Transformation Frameworks for Machine Translation: Strings, Trees, and Graphs

kevin knight

university of southern california

TAG+, Sept. 28, 2012

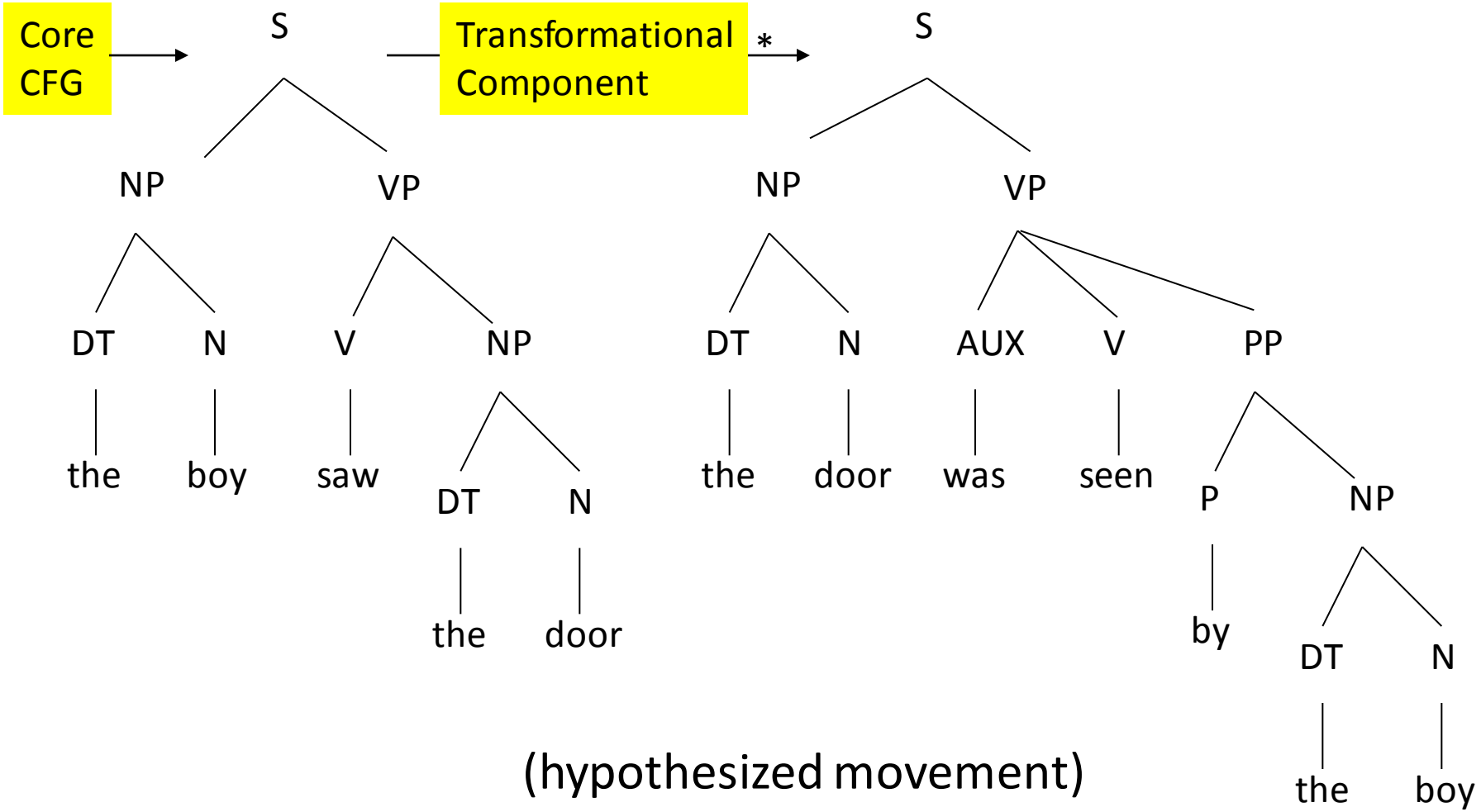
NOAM CHOMSKY

Modern Linguistics

How to characterize all and only strings of English?

NOAM CHOMSKY

Transformational Grammar



(hypothesized movement)

NOAM CHOMSKY

Tree Automata

[Rounds 1970] & [Thatcher 1970]:

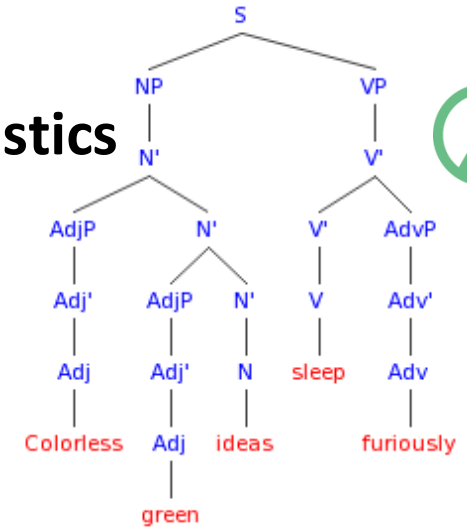
Invented tree transducers to formalize Transformational Grammar

[Thatcher 1973]:

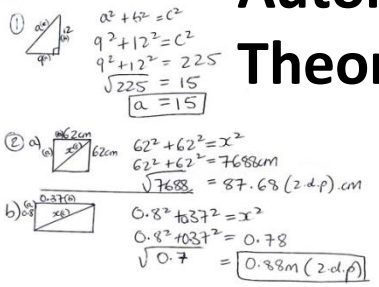
"The number one priority in the area [of tree automata theory] is a careful assessment of the significant problems concerning **natural language and programming language semantics and translation**. If such problems can be found and formulated, I am convinced that the approach informally surveyed here can provide a unifying framework within which to study them."

1960s & 1970s

Linguistics

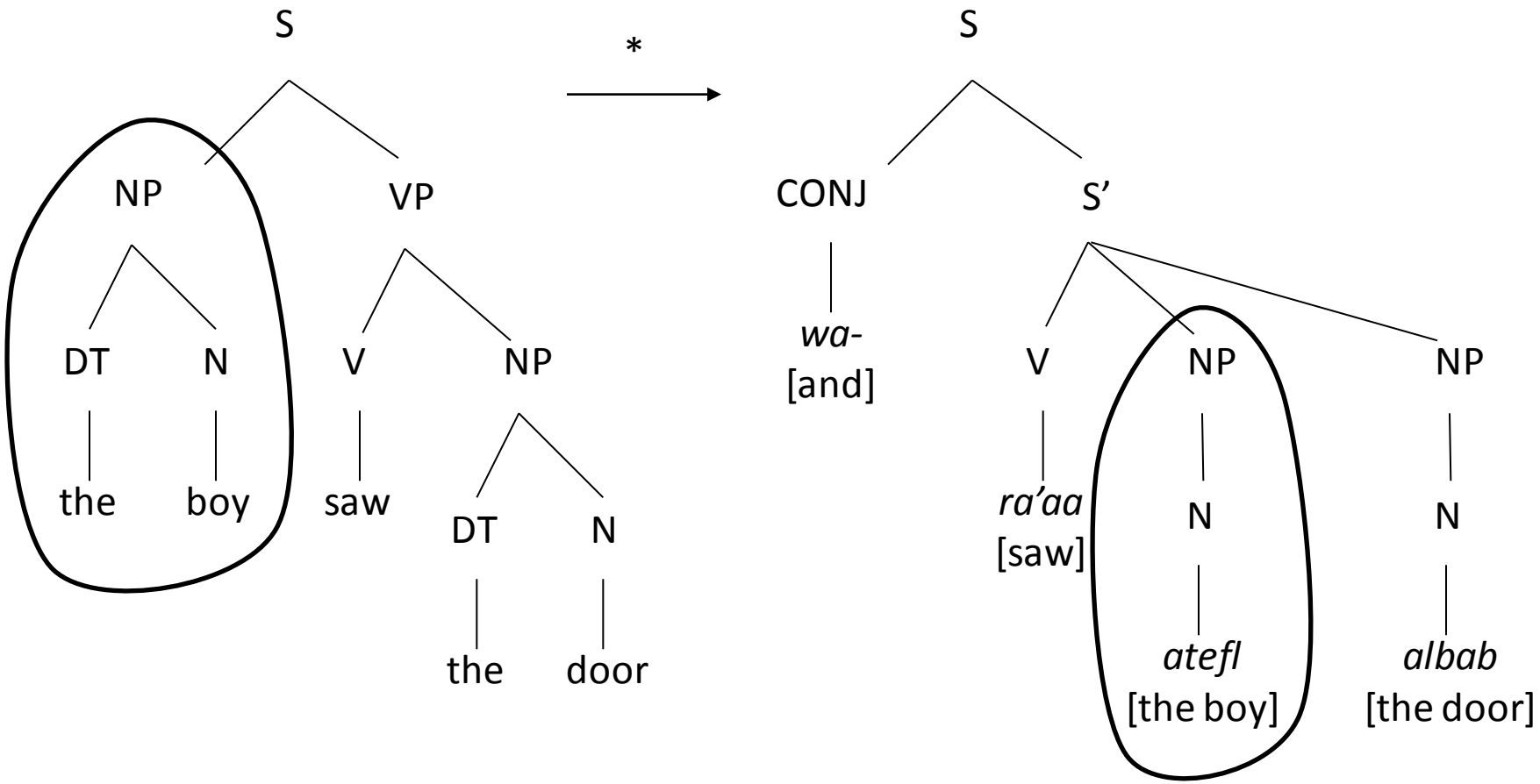


Automata Theory



Computers

Language Translation

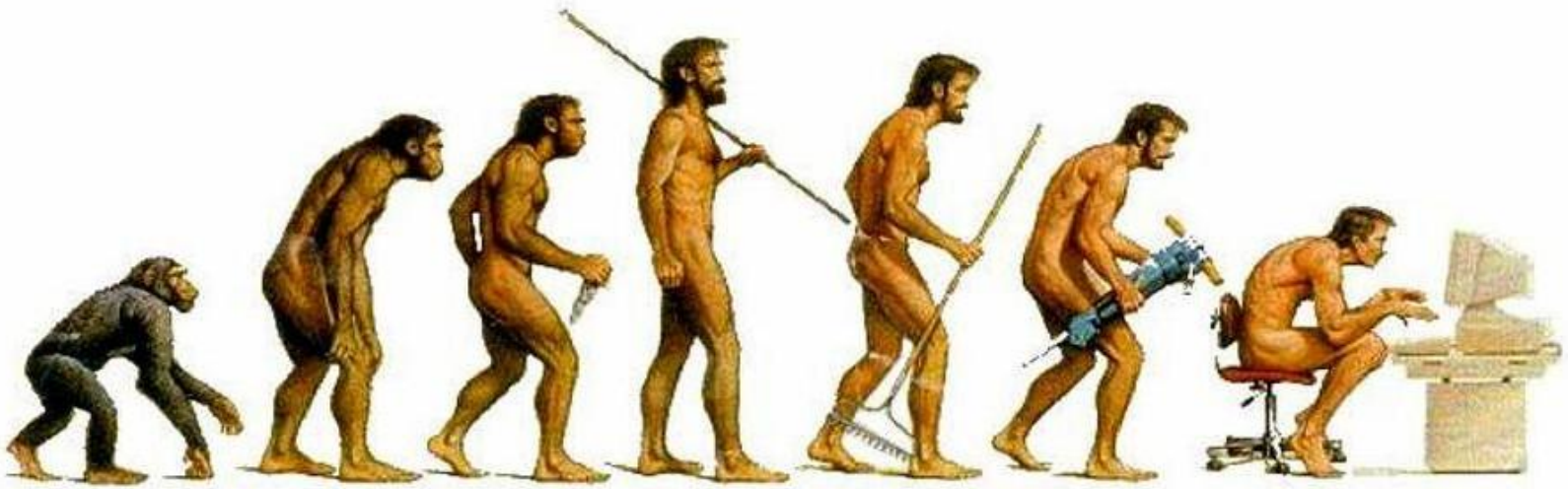


(real movement!)

Language Translation is Hard

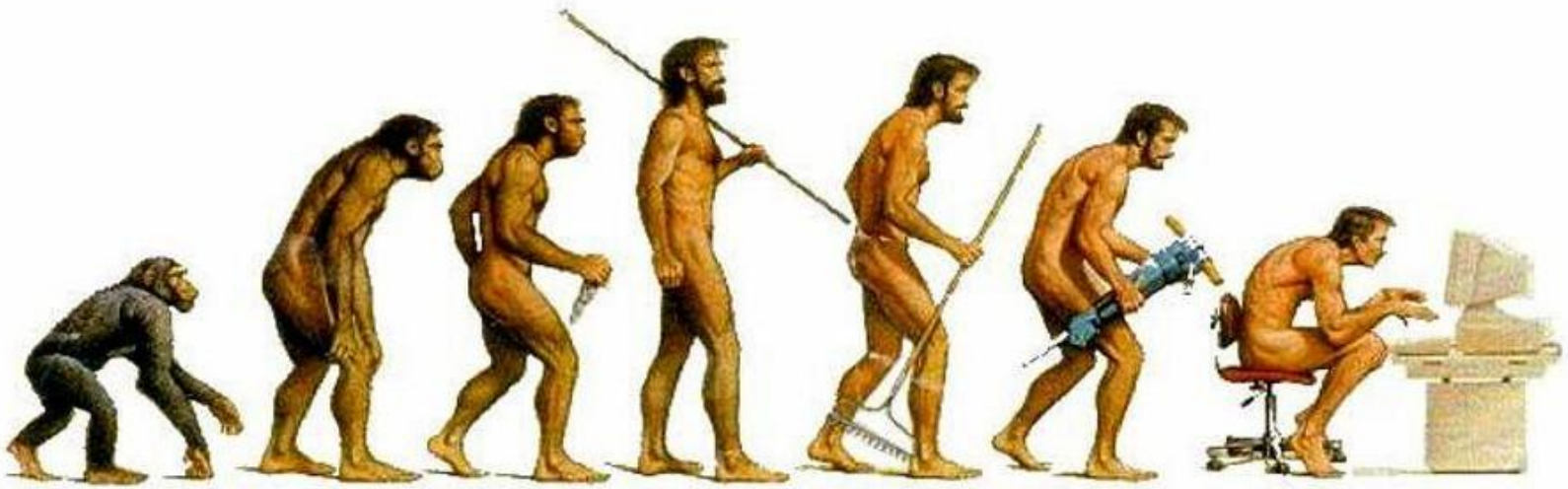
- Each word has tons of meanings
  - I'll **get** a cup of coffee → ?
  - I didn't **get** that joke → ?
  - I **get** up at 8am → ?
  - I **get** nervous → ?
  - Yeah, I **get** around … → ?
- Each word has zillions of contexts
- Word order is very different
- Machine must produce good sentences, not just consume them

