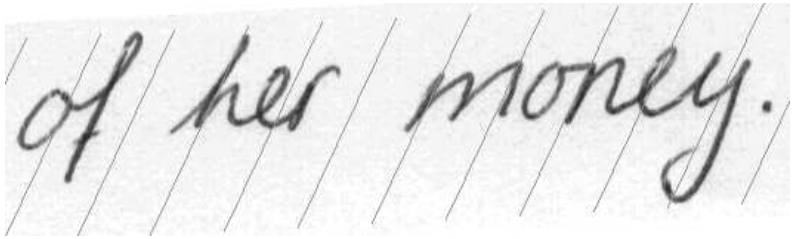


Appendix The Recognizer

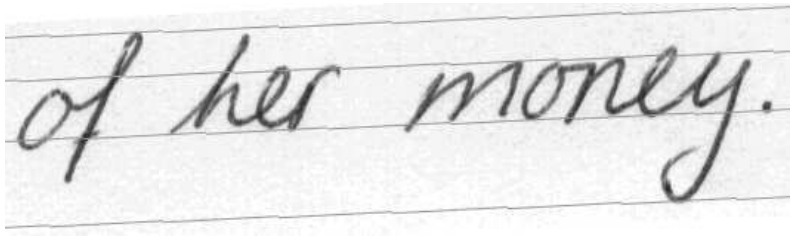
Text Line Normalization



slant estimation

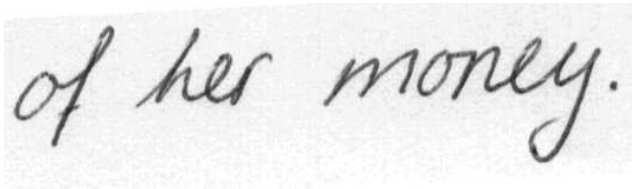


reference line estimation

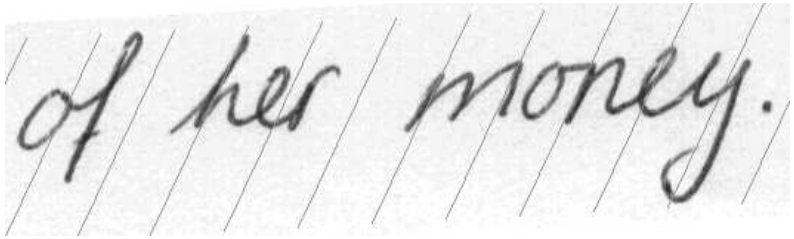


Appendix The Recognizer

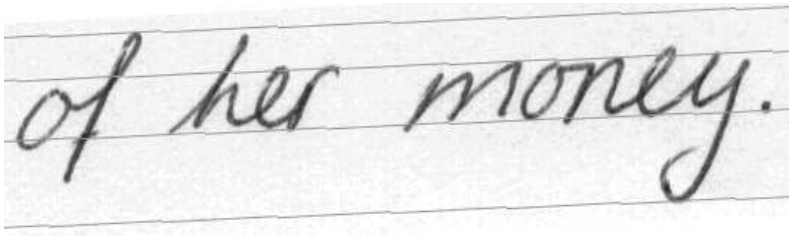
Text Line Normalization



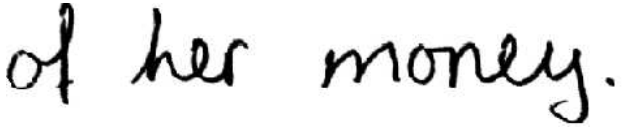
slant estimation



reference line estimation

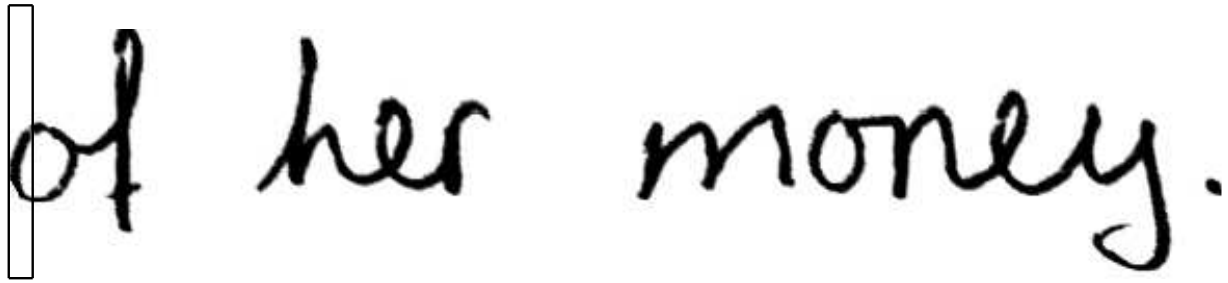


normalized text line



