Trade-off between incrementality and accuracy

Seminar on Incremental Processing

Benjamin Weitz

June 30, 2011

Outline

Introduction

Incremental Dialogue Systems:

Introduction

Incremental Dialogue Systems:

- fast(er)

Introduction

Incremental Dialogue Systems:

- fast(er)
- sometimes wrong decisions

Introduction

Incremental Dialogue Systems:

- fast(er)
- sometimes wrong decisions $\Rightarrow$ revisions

Introduction

Incremental Dialogue Systems:

- fast(er)
- sometimes wrong decisions $\Rightarrow$ revisions
- components depend on each other

Introduction

Incremental Dialogue Systems:

- fast(er)
- sometimes wrong decisions $\Rightarrow$ revisions
- components depend on each other

Idea: decrease incrementality a bit to reduce revisions

Introduction

Incremental Dialogue Systems:

- fast(er)
- sometimes wrong decisions $\Rightarrow$ revisions
- components depend on each other

Idea: decrease incrementality a bit to reduce revisions

- Baumann et. al: Automatic Speech Recognition

Introduction

Incremental Dialogue Systems:

- fast(er)
- sometimes wrong decisions $\Rightarrow$ revisions
- components depend on each other

Idea: decrease incrementality a bit to reduce revisions

- Baumann et. al: Automatic Speech Recognition
- Kato et. al: Parsing

Baumann et al.

Baumann et al.

- Define measures to evaluate incremental ASR-Systems

Baumann et al.

- Define measures to evaluate incremental ASR-Systems
- Evaluate an existing system with these measures

Baumann et al.

- Define measures to evaluate incremental ASR-Systems
- Evaluate an existing system with these measures
- Use the measures to improve ASR

Baumann et al.

- Define measures to evaluate incremental ASR-Systems
- Evaluate an existing system with these measures
- Use the measures to improve ASR
- Conclusions

Baumann et al.

- Define measures to evaluate incremental ASR-Systems
- Evaluate an existing system with these measures
- Use the measures to improve ASR
- Conclusions

Definitions

Hypothesis at time t:

$hyp_t = w_{hyp_t}$

Definitions

Hypothesis at time t:

$hyp_t = w_{hyp\,t}$

How to evaluate the quality of $\boldsymbol{hyp_t}$?

use **actually spoken input** as gold standard

use **final hypothesis of ASR** as gold standard

Definitions

Hypothesis at time t:

$hyp_t = w_{hyp\,t}$

How to evaluate the quality of hyp_t?

use **actually spoken input** as gold standard

use **final hypothesis of ASR** as gold standard

Definitions

Hypothesis at time t:

$hyp_t = w_{hyp\,t}$

How to evaluate the quality of hyp_t?

use **actually spoken input** as gold standard

use **final hypothesis of ASR** as gold standard

Why?

more meaningful:

- relates partial hypothesis to what can be expected from ASR
- correct interpretation might never be recognized

Relative Correctness

w_{gold}: final, non incremental hypothesis

Relative Correctness

w_{gold}: final, non incremental hypothesis

Relative Correctness

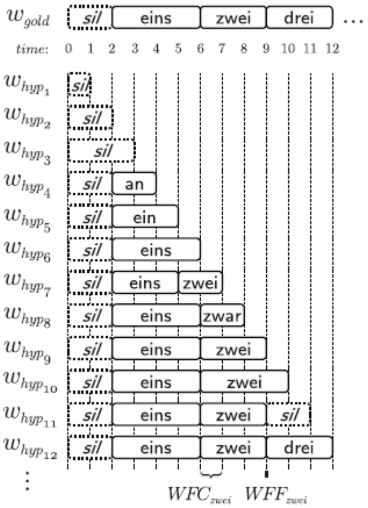
$w_{hyp\,t}$ is *relatively correct*, **iff** $w_{hyp\,t} = w_{gold\,t}$

Relative Correctness

w_{gold}: final, non incremental hypothesis

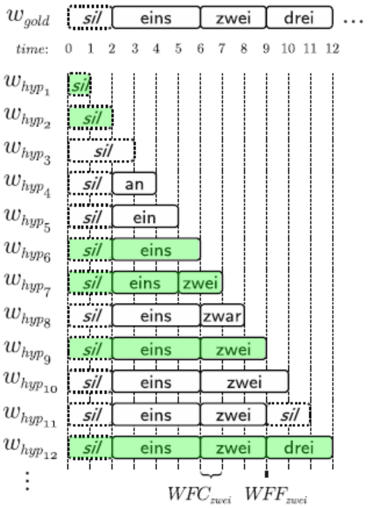
Relative Correctness

$w_{hyp\,t}$ is *relatively correct*, **iff** $w_{hyp\,t} = w_{gold\,t}$



Relative Correctness

w_{gold}: final, non incremental hypothesis



Relative Correctness

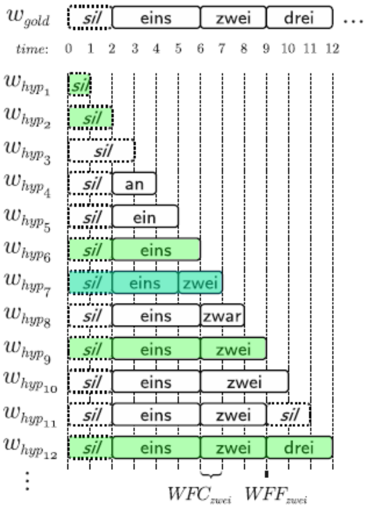
w_{hyp_t} is *relatively correct*, **iff**
$w_{hyp_t} = w_{gold_t}$

Relative Correctness

w_{gold}: final, non incremental hypothesis

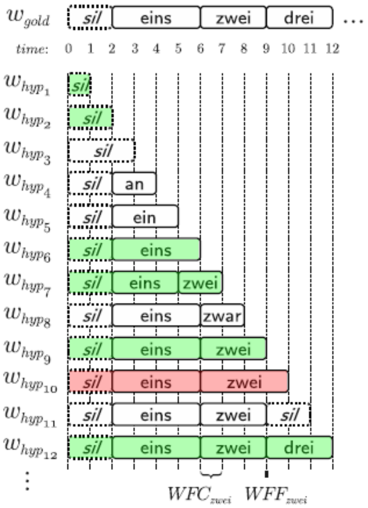
Relative Correctness

$w_{hyp\,t}$ is *relatively correct*, **iff**
$w_{hyp\,t} = w_{gold\,t}$



Relative Correctness

w_{gold}: final, non incremental hypothesis



Relative Correctness

$w_{hyp\,t}$ is *relatively correct*, **iff** $w_{hyp\,t} = w_{gold\,t}$

Relative Correctness

w_{gold}: final, non incremental hypothesis

Relative Correctness

w_{hyp_t} is *relatively correct*, **iff** $w_{hyp_t} = w_{gold_t}$

Prefix Correctness

w_{hyp_t} is *prefix-correct*, **iff** w_{hyp_t} is a prefix of w_{gold_t}

Relative Correctness

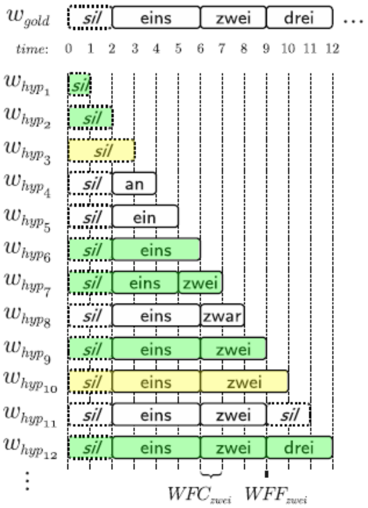
w_{gold}: final, non incremental hypothesis

Relative Correctness

$w_{hyp\,t}$ is *relatively correct*, **iff** $w_{hyp\,t} = w_{gold\,t}$

Prefix Correctness

$w_{hyp\,t}$ is *prefix-correct*, **iff** $w_{hyp\,t}$ is a prefix of $w_{gold\,t}$



Edit Overhead

Three ways for hyp_{t+1} to differ from hyp_t:

- extension
- revokation
- revision

Edit Overhead

Three ways for hyp_{t+1} to differ from hyp_t:

- extension
- revokation
- revision

Two types of *edit messages*:

- *add* message: $\oplus$
- *revoke* message: $\ominus$

Edit Overhead

Three ways for hyp_{t+1} to differ from hyp_t:

- extension: $\oplus$
- revokation
- revision

Two types of *edit messages*:

- *add* message: $\oplus$
- *revoke* message: $\ominus$

Edit Overhead

Three ways for hyp_{t+1} to differ from hyp_t:

- extension: $\oplus$
- revokation: $\ominus$
- revision

Two types of *edit messages*:

- *add* message: $\oplus$
- *revoke* message: $\ominus$

Edit Overhead

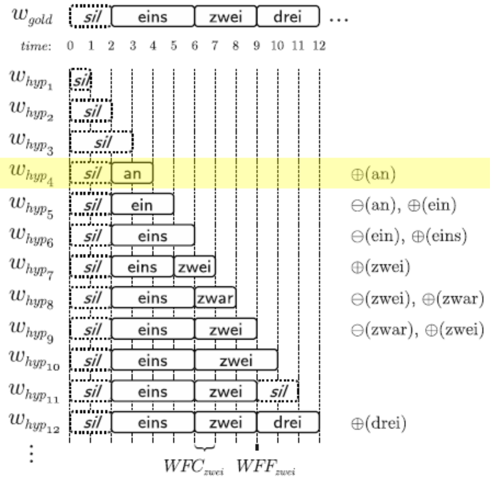
Three ways for hyp_{t+1} to differ from hyp_t:

- extension: $\oplus$
- revokation: $\ominus$
- revision: $\ominus$, $\oplus$

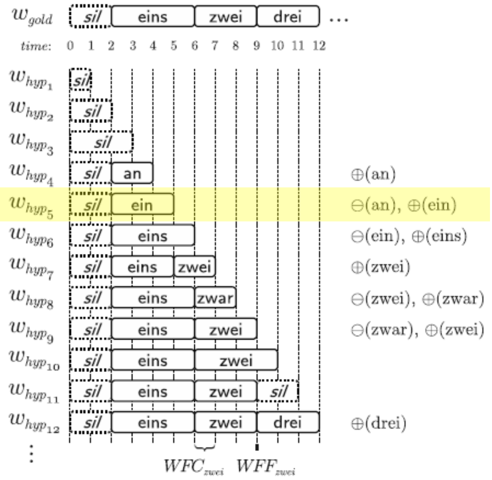
Two types of *edit messages*:

- *add* message: $\oplus$
- *revoke* message: $\ominus$

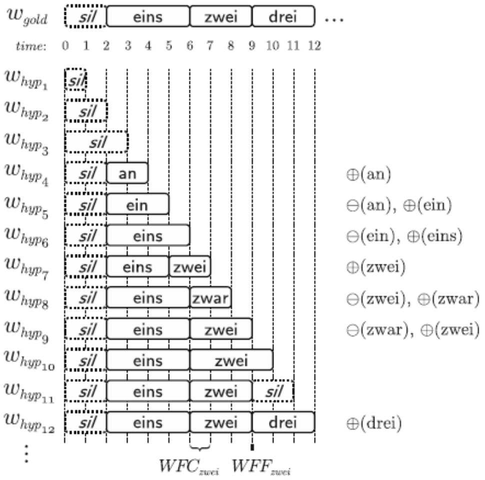
Edit Overhead



Edit Overhead



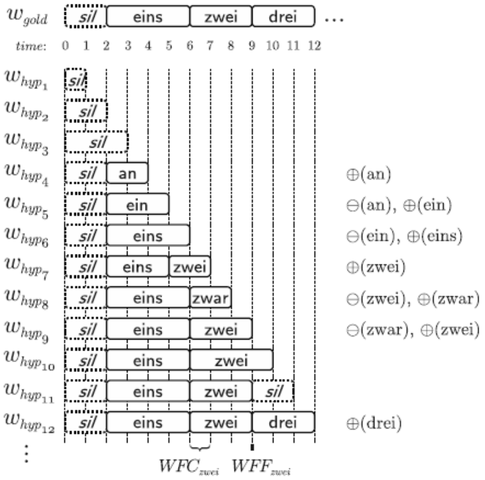
Edit Overhead



Perfect ASR-System:

1 extension for each word

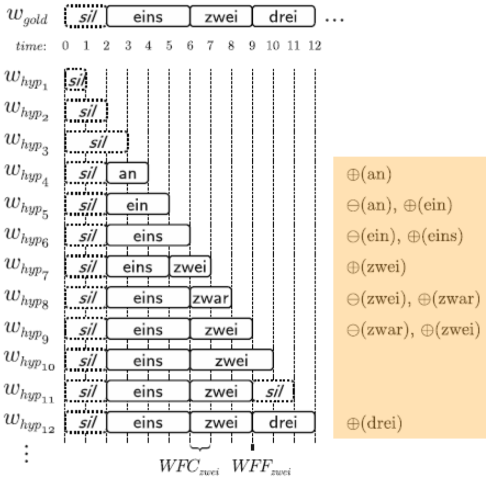
Edit Overhead



Perfect ASR-System:
1 extension for each word

Edit Overhead (EO)
rate of spurious edits

Edit Overhead



Perfect ASR-System:

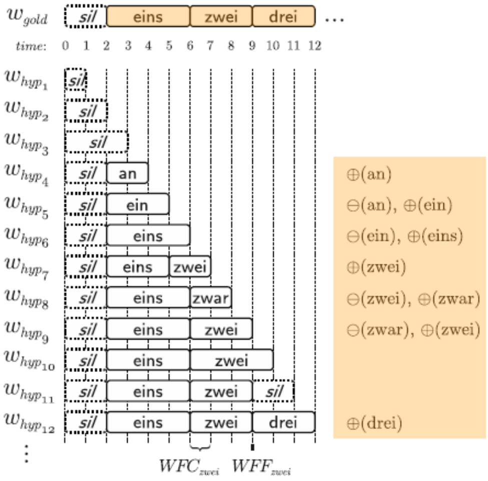
1 extension for each word

Edit Overhead (EO)

rate of spurious edits

11 edits

Edit Overhead



Perfect ASR-System:

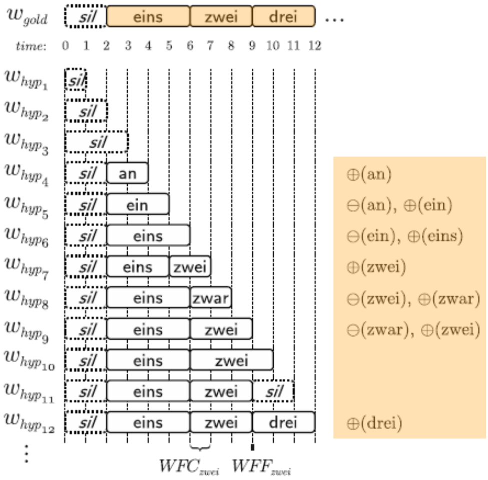
1 extension for each word

Edit Overhead (EO)

rate of spurious edits

11 edits, 3 words

Edit Overhead



Perfect ASR-System:

1 extension for each word

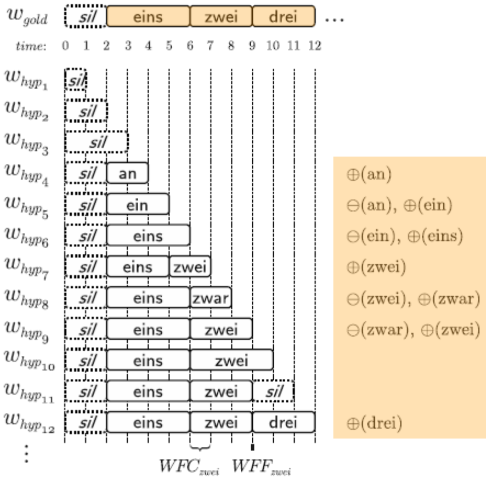
Edit Overhead (EO)

rate of spurious edits

11 edits, 3 words

$\Rightarrow$ 8 unnecessary edits

Edit Overhead



Perfect ASR-System:
1 extension for each word

Edit Overhead (EO)
rate of spurious edits

11 edits, 3 words
$\Rightarrow$ 8 unnecessary edits
$\Rightarrow EO = \frac{8}{11}$

Timing Measures

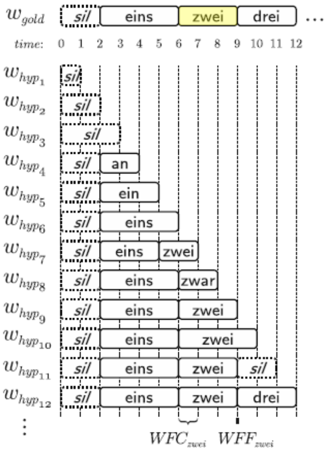
Word First Correct Response (WFC)

The first time a word appears in the correct position

Timing Measures

Word First Correct Response (WFC)

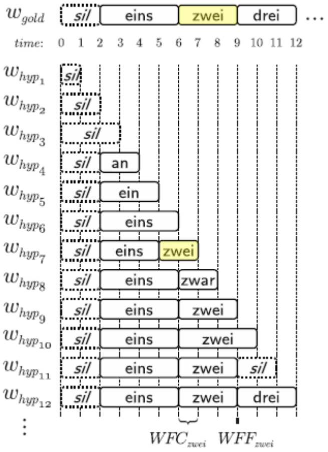
The first time a word appears in the correct position



Timing Measures

Word First Correct Response (WFC)

The first time a word appears in the correct position



$WFC_{zwei} = 7$

Timing Measures

Word First Correct Response (WFC)

The first time a word appears in the correct position

Word First Final Response (WFF)

The time a hypothesis remains stable / doesn't change anymore

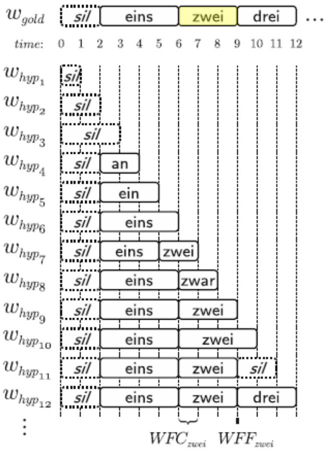
Timing Measures

Word First Correct Response (WFC)

The first time a word appears in the correct position

Word First Final Response (WFF)

The time a hypothesis remains stable / doesn't change anymore



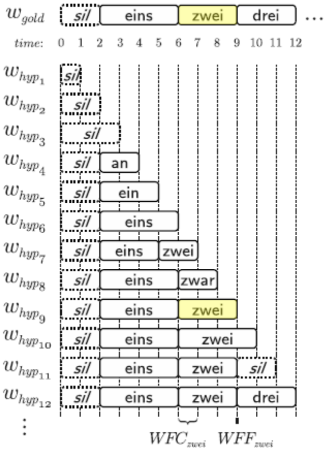
Timing Measures

Word First Correct Response (WFC)

