Offline Handwriting Recognition and Grammar based Syntax Analysis

PhD Defense, 13.11.2003

Matthias Zimmermann

Department of Computer Science

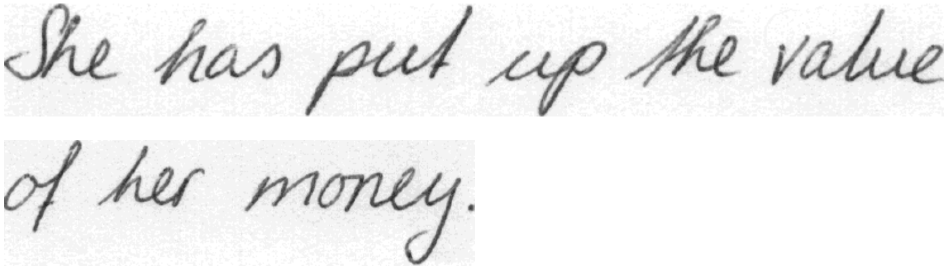
University of Bern, Switzerland

Organization

1. Introduction
2. Problem Statement
3. The Grammar
4. The Parser
5. IAM Database and Recognizer
6. The Combination
7. Setup, Optimization and Evaluation
8. Conclusions and Outlook

Introduction

Perfect Transcription of general Handwritten Text



↓

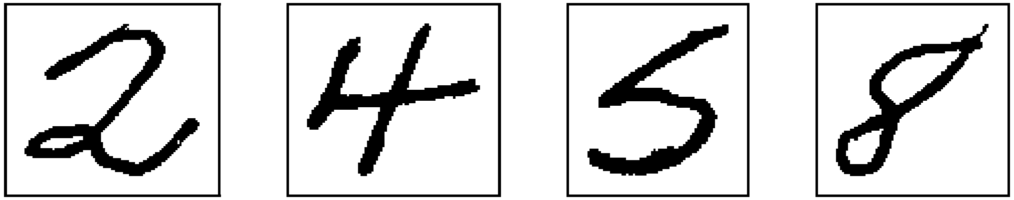
Recognizer

↓

She|has|put|up|the|value|of|her|money|.

Introduction

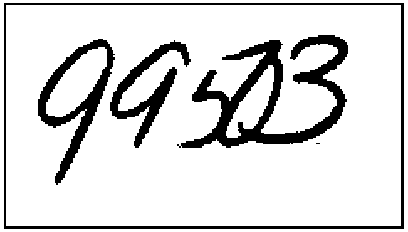
Recognition of Isolated digits

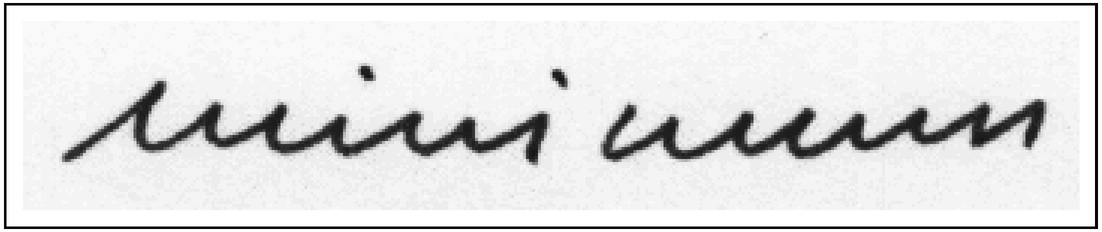


- human like performance, or better
- recognition rates: around 99.5%

Introduction

Recognition of Numbers and Words

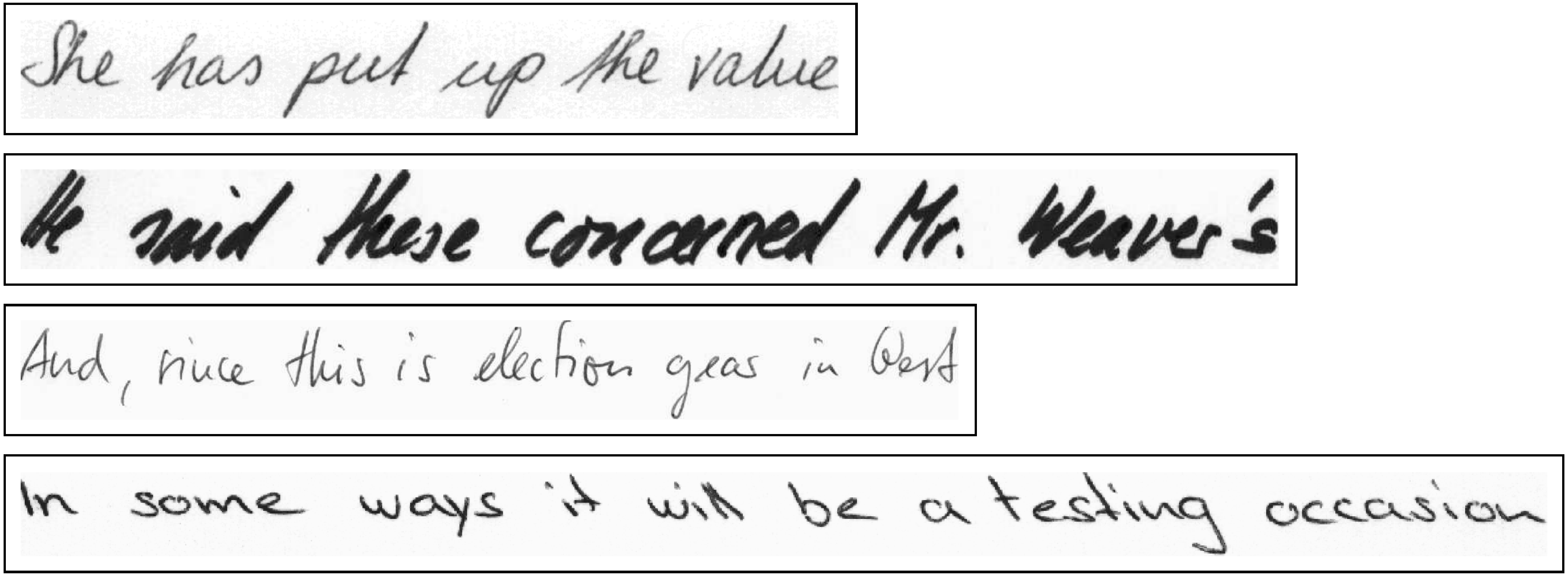




- worse performance than humans
- recognition rates: around 80-90%

Introduction

Recognition of general Text



- significantly worse performance than humans
- recognition rates: around 70%

Introduction

Why is it so difficult?

Introduction

Why is it so difficult?

- large variation of writing styles

Introduction

Why is it so difficult?

- large variation of writing styles
- variation of writing instruments

Introduction

Why is it so difficult?

- large variation of writing styles
- variation of writing instruments
- segmentation problem

Introduction

Why is it so difficult?

- large variation of writing styles
- variation of writing instruments
- segmentation problem
- lack of task-specific knowledge (in contrast to check amounts and postal addresses)

Introduction

Why is it so difficult?

- large variation of writing styles
- variation of writing instruments
- segmentation problem
- lack of task-specific knowledge (in contrast to check amounts and postal addresses)

Advantage of humans over computers:

- performance heavily relies on text understanding

Introduction

Why is it so difficult?

- large variation of writing styles
- variation of writing instruments
- segmentation problem
- lack of task-specific knowledge (in contrast to check amounts and postal addresses)

Advantage of humans over computers:

- performance heavily relies on text understanding, combining
  - optical recognition of characters
  - language syntax
  - semantics and "world knowledge"

Introduction

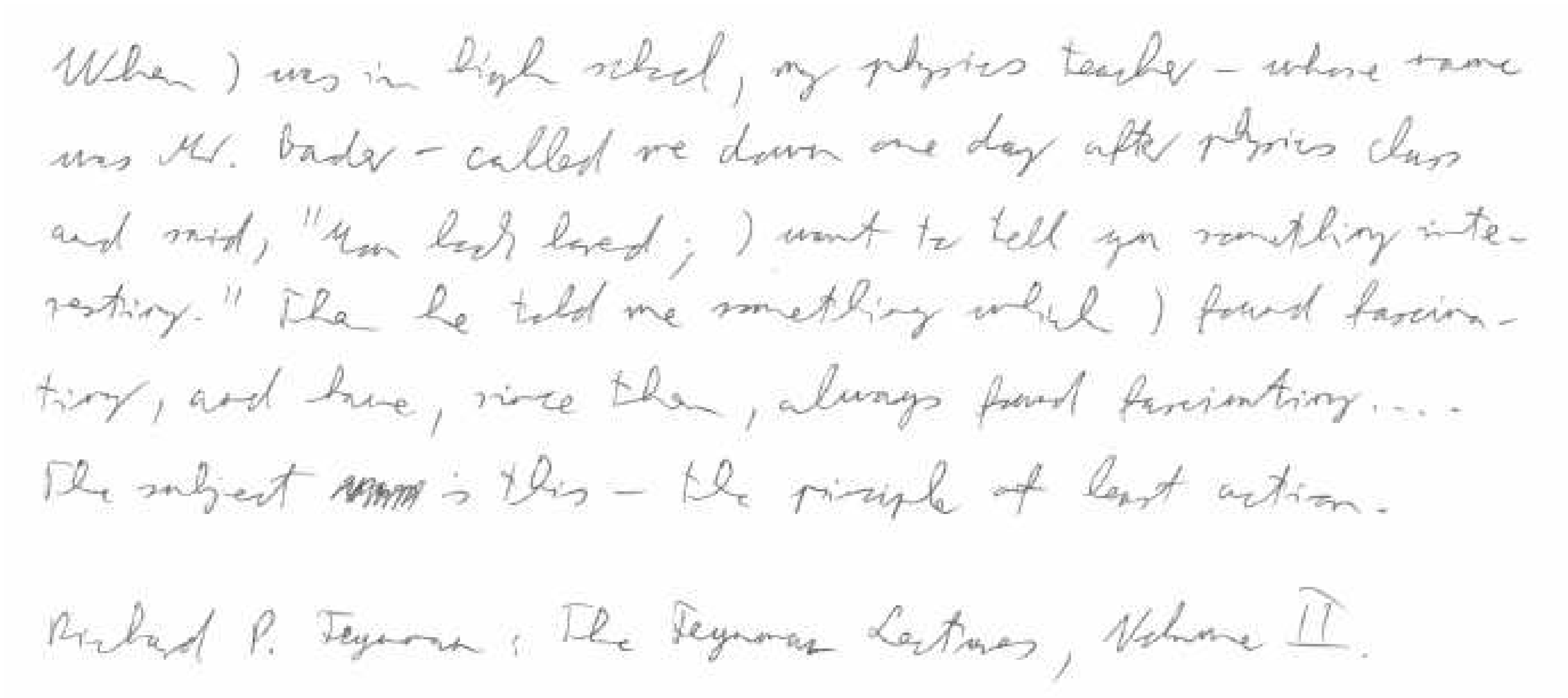
Handwritten Text

When I was in high school, my physics teacher — whose name was Mr. Bader — called me down one day after physics class and said, "You look bored; I want to tell you something interesting." Then he told me something which I found fascinating, and have, since then, always found fascinating.... The subject is this — the principle of least action.

Richard P. Feynman: The Feynman Lectures, Volume II.

Introduction

Handwritten Text (English)

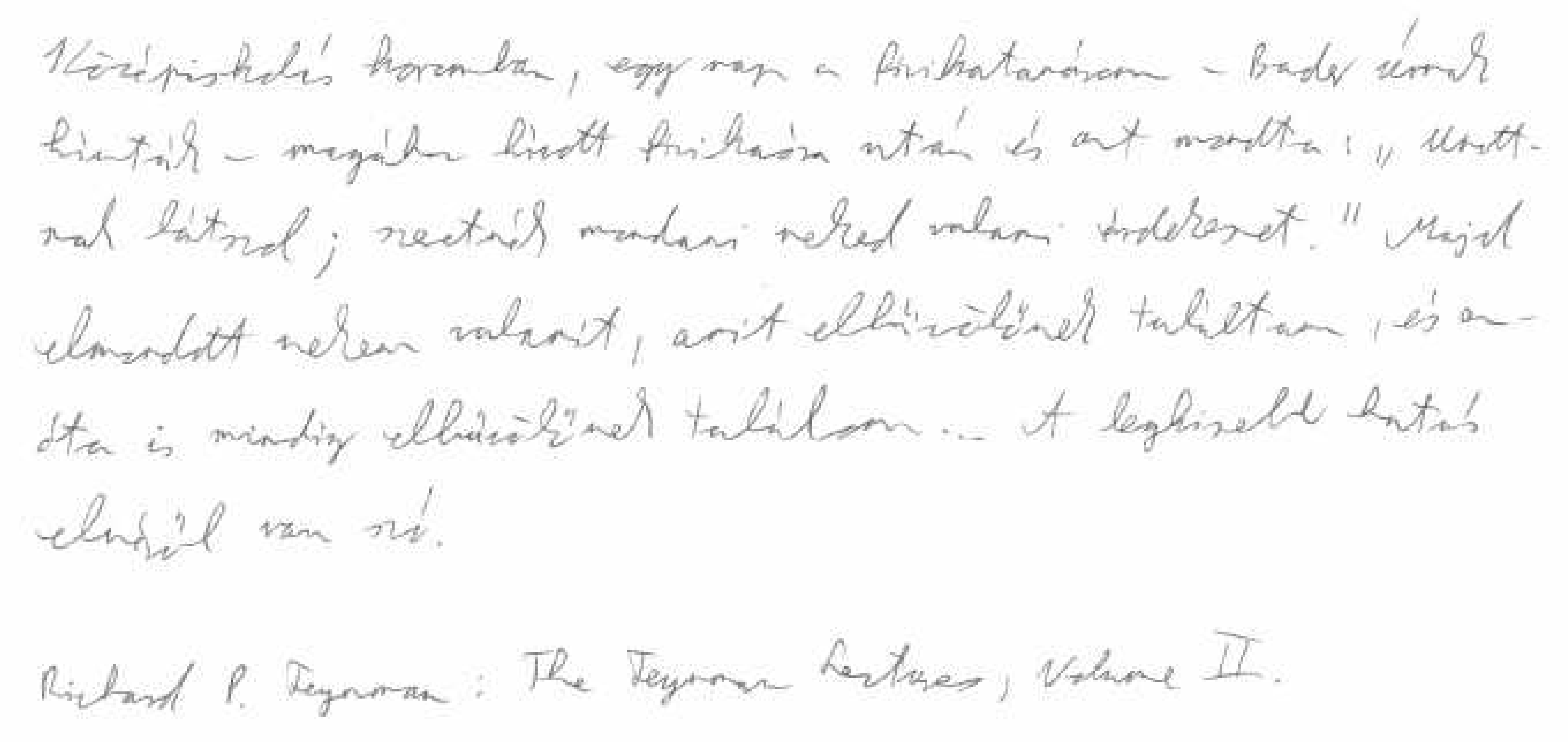

When I was in high school, my physics teacher – whose name was Mr. Bader – called me down one day after physics class and said, "You look bored; I want to tell you something interesting." Then he told me something which I found fascinating, and have, since then, always found fascinating.... The subject is this – the principle of least action.

Richard P. Feynman: The Feynman Lectures, Volume II.

- human performance 100% (character level)

Introduction

Handwritten Text

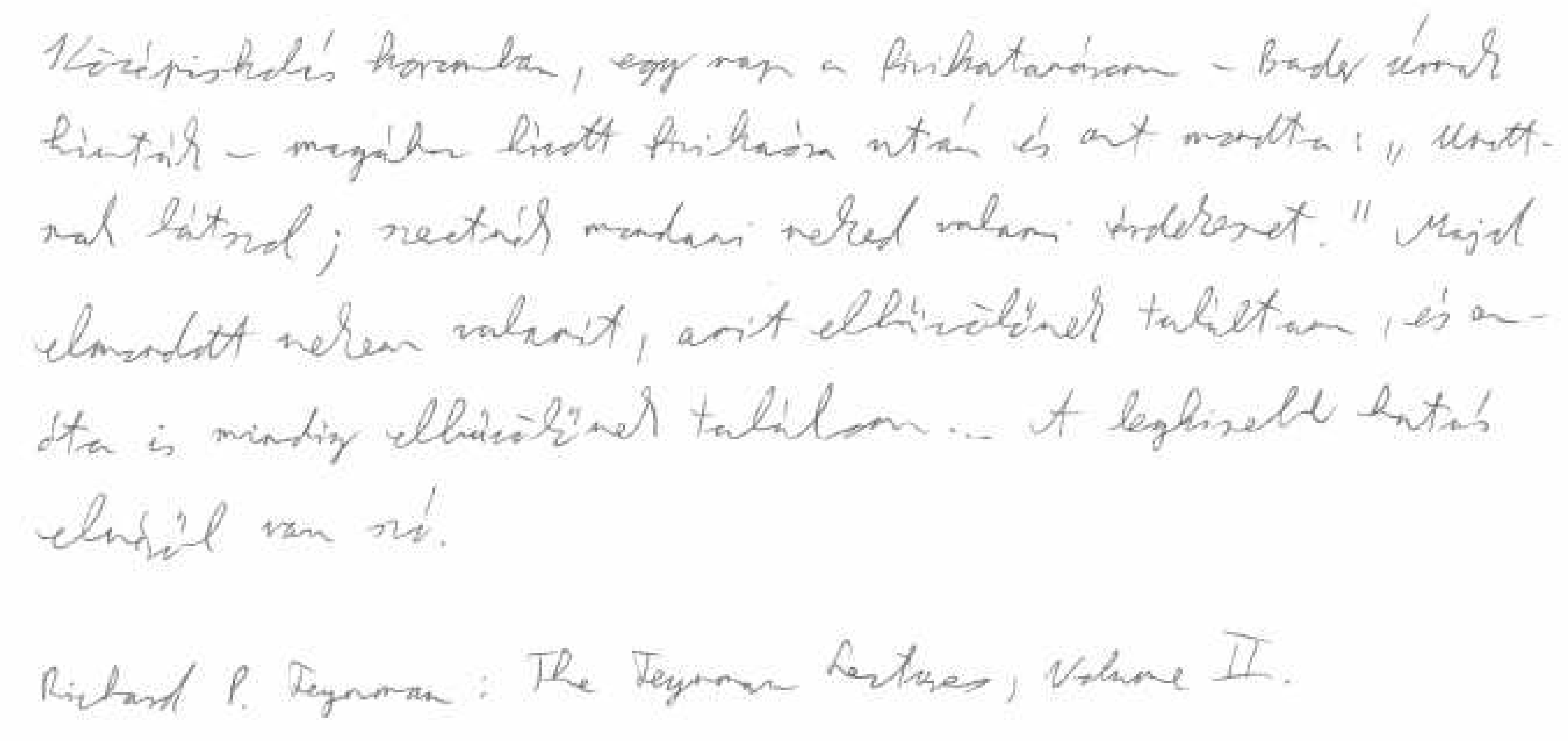


Középiskolás koromban, egy nap a fizikatanárom – Bader úrnak hívták – magához hívott fizikaóra után és azt mondta: „Unott-
nak látszol; mesélek neked valami érdekeset." Majd
elmondott nekem valamit, amit elbűvölőnek találtam, és az-
óta is mindig elbűvölőnek találom... A legkisebb hatás
elvéről van szó.