Appendix The Recognizer

Extraction of the Feature Vector Sequence

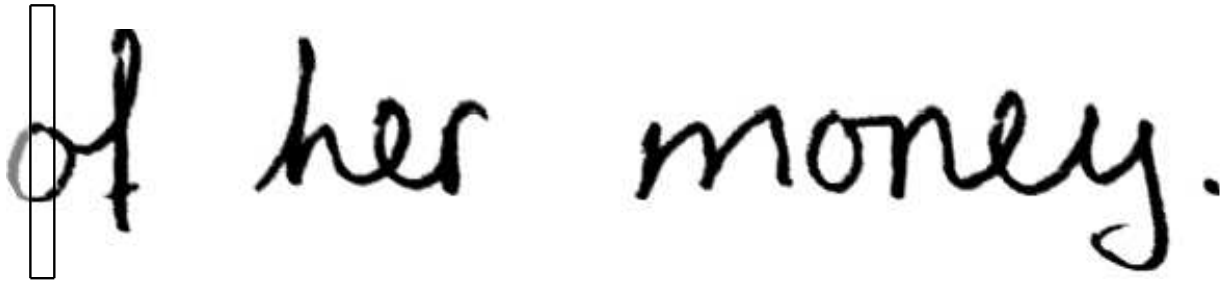


$$\begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1m} \end{pmatrix}$$

$$X = (X_1,$$

Appendix The Recognizer

Extraction of the Feature Vector Sequence



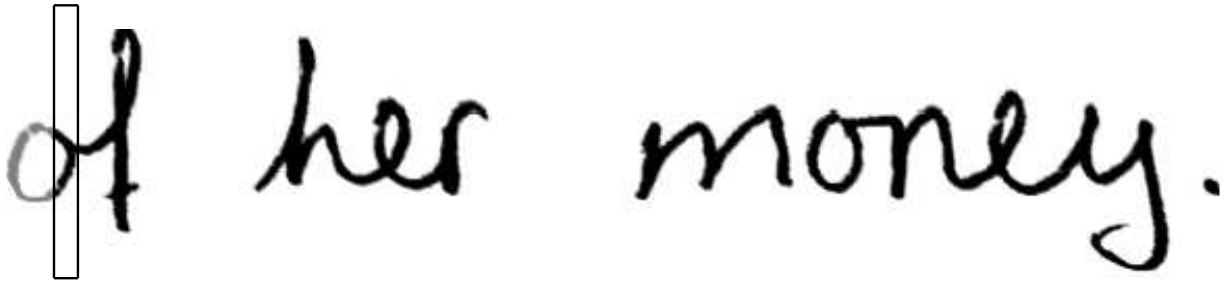
$$\downarrow$$

$$\begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2m} \end{pmatrix}$$

$$X = (X_1, X_2,$$

Appendix The Recognizer

Extraction of the Feature Vector Sequence

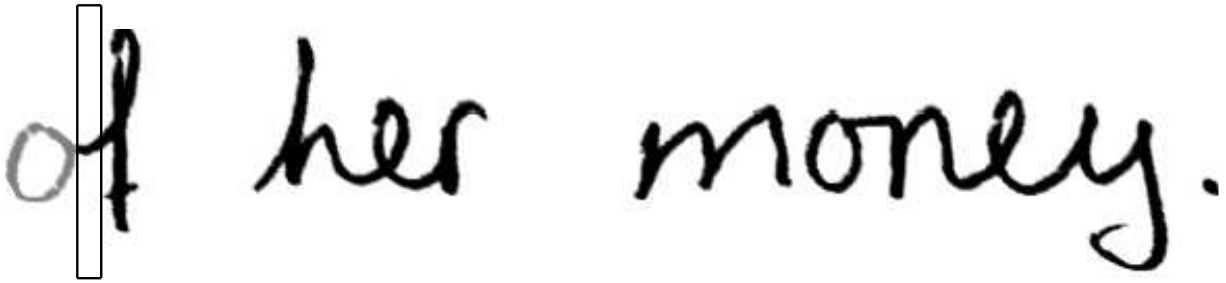