The first time a word appears in the correct position

Word First Final Response (WFF)

The time a hypothesis remains stable / doesn't change anymore



$$WFF_{zwei} = 9$$

Timing Measures

Word First Correct Response (WFC)

The first time a word appears in the correct position

Word First Final Response (WFF)

The time a hypothesis remains stable / doesn't change anymore

Correction Time (CT)

$CT = WFF - WFC$

Timing Measures

Word First Correct Response (WFC)

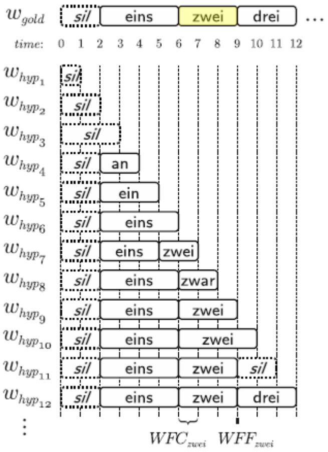
The first time a word appears in the correct position

Word First Final Response (WFF)

The time a hypothesis remains stable / doesn't change anymore

Correction Time (CT)

$CT = WFF - WFC$



Timing Measures

Word First Correct Response (WFC)

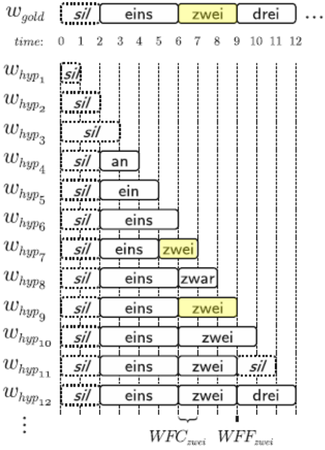
The first time a word appears in the correct position

Word First Final Response (WFF)

The time a hypothesis remains stable / doesn't change anymore

Correction Time (CT)

$CT = WFF - WFC$



$$CT_{zwei} = 9 - 7 = 2$$

Baumann et al.

- Define measures to evaluate incremental ASR-Systems
- Evaluate an existing system with these measures
- Use the measures to improve ASR
- Conclusions

Setup and Data

- continious speech framework *Sphinx-4*
- acoustic model
  - German
  - instructions in a puzzle domain
- trigram language model
- test data
  - 85 recodings
  - two speakers
  - sentence similar to training sentences

Measurements

SER	68.2%
WER	18.8%
r-correct	30.9%
p-correct	53.1%
edit overhead	90.5%
mean word duration	0.378 s
WFC	mean: 0.276 s
WFF	mean: 0.004 s
immediately correct	58.6%

Measurements

SER	68.2%
WER	18.8%
r-correct	30.9%
p-correct	53.1%
edit overhead	90.5%
mean word duration	0.378 s
WFC	mean: 0.276 s
WFF	mean: 0.004 s
immediately correct	58.6%

- rather low correctness

Measurements

SER	68.2%
WER	18.8%
r-correct	30.9%
p-correct	53.1%
edit overhead	90.5%
mean word duration	0.378 s
WFC	mean: 0.276 s
WFF	mean: 0.004 s
immediately correct	58.6%

- rather low correctness
- very high edit overhead!

Measurements

SER	68.2%
WER	18.8%
r-correct	30.9%
p-correct	53.1%
edit overhead	90.5%
mean word duration	0.378 s
WFC	mean: 0.276 s
WFF	mean: 0.004 s
immediately correct	58.6%

- rather low correctness
- very high edit overhead!
- information available after $\frac{3}{4}$ have been spoken

Measurements

SER	68.2%
WER	18.8%
r-correct	30.9%
p-correct	53.1%
edit overhead	90.5%
mean word duration	0.378 s
WFC	mean: 0.276 s
WFF	mean: 0.004 s
immediately correct	58.6%

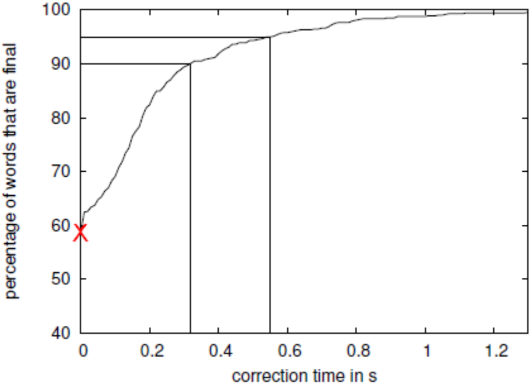
- rather low correctness
- very high edit overhead!
- information available after $\frac{3}{4}$ have been spoken
- word becomes final when it has ended

Measurements

SER	68.2%
WER	18.8%
r-correct	30.9%
p-correct	53.1%
edit overhead	90.5%
mean word duration	0.378 s
WFC	mean: 0.276 s
WFF	mean: 0.004 s
immediately correct	58.6%

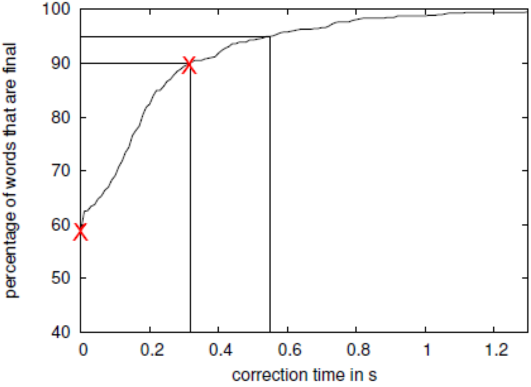
- rather low correctness
- very high edit overhead!
- information available after $\frac{3}{4}$ have been spoken
- word becomes final when it has ended

Measurements



58.6% immediately correct

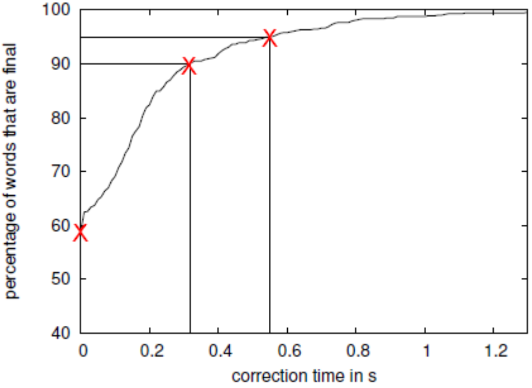
Measurements



58.6% immediately correct

90% correct after a correction time of 320 ms

Measurements



58.6% immediately correct

90% correct after a correction time of 320 ms

95% correct after a correction time of 550 ms

Independency of the measures?

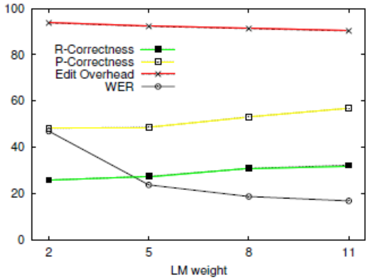
Are the measures independent of specific settings?

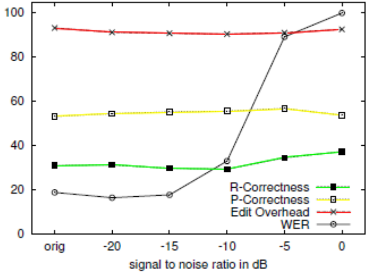
- vary LM/AM-weight
- vary audio quality by adding white noise

Independency of the measures?

Are the measures independent of specific settings?

- vary LM/AM-weight
- vary audio quality by adding white noise





Baumann et al.

- Define measures to evaluate incremental ASR-Systems
- Evaluate an existing system with these measures
- Use the measures to improve ASR
- Conclusions

Which improvements?

main goal: improve edit overhead

- reduce amount of wrong hypotheses
- still as quick as possible

Right Context

allow larger right context of size Δ:

- at time t: take into account output of ASR until $t - \Delta$ only
- $hyp_{t-\Delta}$ has a lookahead up to t

Right Context

allow larger right context of size Δ:

- at time t: take into account output of ASR until $t - \Delta$ only
- $hyp_{t-\Delta}$ has a lookahead up to t

$\Rightarrow$ reduction of edit overhead

Right Context

allow larger right context of size Δ:

- at time t: take into account output of ASR until $t - \Delta$ only
- $hyp_{t-\Delta}$ has a lookahead up to t

$\Rightarrow$ reduction of edit overhead

$\Rightarrow$ hypothesis lags behind the gold standard

- WFC increases by Δ
- effects on correctness, because w_{gold_t} may contain more words

Right Context

allow larger right context of size Δ:

- at time t: take into account output of ASR until $t - \Delta$ only
- $hyp_{t-\Delta}$ has a lookahead up to t

$\Rightarrow$ reduction of edit overhead

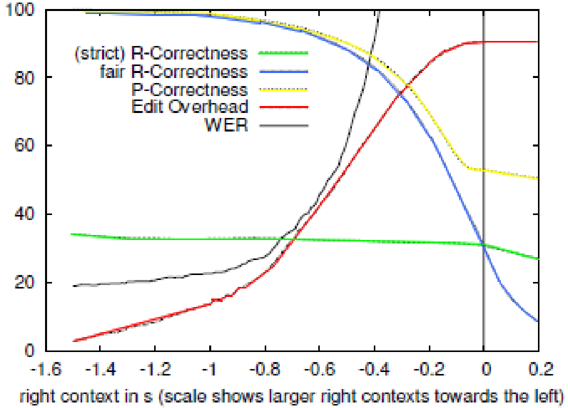
$\Rightarrow$ hypothesis lags behind the gold standard