Richard P. Feynman: The Feynman Lectures, Volume II.

Introduction

Handwritten Text (Magyar)



- human performance 70% (character level, no understanding)

Introduction

Observations

- performance of current systems far from perfect

Introduction

Observations

- performance of current systems far from perfect
- computers can compete on the character level

Introduction

Observations

- performance of current systems far from perfect
- computers can compete on the character level
- text understanding is responsible for high recognition rates

Introduction

Observations

- performance of current systems far from perfect
- computers can compete on the character level
- text understanding is responsible for high recognition rates
- humans can combine optical recognition with
  - syntax of the language
  - semantics and "world knowledge"

Introduction

Observations

- performance of current systems far from perfect
- computers can compete on the character level
- text understanding is responsible for high recognition rates
- humans can combine optical recognition with
  - syntax of the language
  - semantics and "world knowledge"
- formal descriptions of semantics are missing, but

Introduction

Observations

- performance of current systems far from perfect
- computers can compete on the character level
- text understanding is responsible for high recognition rates
- humans can combine optical recognition with
  - syntax of the language
  - semantics and "world knowledge"
- formal descriptions of semantics are missing, but
- syntax analysis could help to boost recognition rates

Introduction

Syntax Analysis

- formal descriptions of language syntax (Chomsky 1956)

Introduction

Syntax Analysis

- formal descriptions of language syntax (Chomsky 1956)
- formal grammars most frequently used to specify
  - programming languages
  - communication protocols

Introduction

Syntax Analysis

- formal descriptions of language syntax (Chomsky 1956)
- formal grammars most frequently used to specify
  - programming languages
  - communication protocols
- special programs (parsers) help to
  - check syntactical correctness
  - analyze syntactical structure

Introduction

Syntax Analysis

- formal descriptions of language syntax (Chomsky 1956)
- formal grammars most frequently used to specify
  - programming languages
  - communication protocols
- special programs (parsers) help to
  - check syntactical correctness
  - analyze syntactical structure
- more recently, formal grammars for natural languages
  - Susanne corpus (1992)
  - Penn treebank (1993)
  - Lancaster parsed corpus (1995)

Introduction

Lancaster Parsed Corpus

- covers a variety of 15 text categories:
  - press: reportage
  - press: editorial
  - religion
  - general fiction
  - romance
  - humor
  - …

Introduction

Lancaster Parsed Corpus

- covers a variety of 15 text categories:
  - press: reportage
  - press: editorial
  - religion
  - general fiction
  - romance
  - humor
  - ...
- contains 12,000 parsed English sentences

Introduction

Lancaster Parsed Corpus

- covers a variety of 15 text categories:
  - press: reportage
  - press: editorial
  - religion
  - general fiction
  - romance
  - humor
  - ...
- contains 12,000 parsed English sentences
- parse trees as bracketed sentences

Introduction

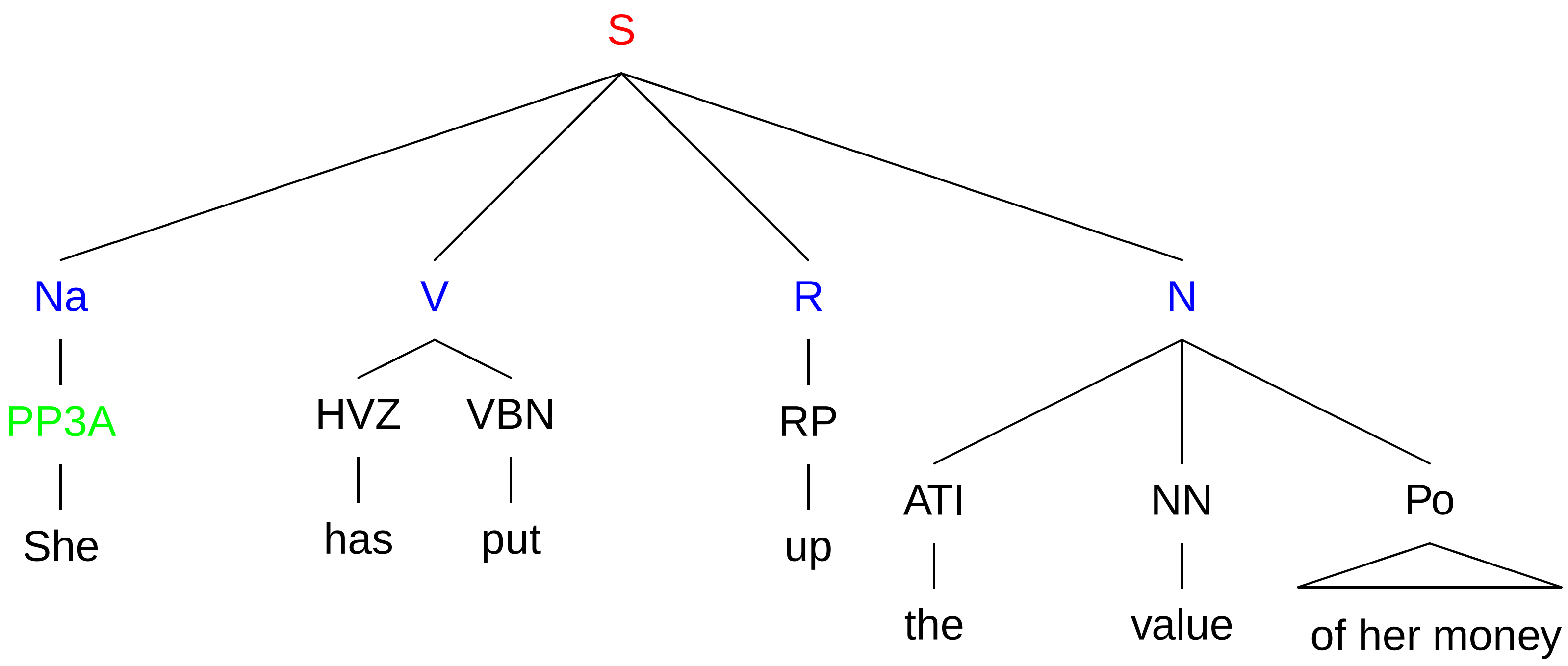
A Bracketed Sentence

$[S\ [Na\ [PP3A$ She $]][V\ [HVZ$ has $][VBN$ put $]][R\ [RP$ up $]][N\ [ATI$ the $][NN$ value $][Po\ [INO$ of $][N\ [PP\$$ her $][NN$ money $]]]]]$

S	independent sentence
Na	noun phrase (subject of verb)
PP3A	personal pronoun, 3rd pers. singular nom
V	verb phrase
R	adverb phrase
N	noun phrase

Introduction

The Parse Tree



Problem Statement

Find Answers to the following Questions:

Problem Statement

Find Answers to the following Questions:

- what type of grammar should be used?

Problem Statement

Find Answers to the following Questions:

- what type of grammar should be used?
- how shoud syntax anlaysis and text recognition be combined?

Problem Statement

Find Answers to the following Questions:

- what type of grammar should be used?
- how shoud syntax anlaysis and text recognition be combined?
- can syntax analysis help handwriting recognition?

Organization

1. Introduction
2. Problem Statement
3. The Grammar
4. The Parser
5. IAM Database and Recognizer
6. The Combination
7. Setup, Optimization and Evaluation
8. Conclusions and Outlook

The Grammar

What Type of Grammar should be used?

requirements:

The Grammar

What Type of Grammar should be used?

requirements:

- acceptable computational complexity

The Grammar

What Type of Grammar should be used?

requirements:

- acceptable computational complexity
- powerful description of syntax

The Grammar

What Type of Grammar should be used?

requirements:

- acceptable computational complexity
- powerful description of syntax
- support of ranking a candidate list (provided by the recognizer)

The Grammar

What Type of Grammar should be used?

requirements:

- acceptable computational complexity
- powerful description of syntax
- support of ranking a candidate list (provided by the recognizer)
- available linguistic resources

The Grammar

What Type of Grammar should be used?

requirements:

- acceptable computational complexity
- powerful description of syntax
- support of ranking a candidate list (provided by the recognizer)
- available linguistic resources

therefore:

- Stochastic Context-Free Grammars (SCFG)

The Grammar

Why SCFG?

The Grammar

Why SCFG?

- SCFG offer a compromise between
  - computational complexity
  - description power of syntax

The Grammar

Why SCFG?

- SCFG offer a compromise between
  - computational complexity
  - description power of syntax
- compatible linguistic resources exist

The Grammar

Why SCFG?

- SCFG offer a compromise between
  - computational complexity
  - description power of syntax
- compatible linguistic resources exist
- SCFG can be extracted from available resources

The Grammar

Why SCFG?

- SCFG offer a compromise between
  - computational complexity
  - description power of syntax
- compatible linguistic resources exist
- SCFG can be extracted from available resources
- SCFG allow candidate ranking

The Grammar

An Example

$$\begin{array}{lll l} S & \longrightarrow & A\ B & (0.8) \\ S & \longrightarrow & A\ S\ B & (0.2) \\ A & \longrightarrow & a & (1.0) \\ B & \longrightarrow & b & (1.0) \end{array}$$

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols: S,A,B

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

sentence generation

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

sentence generation

S

$[S\ \ ]$ (1.0)

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

sentence generation

S $\stackrel{0.2}{\longrightarrow}$ ASB

$[S\ [A\ ][S\ ][B\ ]\ ]$ (0.2)

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

sentence generation

$$\text{S} \stackrel{0.2}{\longrightarrow} \text{ASB} \stackrel{0.8}{\longrightarrow} \text{AABB}$$

$$[S\ [A\ ][S\ [A\ ][B\ ]\ ][B\ ]]$$ (0.16)

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

sentence generation

S $\stackrel{0.2}{\rightarrow}$ ASB $\stackrel{0.8}{\rightarrow}$ AABB $\stackrel{1.0}{\rightarrow}$ aABB

$[S\ [A\ \text{a}\ ][S\ [A\ ][B\ ]][B\ ]]$ (0.16)

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

sentence generation

$$\text{S} \overset{0.2}{\longrightarrow} \text{ASB} \overset{0.8}{\longrightarrow} \text{AABB} \overset{1.0}{\longrightarrow} \text{aABB} \overset{1.0}{\longrightarrow} \text{aaBB} \longrightarrow \ldots$$

$[S\ [A\ \text{a}\ ][S\ [A\ \text{a}\ ][B\ \ ]][B\ \ ]]$ (0.16)

The Grammar

An Example

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
productions or rules:	see left
rule probabilities:	see left

sentence generation

$$\text{S} \stackrel{0.2}{\longrightarrow} \text{ASB} \stackrel{0.8}{\longrightarrow} \text{AABB} \stackrel{1.0}{\longrightarrow} \text{aABB} \stackrel{1.0}{\longrightarrow} \text{aaBB} \longrightarrow \ldots \longrightarrow \text{aabb}$$

$$[S\ [A\ \text{a}\ ][S\ [A\ \text{a}\ ][B\ \text{b}\ ]][B\ \text{b}\ ]] \quad (0.16)$$

sentence complete, no more nonterminal symbols to expand

The Grammar

An Example, continued

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
rules or productions:	see left
rule probabilities:	see left

syntax analysis

The Grammar

An Example, continued

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
rules or productions:	see left
rule probabilities:	see left

syntax analysis

aabb

The Grammar

An Example, continued

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
rules or productions:	see left
rule probabilities:	see left

syntax analysis

aabb correct syntax, sentence probability 0.16

The Grammar

An Example, continued

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
rules or productions:	see left
rule probabilities:	see left

syntax analysis

aabb correct syntax, sentence probability 0.16

ab

The Grammar

An Example, continued

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
rules or productions:	see left
rule probabilities:	see left

syntax analysis

aabb	correct syntax, sentence probability 0.16
ab	correct syntax, sentence probability 0.8

The Grammar

An Example, continued

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
rules or productions:	see left
rule probabilities:	see left

syntax analysis

aabb	correct syntax, sentence probability 0.16
ab	correct syntax, sentence probability 0.8
aab	

The Grammar

An Example, continued

S	$\longrightarrow$	A B	(0.8)
S	$\longrightarrow$	A S B	(0.2)
A	$\longrightarrow$	a	(1.0)
B	$\longrightarrow$	b	(1.0)

nonterminal symbols:	S,A,B
terminal symbols (words):	a,b
start symbol:	S
rules or productions:	see left
rule probabilities:	see left

syntax analysis

aabb	correct syntax, sentence probability 0.16
ab	correct syntax, sentence probability 0.8
aab	syntax not correct, cannot be produced by grammar

The Grammar

Extraction of Rules from Bracketed Sentences

$[S$ $[Na$ $[PP3A$ She $]]$$[V$ $[HVZ$ has $]$$[VBN$ put $]]$$[R$ $[RP$ up $]]$$[N$ $[ATI$ the $]$$[NN$ value $]$$[Po$ $[INO$ of $]$$[N$ $[PP\$$ her $]$$[NN$ money $]]]]]$

The Grammar

Extraction of Rules from Bracketed Sentences

$[S$ $[Na$ $[PP3A$ She $]]$$[V$ $[HVZ$ has $]$$[VBN$ put $]]$$[R$ $[RP$ up $]]$$[N$ $[ATI$ the $]$$[NN$ value $]$$[Po$ $[INO$ of $]$$[N$ $[PP\$$ her $]$$[NN$ money $]]]]]$

S $\longrightarrow$ Na V R N

The Grammar

Extraction of Rules from Bracketed Sentences

$[S\ [Na\ [PP3A$ She $]][V\ [HVZ$ has $][VBN$ put $]][R\ [RP$ up $]][N$ $[ATI$ the $][NN$ value $][Po\ [INO$ of $][N\ [PP\$$ her $][NN$ money $]]]]]$

S	$\longrightarrow$	Na V R N
Na	$\longrightarrow$	PP3A

The Grammar

Extraction of Rules from Bracketed Sentences

$[S\ [Na\ [PP3A$ She $]][V\ [HVZ$ has $][VBN$ put $]][R\ [RP$ up $]][N\ [ATI$ the $][NN$ value $][Po\ [INO$ of $][N\ [PP\$$ her $][NN$ money $]]]]]$

S	$\longrightarrow$	Na V R N
Na	$\longrightarrow$	PP3A
PP3A	$\longrightarrow$	She

The Grammar

Extraction of Rules from Bracketed Sentences

$[S\ [Na\ [PP3A$ She $]][V\ [HVZ$ has $][VBN$ put $]][R\ [RP$ up $]][N$ $[ATI$ the $][NN$ value $][Po\ [INO$ of $][N\ [PP\$$ her $][NN$ money $]]]]]$

S	$\longrightarrow$	Na V R N
Na	$\longrightarrow$	PP3A
PP3A	$\longrightarrow$	She
V	$\longrightarrow$	HVZ VBN

The Grammar

Extraction of Rules from Bracketed Sentences

$[S\ [Na\ [PP3A$ She $]][V\ [HVZ$ has $][VBN$ put $]][R\ [RP$ up $]][N$
$[ATI$ the $][NN$ value $][Po\ [INO$ of $][N\ [PP\$$ her $][NN$ money $]]]]]$

S	$\longrightarrow$	Na V R N
Na	$\longrightarrow$	PP3A
PP3A	$\longrightarrow$	She
V	$\longrightarrow$	HVZ VBN
HVZ	$\longrightarrow$	has
VBN	$\longrightarrow$	put
...		

The Grammar

Extraction of Rules from Bracketed Sentences

$[S\ [Na\ [PP3A$ She $]][V\ [HVZ$ has $][VBN$ put $]][R\ [RP$ up $]][N$ $[ATI$ the $][NN$ value $][Po\ [INO$ of $][N\ [PP\$$ her $][NN$ money $]]]]]$

	S	$\longrightarrow$	Na V R N
	Na	$\longrightarrow$	PP3A
lexical	PP3A	$\longrightarrow$	She
	V	$\longrightarrow$	HVZ VBN
lexical	HVZ	$\longrightarrow$	has
lexical	VBN	$\longrightarrow$	put
	...		