Natural Language Processing

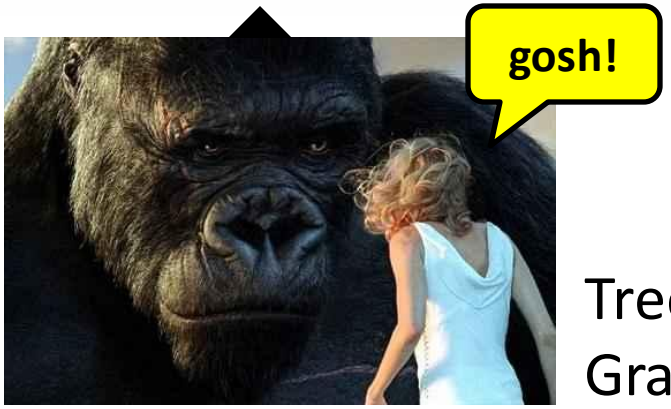


FSA CFG Unification Grammar CFG FSA

Natural Language Processing



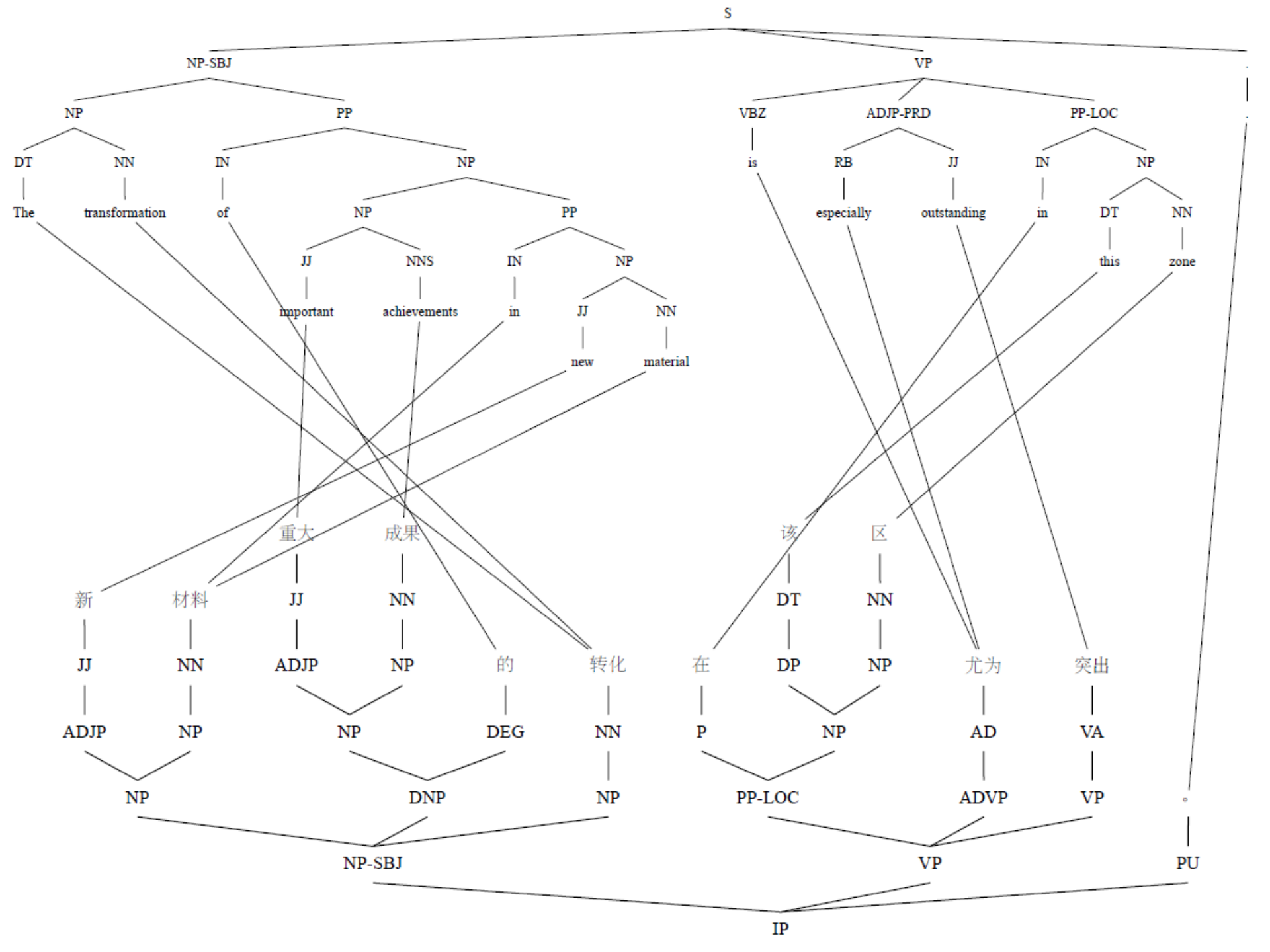
FSA CFG Unification Grammar CFG FSA

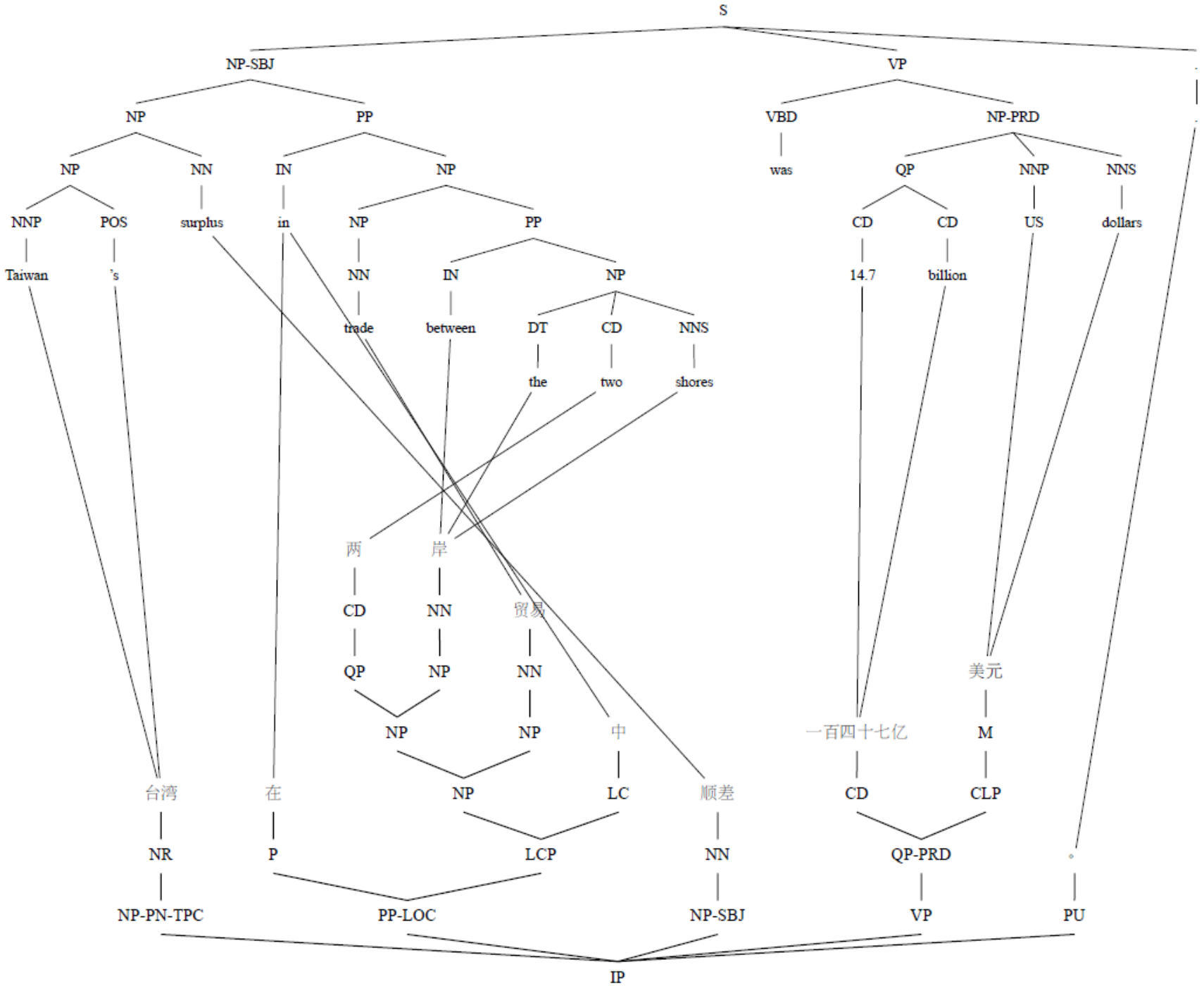


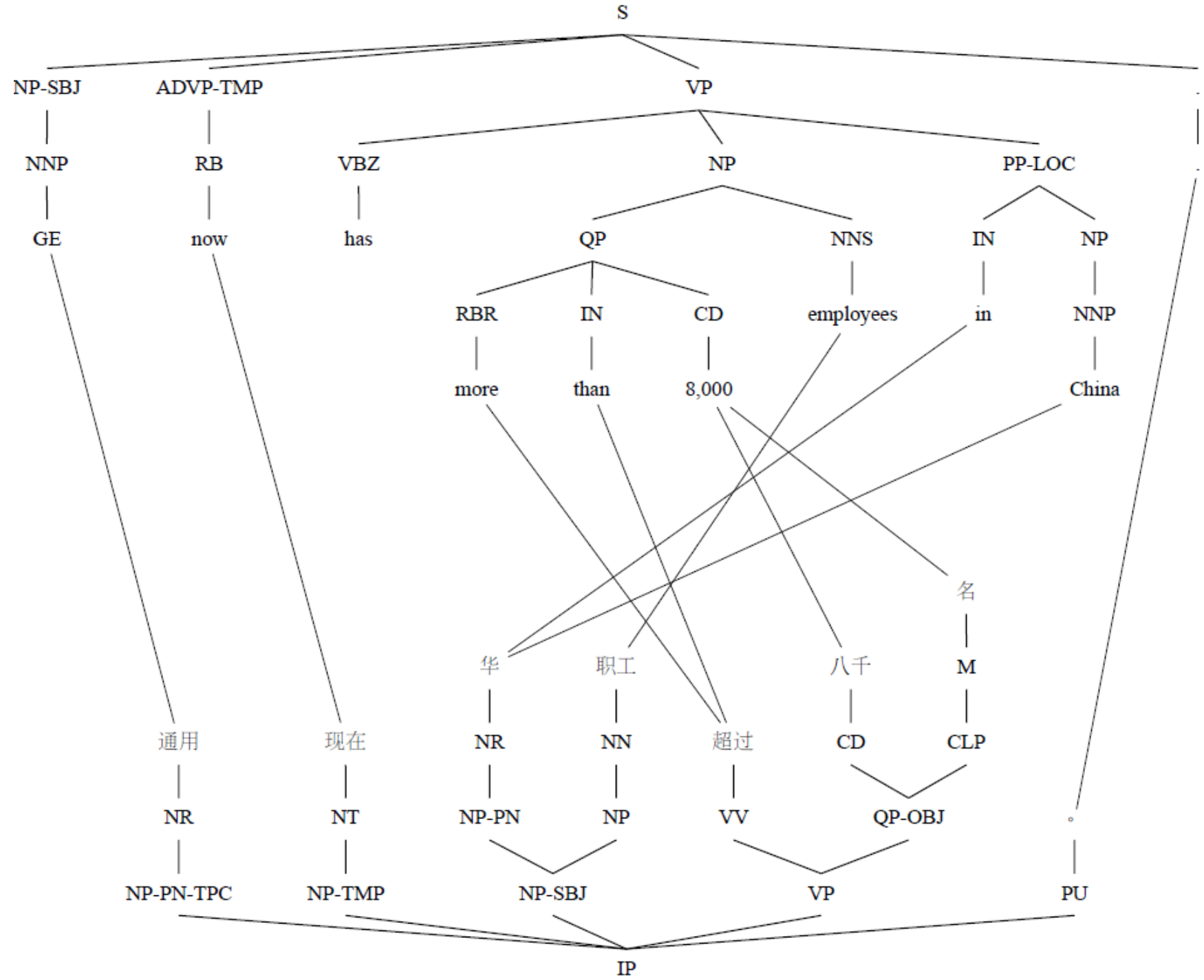
Tree Adjoining Grammar

Linguistic Data

- We have a lot!
- Can train an English FSA on one trillion words.
  - The cup is on the table >> Cup the table on is the
  - The player is on the field >> The player is in the field
- Translation data is especially tantalizing:
  - Billions of words, for some language pairs
- Let's **explain** translation data,
  search for models that **fit** the data,
  use those models to **translate** new data …


S
NP-SBJ
VP
NP
PP
VBZ
ADJP-PRD
PP-LOC
DT
NN
IN
NP
is
RB
JJ
IN
NP
The
transformation
of
NP
PP
especially
outstanding
in
DT
NN
JJ
NNS
IN
NP
this
zone
important
achievements
in
JJ
NN
new
material
重大
成果
该
区
新
材料
JJ
NN
DT
NN
JJ
NN
ADJP
NP
的
转化
在
DP
NP
尤为
突出
ADJP
NP
NP
DEG
NN
P
NP
AD
VA
NP
DNP
NP
PP-LOC
ADVP
VP
。
NP-SBJ
VP
PU
IP


S
NP-SBJ
VP
NP
PP
VBD
NP-PRD
NP
NN
IN
NP
was
QP
NNP
NNS
NNP
POS
surplus
in
NP
PP
CD
CD
US
dollars
Taiwan
's
NN
IN
NP
14.7
billion
trade
between
DT
CD
NNS
the
two
shores
两
岸
CD
NN
贸易
QP
NP
NN
美元
NP
NP
中
一百四十七亿
M
台湾
在
NP
LC
顺差
CD
CLP
NR
P
LCP
NN
QP-PRD
NP-PN-TPC
PP-LOC
NP-SBJ
VP
PU
IP


S
NP-SBJ
ADVP-TMP
VP
.
NNP
RB
VBZ
NP
PP-LOC
.
GE
now
has
QP
NNS
IN
NP
RBR
IN
CD
employees
in
NNP
more
than
8,000
China
名
华
职工
八千
M
通用
现在
NR
NN
超过
CD
CLP
NR
NT
NP-PN
NP
VV
QP-OBJ
。
NP-PN-TPC
NP-TMP
NP-SBJ
VP
PU
IP

Model Should Fit Data

What does it mean for a translation model to **fit the observed translation data**?

#1 Theory approach

#2 Linguistics approach

#3 Statistical approach

#4 Heroic approach

Fit to Data #1: Theory Approach

- Goal: Create an underlying formalism and prove that it has certain formal properties necessary to explain data.

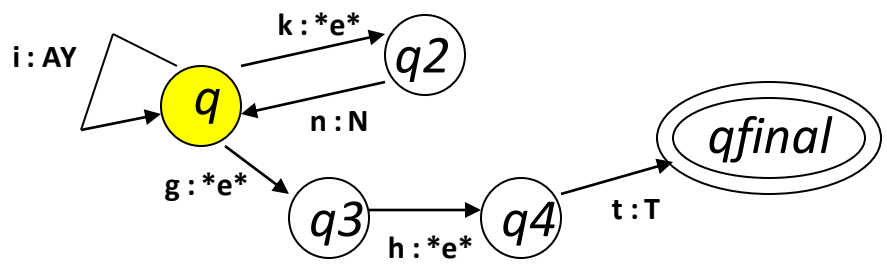
Finite-State Transducer (FST)

Original input:

k n i g h t

FST

q k → *q2* *e*
q i → *q* AY
q g → *q3* *e*
q2 n → *q* N
q3 h → *q4* *e*
q4 t → *qfinal* T



Transformation:

q k n i g h t

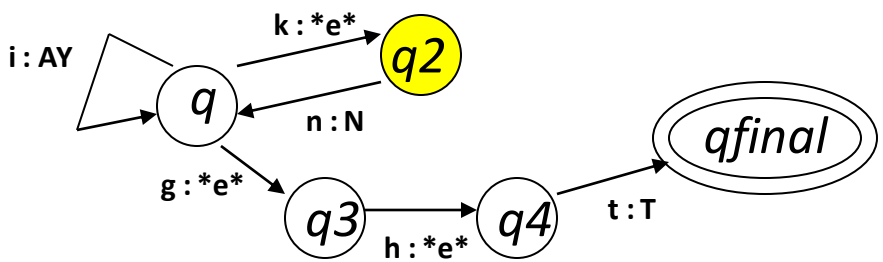
Finite-State Transducer (FST)

Original input:

k n i g h t

FST

q k → *q2* *e*
q i → *q* AY
q g → *q3* *e*
q2 n → *q* N
q3 h → *q4* *e*
q4 t → *qfinal* T



Transformation:

q2 n i g h t

Finite-State Transducer (FST)

Original input:

k n i g h t

FST

q k → *q2* *e*

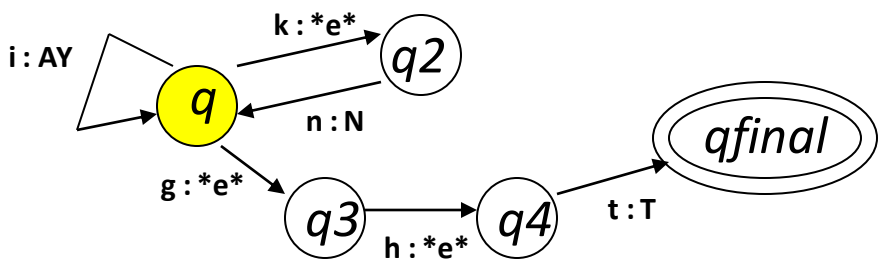
q i → *q* AY

q g → *q3* *e*

q2 n → *q* N

q3 h → *q4* *e*

q4 t → *qfinal* T



Transformation:

N i g h t

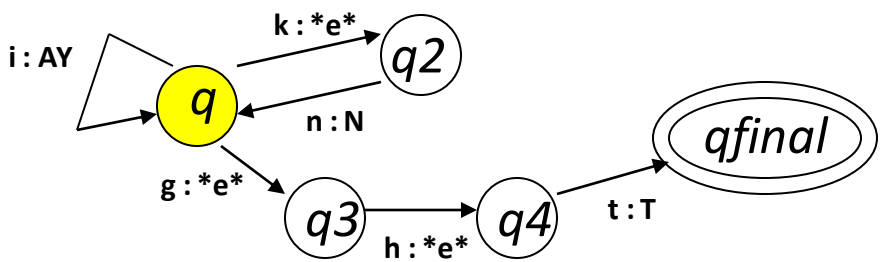
Finite-State Transducer (FST)

Original input:

k
n
i
g
h
t

FST

q k → *q2* *e*
q i → *q* AY
q g → *q3* *e*
q2 n → *q* N
q3 h → *q4* *e*
q4 t → *qfinal* T



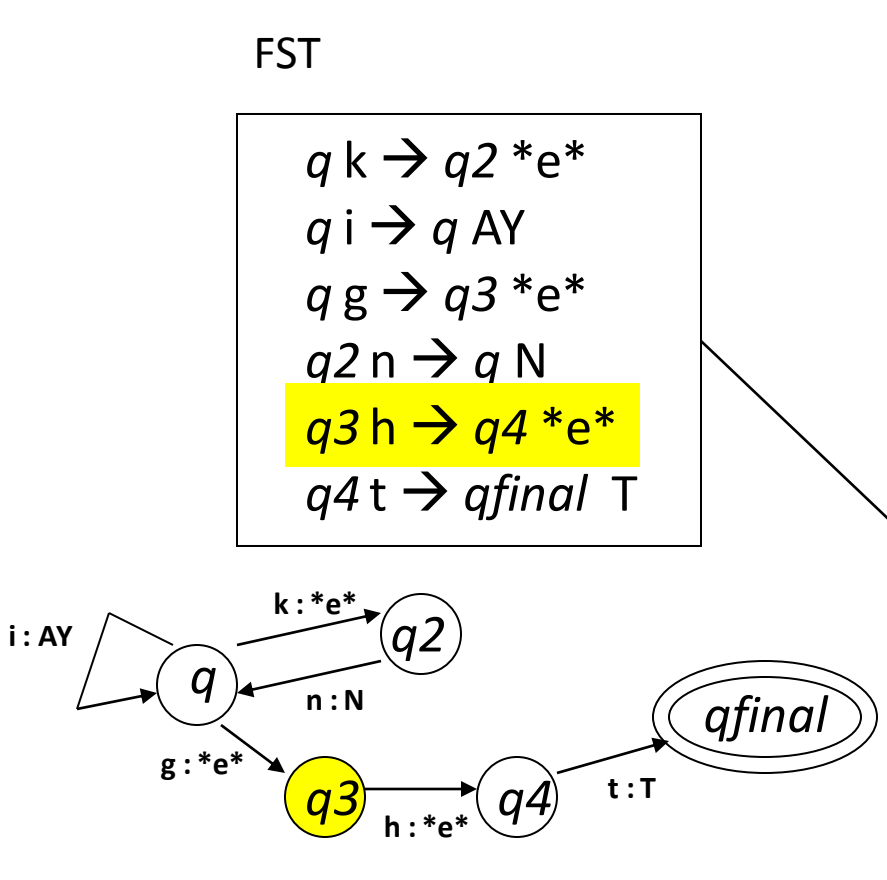
Transformation:

N
AY
q g
h
t

Finite-State Transducer (FST)

Original input:

k
n
i
g
h
t



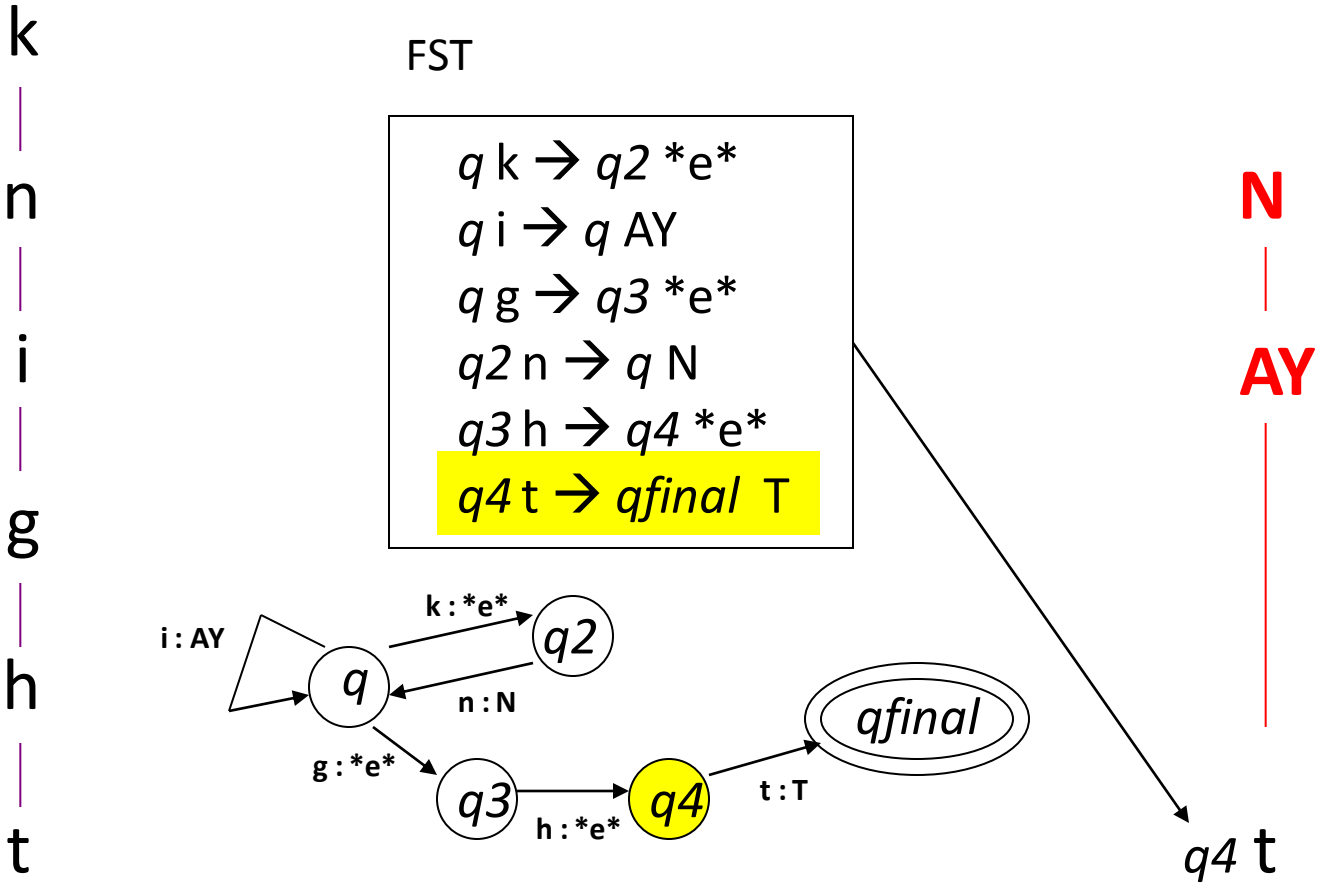
Transformation:

N
AY
q3 h
t

Finite-State Transducer (FST)

Original input:

k
n
i
g
h
t



Transformation:

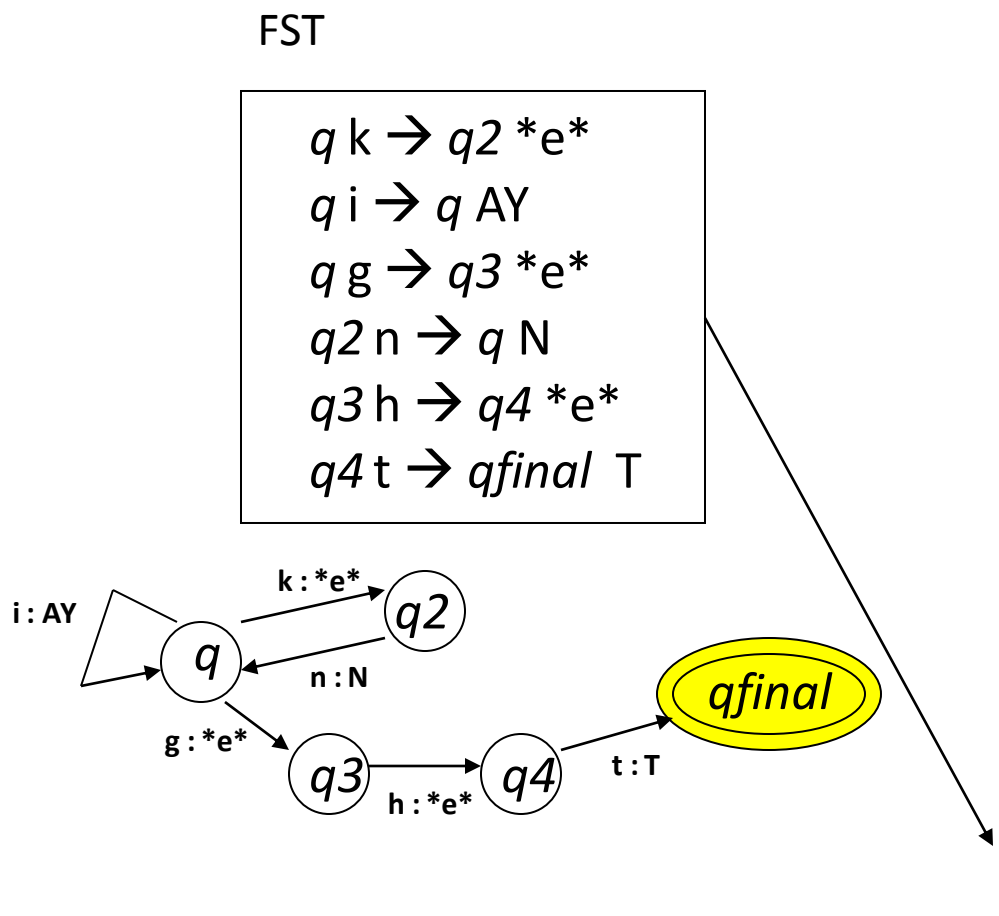
N
AY

q4 t

Finite-State Transducer (FST)

Original input:

k
n
i
g
h
t

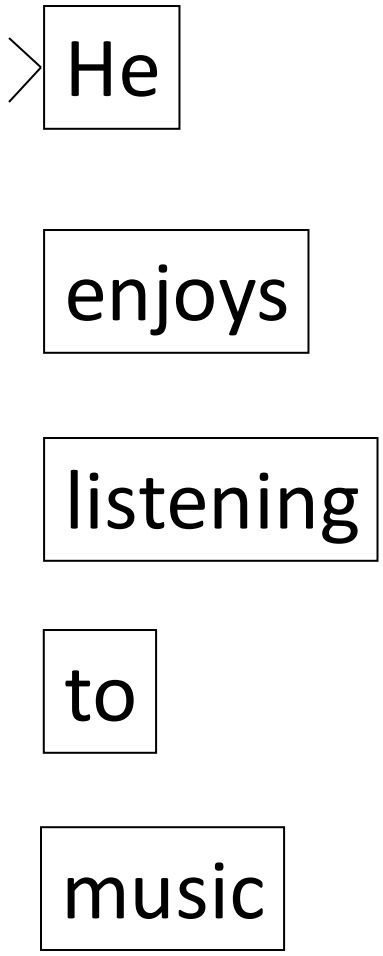


Transformation:

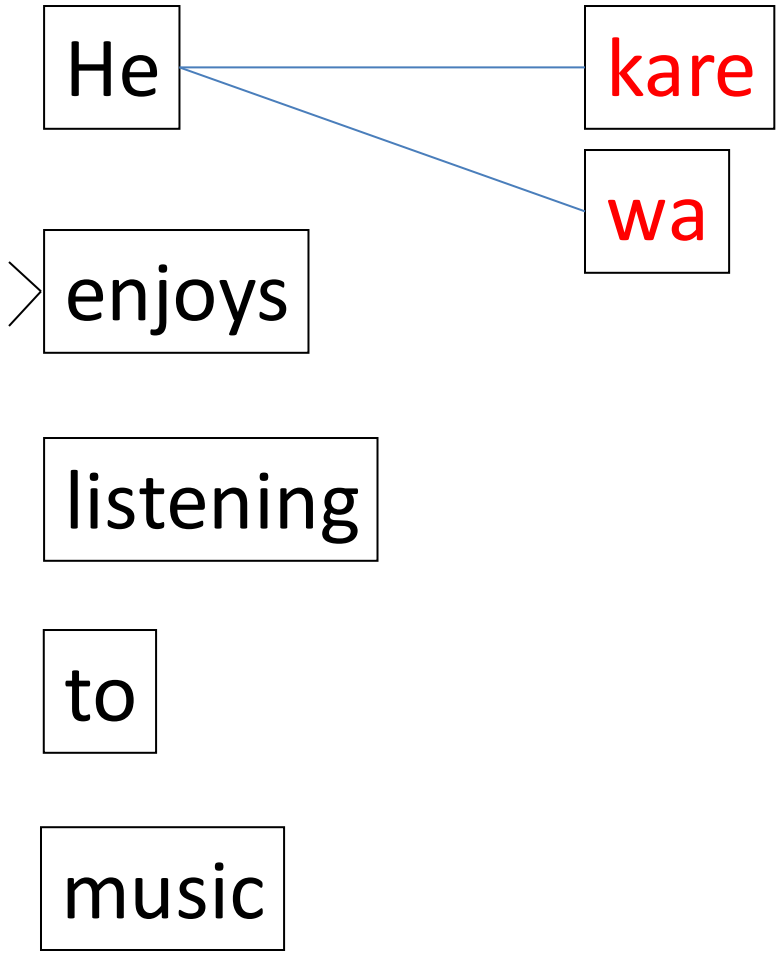
General-Purpose Algorithms

	String Automata Algorithms
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)
EM training	Forward-backward EM (Baum/Welch, 1971; Eisner 2003)
Determinization…	… of weighted string acceptors (Mohri, 1997)
Intersection	WFSA intersection
Applying transducers	string → WFST → WFSA
Transducer composition	WFST composition (Pereira & Riley, 1996)
General tools	FSM, Carmel, OpenFST