- WFC increases by Δ
- effects on correctness, because w_{gold_t} may contain more words

Fair R-Correctness

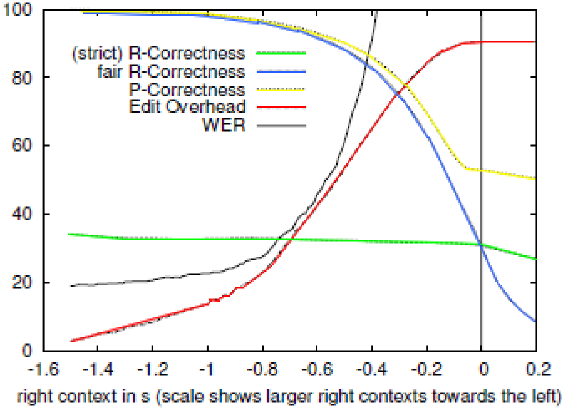
w_{hyp_t} is *fairly r-correct*, **iff** $w_{hyp_t}{}_{t-\Delta} = w_{gold}{}_{t-\Delta}$

Right Context



correctness and EO improve with more right context

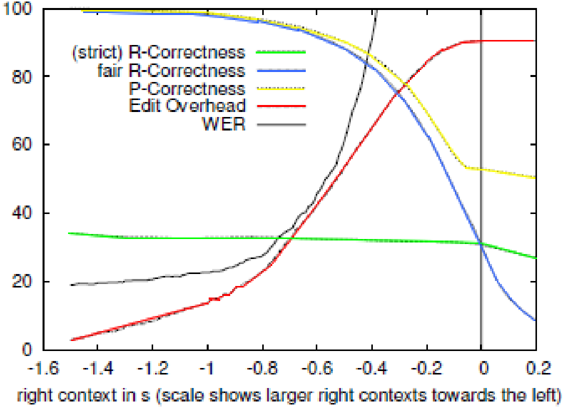
Right Context



correctness and EO improve with more right context

timing measures increase with larger right context

Right Context



correctness and EO improve with more right context

timing measures increase with larger right context

percentage of immediately correct hypotheses increases:
90% @ 580 ms
98% @ 1060 ms

Message Smoothing

edit message must be result of N consecutive hypotheses before commitment

Message Smoothing

edit message must be result of N consecutive hypotheses before commitment

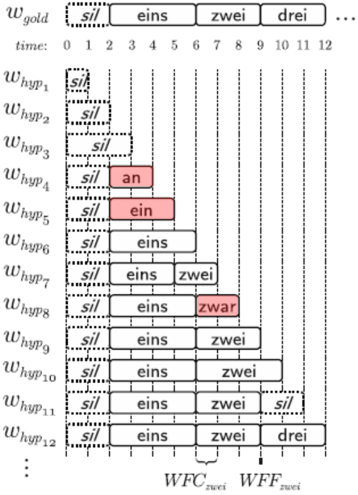
$N = 2$:

Message Smoothing

edit message must be result of N consecutive hypotheses before commitment

$N = 2$:

- *an*, *ein* and *zwar* would never be committed

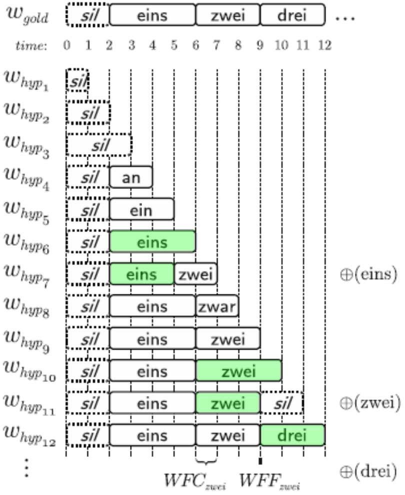


Message Smoothing

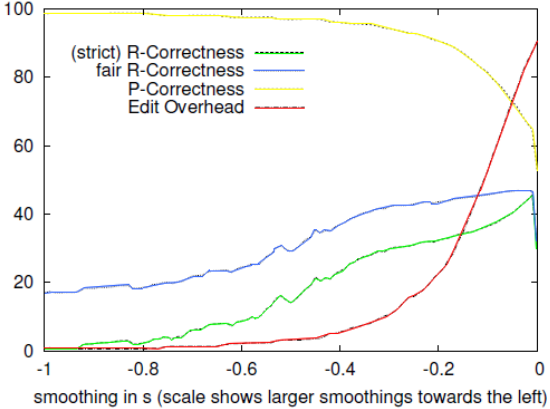
edit message must be result of N consecutive hypotheses before commitment

$N = 2$:

- *an*, *ein* and *zwar* would never be committed
- only 3 edit messages



Message Smoothing

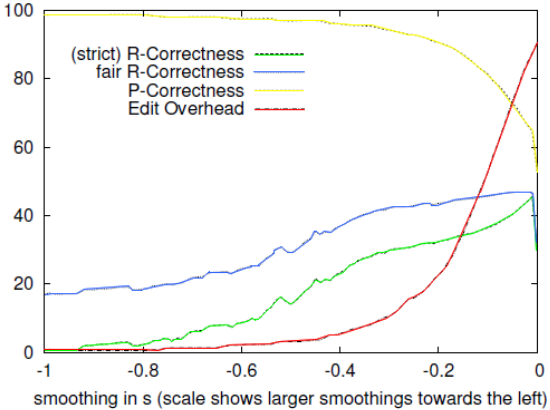


edit overhead falls rapidly:

50% after 110 ms

10% after 320 ms

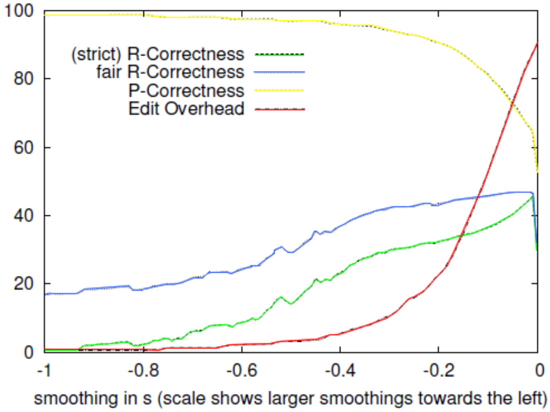
Message Smoothing



edit overhead falls rapidly:
50% after 110 ms
10% after 320 ms

decreasing (strict and fair) r-correctness

Message Smoothing



edit overhead falls rapidly:
50% after 110 ms
10% after 320 ms

decreasing (strict and fair) r-correctness

increasing timing measures

Baumann et al.

- Define measures to evaluate incremental ASR-Systems
- Evaluate an existing system with these measures
- Use the measures to improve ASR
- Conclusions

Conclusions

goal: improve edit overhead

Conclusions

goal: improve edit overhead

Right Context

improvements with larger delays, increasing correctness

Message Smoothing

improvements with shorter delays, decreasing correctness

Conclusions

goal: improve edit overhead

Right Context

improvements with larger delays, increasing correctness

Message Smoothing

improvements with shorter delays, decreasing correctness

could be combined to yield a good effect

Kato et. al

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar
- PITAG: probabilistic ITAG

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar
- PITAG: probabilistic ITAG
- Validity of Partial Parse Trees

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar
- PITAG: probabilistic ITAG
- Validity of Partial Parse Trees
- Experimental Results and Conclusions

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar
- PITAG: probabilistic ITAG
- Validity of Partial Parse Trees
- Experimental Results and Conclusions

Elementary Trees

Initial Trees

must be *leftmost-expanded*:

Elementary Trees

Initial Trees

must be *leftmost-expanded*:

1. $[t]_X$

   t: terminal symbol

   X: nonterminal symbol

Elementary Trees

Initial Trees

must be *leftmost-expanded*:

1. $[t]_X$
   t: terminal symbol
   X: nonterminal symbol

$$\begin{array}{c} X \\ | \\ t \end{array}$$

Elementary Trees

Initial Trees

must be *leftmost-expanded*:

1. $[t]_X$
   t: terminal symbol
   X: nonterminal symbol

$$\begin{array}{c} X \\ | \\ t \end{array}$$

2. $[\sigma X_1 \cdots X_k]_X$
   σ: leftmost expanded tree
   $X, X_1, \ldots, X_k$: nonterminal symbols

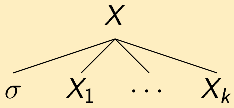
Elementary Trees

Initial Trees

must be *leftmost-expanded*:

1. $[t]_X$
   t: terminal symbol
   X: nonterminal symbol

2. $[\sigma X_1 \cdots X_k]_X$
   σ: leftmost expanded tree
   $X, X_1, \ldots, X_k$: nonterminal symbols



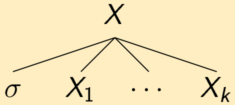
Elementary Trees

Initial Trees

must be *leftmost-expanded*:

1. $[t]_X$
   t: terminal symbol
   X: nonterminal symbol

2. $[\sigma X_1 \cdots X_k]_X$
   σ: leftmost expanded tree
   $X, X_1, \ldots, X_k$: nonterminal symbols