The Grammar

Extraction of Symbols

The Grammar

Extraction of Symbols

- start symbol

The Grammar

Extraction of Symbols

- start symbol (top level symbol)
  - S

The Grammar

Extraction of Symbols

- start symbol (top level symbol)
  - S
- nonterminal symbols

The Grammar

Extraction of Symbols

- start symbol (top level symbol)
  - S
- nonterminal symbols (constituent tags, LHS of non-lexical rules)
  - Na
  - V

The Grammar

Extraction of Symbols

- start symbol (top level symbol)
  - S
- nonterminal symbols (constituent tags, LHS of non-lexical rules)
  - Na
  - V
- preterminal symbols

The Grammar

Extraction of Symbols

- start symbol (top level symbol)
  - S
- nonterminal symbols (constituent tags, LHS of non-lexical rules)
  - Na
  - V
- preterminal symbols (word tags, LHS of lexical rules)
  - PP3A
  - HVZ

The Grammar

Extraction of Symbols

- start symbol (top level symbol)
  - S
- nonterminal symbols (constituent tags, LHS of non-lexical rules)
  - Na
  - V
- preterminal symbols (word tags, LHS of lexical rules)
  - PP3A
  - HVZ
- terminal symbols

The Grammar

Extraction of Symbols

- start symbol (top level symbol)
  - S
- nonterminal symbols (constituent tags, LHS of non-lexical rules)
  - Na
  - V
- preterminal symbols (word tags, LHS of lexical rules)
  - PP3A
  - HVZ
- terminal symbols (words, RHS of lexical rules)
  - She
  - has

The Grammar

Assignment of Probabilities

A	$\rightarrow$	α
V	$\rightarrow$	VBD
V	$\rightarrow$	HVZ VBN
V	$\rightarrow$	HVZ XNOT VBN

$$p(A \rightarrow \alpha) = \frac{N(A \rightarrow \alpha)}{\sum_{\beta} N(A \rightarrow \beta)}$$

The Grammar

Assignment of Probabilities

A	$\rightarrow$	α	$N(A \rightarrow \alpha)$
V	$\rightarrow$	VBD	4,578
V	$\rightarrow$	HVZ VBN	130
V	$\rightarrow$	HVZ XNOT VBN	5

$$p(A \rightarrow \alpha) = \frac{N(A \rightarrow \alpha)}{\sum_{\beta} N(A \rightarrow \beta)}$$

- count number of occurances of each production

The Grammar

Assignment of Probabilities

A	$\rightarrow$	α	$N(A \rightarrow \alpha)$	$\sum N(A \rightarrow \beta)$	$p(A \rightarrow \alpha)$
V	$\rightarrow$	VBD	4,578	16,158	0.2833
V	$\rightarrow$	HVZ VBN	130	16,158	0.0081
V	$\rightarrow$	HVZ XNOT VBN	5	16,158	0.0003

$$p(A \rightarrow \alpha) = \frac{N(A \rightarrow \alpha)}{\sum_\beta N(A \rightarrow \beta)}$$

- count number of occurances of each production
- normalize counts by the LHS

The Parser

Task of the Parser

She|has|put|up|the|value|of|her|money|.

$\downarrow$

SCFG $\longrightarrow$ Parser

$\downarrow$

most probable parse:

7.691e-23 $[S\ [Na\ [PP3A$ She . . .

The Parser

Which Parser?

The Parser

Which Parser?

- CYK Parser (Younger ’67)

The Parser

Which Parser?

- CYK Parser (Younger '67)
- Early Parser (Early '80)

The Parser

Which Parser?

- CYK Parser (Younger ’67)
- Early Parser (Early ’80)
- Prob. Eearly Parser (Stolcke ’94)

The Parser

Which Parser?

- CYK Parser (Younger '67)
- Early Parser (Early '80)
- Prob. Eearly Parser (Stolcke '94)
- CYK+ Parser (Chappelier & Rajman '98)

The Parser

Which Parser?

- CYK Parser (Younger '67)
- Early Parser (Early '80)
- Prob. Eearly Parser (Stolcke '94)
- CYK+ Parser (Chappelier & Rajman '98)
- ...

The Parser

The CYK+ Parser

the cat ate the mouse

S	$\longrightarrow$	NP VP(1.0)
VP	$\longrightarrow$	V(0.3)|V NP(0.7)
NP	$\longrightarrow$	Det N(1.0)
Det	$\longrightarrow$	the(1.0)
N	$\longrightarrow$	cat(0.5)|mouse(0.5)
V	$\longrightarrow$	ate(1.0)

The Parser

The CYK+ Parser

active rule: Det $\longrightarrow$ the

the	cat	ate	the	mouse
Det				

$$
\begin{aligned}
S &\longrightarrow NP\ VP(1.0) \\
VP &\longrightarrow V(0.3) | V\ NP(0.7) \\
NP &\longrightarrow Det\ N(1.0) \\
Det &\longrightarrow the(1.0) \\
N &\longrightarrow cat(0.5) | mouse(0.5) \\
V &\longrightarrow ate(1.0)
\end{aligned}
$$

The Parser

The CYK+ Parser

active rule: NP $\longrightarrow$Det $\bullet$ N

the cat ate the mouse

Det

Det$\bullet$

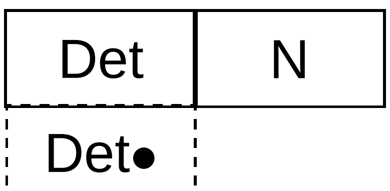
S	$\longrightarrow$	NP VP(1.0)
VP	$\longrightarrow$	V(0.3)|V NP(0.7)
NP	$\longrightarrow$	Det N(1.0)
Det	$\longrightarrow$	the(1.0)
N	$\longrightarrow$	cat(0.5)|mouse(0.5)
V	$\longrightarrow$	ate(1.0)

The Parser

The CYK+ Parser

active rule: N →cat

the cat ate the mouse



S → NP VP(1.0)

VP → V(0.3)|V NP(0.7)

NP → Det N(1.0)

Det → the(1.0)

N → cat(0.5)|mouse(0.5)

V → ate(1.0)

The Parser

The CYK+ Parser

active rule:

the cat ate the mouse

Det	N
Det●	

$S \longrightarrow$ NP VP(1.0)

$VP \longrightarrow$ V(0.3)|V NP(0.7)

$NP \longrightarrow$ Det N(1.0)

$Det \longrightarrow$ the(1.0)

$N \longrightarrow$ cat(0.5)|mouse(0.5)

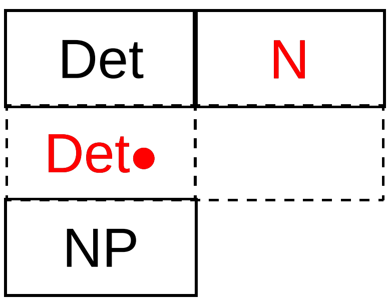
$V \longrightarrow$ ate(1.0)

The Parser

The CYK+ Parser

active rule: NP $\longrightarrow$ Det N

the cat ate the mouse



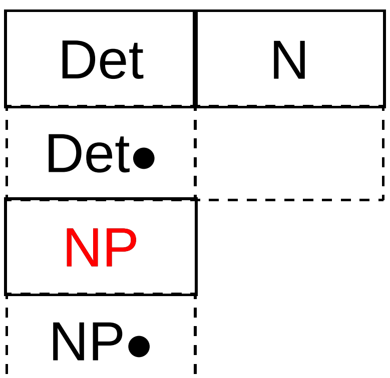
S	$\longrightarrow$	NP VP(1.0)
VP	$\longrightarrow$	V(0.3)|V NP(0.7)
NP	$\longrightarrow$	Det N(1.0)
Det	$\longrightarrow$	the(1.0)
N	$\longrightarrow$	cat(0.5)|mouse(0.5)
V	$\longrightarrow$	ate(1.0)

The Parser

The CYK+ Parser

active rule: S $\longrightarrow$ NP • VP

the cat ate the mouse



S $\longrightarrow$ NP VP(1.0)

VP $\longrightarrow$ V(0.3)|V NP(0.7)

NP $\longrightarrow$ Det N(1.0)

Det $\longrightarrow$ the(1.0)

N $\longrightarrow$ cat(0.5)|mouse(0.5)

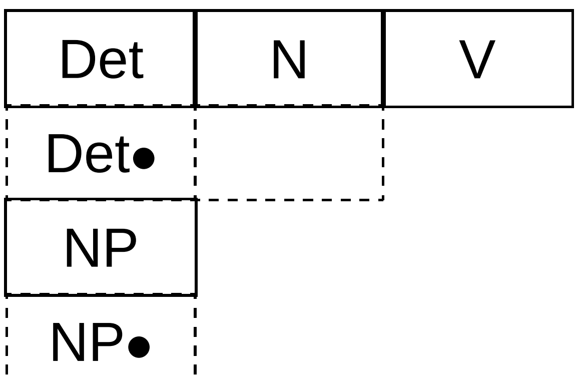
V $\longrightarrow$ ate(1.0)

The Parser

The CYK+ Parser

active rule: V $\rightarrow$ate

the cat ate the mouse



S $\longrightarrow$ NP VP(1.0)

VP $\longrightarrow$ V(0.3)|V NP(0.7)

NP $\longrightarrow$ Det N(1.0)

Det $\longrightarrow$ the(1.0)

N $\longrightarrow$ cat(0.5)|mouse(0.5)

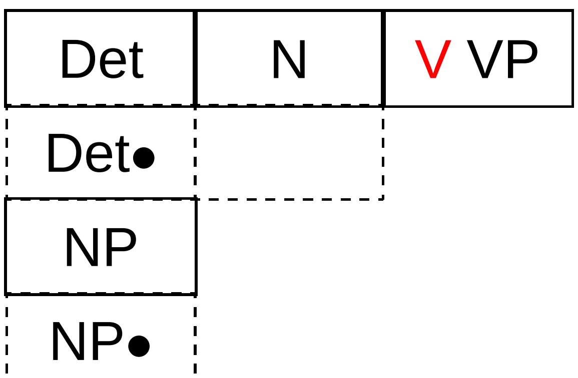
V $\longrightarrow$ ate(1.0)

The Parser

The CYK+ Parser

active rule: VP $\longrightarrow$ V

the cat ate the mouse



S	$\longrightarrow$	NP VP(1.0)
VP	$\longrightarrow$	V(0.3)|V NP(0.7)
NP	$\longrightarrow$	Det N(1.0)
Det	$\longrightarrow$	the(1.0)
N	$\longrightarrow$	cat(0.5)|mouse(0.5)
V	$\longrightarrow$	ate(1.0)

The Parser

The CYK+ Parser

active rule: VP $\longrightarrow$ V • NP

the cat ate the mouse

Det	N	V VP
Det•		V•
NP		
NP•		

S $\longrightarrow$ NP VP(1.0)

VP $\longrightarrow$ V(0.3)|V NP(0.7)

NP $\longrightarrow$ Det N(1.0)

Det $\longrightarrow$ the(1.0)

N $\longrightarrow$ cat(0.5)|mouse(0.5)

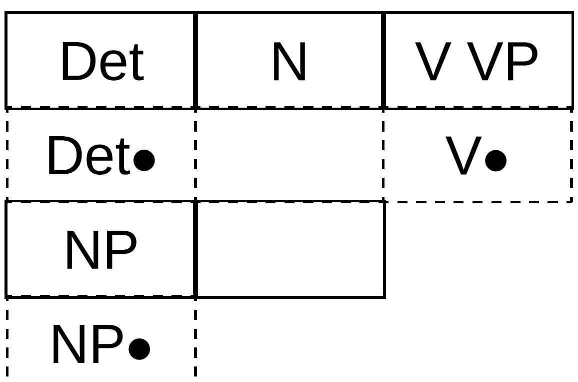
V $\longrightarrow$ ate(1.0)

The Parser

The CYK+ Parser

active rule:

the cat ate the mouse



S $\rightarrow$ NP VP(1.0)

VP $\rightarrow$ V(0.3)|V NP(0.7)

NP $\rightarrow$ Det N(1.0)

Det $\rightarrow$ the(1.0)

N $\rightarrow$ cat(0.5)|mouse(0.5)

V $\rightarrow$ ate(1.0)

The Parser

The CYK+ Parser

active rule:

the	cat	ate	the	mouse
Det	N	V VP		
Det•		V•		
NP				
NP•				

$S \longrightarrow$ NP VP(1.0)

$VP \longrightarrow$ V(0.3)|V NP(0.7)

$NP \longrightarrow$ Det N(1.0)

$Det \longrightarrow$ the(1.0)

$N \longrightarrow$ cat(0.5)|mouse(0.5)

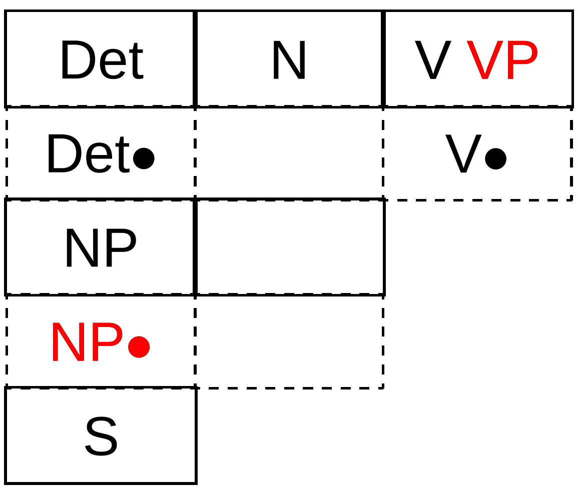
$V \longrightarrow$ ate(1.0)

The Parser

The CYK+ Parser

active rule: S →NP VP

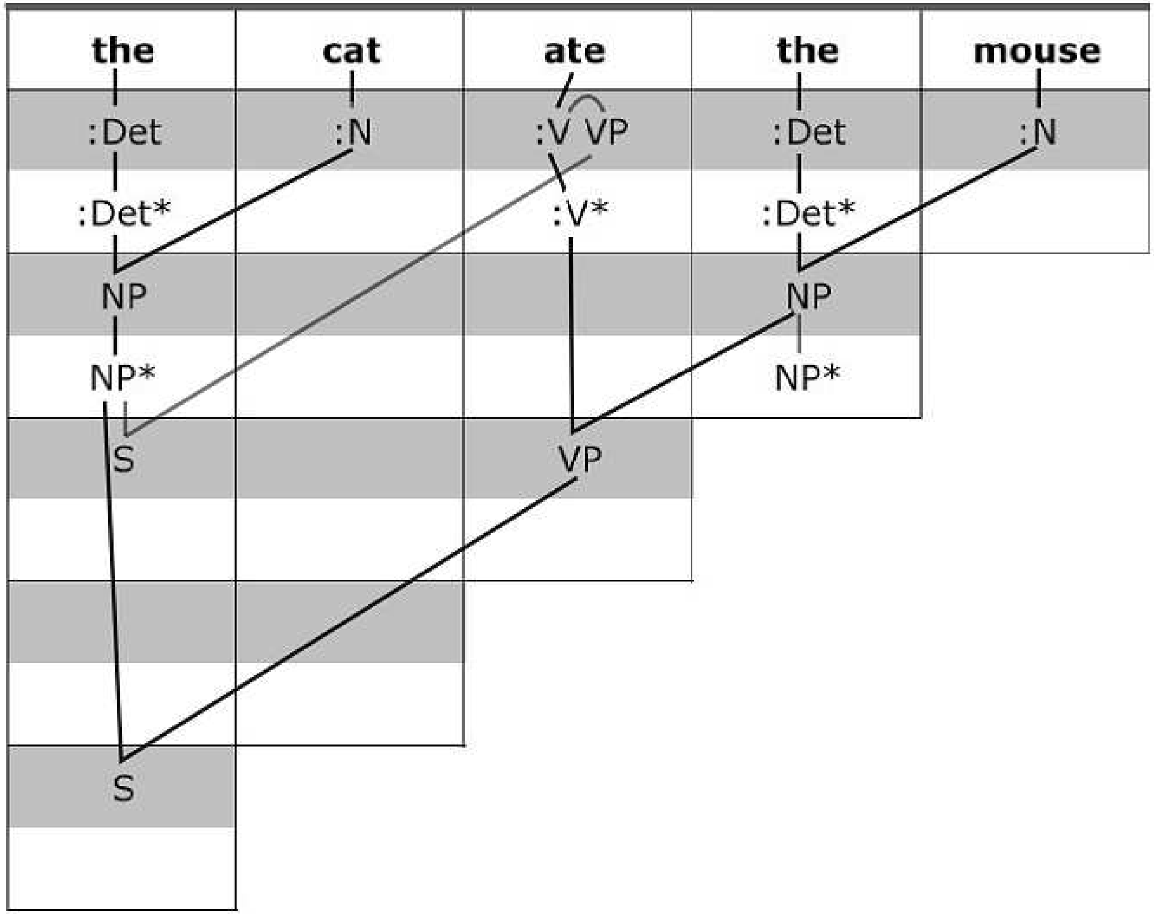
the cat ate the mouse



S → NP VP(1.0)
VP → V(0.3)|V NP(0.7)
NP → Det N(1.0)
Det → the(1.0)
N → cat(0.5)|mouse(0.5)
V → ate(1.0)

The Parser

Complete Parse Chart



The Parser

Computing the Most Probable Parse

initialization (1st row of chart)

- assign probs of lexial rules to elements in closed list

new elements

- multiply probs of rule with probs of elements
- assign product to new element

existing elements

- multiply probs of rule with probs of elements
- if new prob is higher assign new prob and change back pointers

Organization

1. Introduction
2. Problem Statement
3. The Grammar
4. The Parser
5. IAM Database and Recognizer
6. The Combination
7. Setup, Optimization and Evaluation
8. Conclusions and Outlook

The IAM Database

Current State

- more than 1,500 scanned pages of handwritten text

The IAM Database

Current State

- more than 1,500 scanned pages of handwritten text
- material from over 600 different writers
  - 95,000 correctly segmented words
  - 13,000 lines of handwritten text
  - over 5,000 complete sentences