String Transduction for MT

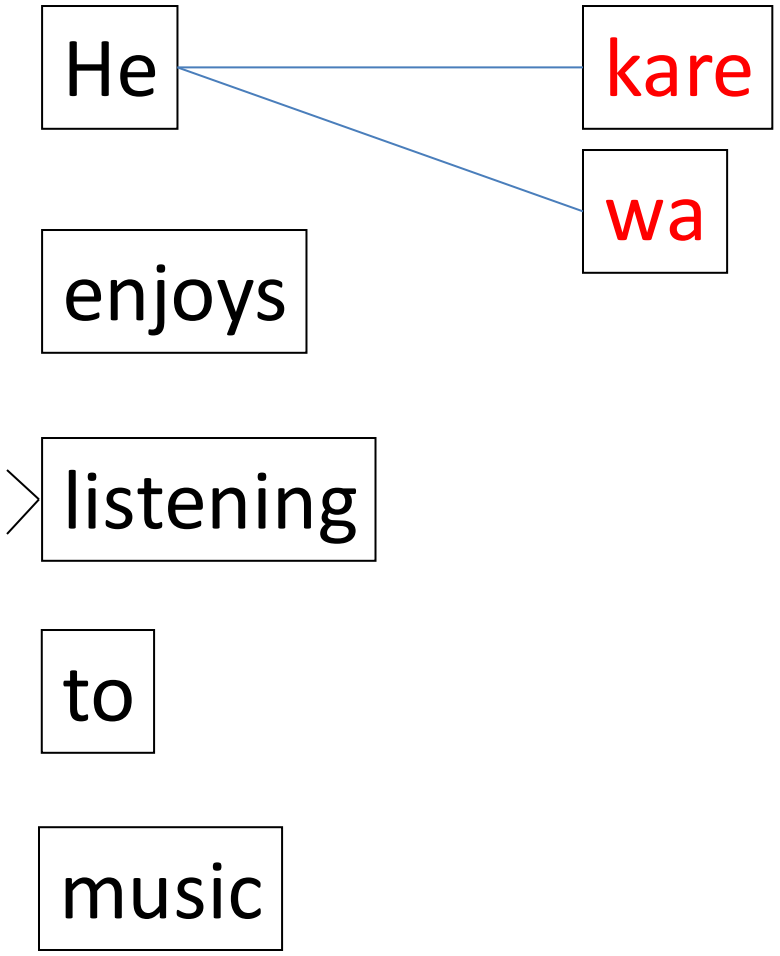


String Transduction for MT

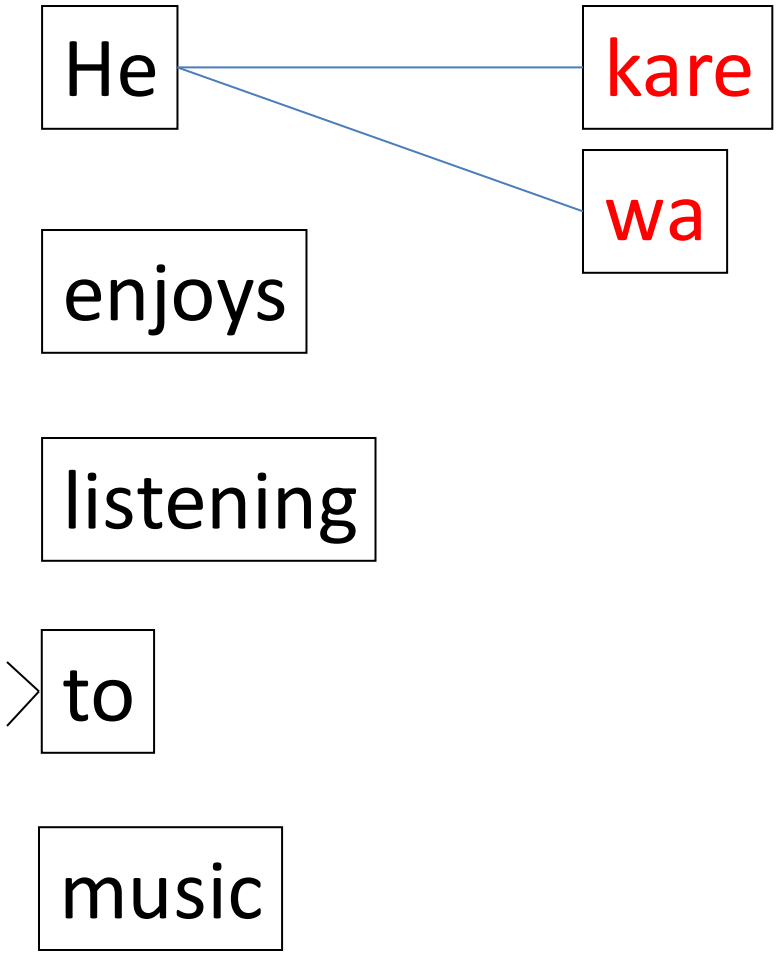


String Transduction for MT

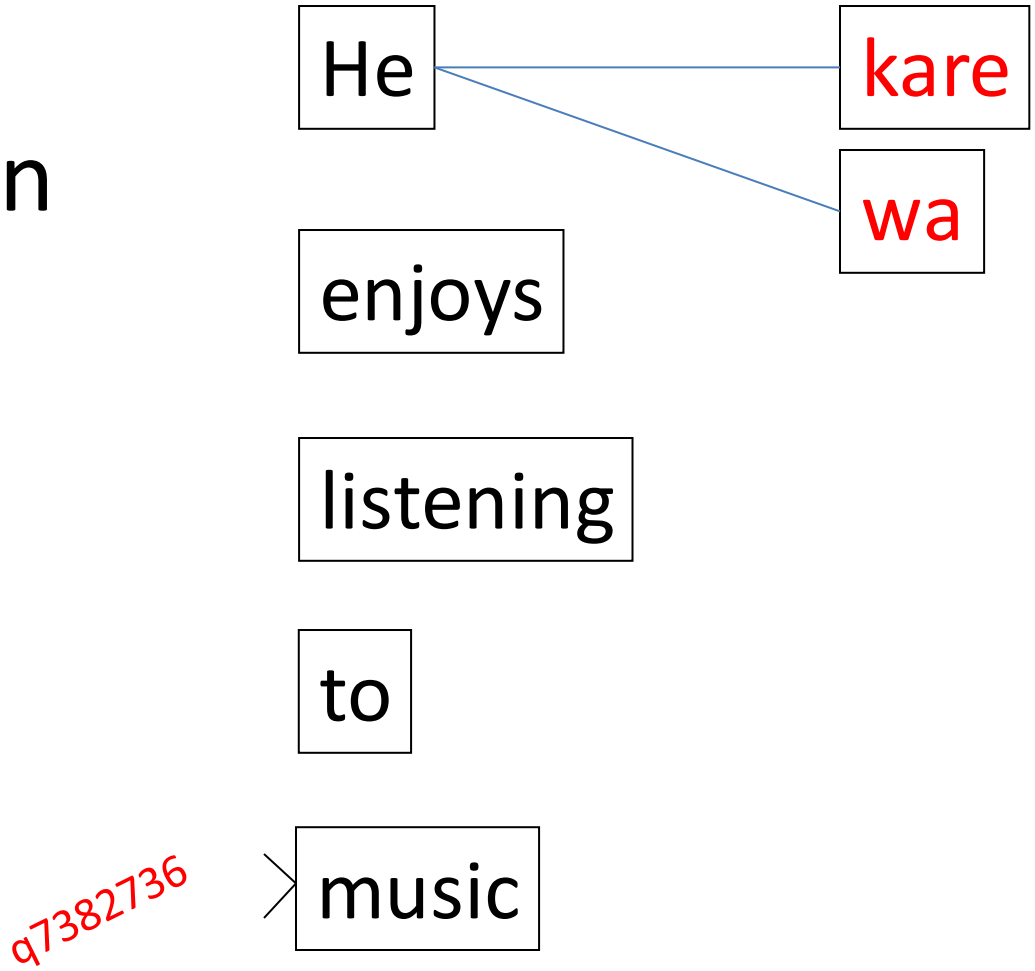
String Transduction for MT



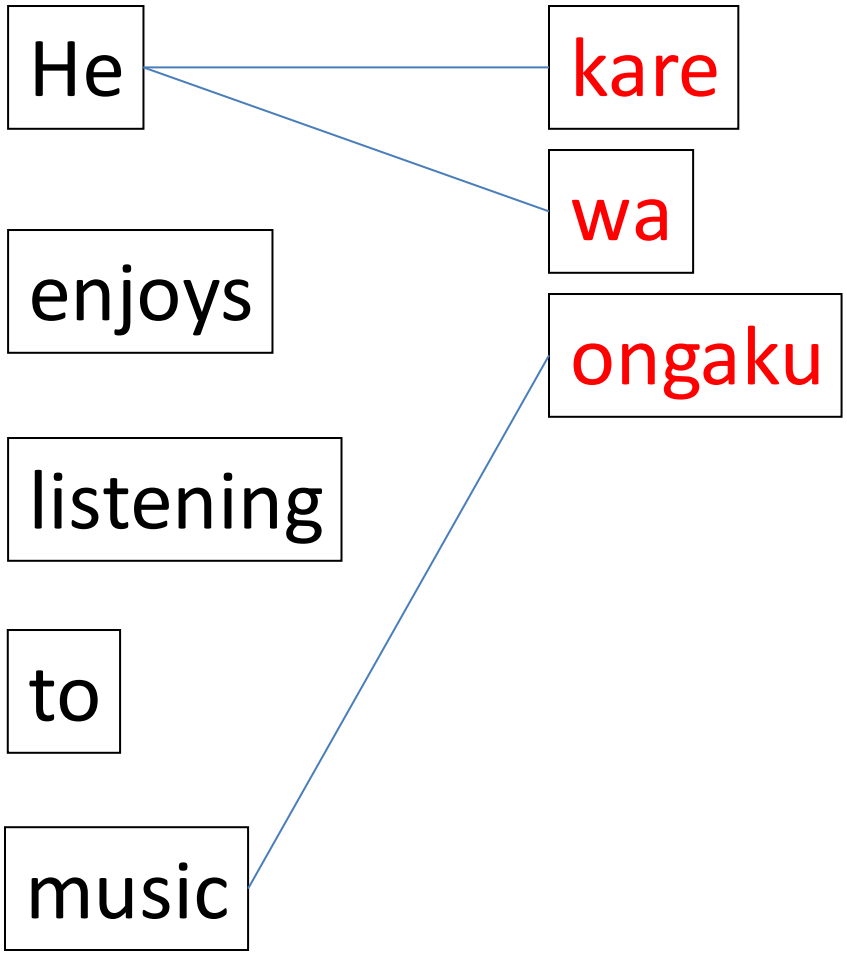
String Transduction for MT



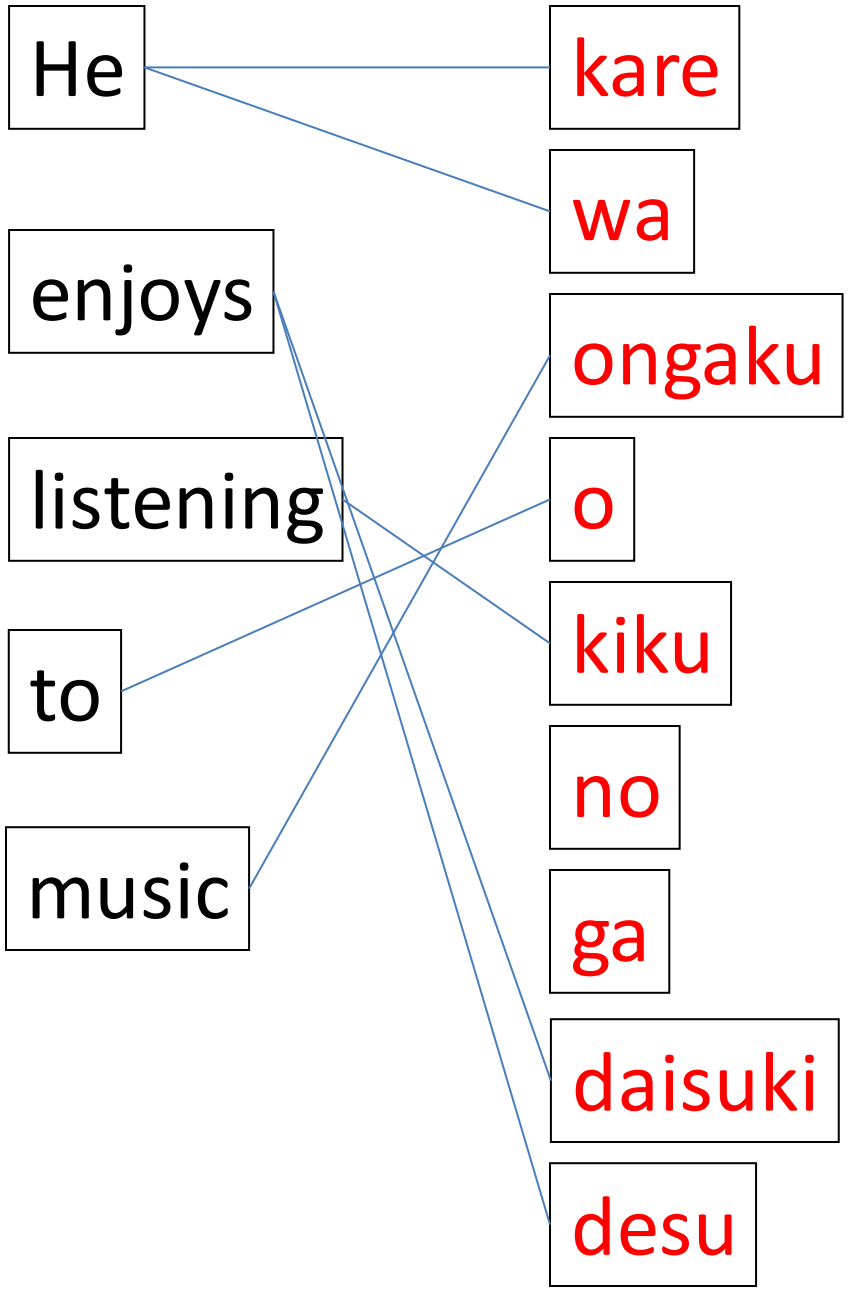
String Transduction for MT



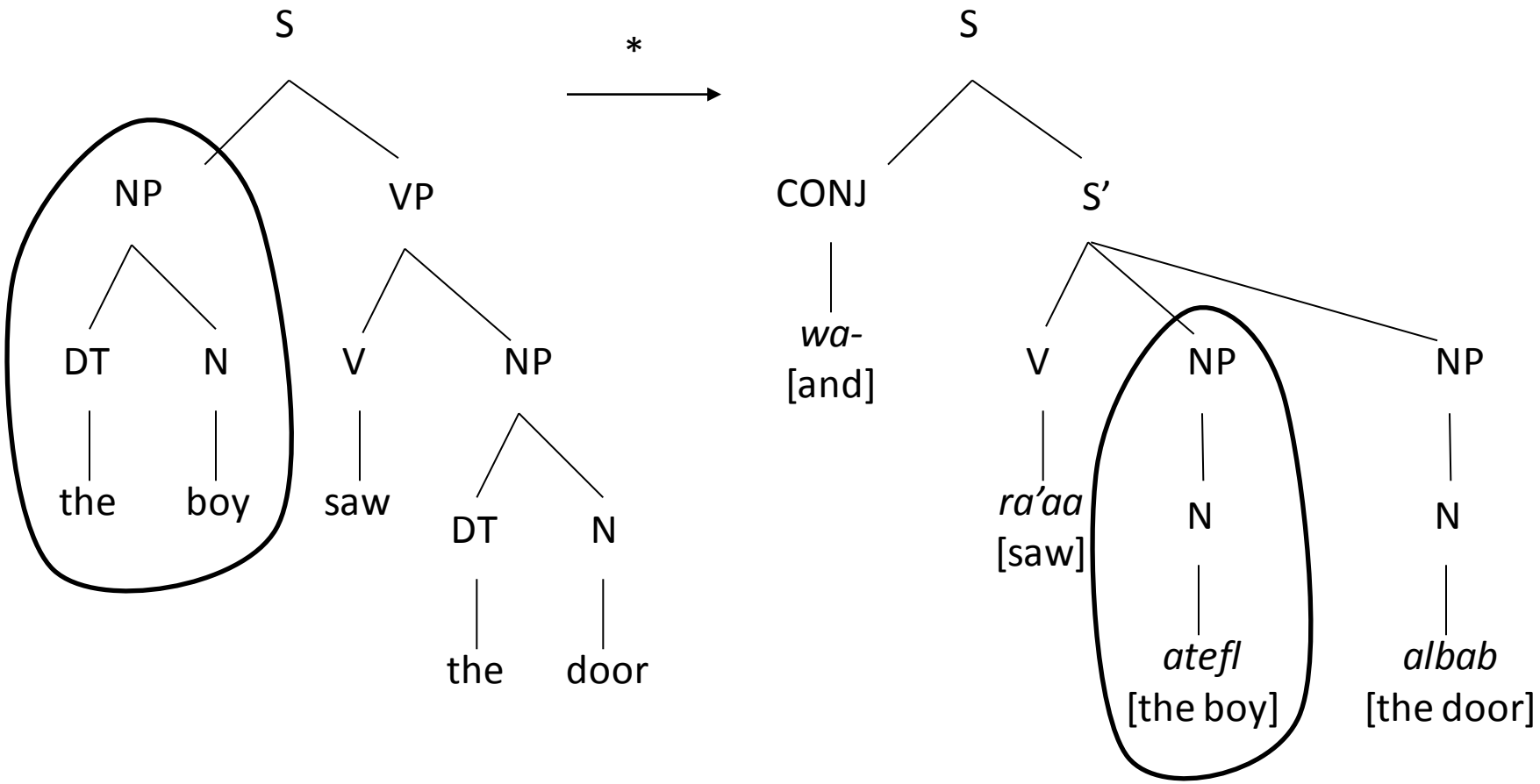
String Transduction for MT



String Transduction for MT



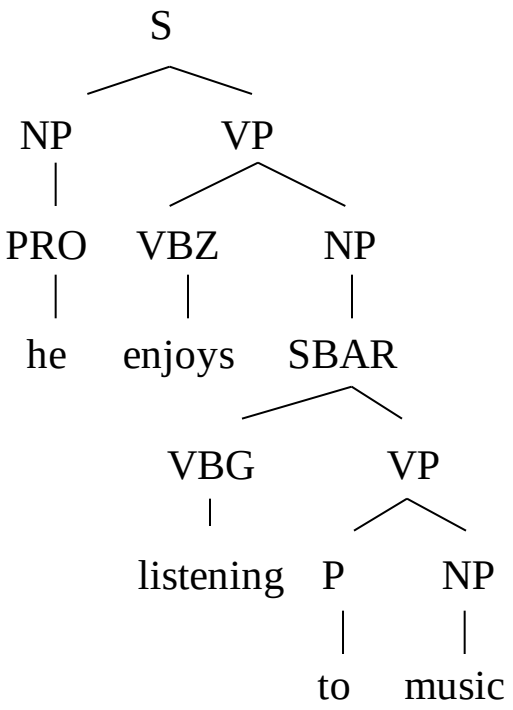
Top-Down Tree Transducer

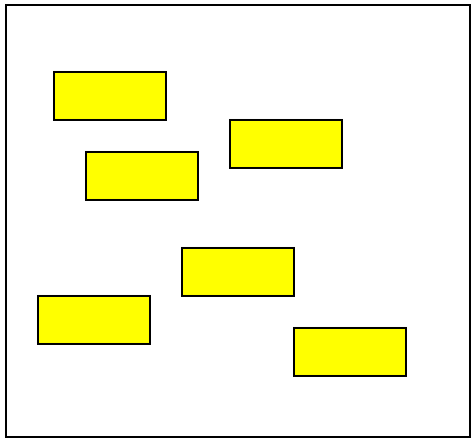


Top-Down Tree Transducer

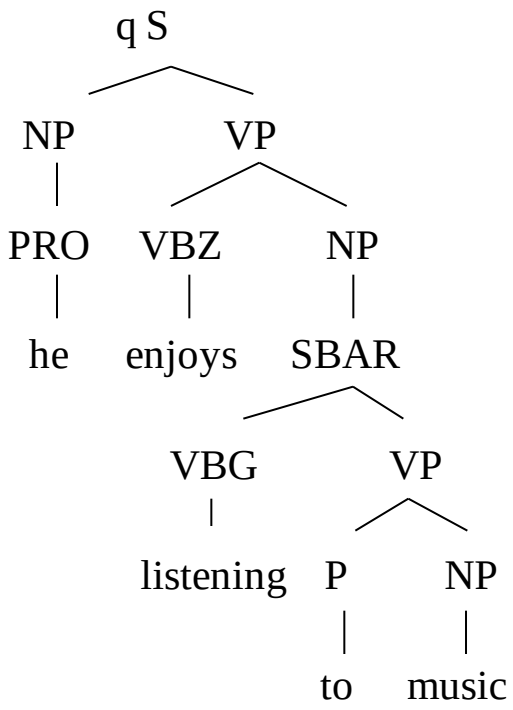
(W. Rounds 1970; J. Thatcher 1970)