Auxiliary Trees

$[X^* \sigma X_1 \cdots X_k]_X$
σ: leftmost expanded tree
$X, X_1, \ldots, X_k$: nonterminal symbols

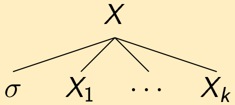
Elementary Trees

Initial Trees

must be *leftmost-expanded*:

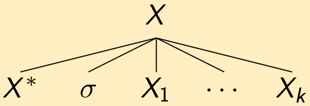
1. $[t]_X$
   t: terminal symbol
   X: nonterminal symbol

2. $[\sigma X_1 \cdots X_k]_X$
   σ: leftmost expanded tree
   $X, X_1, \ldots, X_k$: nonterminal symbols

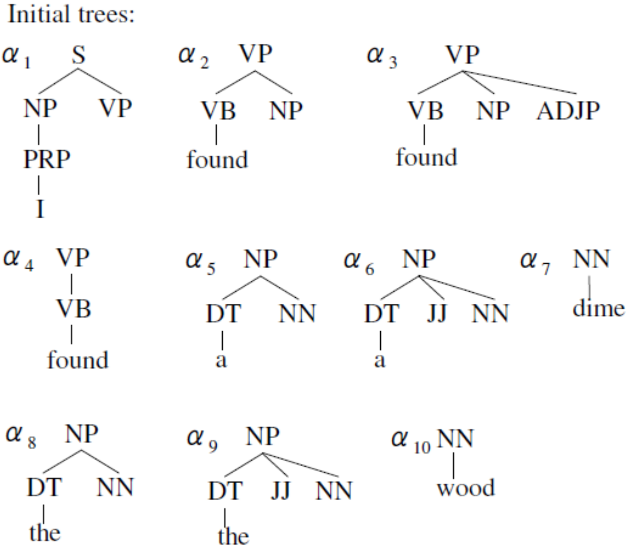


Auxiliary Trees

$[X^* \sigma X_1 \cdots X_k]_X$
σ: leftmost expanded tree
$X, X_1, \ldots, X_k$: nonterminal symbols

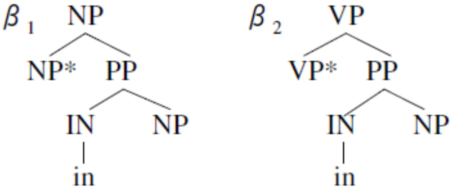


Elementary Trees



Elementary Trees

Auxiliary trees:



Operations

Substitution

replaces a **leftmost nonterminal leaf** of a partial parse tree σ with an **initial tree** α having the **same nonterminal symbol** at its **root**

s_α: substituting α

$s_\alpha(\sigma)$: result of applying s_α to σ

Operations

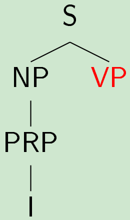
Substitution

replaces a **leftmost nonterminal leaf** of a partial parse tree σ with an **initial tree** α having the **same nonterminal symbol** at its **root**

s_α: substituting α

$s_\alpha(\sigma)$: result of applying s_α to σ

Example



Operations

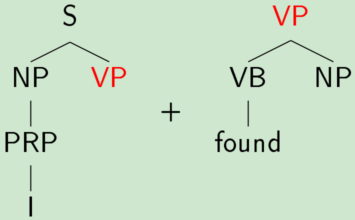
Substitution

replaces a **leftmost nonterminal leaf** of a partial parse tree σ with an **initial tree** α having the **same nonterminal symbol** at its **root**

s_α: substituting α

$s_\alpha(\sigma)$: result of applying s_α to σ

Example



Operations

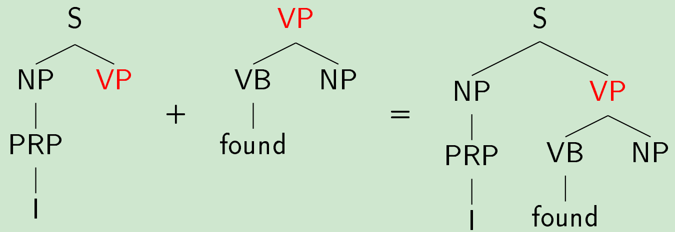
Substitution

replaces a **leftmost nonterminal leaf** of a partial parse tree σ with an **initial tree** α having the **same nonterminal symbol** at its **root**

s_α: substituting α

$s_\alpha(\sigma)$: result of applying s_α to σ

Example



Operations

Adjunction

splits a partial parse tree σ **at a nonterminal node having no nonterminal leaf** and **inserts** an **auxiliary tree** β having the **same nonterminal symbol** at its **root**

a_β: adjoining β

$a_\beta(\sigma)$: result of applying a_β to σ

Operations

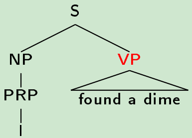
Adjunction

splits a partial parse tree σ **at a nonterminal node having no nonterminal leaf** and **inserts** an **auxiliary tree** β having the **same nonterminal symbol** at its **root**

a_β: adjoining β

$a_\beta(\sigma)$: result of applying a_β to σ

Example



Operations

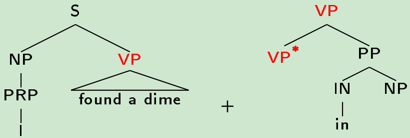
Adjunction

splits a partial parse tree σ **at a nonterminal node having no nonterminal leaf** and **inserts** an **auxiliary tree** β having the **same nonterminal symbol** at its **root**

a_β: adjoining β

$a_\beta(\sigma)$: result of applying a_β to σ

Example



Operations

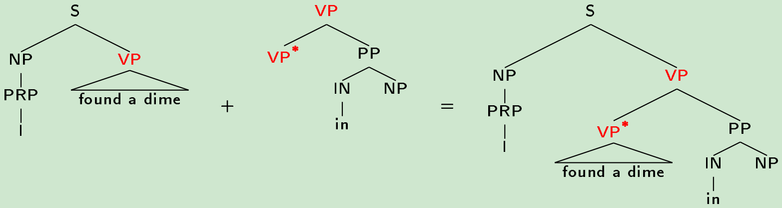
Adjunction

splits a partial parse tree σ **at a nonterminal node having no nonterminal leaf** and **inserts** an **auxiliary tree** β having the **same nonterminal symbol** at its **root**

a_β: adjoining β

$a_\beta(\sigma)$: result of applying a_β to σ

Example



Parsing with ITAG

at i-th word w_i:

Parsing with ITAG

at i-th word w_i:

- combine (substitution, adjunction) elementary trees for w_i with partial parse trees for $w_1 \cdots w_{i-1}$

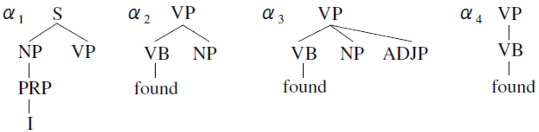
Parsing with ITAG

at i-th word w_i:

- combine (substitution, adjunction) elementary trees for w_i with partial parse trees for $w_1 \cdots w_{i-1}$