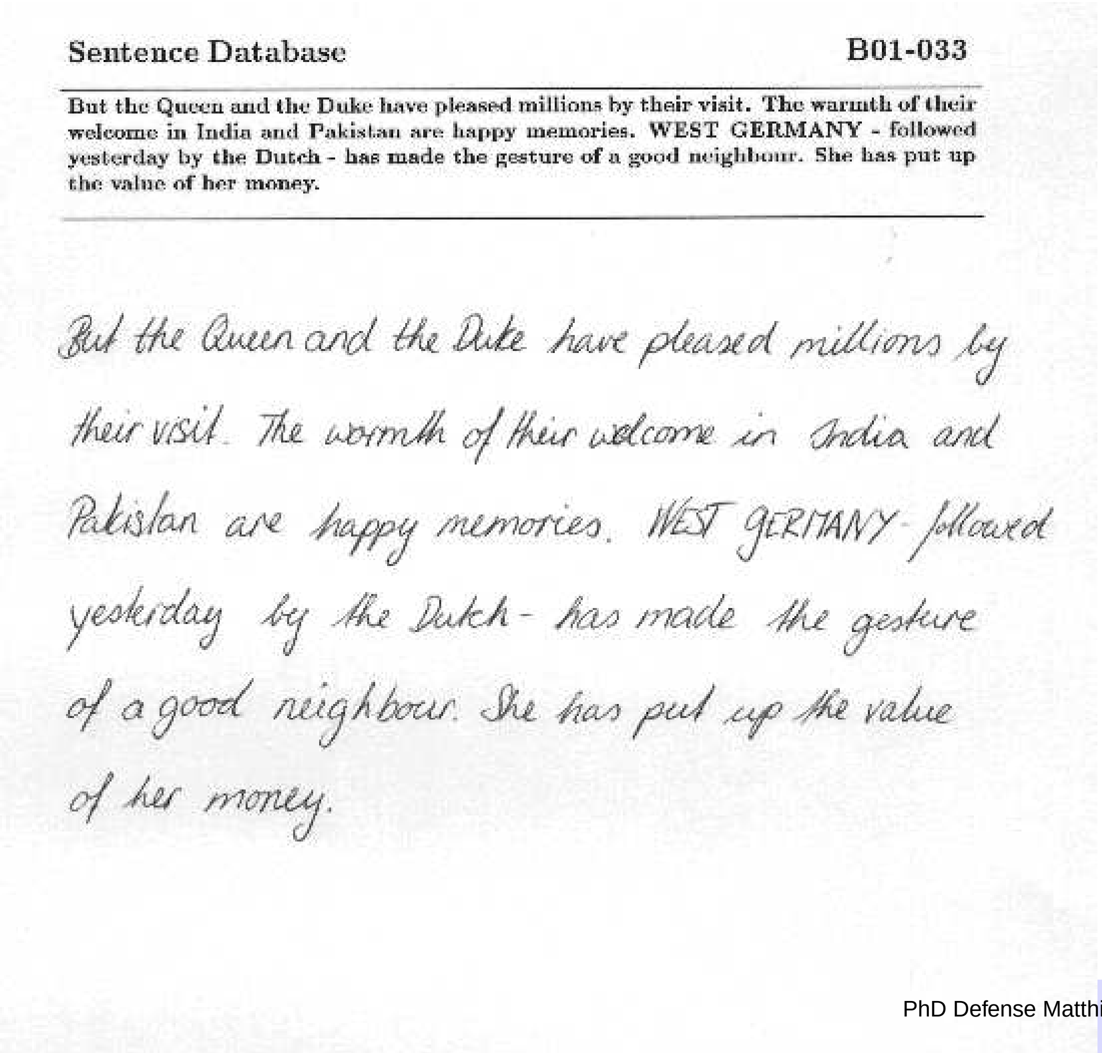
The IAM Database

Current State

- more than 1,500 scanned pages of handwritten text
- material from over 600 different writers
  - 95,000 correctly segmented words
  - 13,000 lines of handwritten text
  - over 5,000 complete sentences
- covering a vocabulary of over 12,000 words

The IAM Database

IAM Database: Form b01-033



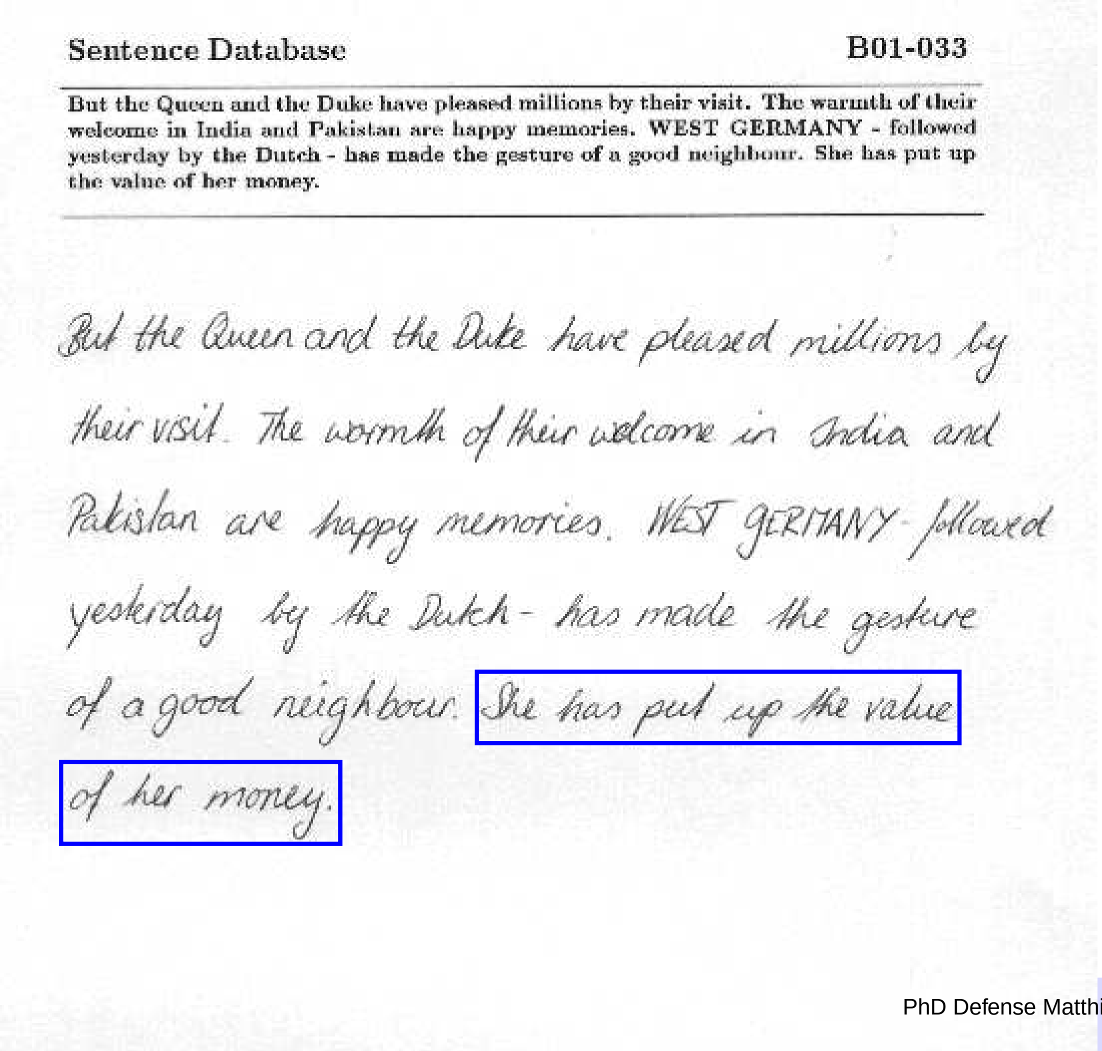
Sentence Database **B01-033**

But the Queen and the Duke have pleased millions by their visit. The warmth of their welcome in India and Pakistan are happy memories. WEST GERMANY - followed yesterday by the Dutch - has made the gesture of a good neighbour. She has put up the value of her money.

But the Queen and the Duke have pleased millions by
their visit. The warmth of their welcome in India and
Pakistan are happy memories. WEST GERMANY - followed
yesterday by the Dutch - has made the gesture
of a good neighbour. She has put up the value
of her money.

The IAM Database

IAM Database: Sentence b01-033-s3



Sentence Database **B01-033**

But the Queen and the Duke have pleased millions by their visit. The warmth of their welcome in India and Pakistan are happy memories. WEST GERMANY - followed yesterday by the Dutch - has made the gesture of a good neighbour. She has put up the value of her money.

But the Queen and the Duke have pleased millions by
their visit. The warmth of their welcome in India and
Pakistan are happy memories. WEST GERMANY - followed
yesterday by the Dutch - has made the gesture
of a good neighbour. She has put up the value
of her money.

The Recognizer

Based on Existing System (Marti ’00)

The Recognizer

Based on Existing System (Marti '00)

- text line normalization (skew, reference lines)

The Recognizer

Based on Existing System (Marti ’00)

- text line normalization (skew, reference lines)
- feature extraction (sliding window)

The Recognizer

Based on Existing System (Marti '00)

- text line normalization (skew, reference lines)
- feature extraction (sliding window)
- HMM for each character
  - linear topology
  - multi Gaussian
  - variable length modeling
- Baum-Welch training
- bigram language model

The Recognizer

Based on Existing System (Marti ’00)

- text line normalization (skew, reference lines)
- feature extraction (sliding window)
- HMM for each character
  - linear topology
  - multi Gaussian
  - variable length modeling
- Baum-Welch training
- bigram language model
- Viterbi decoding

The Recognizer

Based on Existing System (Marti '00)

- text line normalization (skew, reference lines)
- feature extraction (sliding window)
- HMM for each character
  - linear topology
  - multi Gaussian
  - variable length modeling
- Baum-Welch training
- bigram language model
- Viterbi decoding
- recognition lattice generation

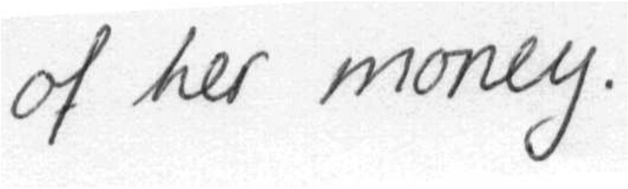
The Recognizer

Based on Existing System (Marti '00)

- text line normalization (skew, reference lines)
- feature extraction (sliding window)
- HMM for each character
  - linear topology
  - multi Gaussian
  - variable length modeling
- Baum-Welch training
- bigram language model
- Viterbi decoding
- recognition lattice generation
- n-best list generation