$$\begin{pmatrix} x_{31} \\ x_{32} \\ \vdots \\ x_{3m} \end{pmatrix}$$

$$X = (X_1, X_2, X_3,$$

Appendix The Recognizer

Extraction of the Feature Vector Sequence

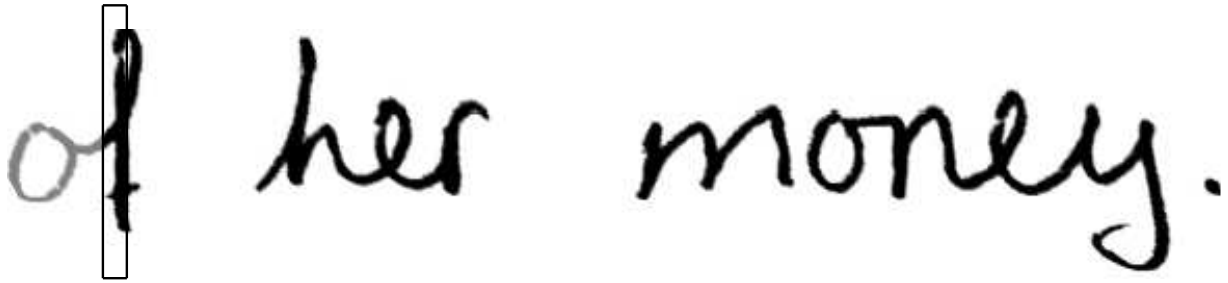


$$\begin{pmatrix} x_{41} \\ x_{42} \\ \vdots \\ x_{4m} \end{pmatrix}$$

$$X = (X_1, X_2, X_3, X_4,$$

Appendix The Recognizer

Extraction of the Feature Vector Sequence

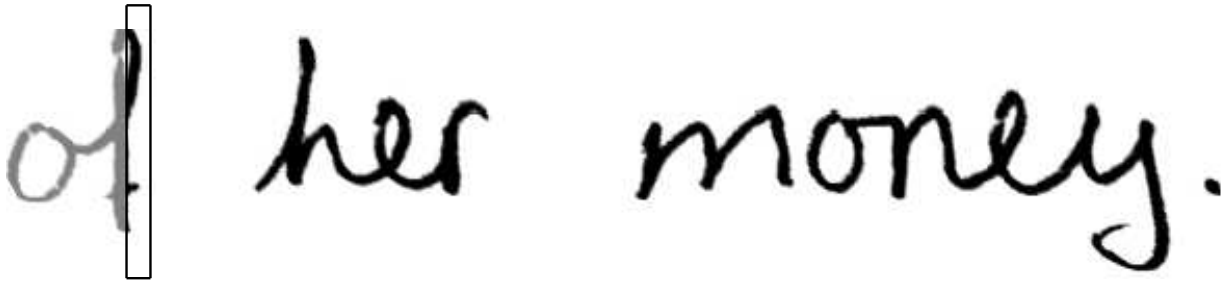


$$\begin{pmatrix} x_{51} \\ x_{52} \\ \vdots \\ x_{5m} \end{pmatrix}$$

$$X = (X_1, X_2, X_3, X_4, X_5,$$