$\Rightarrow$ partial parse trees for $w_1 \cdots w_i$

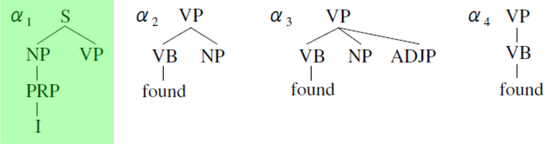
Example

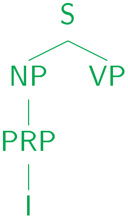

α1 S
NP VP
PRP
I
α2 VP
VB NP
found
α3 VP
VB NP ADJP
found
α4 VP
VB
found

S

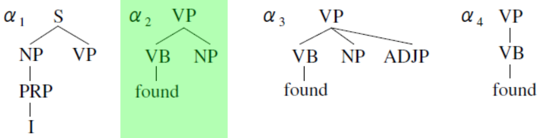
Example

I

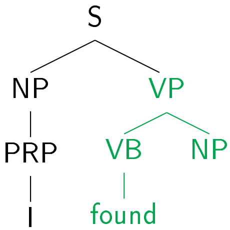

α1 S
NP VP
PRP
I
α2 VP
VB NP
found
α3 VP
VB NP ADJP
found
α4 VP
VB
found


S
NP VP
PRP
I

Example

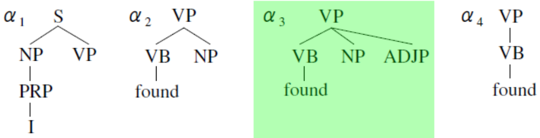


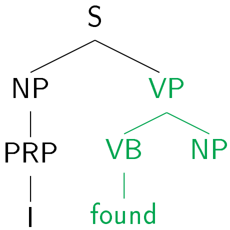
I found

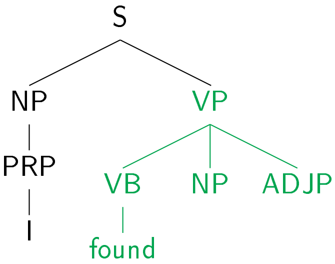


Example

I found

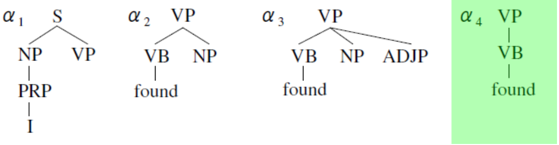


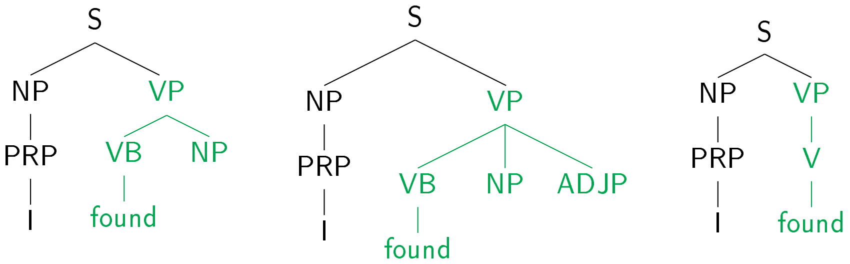




Example

I found





Example

word	#	partial parse tree
	1	s
I	2	$[[[I]_{prp}]_{np} vp]_s$
found	3	$[[[I]_{prp}]_{np}[[found]_{vb} np]_{vp}]_s$
	4	$[[[I]_{prp}]_{np}[[found]_{vb} np\ adjp]_{vp}]_s$
	5	$[[[I]_{prp}]_{np}[[found]_{vb}]_{vp}]_s$
a	6	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} nn]_{np}]_{vp}]_s$
	7	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} jj\ nn]_{np}]_{vp}]_s$
	8	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} nn]_{np} adjp]_{vp}]_s$
	9	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} jj\ nn]_{np} adjp]_{vp}]_s$
dime	10	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}]_s$
	11	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np} adjp]_{vp}]_s$
in	12	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in} np]_{pp}]_{vp}]_s$
	13	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in} np]_{pp}]_{np}]_{vp}]_s$
the	14	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt} nn]_{np}]_{pp}]_{vp}]_s$
	15	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt} jj\ nn]_{np}]_{pp}]_{vp}]_s$
	16	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt} nn]_{np}]_{pp}]_{np}]_{vp}]_s$
	17	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt} jj\ nn]_{np}]_{pp}]_{np}]_{vp}]_s$
wood	18	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt}[wood]_{nn}]_{np}]_{pp}]_{vp}]_s$
	19	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt}[wood]_{nn}]_{np}]_{pp}]_{np}]_{vp}]_s$

Example

word	#	partial parse tree
	1	s
I	2	$[[[I]_{prp}]_{np} vp]_s$
found	3	$[[[I]_{prp}]_{np}[[found]_{vb} np]_{vp}]_s$
	4	$[[[I]_{prp}]_{np}[[found]_{vb} np\ adjp]_{vp}]_s$
	5	$[[[I]_{prp}]_{np}[[found]_{vb}]_{vp}]_s$
a	6	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} nn]_{np}]_{vp}]_s$
	7	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} jj\ nn]_{np}]_{vp}]_s$
	8	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} nn]_{np} adjp]_{vp}]_s$
	9	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} jj\ nn]_{np} adjp]_{vp}]_s$
dime	10	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}]_s$
	11	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np} adjp]_{vp}]_s$
in	12	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in} np]_{pp}]_{vp}]_s$
	13	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in} np]_{pp}]_{np}]_{vp}]_s$
the	14	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt} nn]_{np}]_{pp}]_{vp}]_s$
	15	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt} jj\ nn]_{np}]_{pp}]_{vp}]_s$
	16	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt} nn]_{np}]_{pp}]_{np}]_{vp}]_s$
	17	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt} jj\ nn]_{np}]_{pp}]_{np}]_{vp}]_s$
wood	18	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt}[wood]_{nn}]_{np}]_{pp}]_{vp}]_s$
	19	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt}[wood]_{nn}]_{np}]_{pp}]_{np}]_{vp}]_s$

constructing parse trees of initial fragments for every word input possible!

Tree Construction

How to get the elementary trees?

Tree Construction

How to get the elementary trees?

- Extraction from a treebank!

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar
- PITAG: probabilistic ITAG
- Validity of Partial Parse Trees
- Experimental Results and Conclusions

Probabilities

probabilities for events combining an elementary tree and another tree by **substitution and adjunction**

Probabilities

probabilities for events combining an elementary tree and another tree by **substitution and adjunction**

α: initial tree with root symbol X

Probability of substituting α

$$P(s_\alpha) = \frac{|s_\alpha| \text{ in the treebank}}{|\text{substitutions using other initial trees with root X}| \text{ in the treebank}}$$

Probabilities

probabilities for events combining an elementary tree and another tree by **substitution and adjunction**

α: initial tree with root symbol X

Probability of substituting α

$$P(s_\alpha) = \frac{|s_\alpha| \text{ in the treebank}}{|\text{substitutions using other initial trees with root X}| \text{ in the treebank}}$$

β: auxiliary tree with root symbol X

Probability of adjoining β

$$P(a_\beta) = \frac{|a_\beta| \text{ in the treebank}}{|X| \text{ in the treebank}}$$

Probabilities

probability of a parse-tree:

- product of probabilities of operations used for construction

Probabilities

probability of a parse-tree:

- product of probabilities of operations used for construction

S

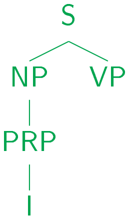
operation	probability
s_{α_1}	1.0
s_{α_2}	0.7
s_{α_5}	0.3
s_{α_7}	0.5

Probabilities

probability of a parse-tree:

- product of probabilities of operations used for construction

operation	probability
s_{α_1}	1.0
s_{α_2}	0.7
s_{α_5}	0.3
s_{α_7}	0.5



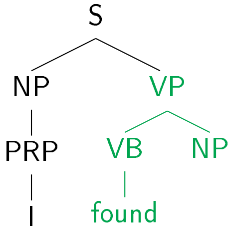
1.0

Probabilities

probability of a parse-tree:

- product of probabilities of operations used for construction

operation	probability
s_{α_1}	1.0
s_{α_2}	0.7
s_{α_5}	0.3
s_{α_7}	0.5



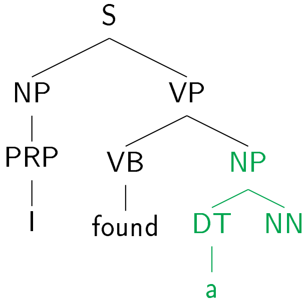
1.0×0.7

Probabilities

probability of a parse-tree:

- product of probabilities of operations used for construction

operation	probability
s_{α_1}	1.0
s_{α_2}	0.7
s_{α_5}	0.3
s_{α_7}	0.5



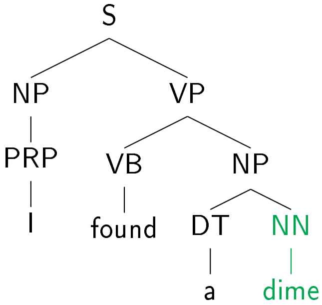
$1.0 \times 0.7 \times 0.3$

Probabilities

probability of a parse-tree:

- product of probabilities of operations used for construction

operation	probability
s_{α_1}	1.0
s_{α_2}	0.7
s_{α_5}	0.3
s_{α_7}	0.5



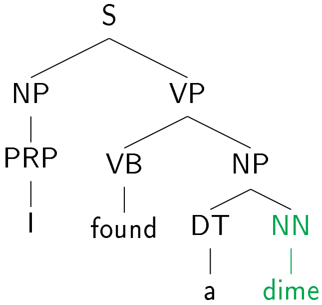
$1.0 \times 0.7 \times 0.3 \times 0.5$

Probabilities

probability of a parse-tree:

- product of probabilities of operations used for construction

operation	probability
s_{α_1}	1.0
s_{α_2}	0.7
s_{α_5}	0.3
s_{α_7}	0.5



$1.0 \times 0.7 \times 0.3 \times 0.5 = 0.105$

Parsing with PITAG

Improve efficiency by:

- discard tree with lower probability when there are parse trees with same possible operations
- only keep n-best partial parse-trees

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar
- PITAG: probabilistic ITAG
- Validity of Partial Parse Trees
- Experimental Results and Conclusions

Definitions

σ, τ: partial parse trees

Subsumption

σ subsumes τ, **iff** τ can be constructed by applying substitution- and adjunction-operations to σ
and every parse tree subsumes itself

Definitions

σ, τ: partial parse trees

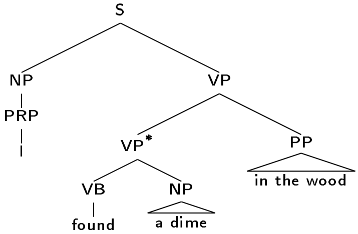
Subsumption

σ subsumes τ, **iff** τ can be constructed by applying substitution- and adjunction-operations to σ
and every parse tree subsumes itself

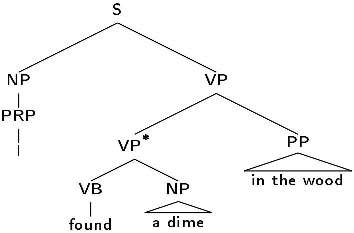
Validity

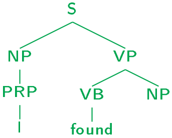
a partial parse tree is valid for an input sentence, **iff** it subsumes the correct parse tree for the input sentence