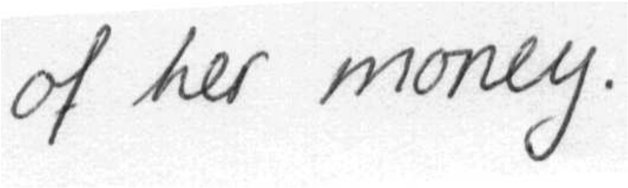
The Recognizer

Text Line Normalization

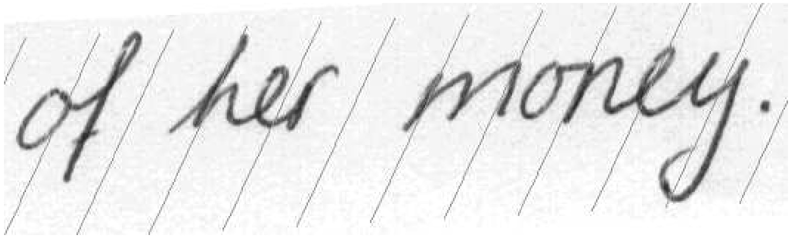


The Recognizer

Text Line Normalization

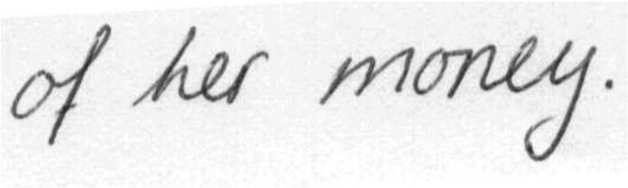


slant estimation

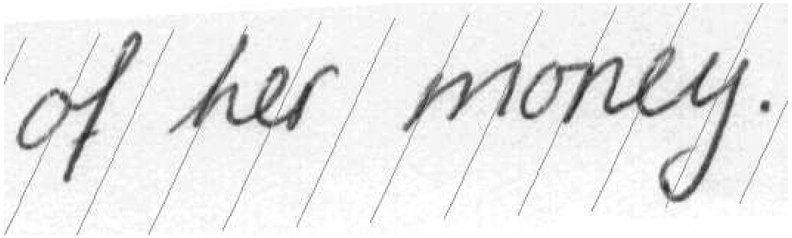


The Recognizer

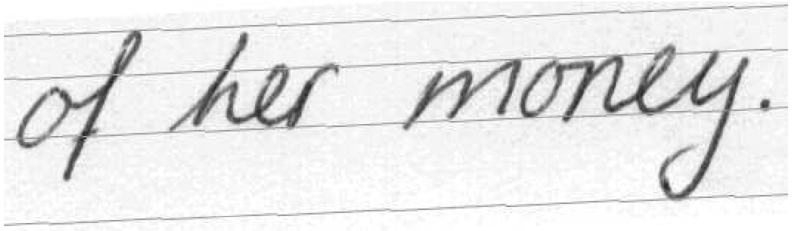
Text Line Normalization



slant estimation

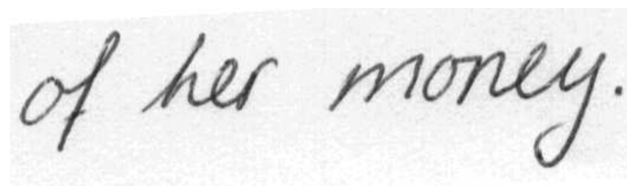


reference line estimation

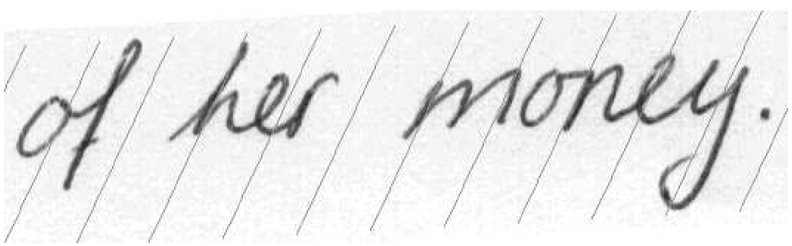


The Recognizer

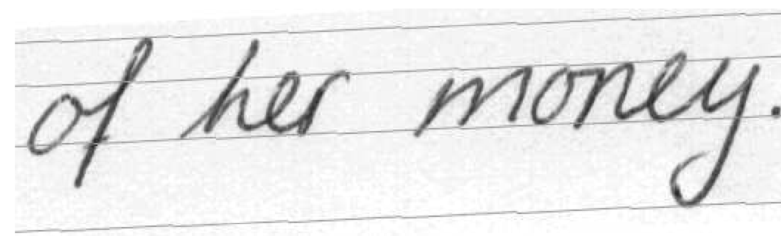
Text Line Normalization



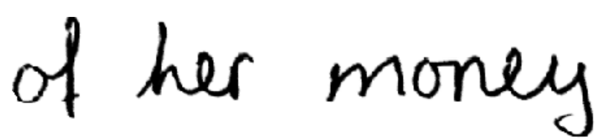
slant estimation



reference line estimation

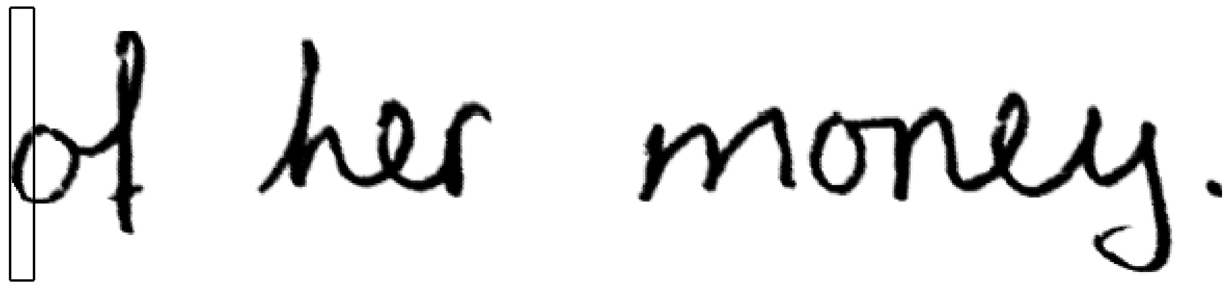


normalized text line



The Recognizer

Extraction of the Feature Vector Sequence



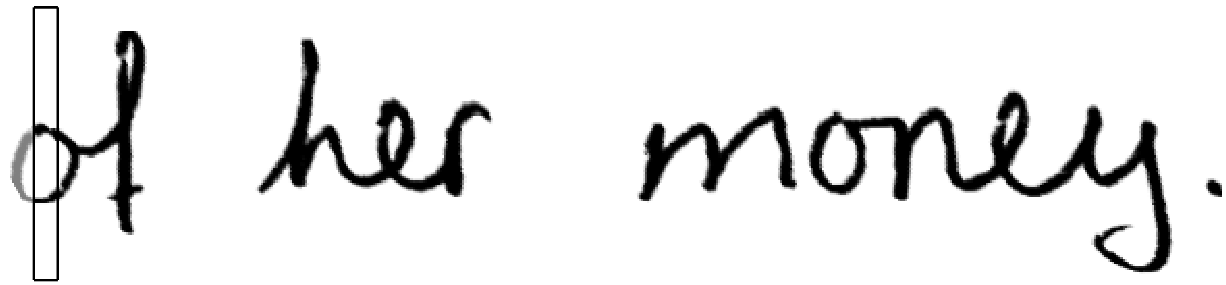
$$\downarrow$$

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

$$X = (X_1,$$

The Recognizer

Extraction of the Feature Vector Sequence



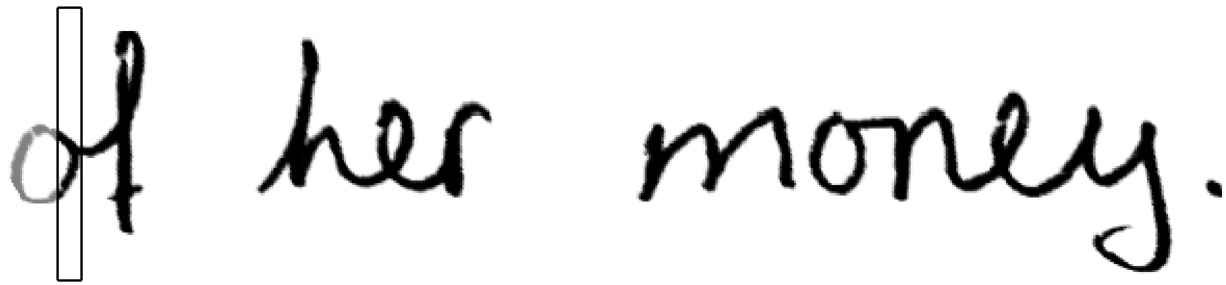
$$\downarrow$$

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

$$X = (X_1, X_2,$$

The Recognizer

Extraction of the Feature Vector Sequence



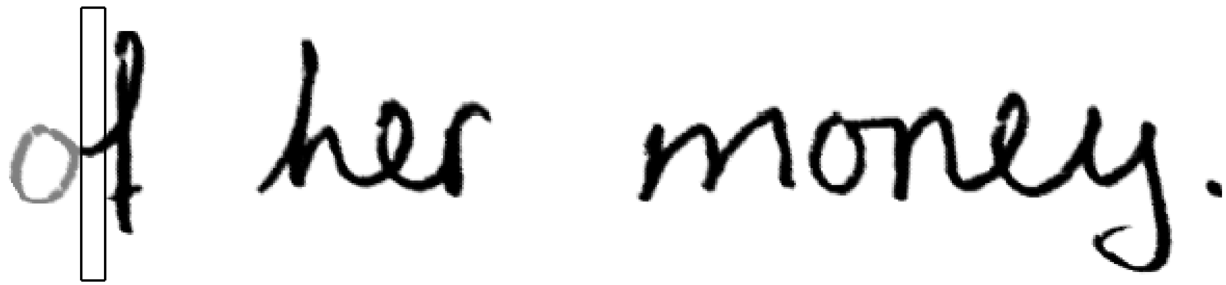
$$\downarrow$$

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

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

The Recognizer

Extraction of the Feature Vector Sequence



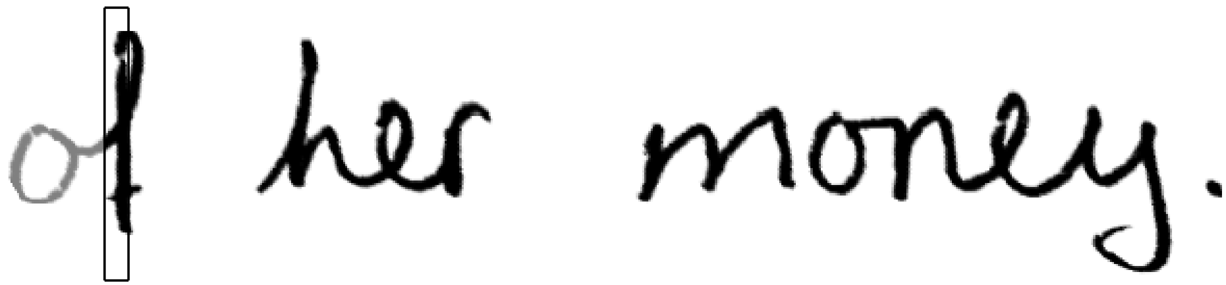
$$\downarrow$$

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

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

The Recognizer

Extraction of the Feature Vector Sequence



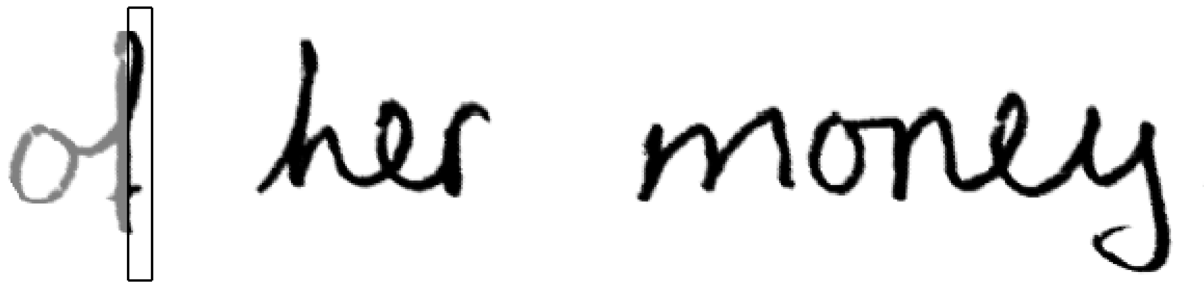
$$\downarrow$$

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

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

The Recognizer

Extraction of the Feature Vector Sequence



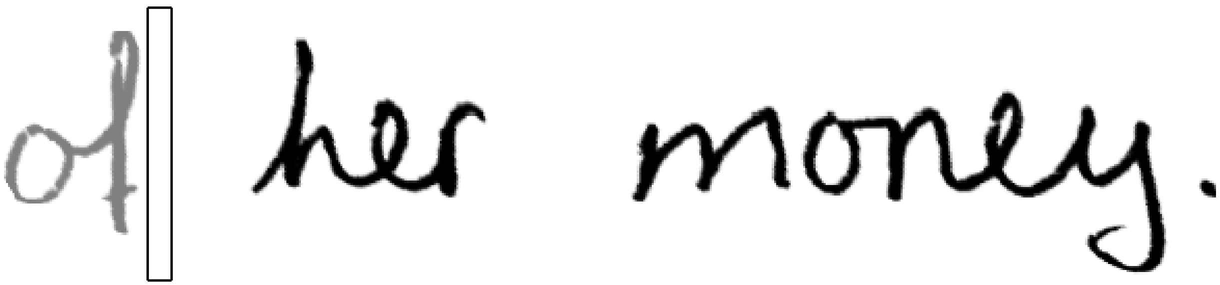
$$\downarrow$$

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

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

The Recognizer

Extraction of the Feature Vector Sequence



$$\downarrow$$

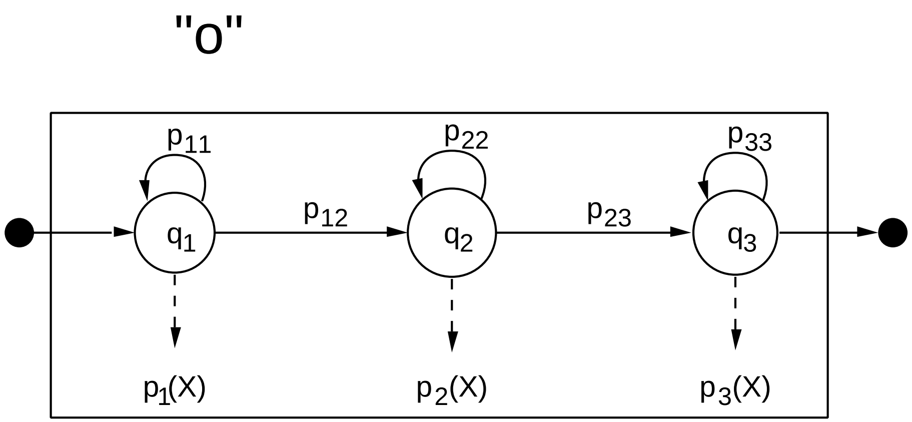
$$\begin{pmatrix} x_{71} \\ x_{72} \\ \vdots \\ x_{7m} \end{pmatrix}$$

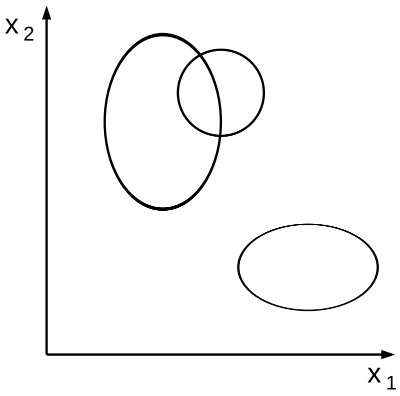
$$X = (X_1, X_2, X_3, X_4, X_5, X_6, X_7, \ldots$$