Appendix The Recognizer

Extraction of the Feature Vector Sequence

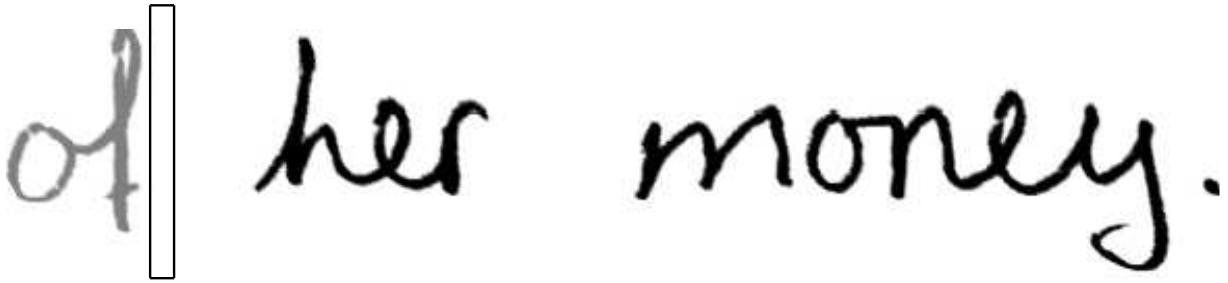


$$\begin{pmatrix} x_{61} \\ x_{62} \\ \vdots \\ x_{6m} \end{pmatrix}$$

$$X = (X_1, X_2, X_3, X_4, X_5, X_6,$$

Appendix The Recognizer

Extraction of the Feature Vector Sequence



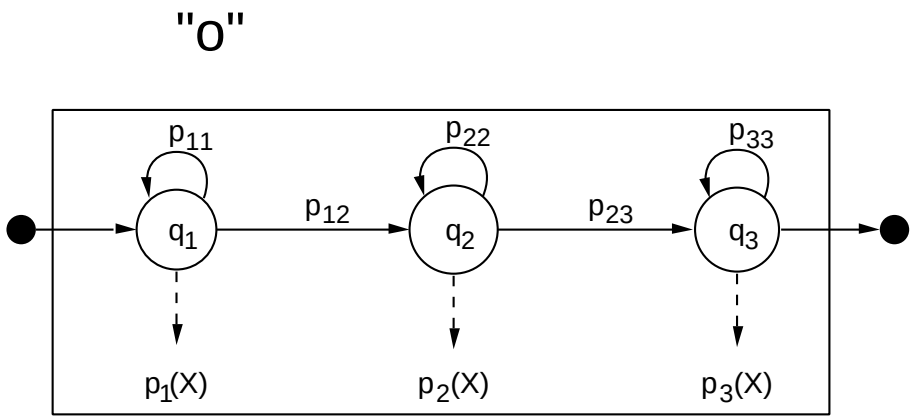
$$\begin{pmatrix} x_{71} \\ x_{72} \\ \vdots \\ x_{7m} \end{pmatrix}$$

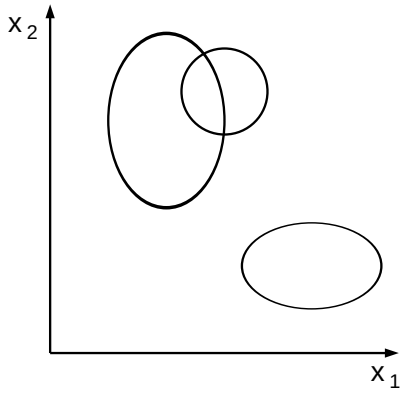
$$X = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, \ldots$$