Example

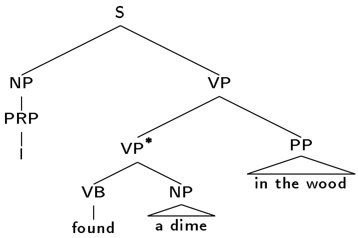


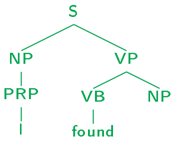
Example

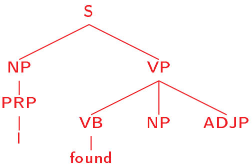

S
NP
VP
PRP
I
VP*
PP
in the wood
VB
NP
found
a dime


S
NP
VP
PRP
VB
NP
I
found

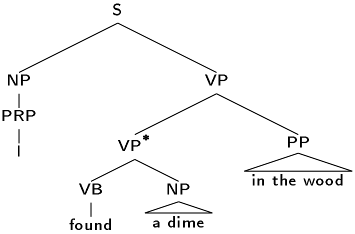
Example

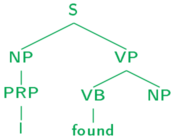

S
NP
VP
PRP
I
VP*
PP
VB
NP
in the wood
found
a dime

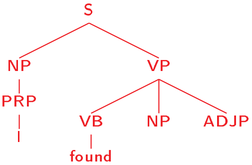

S
NP
VP
PRP
VB
NP
I
found

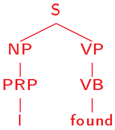

S
NP
VP
PRP
I
VB
NP
ADJP
found

Example


S
NP
PRP
I
VP
VP*
VB
found
NP
a dime
PP
in the wood


S
NP
PRP
I
VP
VB
found
NP


S
NP
PRP
I
VP
VB
found
NP
ADJP


S
NP
PRP
I
VP
VB
found

Evaluating Validity

validity for a partial parse tree depends on the rest of the sentence

Evaluating Validity

validity for a partial parse tree depends on the rest of the sentence
⇒dynamically varies for every input word

Evaluating Validity

validity for a partial parse tree depends on the rest of the sentence
⇒dynamically varies for every input word

σ: partial parse tree for initial fragment $w_1 \cdots w_i (i \leq j)$

Conditional Validity of partial parse tree σ

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of partial parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

Evaluating Validity

validity for a partial parse tree depends on the rest of the sentence
$\Rightarrow$dynamically varies for every input word

σ: partial parse tree for initial fragment $w_1 \cdots w_i (i \leq j)$

Conditional Validity of partial parse tree σ

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of partial parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

example: later

Output Partial Parse Trees

delay output until validity is high enough:

Output Partial Parse Trees

delay output until validity is high enough:

Parse Tree to be returned

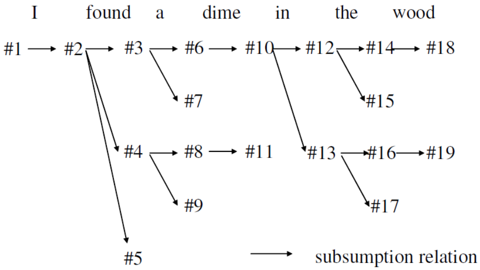
Parse Tree with the longest initial fragment whose validity is greater than threshold θ

θ: threshold between 0 and 1

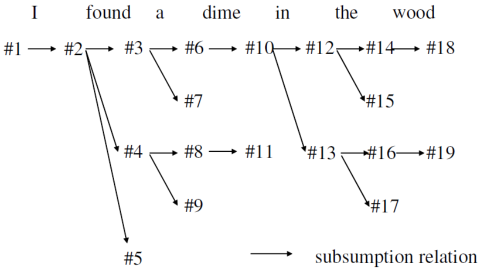
Example

word	#	partial parse tree
	1	s
I	2	$[[[I]_{prp}]_{np} vp]_s$
found	3	$[[[I]_{prp}]_{np}[[found]_{vb} np]_{vp}]_s$
	4	$[[[I]_{prp}]_{np}[[found]_{vb} np\ adjp]_{vp}]_s$
	5	$[[[I]_{prp}]_{np}[[found]_{vb}]_{vp}]_s$
a	6	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} nn]_{np}]_{vp}]_s$
	7	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} jj\ nn]_{np}]_{vp}]_s$
	8	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} nn]_{np} adjp]_{vp}]_s$
	9	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt} jj\ nn]_{np} adjp]_{vp}]_s$
dime	10	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}]_s$
	11	$[[[I]_{prp}]_{np}[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np} adjp]_{vp}]_s$
in	12	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in} np]_{pp}]_{vp}]_s$
	13	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in} np]_{pp}]_{np}]_{vp}]_s$
the	14	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt} nn]_{np}]_{pp}]_{vp}]_s$
	15	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt} jj\ nn]_{np}]_{pp}]_{vp}]_s$
	16	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt} nn]_{np}]_{pp}]_{np}]_{vp}]_s$
	17	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt} jj\ nn]_{np}]_{pp}]_{np}]_{vp}]_s$
wood	18	$[[[I]_{prp}]_{np}[[[found]_{vb}[[a]_{dt}[dime]_{nn}]_{np}]_{vp}[[in]_{in}[[the]_{dt}[wood]_{nn}]_{np}]_{pp}]_{vp}]_s$
	19	$[[[I]_{prp}]_{np}[[found]_{vb}[[[a]_{dt}[dime]_{nn}]_{np}[[in]_{in}[[the]_{dt}[wood]_{nn}]_{np}]_{pp}]_{np}]_{vp}]_s$

Example

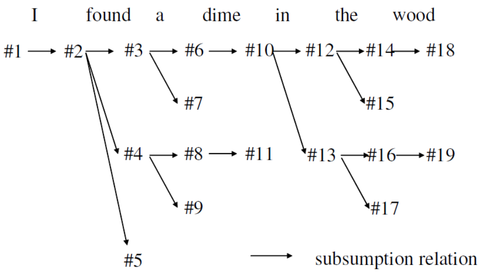

I
found
a
dime
in
the
wood
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
#11
#12
#13
#14
#15
#16
#17
#18
#19
subsumption relation

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

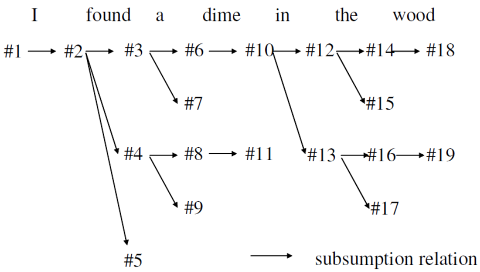
Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

$\theta = 0.8$

Example

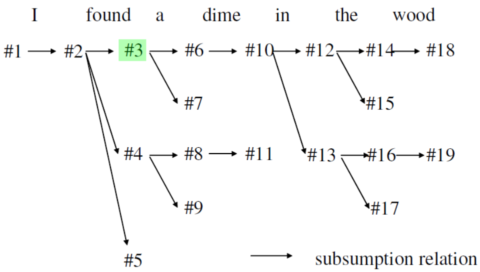


Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

Example



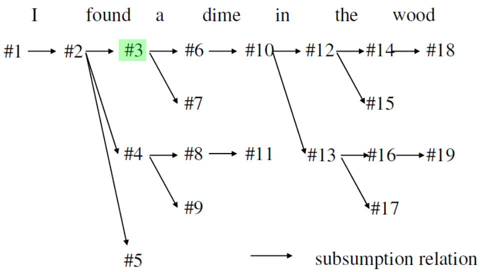
Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output?

Example



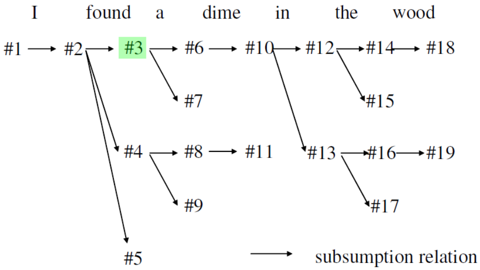
Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found":

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

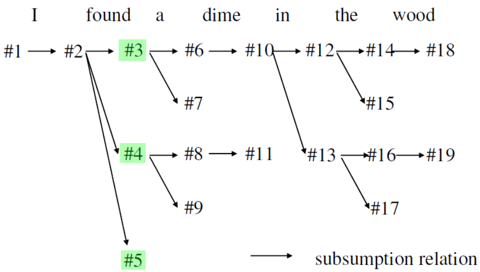
$\theta = 0.8$

$$V(\sigma|w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found":

$$V(\#3, \text{I found}) = \frac{P(\#3)}{P(\#3)+P(\#4)+P(\#5)}$$

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

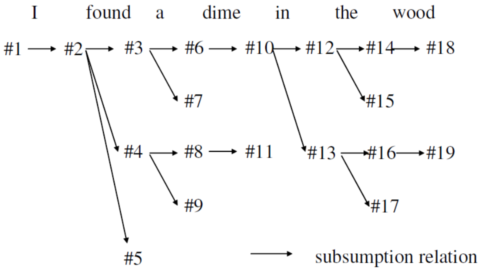
$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found":

$$V(\#3, \text{I found}) = \frac{P(\#3)}{P(\#3)+P(\#4)+P(\#5)}$$

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

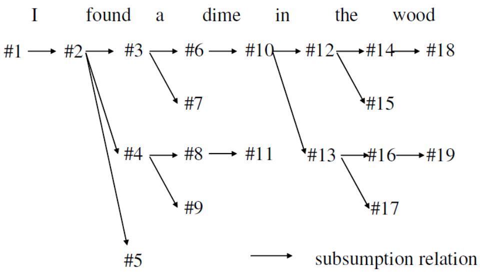
$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found":

$$V(\#3, \text{I found}) = \frac{P(\#3)}{P(\#3)+P(\#4)+P(\#5)} = \frac{0.7}{0.7+0.1+0.2} = 0.7$$