The Recognizer

Viterbi Decoding

$(X_1, X_2, X_3, \ldots$

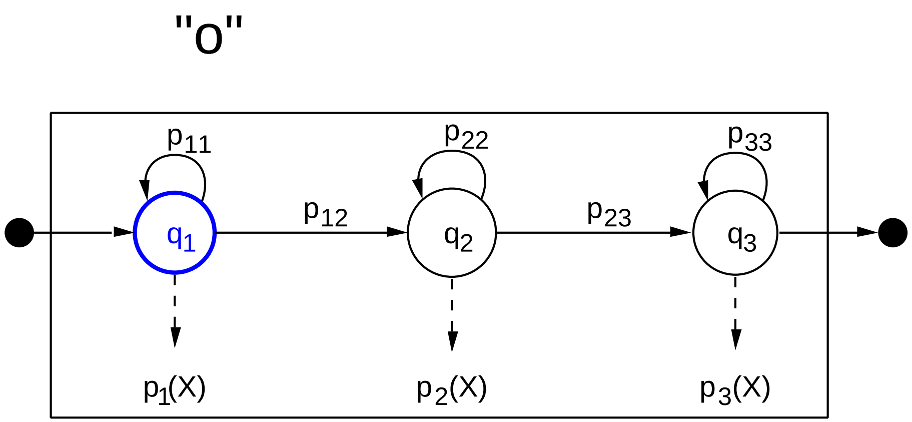


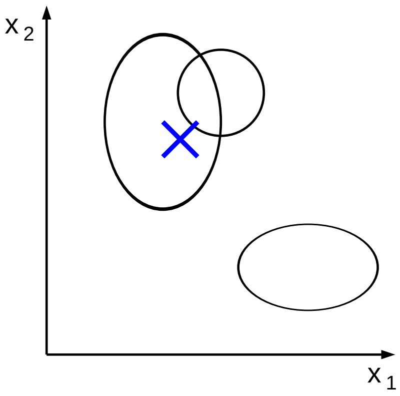


The Recognizer

Viterbi Decoding

$(X_1, X_2, X_3, \ldots$

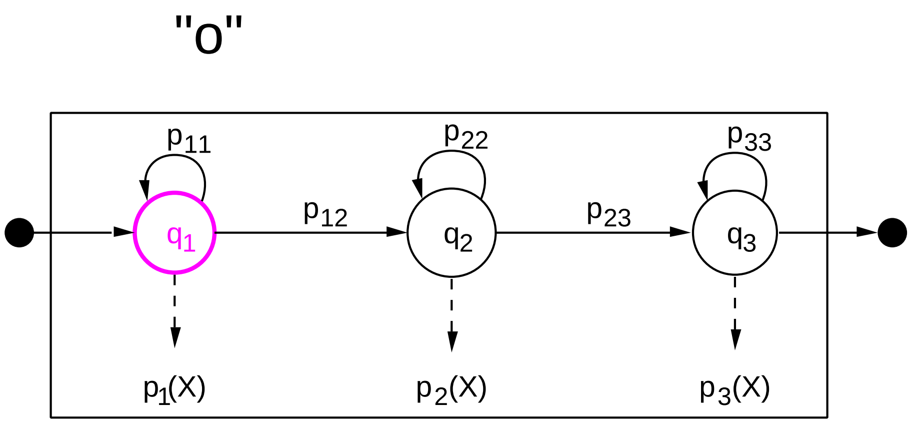


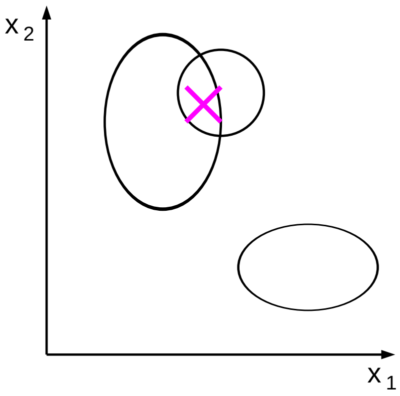


The Recognizer

Viterbi Decoding

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

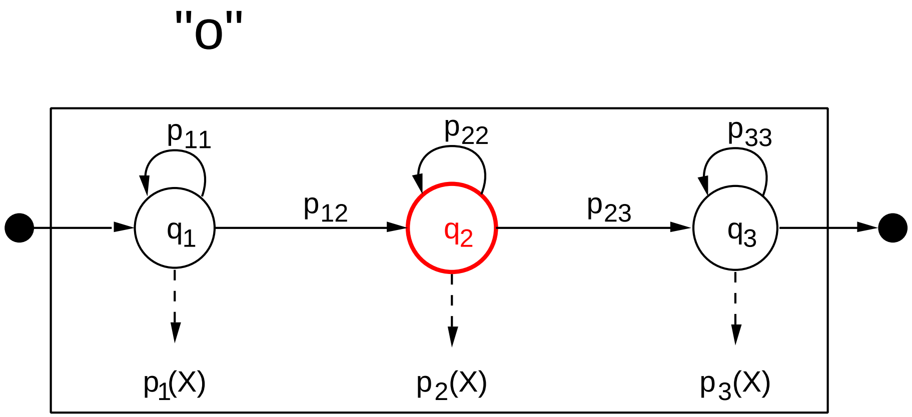


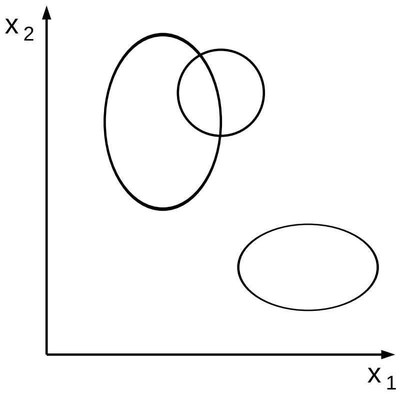


The Recognizer

Viterbi Decoding

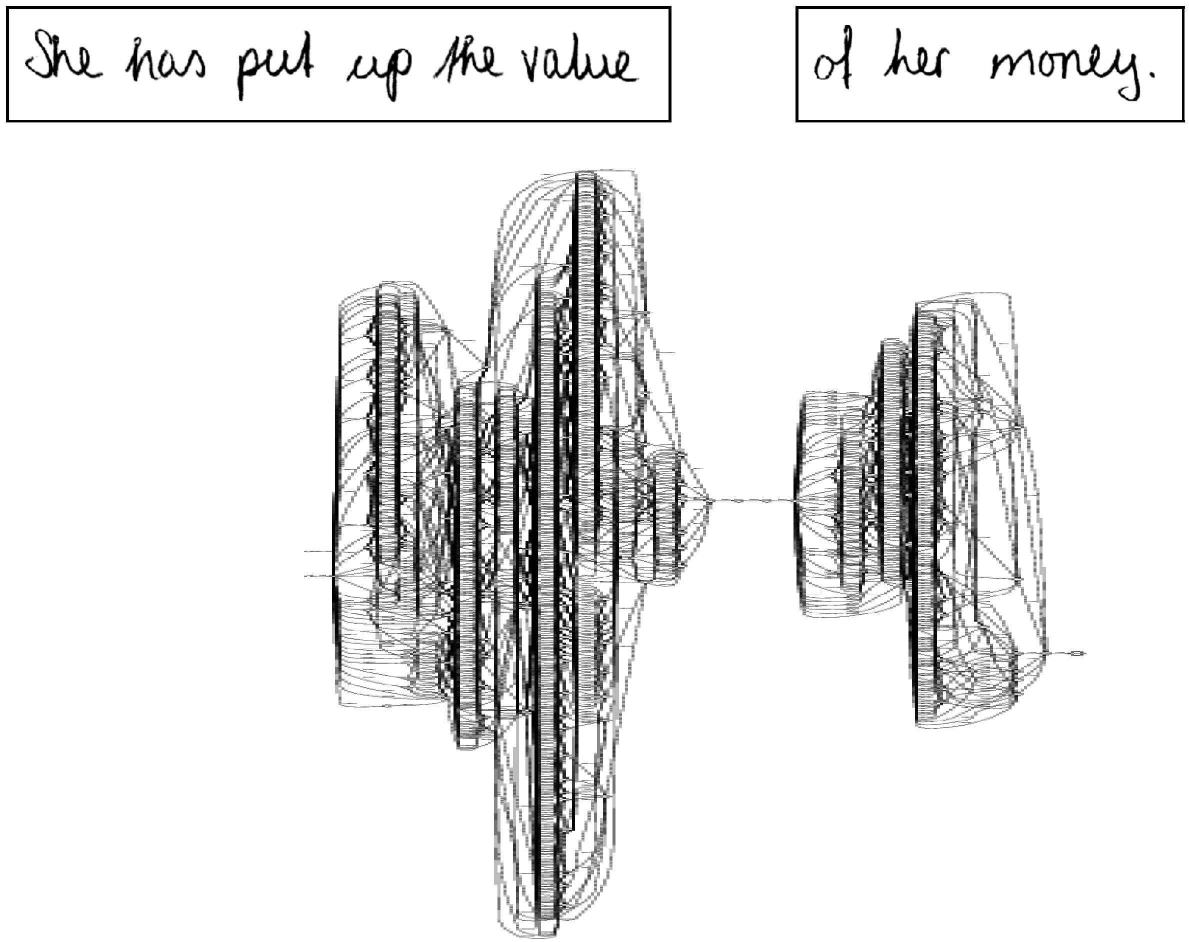
$$(X_1, X_2, {\color{red}X_3}, \ldots$$





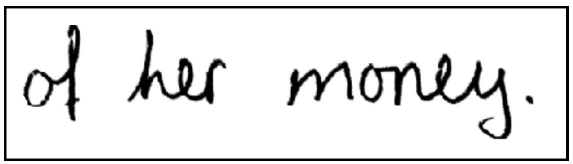
The Recognizer

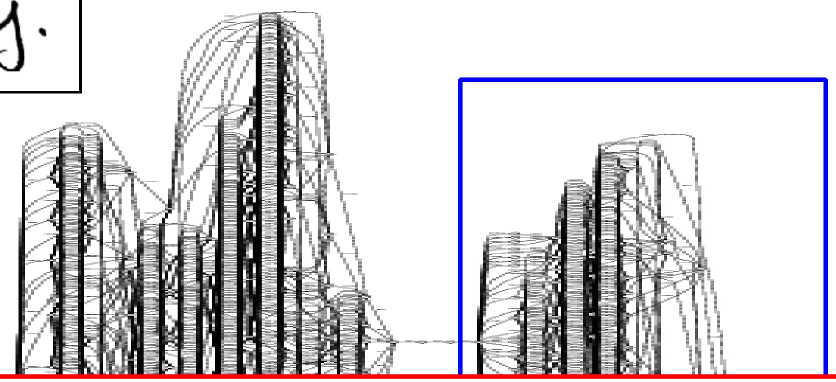
Recognition Lattice Generation

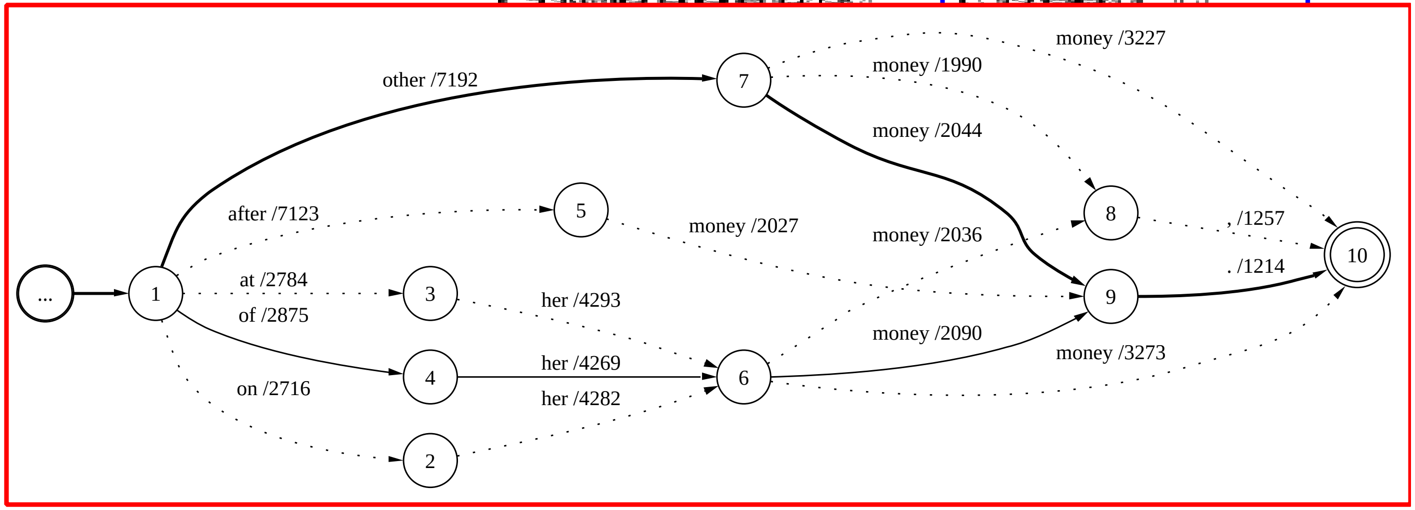


The Recognizer

Sublattice of 2nd Line

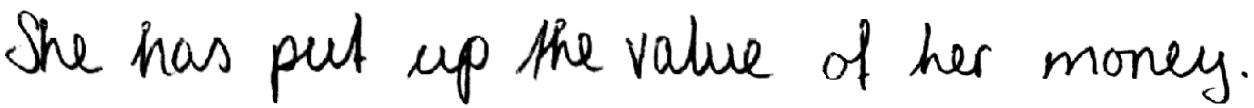






The Recognizer

N-Best List Generation



rank	score	candidate sentence
1	23923.6	She has put up the value other money .
2	23921.8	She has put up the value of her money .
3	23890.3	She had put up the value other money .
4	23888.4	She had put up the value of her money .
5	23854.3	She has put up the value at her money .
...	...	...

The Combination

Combining Recognizer and Parser

The Combination

Combining Recognizer and Parser

tight coupling

The Combination

Combining Recognizer and Parser

tight coupling

- parsing integrated in Viterbi decoding

The Combination

Combining Recognizer and Parser

tight coupling

- parsing integrated in Viterbi decoding
- only grammatically correct paths are expanded

The Combination

Combining Recognizer and Parser

tight coupling

- parsing integrated in Viterbi decoding
- only grammatically correct paths are expanded
- can be very time consuming

The Combination

Combining Recognizer and Parser

tight coupling

- parsing integrated in Viterbi decoding
- only grammatically correct paths are expanded
- can be very time consuming

sequential coupling

The Combination

Combining Recognizer and Parser

tight coupling

- parsing integrated in Viterbi decoding
- only grammatically correct paths are expanded
- can be very time consuming

sequential coupling

- parsing of output of recognizer

The Combination

Combining Recognizer and Parser

tight coupling

- parsing integrated in Viterbi decoding
- only grammatically correct paths are expanded
- can be very time consuming

sequential coupling

- parsing of output of recognizer
- simple to implement and test

The Combination

Combining Recognizer and Parser

tight coupling

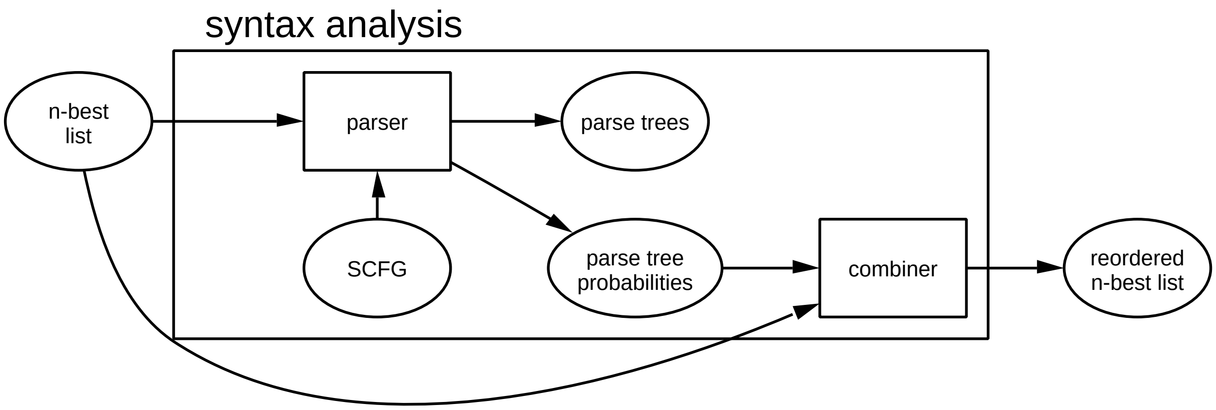
- parsing integrated in Viterbi decoding
- only grammatically correct paths are expanded
- can be very time consuming

sequential coupling

- parsing of output of recognizer
- simple to implement and test
- cannot recover from some errors

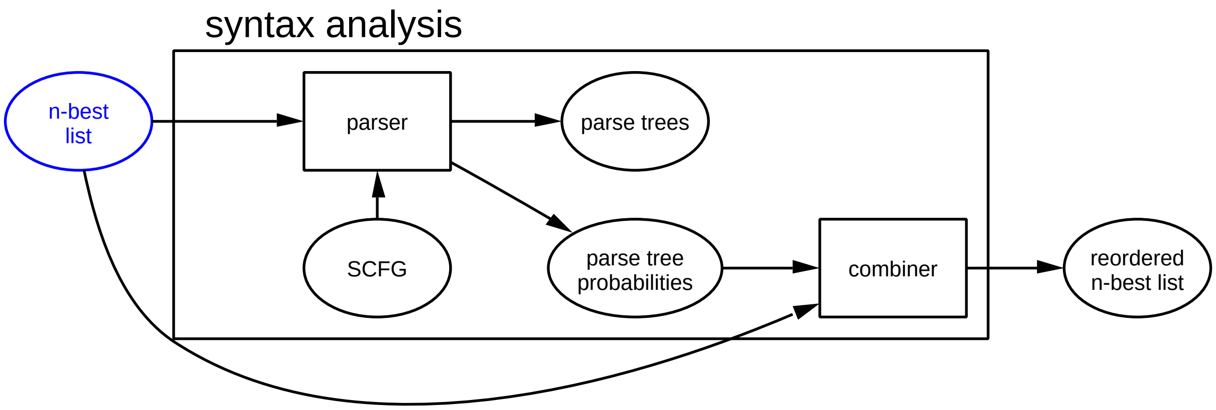
The Combination

Sequential Coupling



The Combination

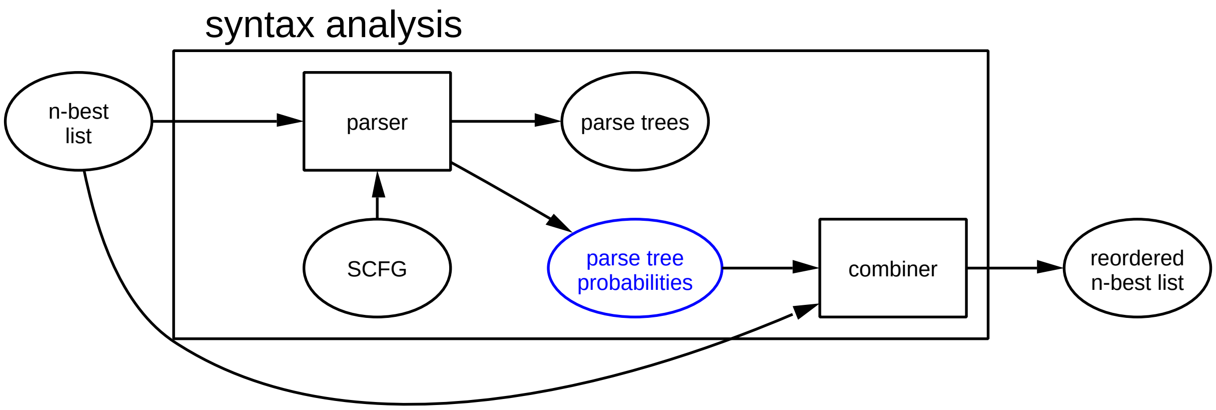
Sequential Coupling



rank	score	candidate sentence
1	23,924	She has put up the value other money .
2	23,922	She has put up the value of her money .
3	23,890	She had put up the value other money .
4	23,888	She had put up the value of her money .
5	23,854	She has put up the value at her money .

The Combination

Sequential Coupling



rank	score	candidate sentence	parse prob.
1	23,924	She has put up the value other money .	7.691e-23
2	23,922	She has put up the value of her money .	4.629e-20
3	23,890	She had put up the value other money .	2.631e-22
4	23,888	She had put up the value of her money .	1.584e-19
5	23,854	She has put up the value at her money .	1.125e-21

The Combination

Combination Scheme

The Combination

Combination Scheme

- recognition score is a log likelyhood

The Combination

Combination Scheme

- recognition score is a log likelyhood
- parse probability can be transformed into that

The Combination

Combination Scheme

- recognition score is a log likelyhood
- parse probability can be transformed into that
- a parse scale factor γ weights the influence of the parser

The Combination

Combination Scheme

- recognition score is a log likelyhood
- parse probability can be transformed into that
- a parse scale factor γ weights the influence of the parser
- a sentence score can be defined as follows:

$$sentence\ score = recognition\ score + \gamma \log(parse\ probability)$$

The Combination

Combination Scheme

- recognition score is a log likelyhood
- parse probability can be transformed into that
- a parse scale factor γ weights the influence of the parser
- a sentence score can be defined as follows:

$$sentence\ score = recognition\ score + \gamma \log(parse\ probability)$$

- the sentence score is used to reorder the n-best list

The Combination

Combination Scheme

- recognition score is a log likelyhood
- parse probability can be transformed into that
- a parse scale factor γ weights the influence of the parser
- a sentence score can be defined as follows:

$$sentence\ score = recognition\ score + \gamma \log(parse\ probability)$$

- the sentence score is used to reorder the n-best list
- the parse scale factor is optimized on a validation set

The Combination

An Example

parse scale factor $\gamma = 0$

score	pp	$\log$ pp	sen scr.	candidate sentence
23,924	7.691e-23	-22.33	23,924	She has put up the value other money .
23,922	4.629e-20	-19.33	23,922	She has put up the value of her money .
23,890	2.631e-22	-21.58	23,890	She had put up the value other money .
23,888	1.584e-19	-18.80	23,888	She had put up the value of her money .
23,854	1.125e-21	-20.95	23,854	She has put up the value at her money .

- $\gamma = 0$: original system, parsing has no effect

The Combination

An Example

parse scale factor $\gamma = 10$

score	pp	$\log$ pp	sen scr.	candidate sentence
23,924	7.691e-23	-22.33	23,701	She has put up the value other money .
23,922	4.629e-20	-19.33	23,729	She has put up the value of her money .
23,890	2.631e-22	-21.58	23,674	She had put up the value other money .
23,888	1.584e-19	-18.80	23,700	She had put up the value of her money .
23,854	1.125e-21	-20.95	23,644	She has put up the value at her money .

- $\gamma = 0$: original system, parsing has no effect
- $\gamma = 10$: desired effect of syntax analysis

The Combination

An Example

parse scale factor $\gamma = 20$

score	pp	$\log$ pp	sen scr.	candidate sentence
23,924	7.691e-23	-22.33	23,478	She has put up the value other money .
23,922	4.629e-20	-19.33	23,536	She has put up the value of her money .
23,890	2.631e-22	-21.58	23,458	She had put up the value other money .
23,888	1.584e-19	-18.80	23,512	She had put up the value of her money .
23,854	1.125e-21	-20.95	23,435	She has put up the value at her money .

- $\gamma = 0$: original system, parsing has no effect
- $\gamma = 10$: desired effect of syntax analysis
- $\gamma = 20$: increasing influence of parsing

Organization

1. Introduction
2. Problem Statement
3. The Grammar
4. The Parser
5. IAM Database and Recognizer
6. The Combination
7. Setup, Optimization and Evaluation
8. Conclusions and Outlook

Experimental Setup

Experimental Setup

Experimental Setup

Experimental Setup

assumptions

Experimental Setup

Experimental Setup

assumptions

- segmented sentences

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

training of character HMM

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

training of character HMM

- 5,799 lines of handwritten text

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

training of character HMM

- 5,799 lines of handwritten text
- including 33,993 words

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

training of character HMM

- 5,799 lines of handwritten text
- including 33,993 words
- written by 448 writers

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

training of character HMM

- 5,799 lines of handwritten text
- including 33,993 words
- written by 448 writers

tasks for optimizations and evaluation

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

training of character HMM

- 5,799 lines of handwritten text
- including 33,993 words
- written by 448 writers

tasks for optimizations and evaluation

- multi writer task (MW)

Experimental Setup

Experimental Setup

assumptions

- segmented sentences
- closed vocabulary

training of character HMM

- 5,799 lines of handwritten text
- including 33,993 words
- written by 448 writers

tasks for optimizations and evaluation

- multi writer task (MW)
- writer independent task (WI)

Experimental Setup

Task Specifications

	multi writer task (MW)
writers	
sentences	
text lines	
words	
lexicon	
SCFG rules	

- MW: same population of writers

Experimental Setup

Task Specifications

multi writer task (MW)

	validation
writers	157
sentences	200
text lines	534
words	3,814
lexicon	8,824
SCFG rules	21,691

- MW: same population of writers

Experimental Setup

Task Specifications

multi writer task (MW)

	validation	test
writers	157	156
sentences	200	200
text lines	534	575
words	3,814	3,933
lexicon	8,824	8,822
SCFG rules	21,691	21,716

- MW: same population of writers

Experimental Setup

Task Specifications

multi writer task (MW) writer independent task (WI)

	validation	test
writers	157	156
sentences	200	200
text lines	534	575
words	3,814	3,933
lexicon	8,824	8,822
SCFG rules	21,691	21,716

- MW: same population of writers
- WI: different writers in training, validation and test set

Experimental Setup

Task Specifications

	multi writer task (MW)		writer independent task (WI)
	validation	test	validation
writers	157	156	100
sentences	200	200	200
text lines	534	575	582
words	3,814	3,933	4,094
lexicon	8,824	8,822	8,827
SCFG rules	21,691	21,716	21,705

- MW: same population of writers
- WI: different writers in training, validation and test set

Experimental Setup

Task Specifications

	multi writer task (MW)		writer independent task (WI)	
	validation	test	validation	test
writers	157	156	100	100
sentences	200	200	200	200
text lines	534	575	582	575
words	3,814	3,933	4,094	3,956
lexicon	8,824	8,822	8,827	8,821
SCFG rules	21,691	21,716	21,705	21,694

- MW: same population of writers
- WI: different writers in training, validation and test set