Appendix The Recognizer

Viterbi Decoding

$(X_1, X_2, X_3, \ldots$

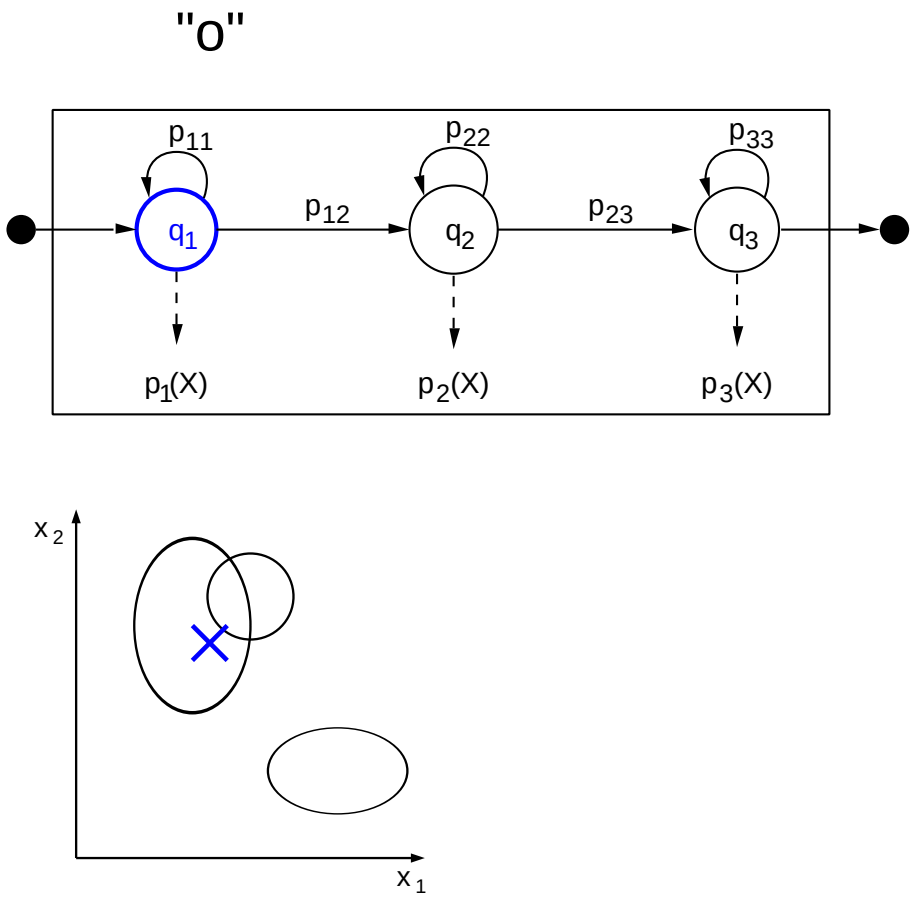




Appendix The Recognizer

Viterbi Decoding

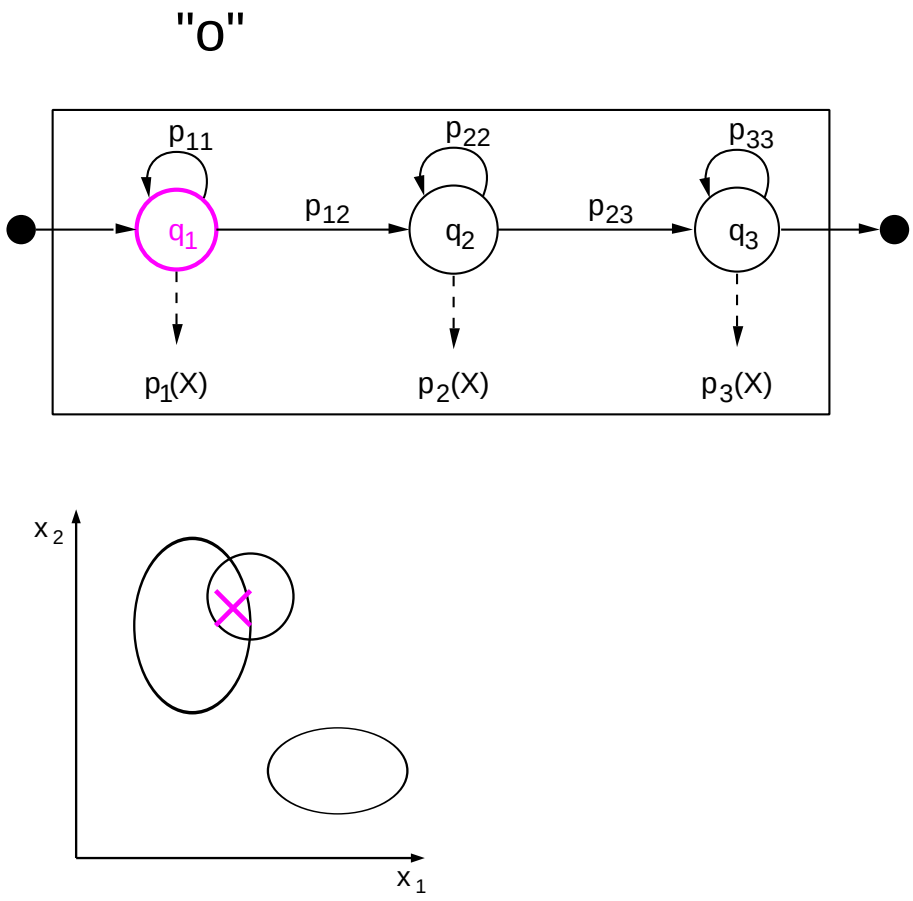