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

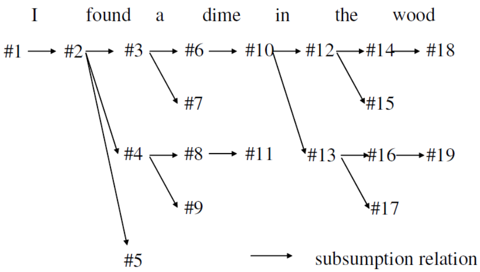
$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found":

$$V(\#3, \text{I found}) = \frac{P(\#3)}{P(\#3)+P(\#4)+P(\#5)} = \frac{0.7}{0.7+0.1+0.2} = 0.7 < \theta$$

Example



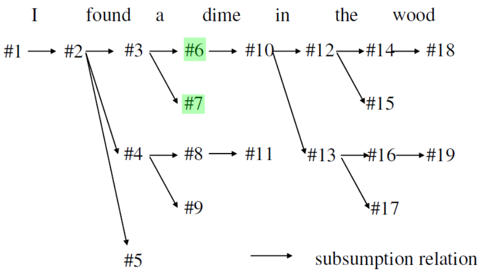
Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? “I found a”:

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

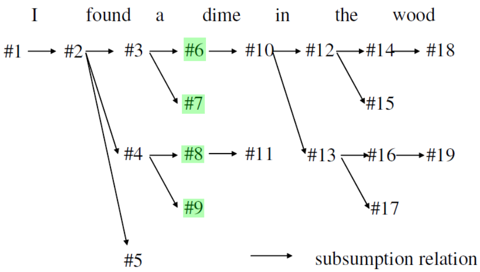
$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found a":

$$V(\#3, \text{I found a}) = \frac{P(\#6)+P(\#7)}{P(\#6)+P(\#7)+P(\#8)+P(\#9)}$$

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

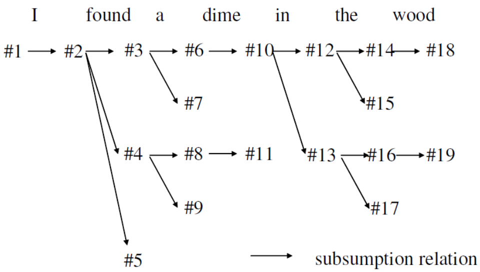
$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found a":

$$V(\#3, \text{I found a}) = \frac{P(\#6)+P(\#7)}{P(\#6)+P(\#7)+P(\#8)+P(\#9)}$$

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

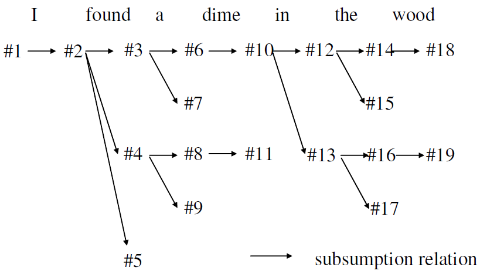
$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found a":

$V(\#3, \text{I found a}) = \frac{P(\#6)+P(\#7)}{P(\#6)+P(\#7)+P(\#8)+P(\#9)} = 0.875$

Example



Parse Tree	Probability
#3	0.7
#4	0.1
#5	0.2
#6	0.21
#7	0.14
#8	0.03
#9	0.02

$\theta = 0.8$

$$V(\sigma | w_1 \cdots w_j) = \frac{\sum(\text{Probabilities of parse trees at } w_j \text{ subsumed by } \sigma)}{\sum(\text{Probabilities of partial parse trees constructed at } w_j)}$$

When will #3 (valid) be output? "I found a":

$$V(\#3, \text{I found a}) = \frac{P(\#6)+P(\#7)}{P(\#6)+P(\#7)+P(\#8)+P(\#9)} = 0.875 \geq \theta$$

Kato et. al

- ITAG: incremental-parsing-oriented tree adjoining grammar
- PITAG: probabilistic ITAG
- Validity of Partial Parse Trees
- Experimental Results and Conclusions

Experimental Setup

- parser implemented in Lisp
- input: POS-Sequences
- elementary trees extracted from Penn Treebank
- only used correctly parsed sentences

Measures

Degree of delay at j-th word

$D(j, s) = j - |\text{output parse tree at } w_j|$

$s = w_1 \cdots w_n$: Input sentence

Measures

Degree of delay at j-th word

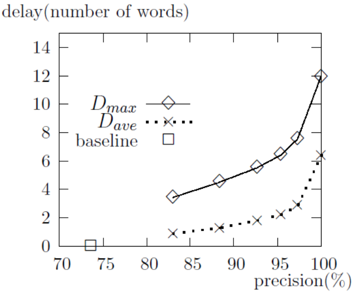
$D(j, s) = j - |\text{output parse tree at } w_j|$

$s = w_1 \cdots w_n$: Input sentence

Precision

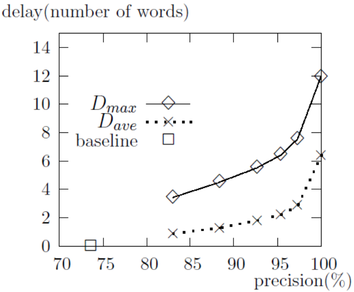
percentage of valid partial parse trees in the output

Results and Conclusions



	precision(%)	D_{max}	D_{ave}
$\theta = 1.0$	100.0	11.9	6.4
$\theta = 0.9$	97.3	7.5	2.9
$\theta = 0.8$	95.4	6.4	2.2
$\theta = 0.7$	92.5	5.5	1.8
$\theta = 0.6$	88.4	4.5	1.3
$\theta = 0.5$	83.0	3.4	0.9
baseline	73.6	0.0	0.0

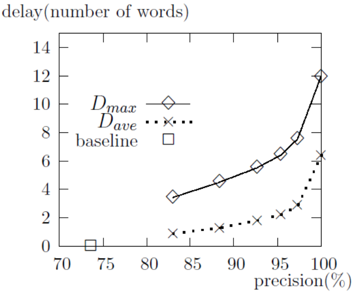
Results and Conclusions



	precision(%)	D_{max}	D_{ave}
$\theta = 1.0$	100.0	11.9	6.4
$\theta = 0.9$	97.3	7.5	2.9
$\theta = 0.8$	95.4	6.4	2.2
$\theta = 0.7$	92.5	5.5	1.8
$\theta = 0.6$	88.4	4.5	1.3
$\theta = 0.5$	83.0	3.4	0.9
baseline	73.6	0.0	0.0

higher threshold

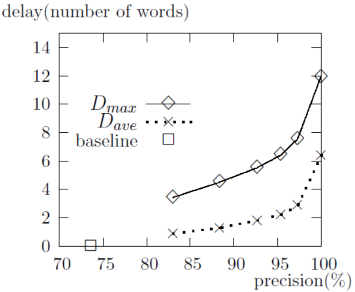
Results and Conclusions



	precision(%)	D_{max}	D_{ave}
$\theta = 1.0$	100.0	11.9	6.4
$\theta = 0.9$	97.3	7.5	2.9
$\theta = 0.8$	95.4	6.4	2.2
$\theta = 0.7$	92.5	5.5	1.8
$\theta = 0.6$	88.4	4.5	1.3
$\theta = 0.5$	83.0	3.4	0.9
baseline	73.6	0.0	0.0

higher threshold
$\Rightarrow$ higher precision, greater delays

Results and Conclusions



	precision(%)	D_{max}	D_{ave}
$\theta = 1.0$	100.0	11.9	6.4
$\theta = 0.9$	97.3	7.5	2.9
$\theta = 0.8$	95.4	6.4	2.2
$\theta = 0.7$	92.5	5.5	1.8
$\theta = 0.6$	88.4	4.5	1.3
$\theta = 0.5$	83.0	3.4	0.9
baseline	73.6	0.0	0.0

higher threshold
$\Rightarrow$ higher precision, greater delays
$\Rightarrow$ trade-off between precision and delay

Summary

Baumann et. al

use right context and message smoothing to reduce/avoid wrong hypotheses in incremental ASR

Summary

Baumann et. al

use right context and message smoothing to reduce/avoid wrong hypotheses in incremental ASR

Kato et. al

use PITAG to reduce/avoid wrong hypotheses in incremental parsing

Summary

Baumann et. al

use right context and message smoothing to reduce/avoid wrong hypotheses in incremental ASR

Kato et. al

use PITAG to reduce/avoid wrong hypotheses in incremental parsing

delay of output

Summary

Baumann et. al

use right context and message smoothing to reduce/avoid wrong hypotheses in incremental ASR

Kato et. al

use PITAG to reduce/avoid wrong hypotheses in incremental parsing

delay of output
$\Rightarrow$ **Trade-off between incrementality/speed and output quality (revisions)**

Thank you!

Thanks for your attention!

References

Timo Baumann, Michaela Atterer, and David Schlangen.
Assessing and Improving the Performance of Speech Recognition for Incremental Systems.
In *Proceedings of NAACL-HLT 2009*, Boulder, USA, June 2009.

Yoshihide Kato, Shigeki Matsubara, and Yasuyoshi Inagaki.
Stochastically evaluating the validity of partial parse trees in incremental parsing.
In *Proceedings of the ACL Workshop Incremental Parsing: Bringing Engineering and Cognition Together*, pages 9–15, Barcelona, Spain, July 2004. Association for Computational Linguistics.