Experimental Setup

Performance Measures

Experimental Setup

Performance Measures

- sentence recognition rate

Experimental Setup

Performance Measures

- sentence recognition rate
- text line recognition rate

Experimental Setup

Performance Measures

- sentence recognition rate
- text line recognition rate
- word recognition rate

$$word\ recognition\ rate = \frac{H}{N} \times 100\%$$

Experimental Setup

Performance Measures

- sentence recognition rate
- text line recognition rate
- word recognition rate

$$word\ recognition\ rate = \frac{H}{N} \times 100\%$$

- word level accuracy

$$word\ level\ accuracy = \frac{H - I}{N} \times 100\%$$

System Optimization

Different Aspects of the Optimization

System Optimization

Different Aspects of the Optimization

- number of states per character model

System Optimization

Different Aspects of the Optimization

- number of states per character model
- number of Gaussians per model state

System Optimization

Different Aspects of the Optimization

- number of states per character model
- number of Gaussians per model state
- number of training iterations for the HMM

System Optimization

Different Aspects of the Optimization

- number of states per character model
- number of Gaussians per model state
- number of training iterations for the HMM
- integration of the bigram language model

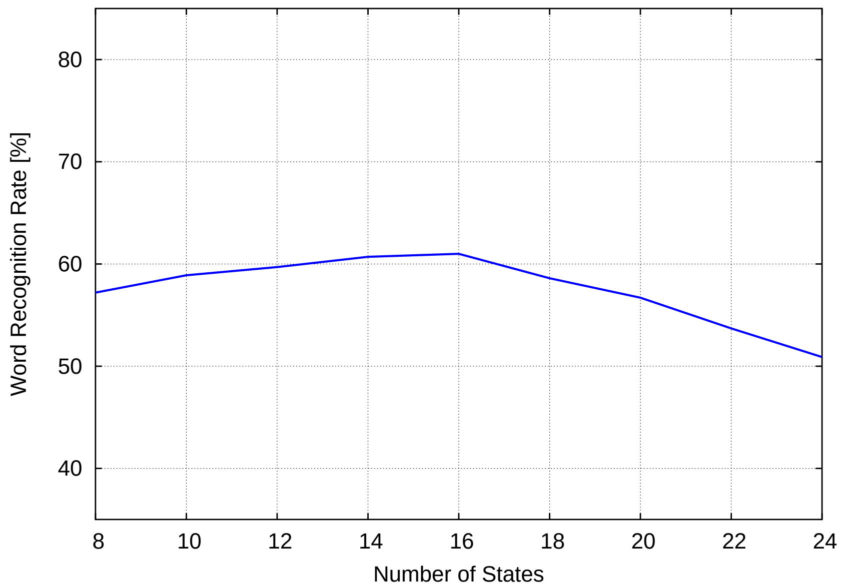
System Optimization

Different Aspects of the Optimization

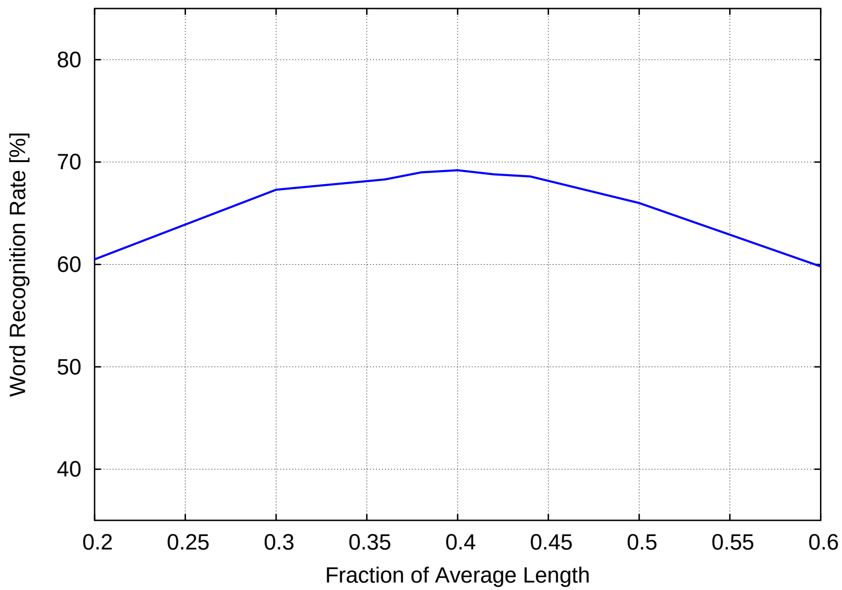
- number of states per character model
- number of Gaussians per model state
- number of training iterations for the HMM
- integration of the bigram language model
- parse scale factor optimization