$(X_1, X_2, X_3, \ldots$



Appendix The Recognizer

Viterbi Decoding

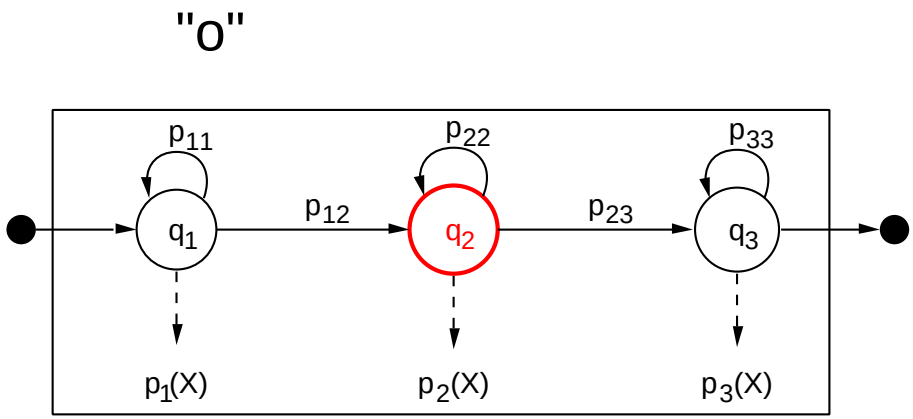
$$(X_1, X_2, X_3, \ldots$$

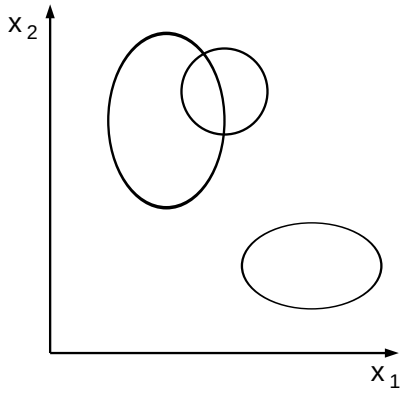


Appendix The Recognizer