Original input:



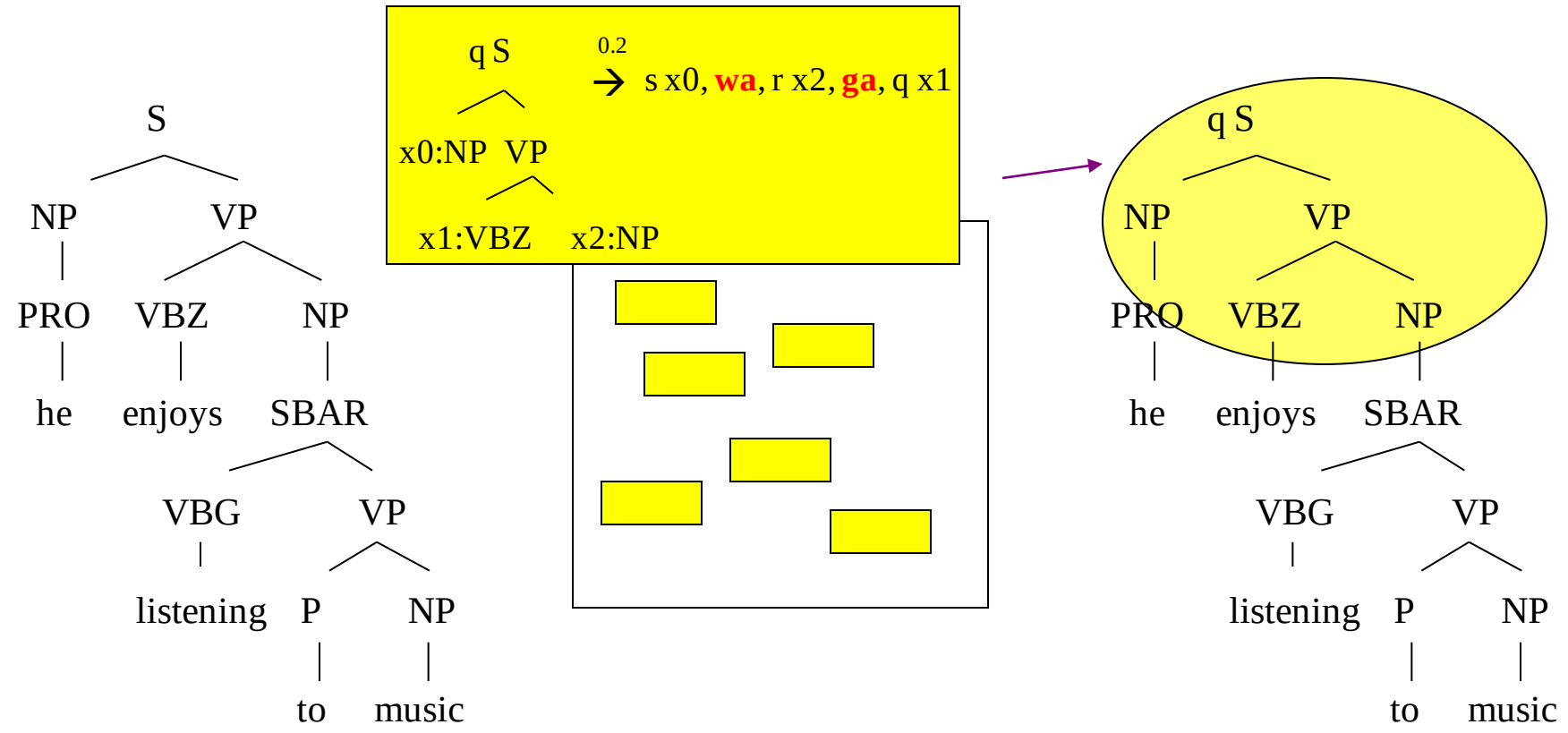


Transformation:



Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

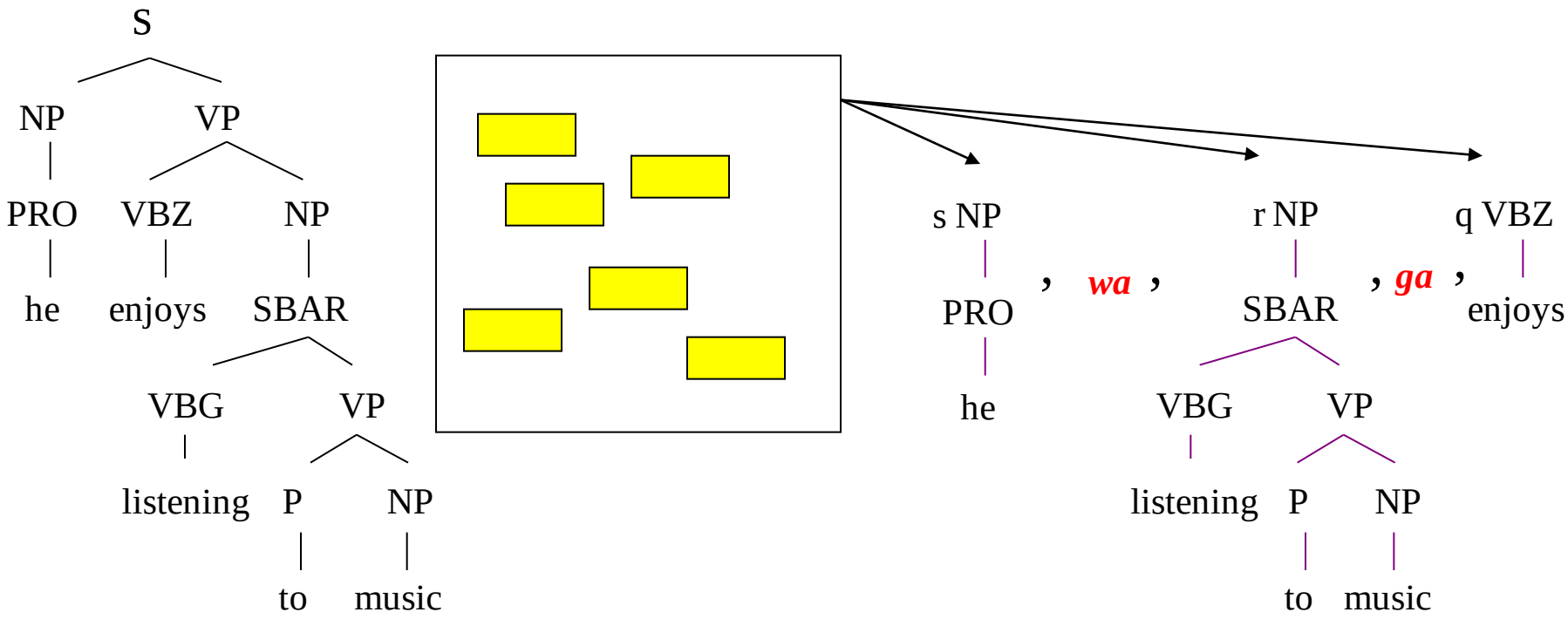


Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

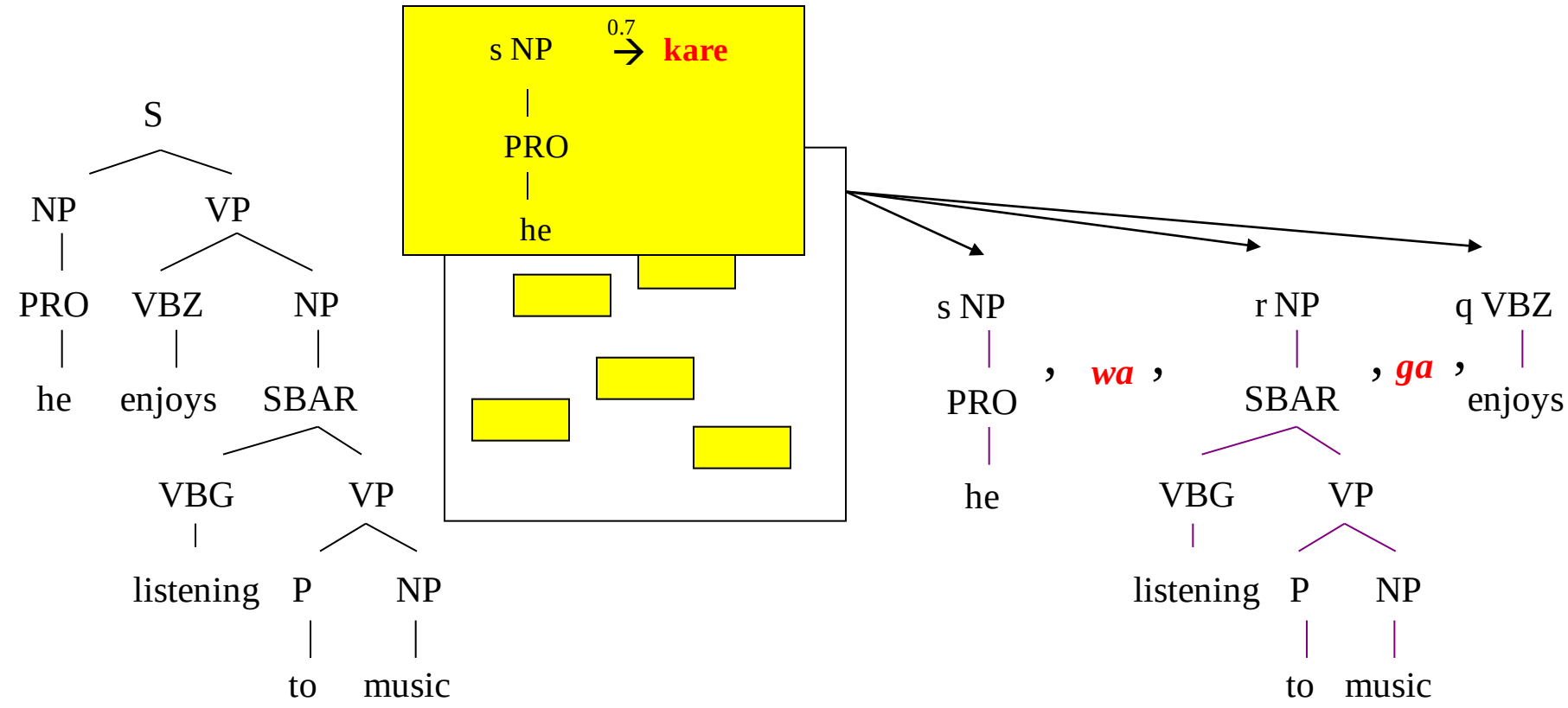
Original input:

Transformation:



Top-Down Tree Transducer

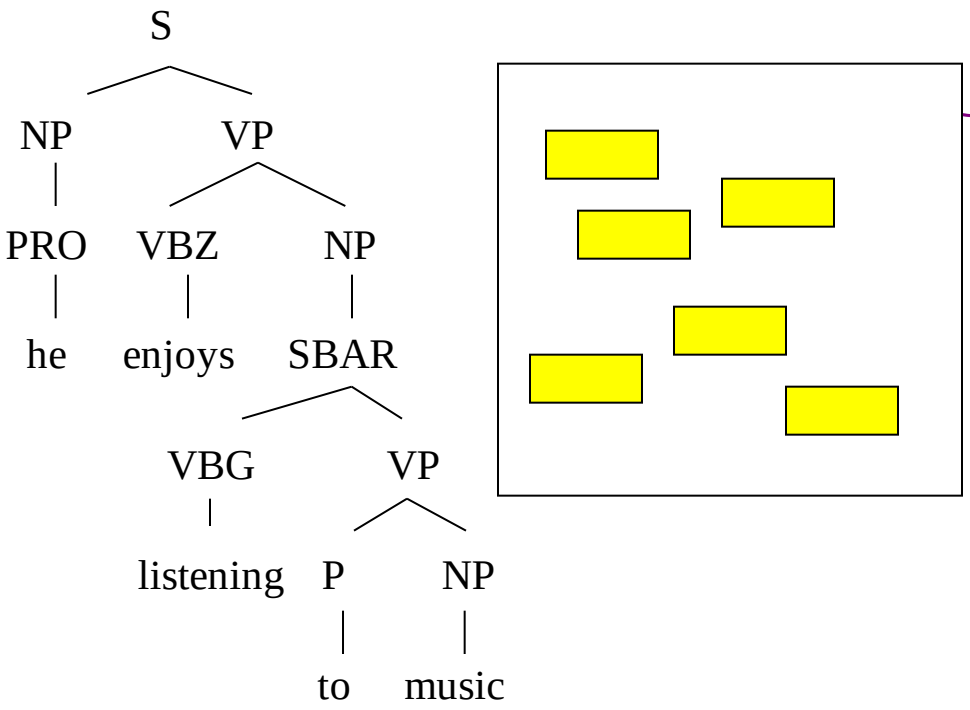
(W. Rounds 1970; J. Thatcher 1970)



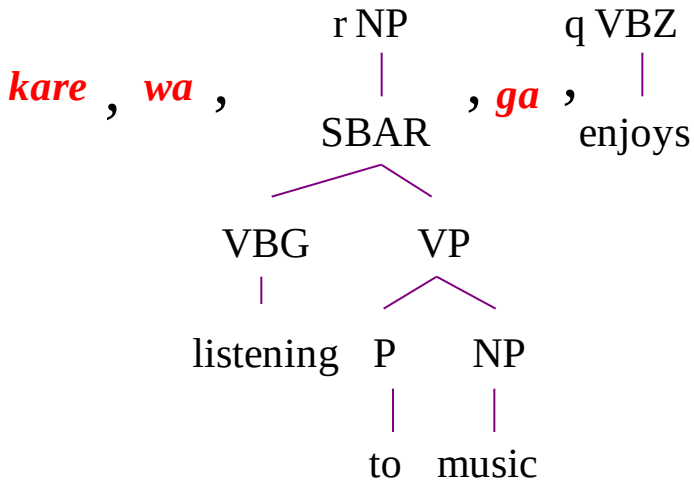
Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

Original input:



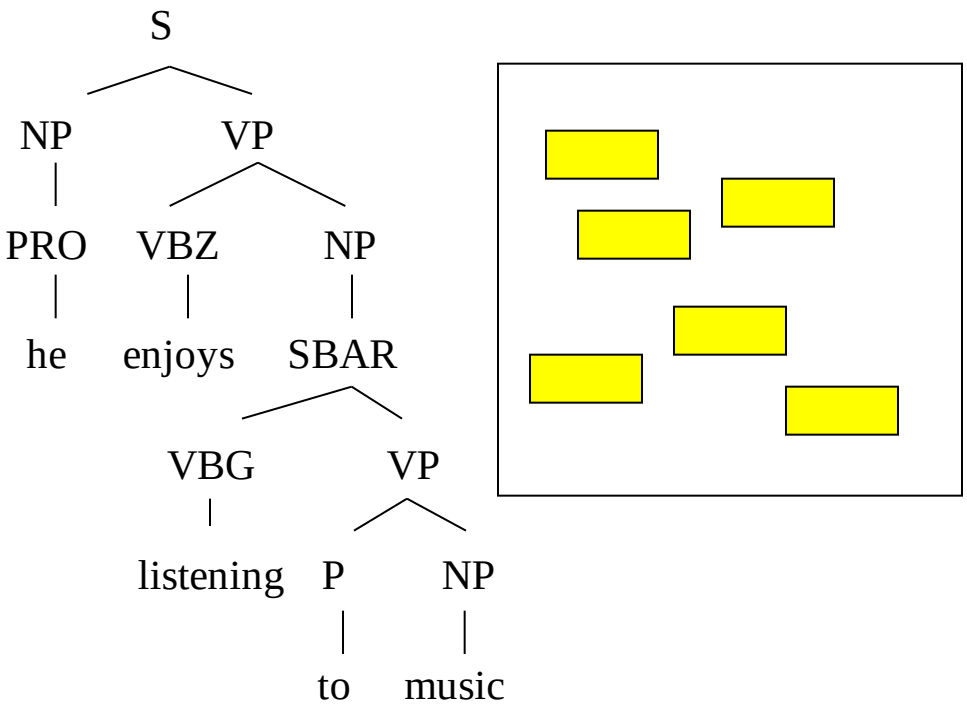
Transformation:



Top-Down Tree Transducer

(W. Rounds 1970; J. Thatcher 1970)

Original input:



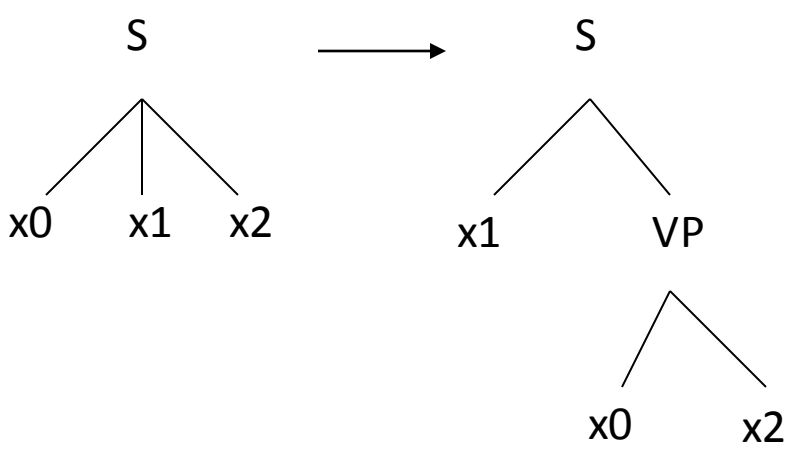
Final output:

kare*, *wa* , *ongaku*, *o* , *kiku*, *no*, *ga*, *daisuki*, *desu

Top-down Tree Transducers

every rule has this form

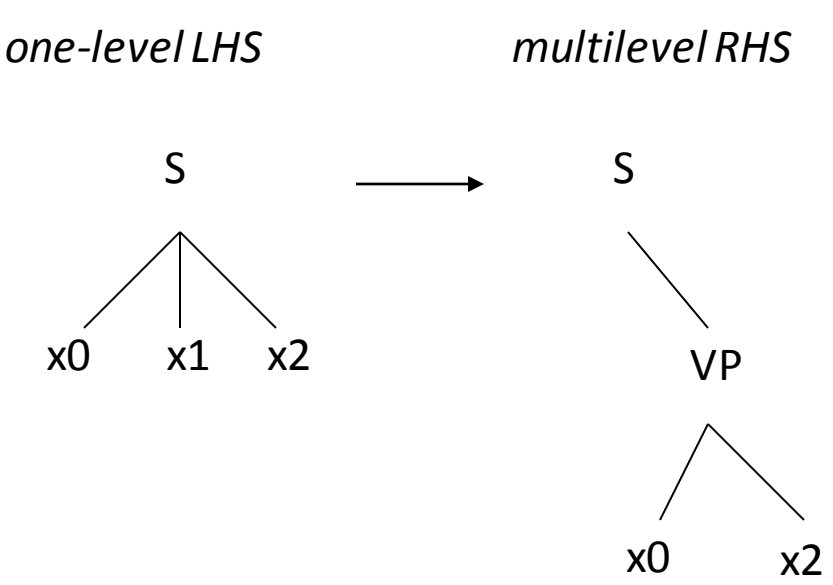
one-level LHS → *multilevel RHS*



T – top-down
L – linear (non-copying)
N – non-deleting

LNT

Top-down Tree Transducers

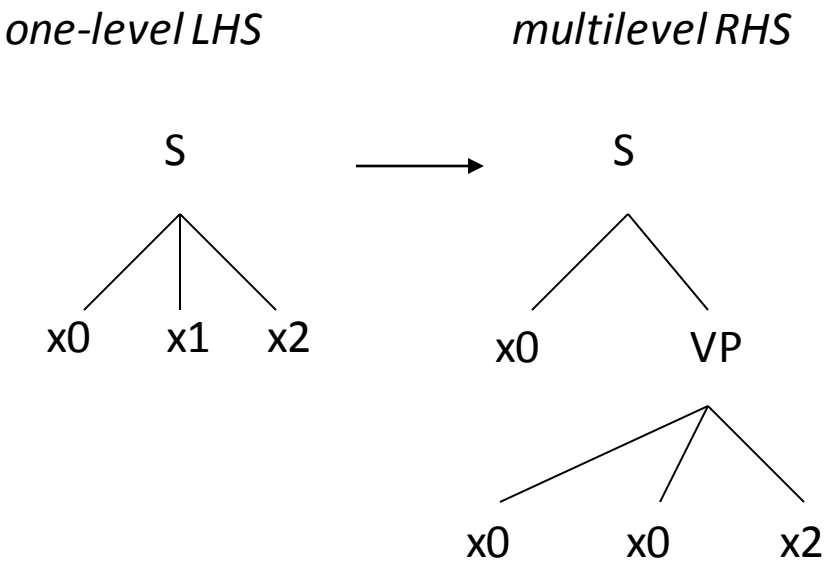


T – top-down
L – linear (non-copying)
N – non-deleting

LT

can delete subtrees

Top-down Tree Transducers

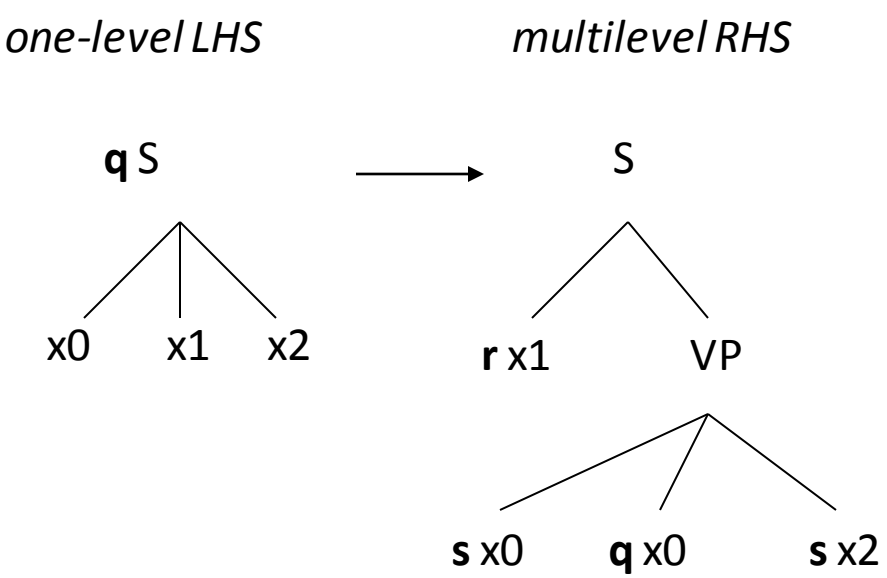


T – top-down
L – linear (non-copying)
N – non-deleting

T

can copy & delete subtrees

Top-down Tree Transducers



T – top-down
L – linear (non-copying)
N – non-deleting

T LT LNT

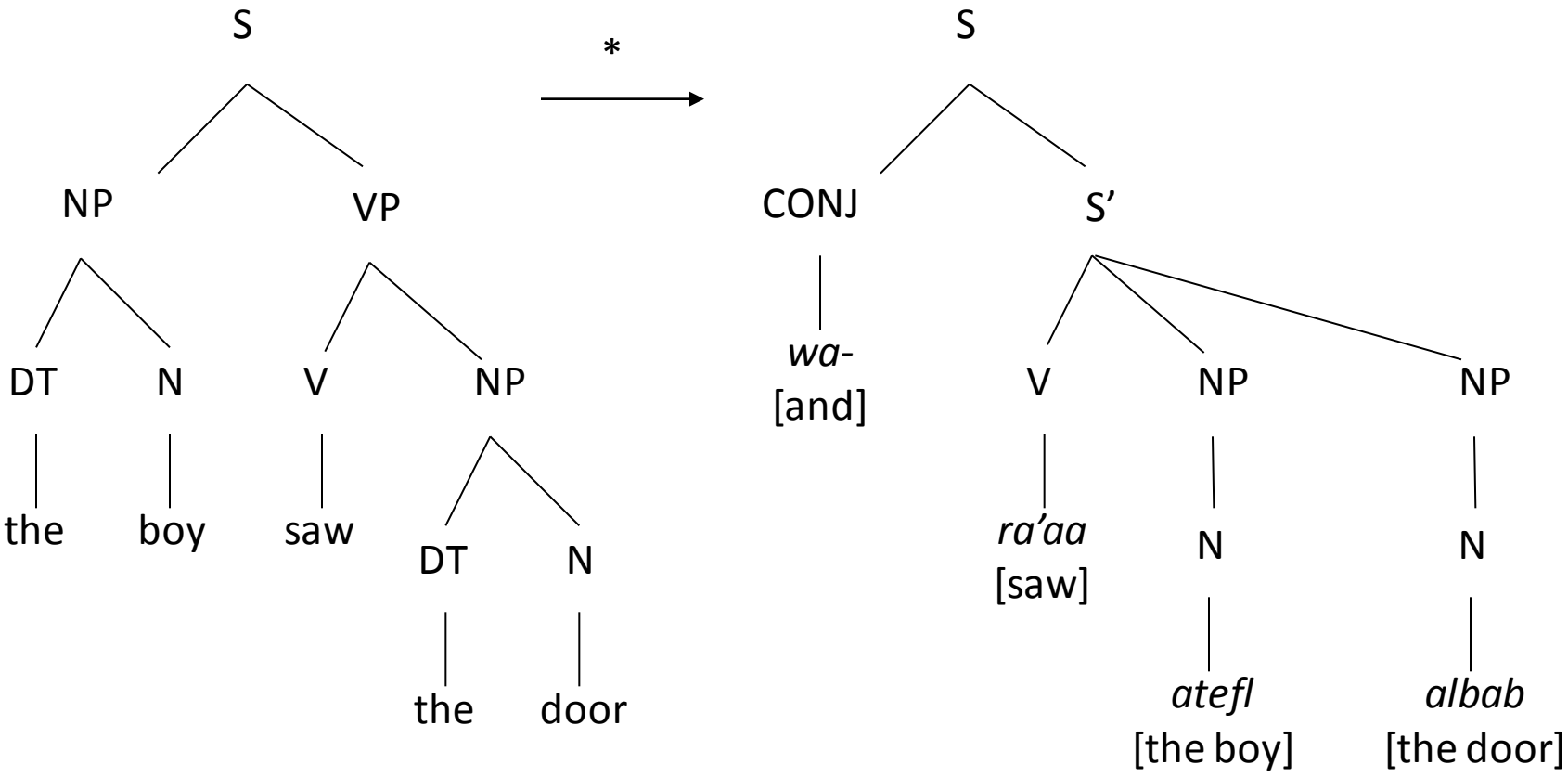
all employ **states**

Desirable Formal Properties

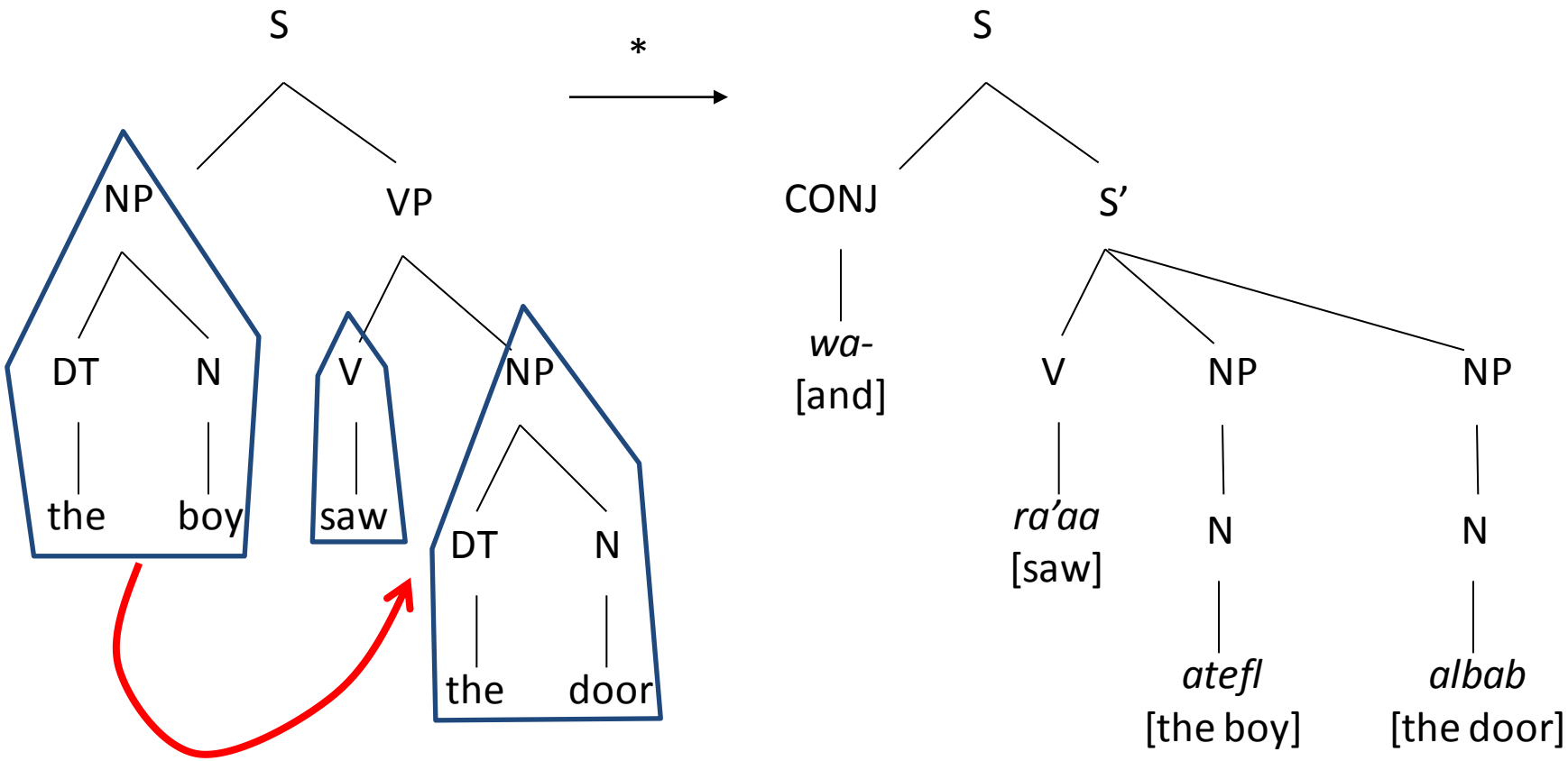
Expressiveness	Do local rotation, plus other stuff
Modularity	Be closed under composition
Inclusiveness	Capture any transformation that a string-based FST can
Teachability	Given input/output pairs, find rule probabilities to maximize likelihood