System Optimization

Optimizing the Number of Model States



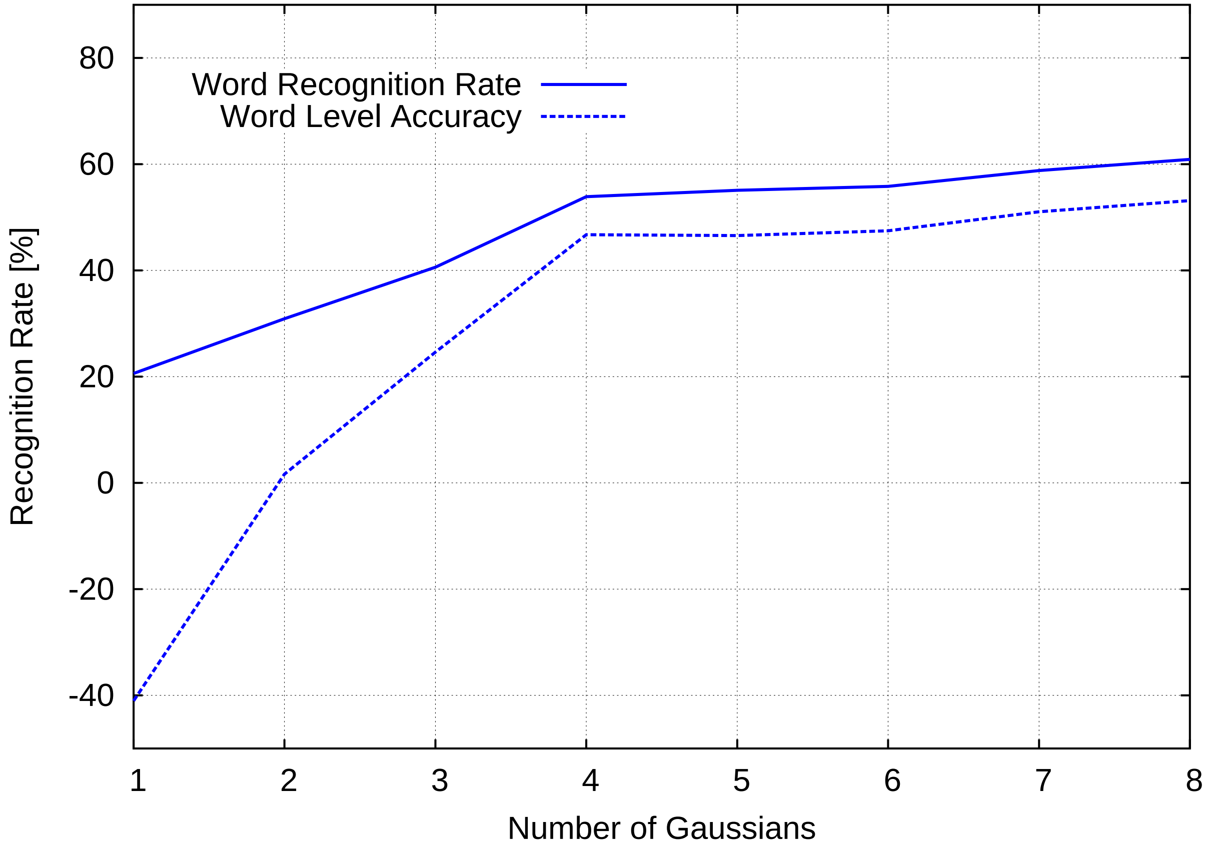
fixed length modeling



variable length modeling

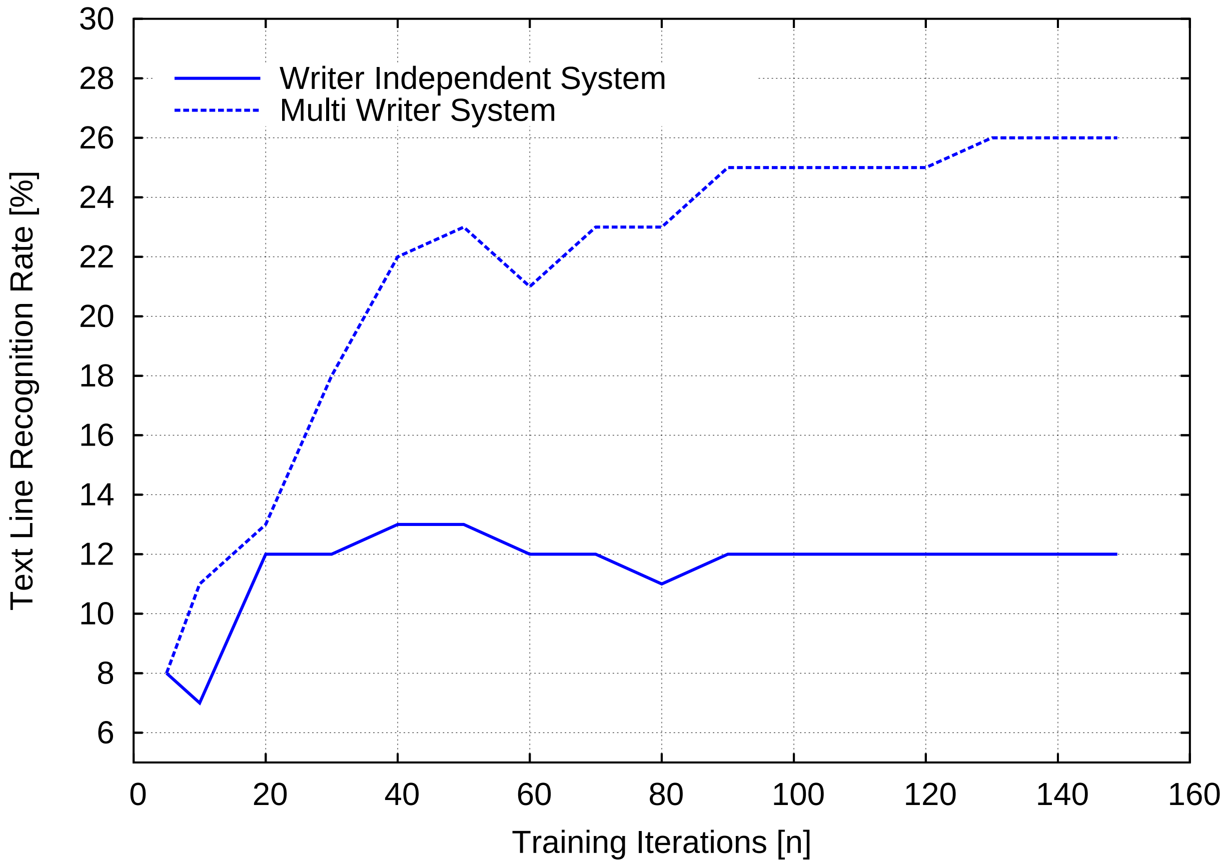
System Optimization

Optimizing the Number of Gaussians



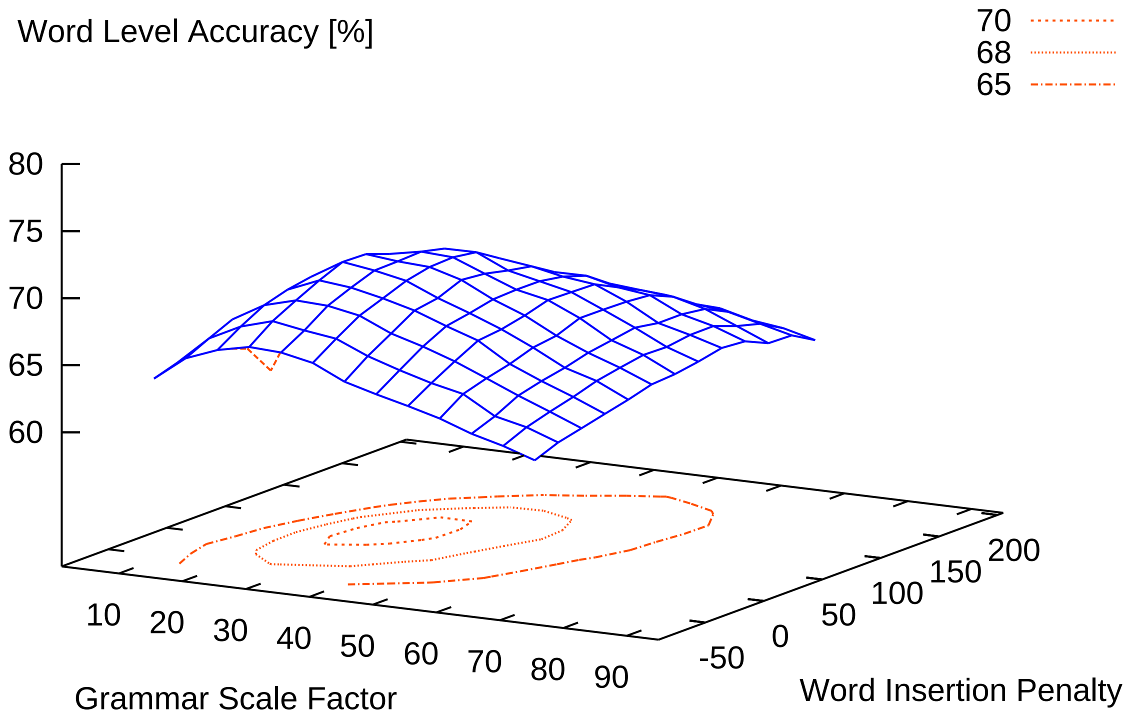
System Optimization

Optimizing the Number of Training Iterations



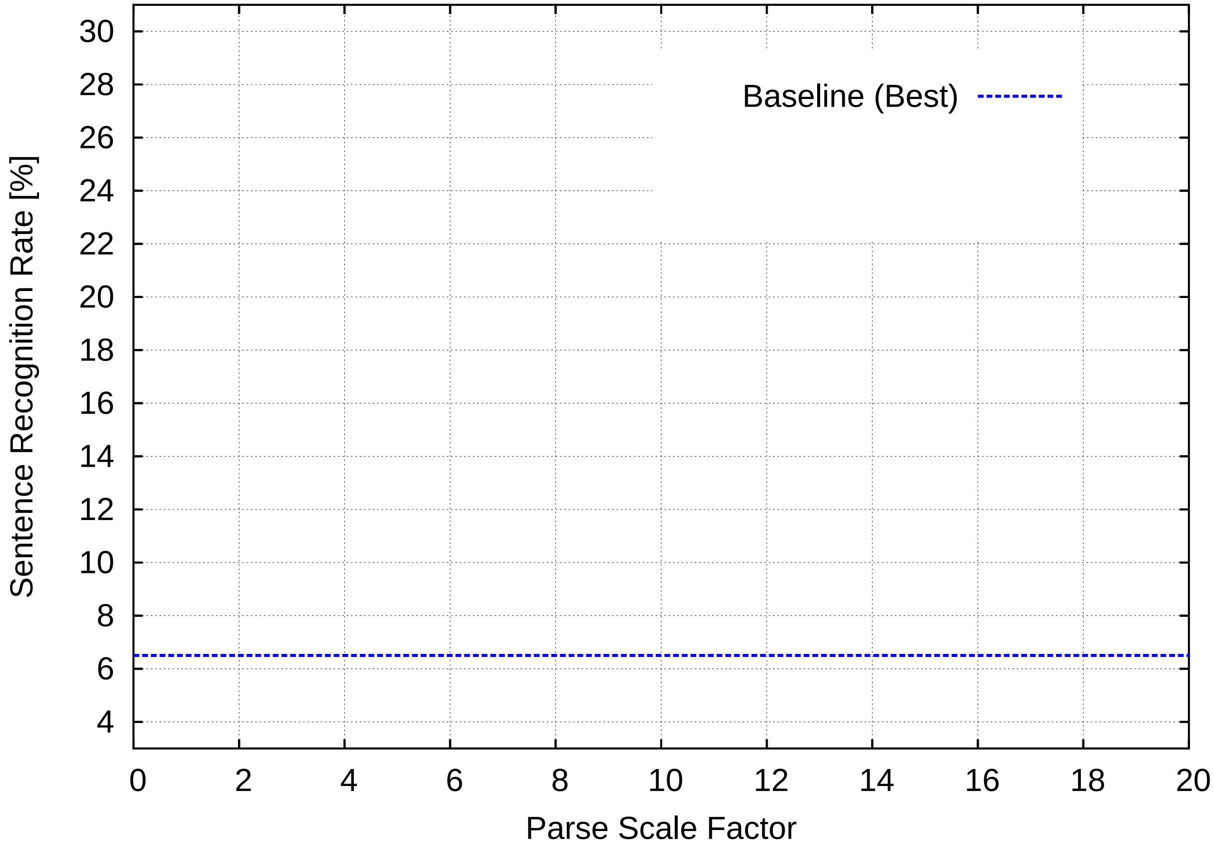
System Optimization

Optimizing the Language Model Integraion



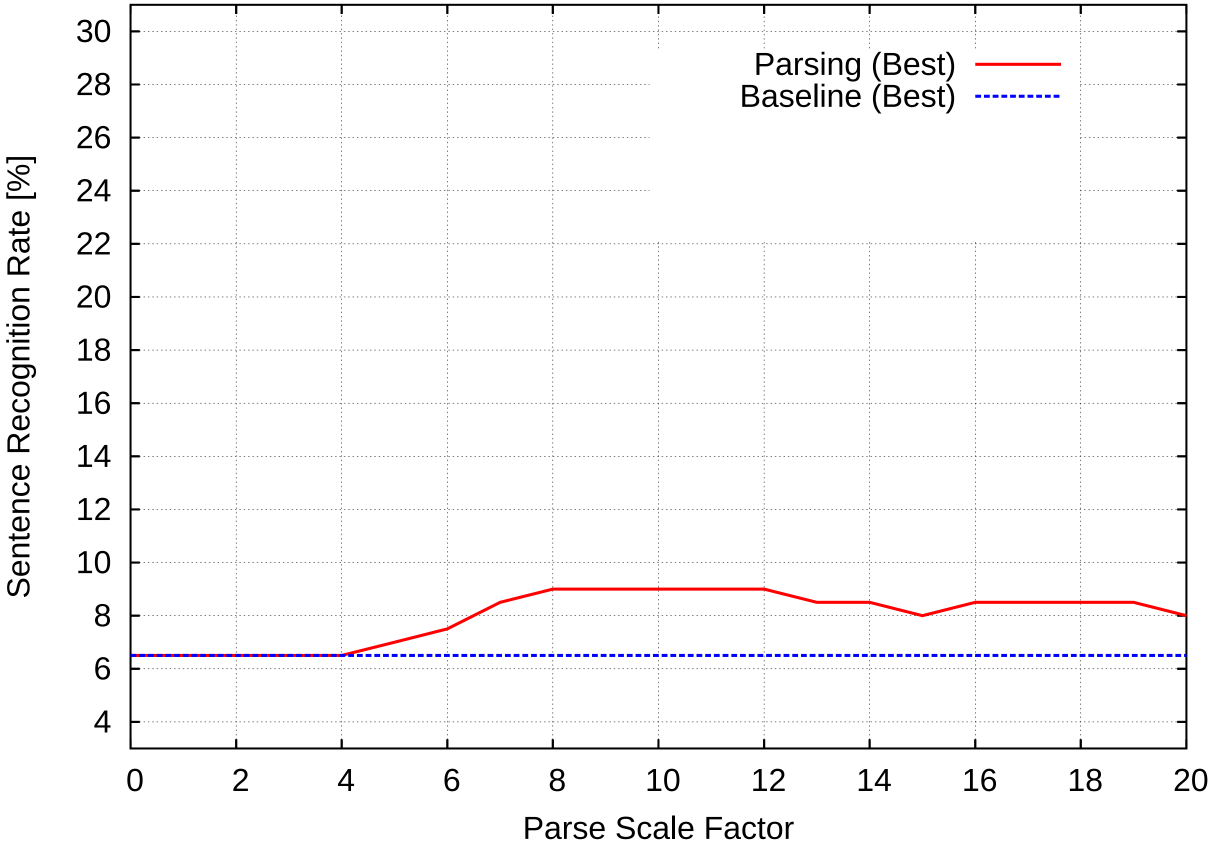
System Optimization

Optimizing the Parse Scale Factor



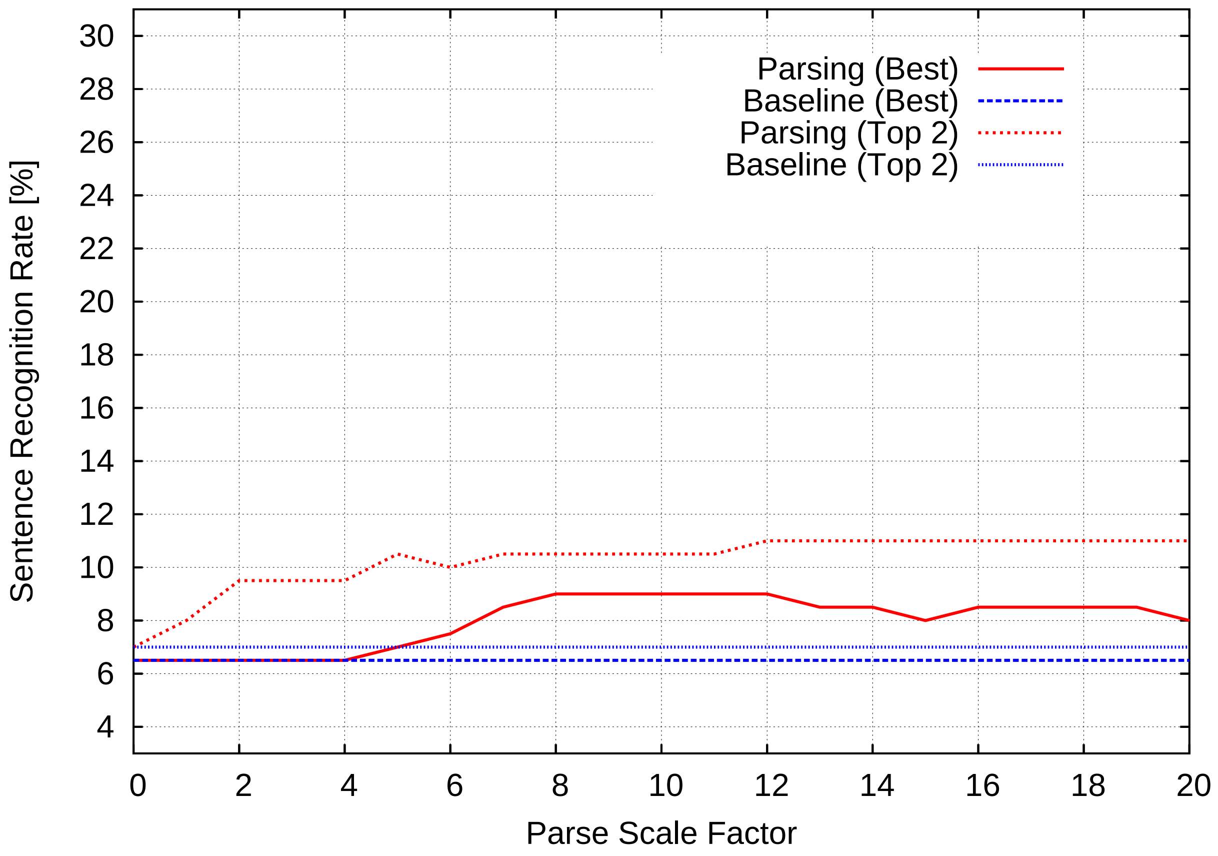
System Optimization

Optimizing the Parse Scale Factor



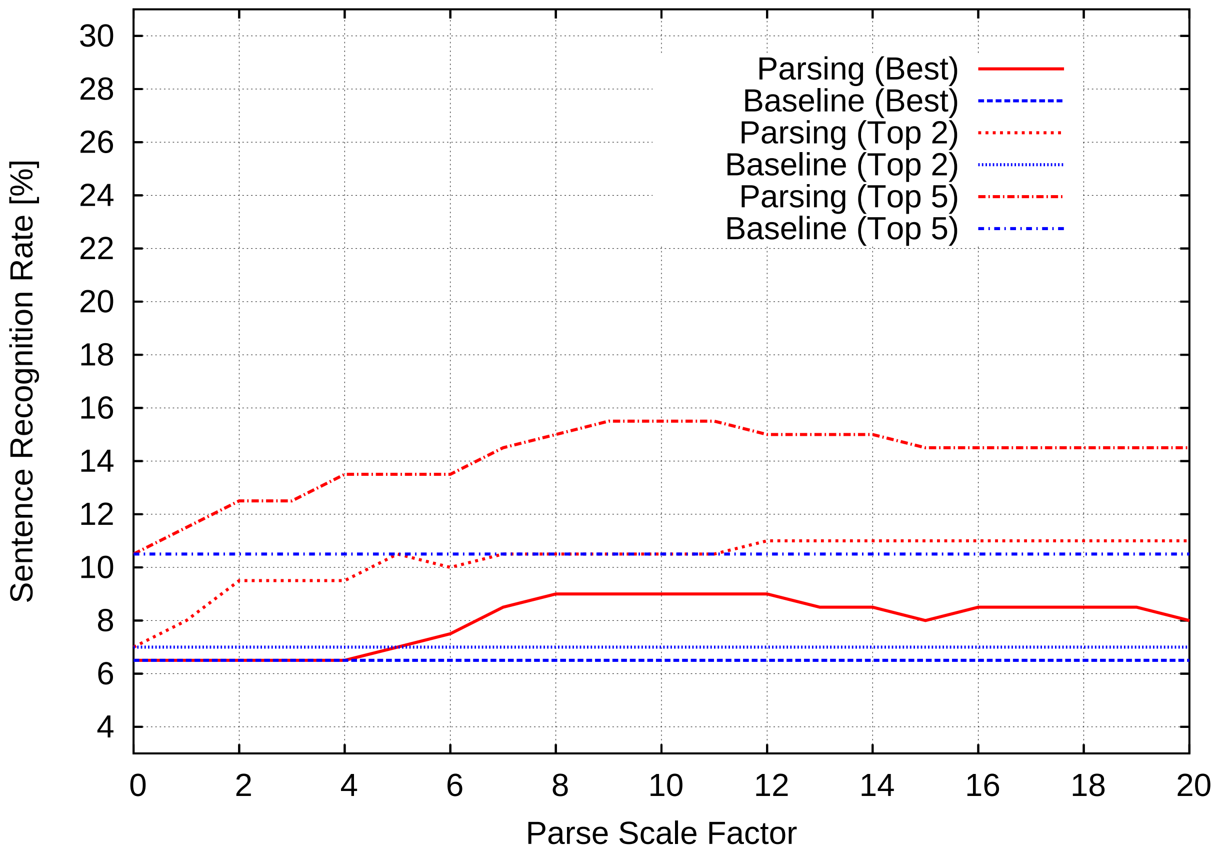
System Optimization

Optimizing the Parse Scale Factor



System Optimization

Optimizing the Parse Scale Factor



System Evaluation

Test Set Results

System Evaluation

Test Set Results

multi writer task

performance measure	baseline	parsing	level
sentence recognition rate	7.5%	8.0%	60%
word recognition rate	76.7%	77.2%	75%
word level accuracy	74.7%	75.6%	$>$ 99%

System Evaluation

Test Set Results

multi writer task

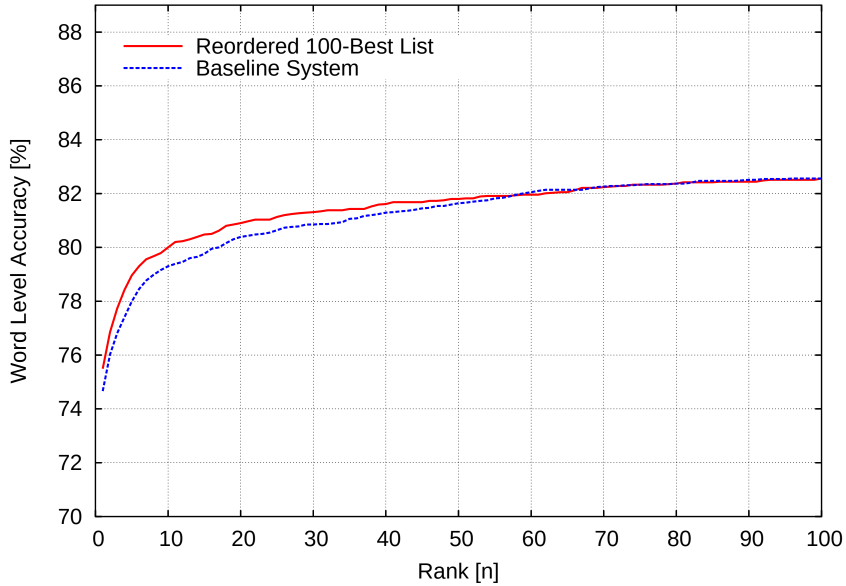
performance measure	baseline	parsing	level
sentence recognition rate	7.5%	8.0%	60%
word recognition rate	76.7%	77.2%	75%
word level accuracy	74.7%	75.6%	$>$ 99%

writer independent task

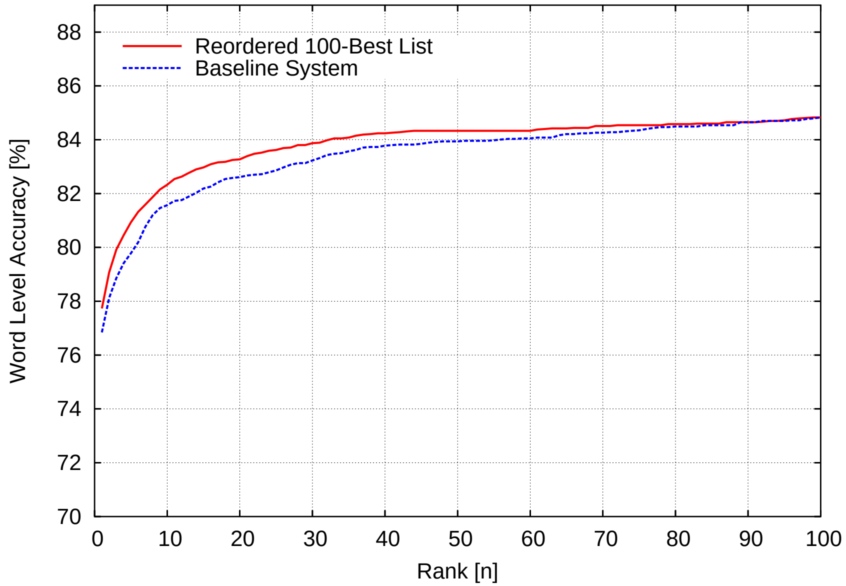
performance measure	baseline	parsing	level
sentence recognition rate	11.0%	12.0%	70%
word recognition rate	79.3%	79.4%	50%
word level accuracy	76.8%	77.6%	91%

System Evaluation

N-best Analysis of Word Level Accuracy



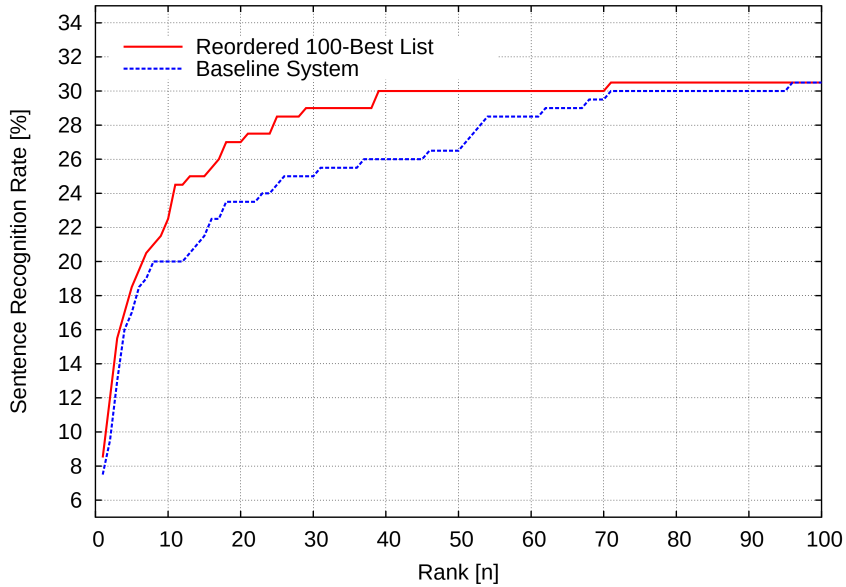
multi writer task



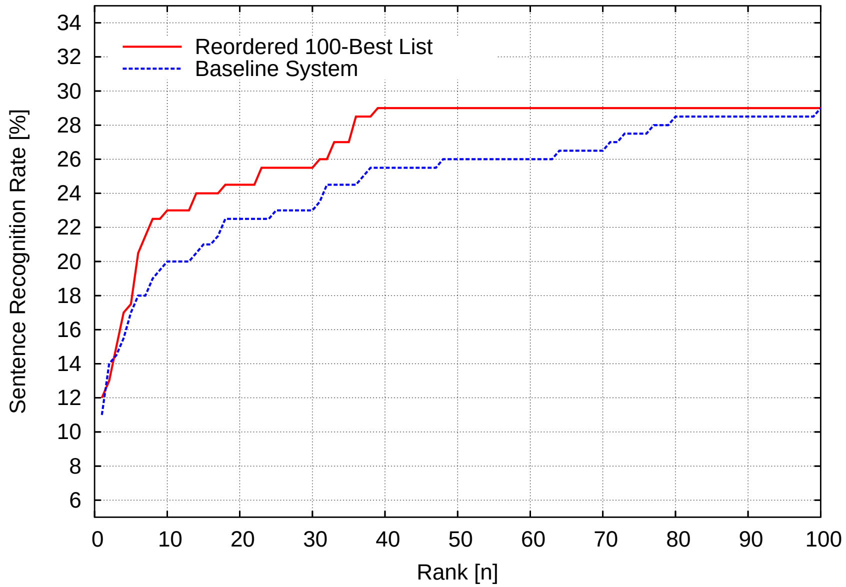
writer independent task

System Evaluation

N-best Analysis of Sentence Recognition Rate



multi writer task



writer independent task

Conclusions

Main Conclusions

Conclusions

Main Conclusions

- substantial improvements of the available recognizer achieved

Conclusions

Main Conclusions

- substantial improvements of the available recognizer achieved
- a SCFG for general English text was extracted

Conclusions

Main Conclusions

- substantial improvements of the available recognizer achieved
- a SCFG for general English text was extracted
- the sequential coupling proved to be effective

Conclusions

Main Conclusions

- substantial improvements of the available recognizer achieved
- a SCFG for general English text was extracted
- the sequential coupling proved to be effective
- syntax analysis can help handwriting recognition

Outlook

Future Directions

Outlook

Future Directions

- further improve recognizer

Outlook

Future Directions

- further improve recognizer
- drop closed vocabulary assumption

Outlook

Future Directions

- further improve recognizer
- drop closed vocabulary assumption
- add reject management to recognizer

Outlook

Future Directions

- further improve recognizer
- drop closed vocabulary assumption
- add reject management to recognizer
- enhance parser to produce n-most probable parses

Outlook

Future Directions

- further improve recognizer
- drop closed vocabulary assumption
- add reject management to recognizer
- enhance parser to produce n-most probable parses
- test alternative SCFG