Viterbi Decoding

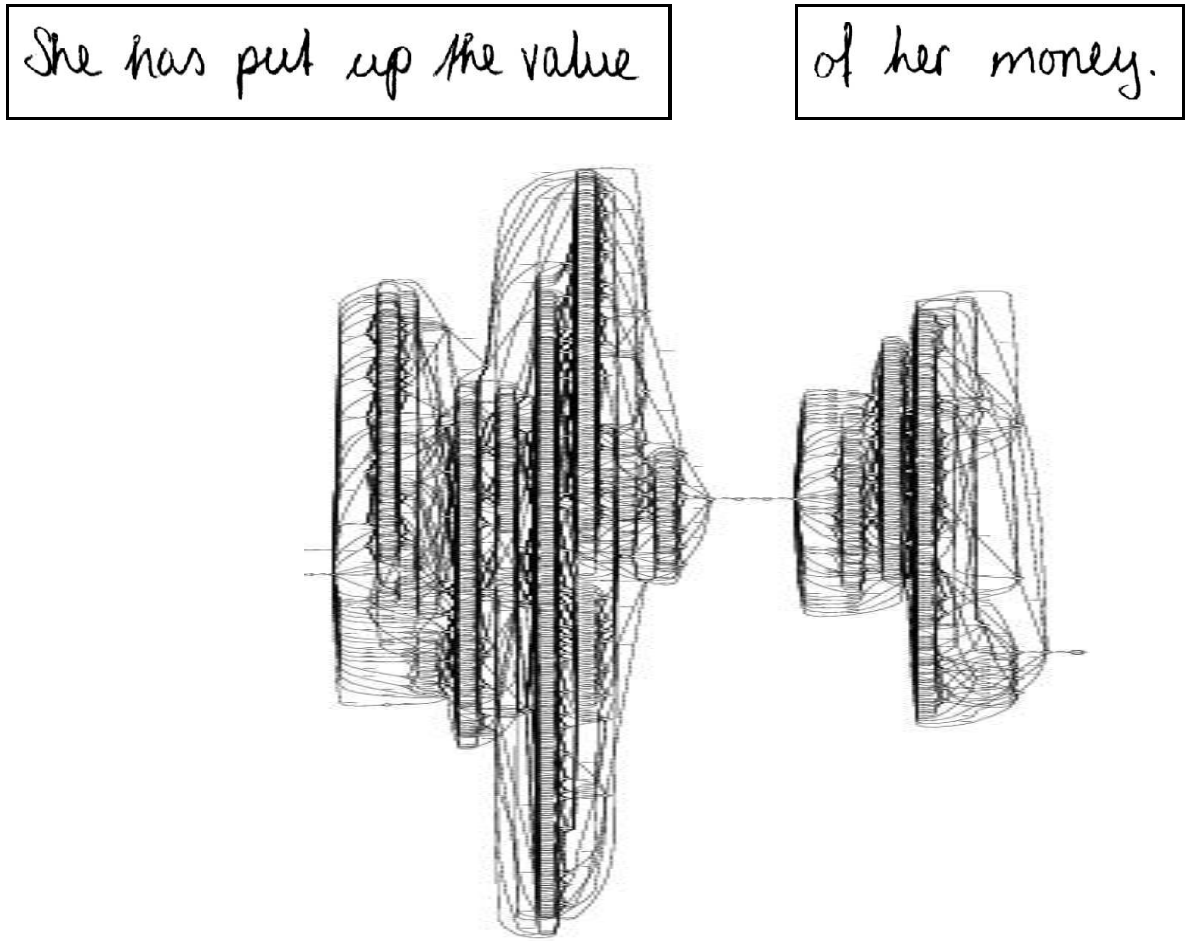
$$(X_1, X_2, X_3, \ldots$$





Appendix The Recognizer

Recognition Lattice Generation



Appendix The Recognizer

Sublattice of 2nd Line