[Knight 07; see also Maletti]

Local Rotation

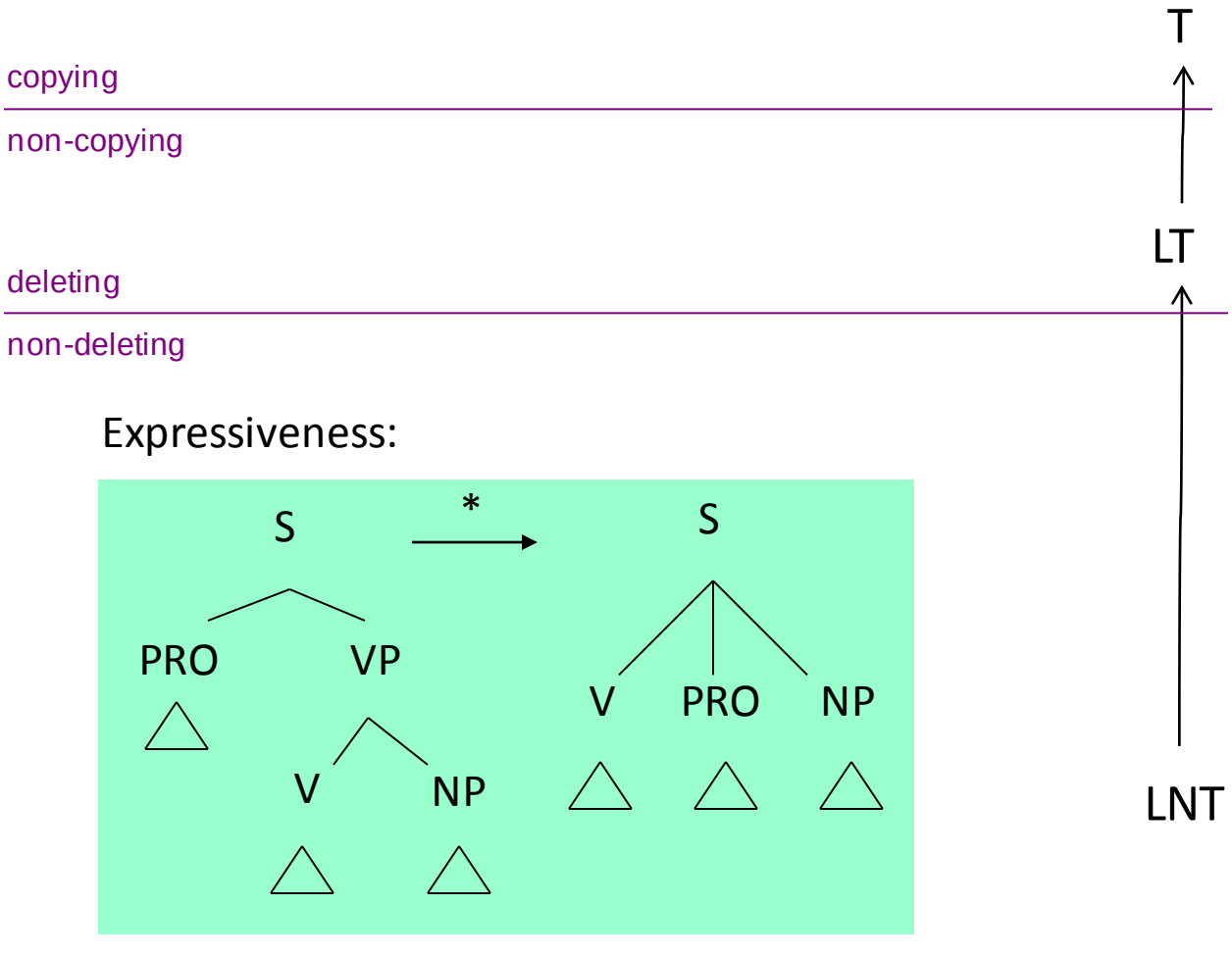


Local Rotation

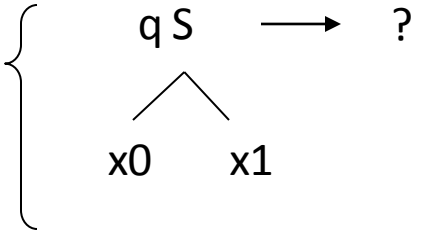


T – top-down
L – linear (non-copying)
N – non-deleting

T – top-down
L – linear (non-copying)
N – non-deleting



Expressiveness:



T – top-down
L – linear (non-copying)
N – non-deleting

copying

non-copying

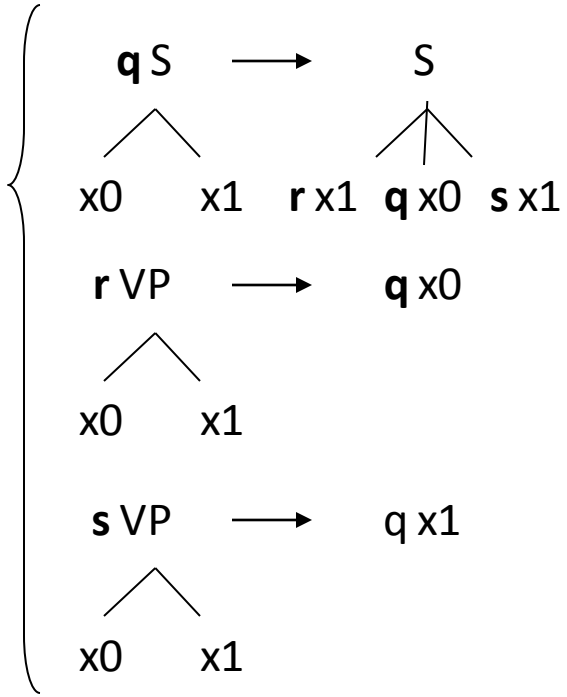
deleting

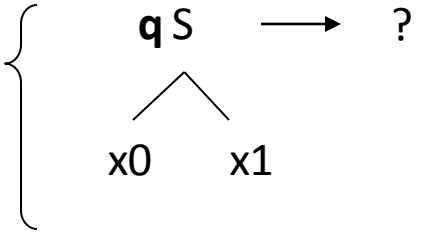
non-deleting

T

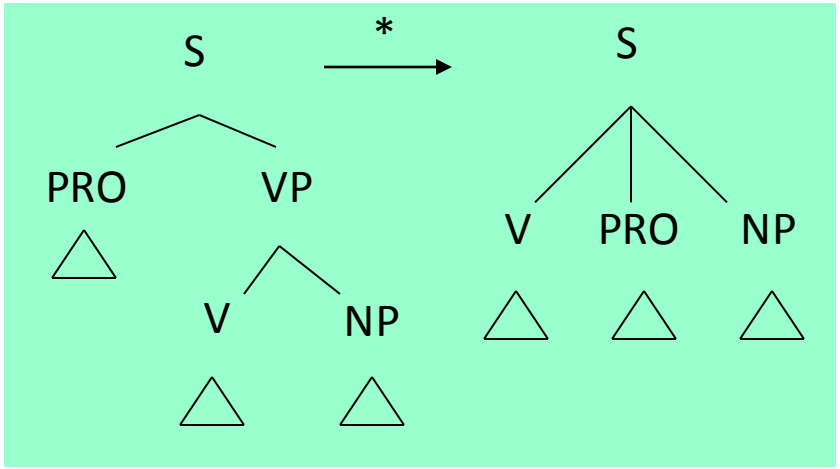
LT

LNT

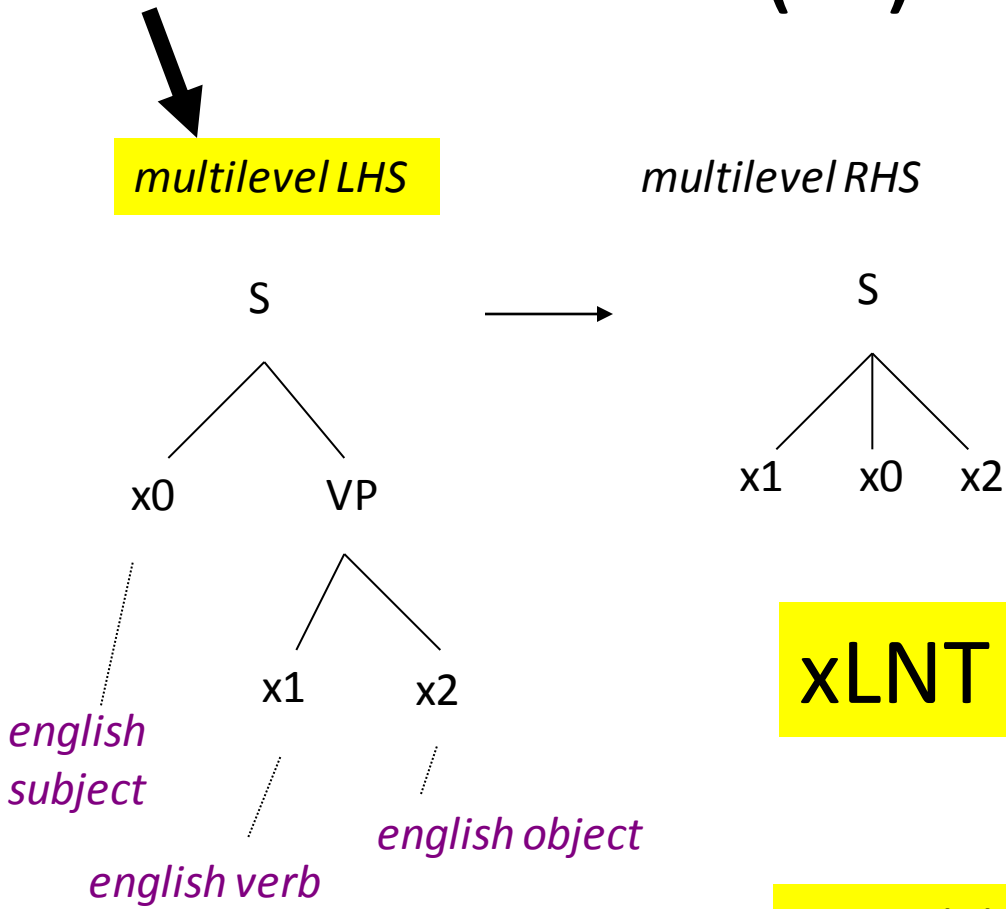




Expressiveness:



Extended (x-) Transducers

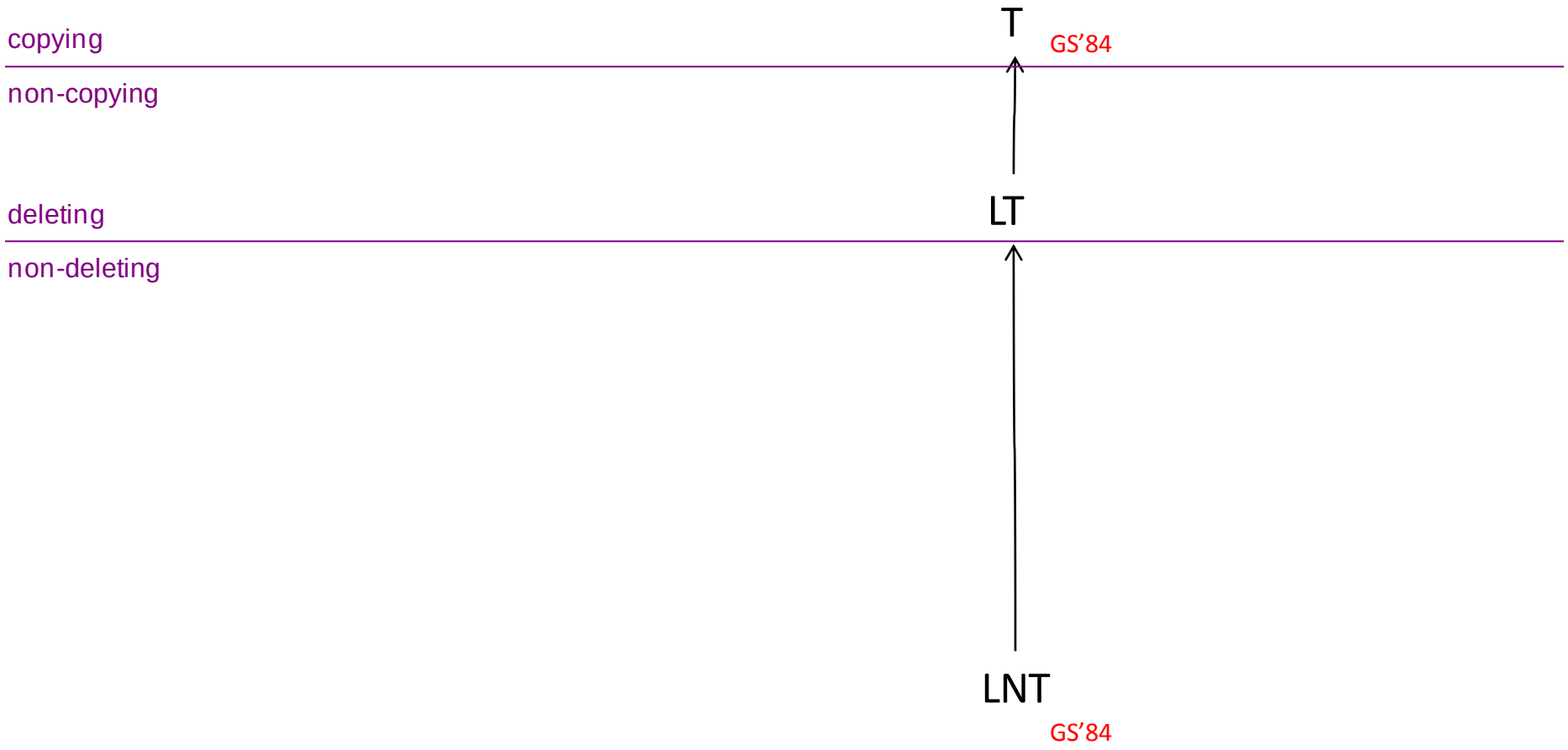


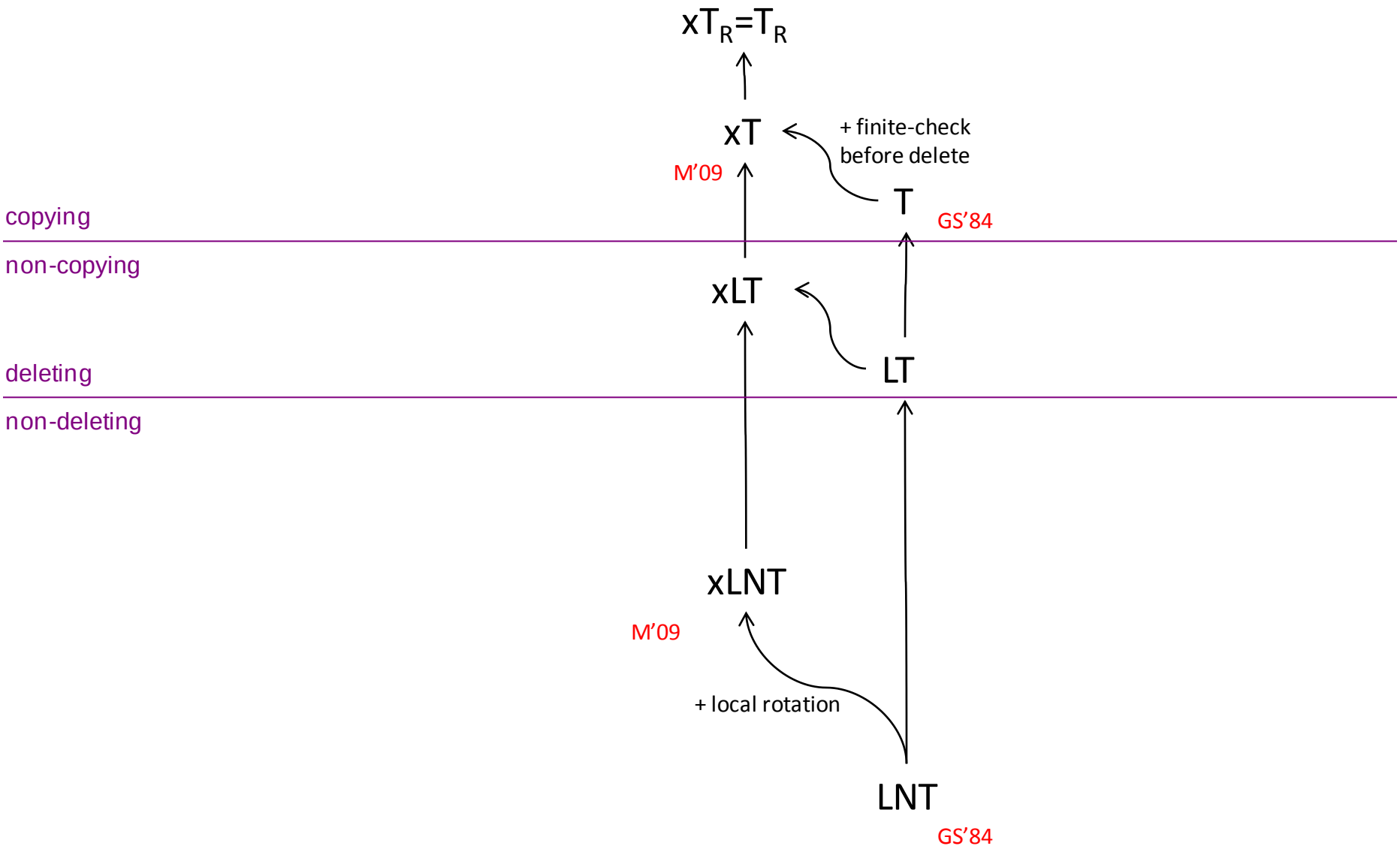
T – top-down
L – linear (non-copying)
N – non-deleting
x – extended LHS

xLNT

can grab more structure

- possibility mentioned in [Rounds 70]
- variant defined in [Dauchet 76]
- used for practical MT by [Galley et al 04, 06]
- studied formally by [Maletti et al 09]

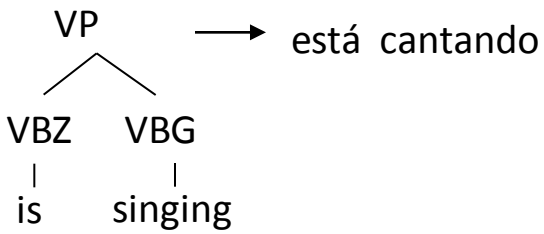

copying
non-copying
T
GS'84
deleting
non-deleting
LT
LNT
GS'84


$xT_R=T_R$
xT
M'09
+ finite-check
before delete
T
GS'84
copying
non-copying
xLT
LT
deleting
non-deleting
xLNT
M'09
+ local rotation
LNT
GS'84

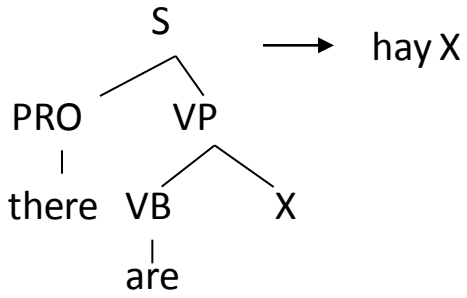
Expressiveness

other necessary things for machine translation

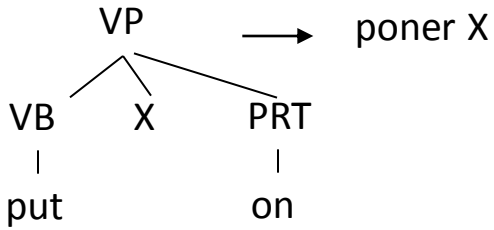
Phrasal Translation



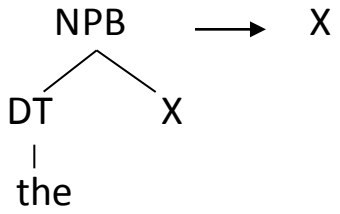
Non-constituent Phrases



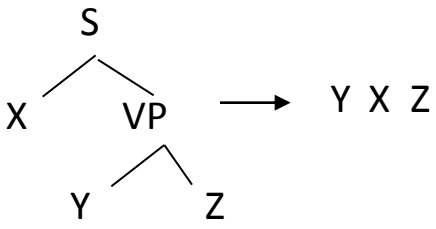
Non-contiguous Phrases



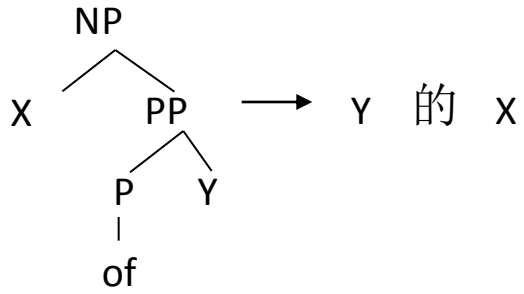
Context-Sensitive Word Insertion/Deletion

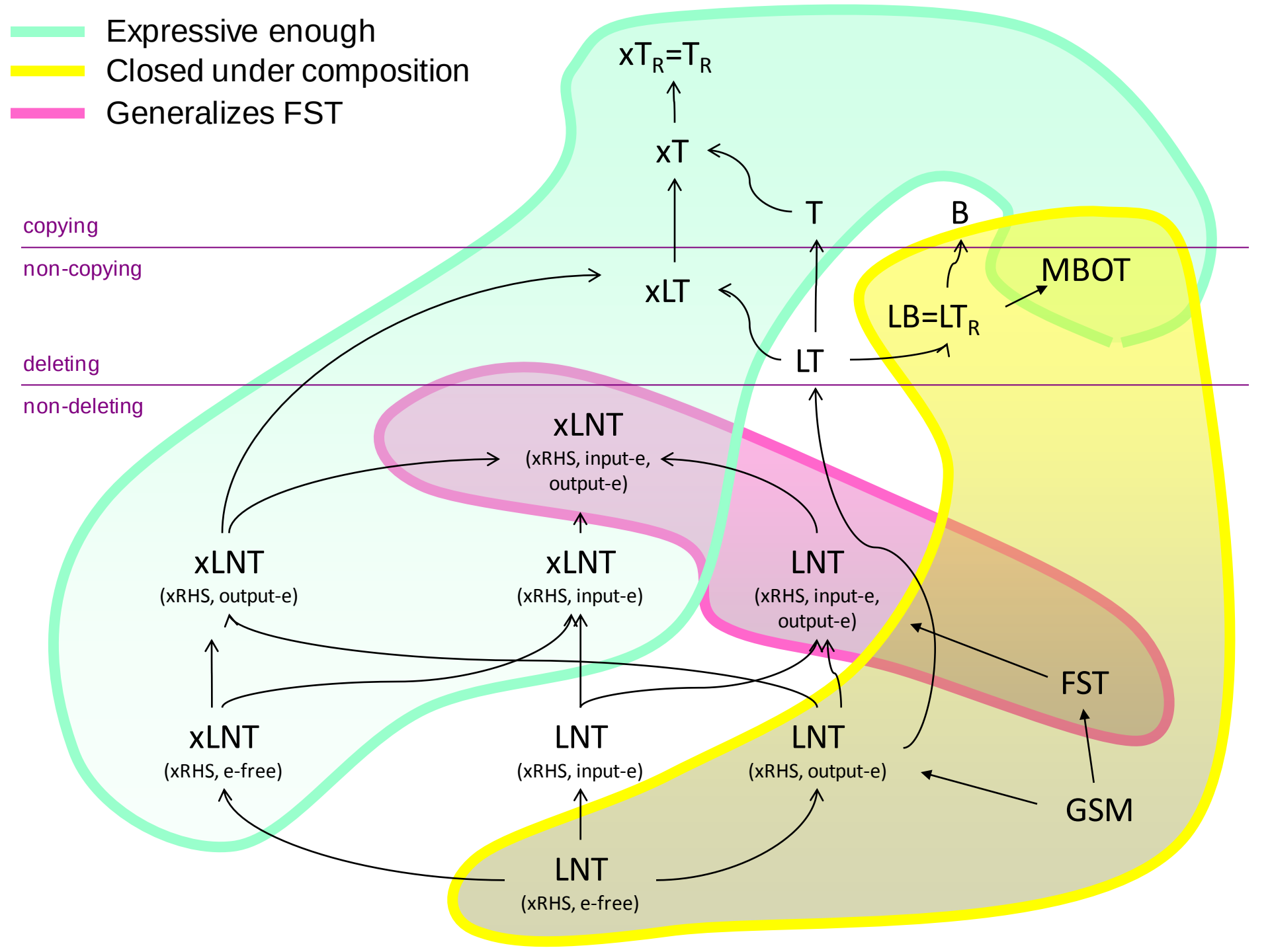


Re-Ordering



Lexicalized Re-Ordering




Expressive enough
Closed under composition
Generalizes FST
copying
non-copying
deleting
non-deleting
xT_R=T_R
xT
T
B
MBOT
xLT
LB=LT_R
LT
xLNT
(xRHS, input-e, output-e)
xLNT
(xRHS, output-e)
xLNT
(xRHS, input-e)
LNT
(xRHS, input-e, output-e)
FST
xLNT
(xRHS, e-free)
LNT
(xRHS, input-e)
LNT
(xRHS, output-e)
GSM
LNT
(xRHS, e-free)

General-Purpose Algorithms

	String Automata Algorithms	Tree Automata Algorithms
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)	… trees in a weighted forest (Jiménez & Marzal, 2000; Huang & Chiang, 2005)
EM training	Forward-backward EM (Baum/Welch, 1971; Eisner 2003)	Tree transducer EM training (Graehl & Knight, 2004)
Determinization…	… of weighted string acceptors (Mohri, 1997)	… of weighted tree acceptors (Borchardt & Vogler, 2003; May & Knight, 2005)
Intersection	WFSA intersection	Tree acceptor intersection
Applying transducers	string → WFST → WFSA	tree → TT → weighted tree acceptor
Transducer composition	WFST composition (Pereira & Riley, 1996)	Many tree transducers not closed under composition (Maletti et al 09)
General tools	FSM, Carmel, OpenFST	Tiburon (May & Knight 10)

Model Should Fit Data

What does it mean for a translation model to **fit the observed translation data**?

#1 Theory approach

#2 Linguistics approach

#3 Statistical approach

#4 Heroic approach

Fit to Data #2: Linguistics Approach

- Goal: Destroy enemy theory by torpedo

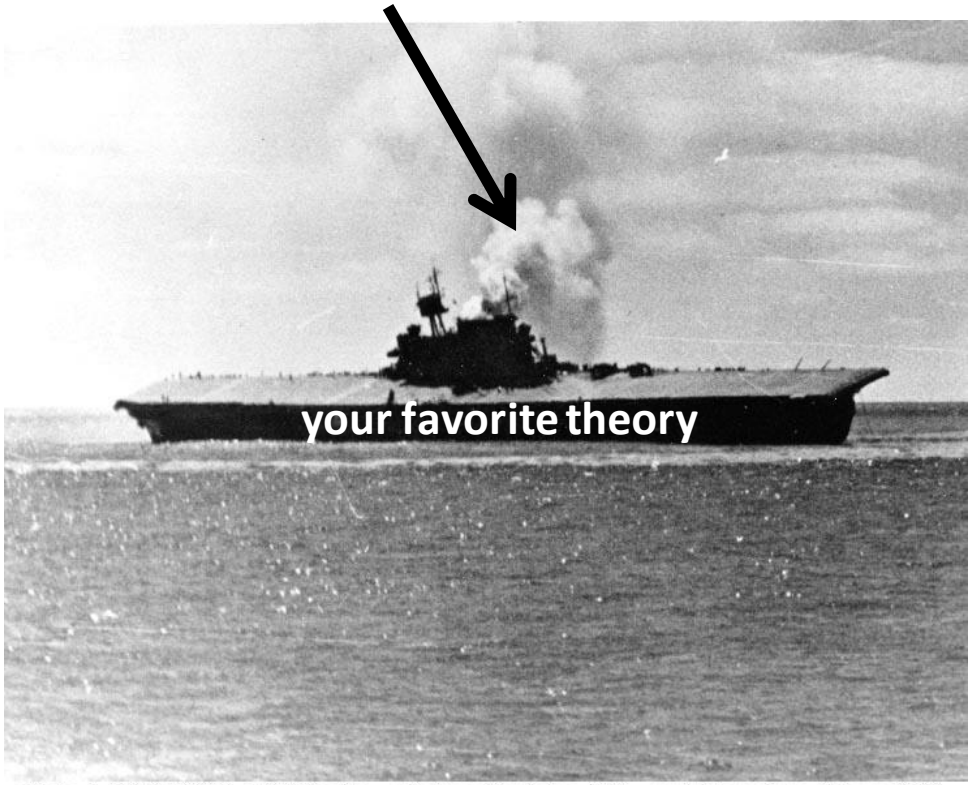
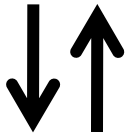


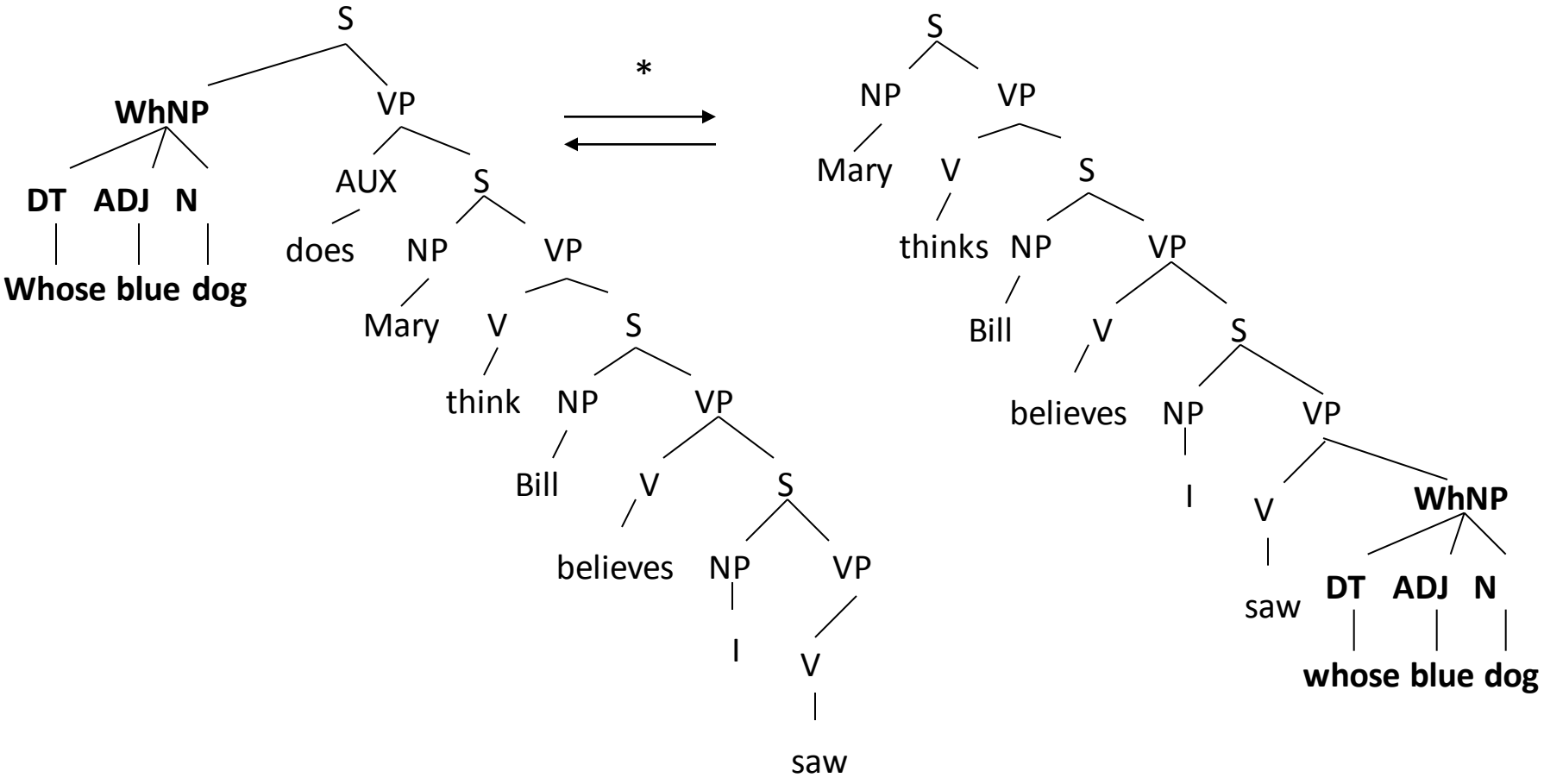
Photo # 80-G-17062 USS Yorktown listing after being hit by aerial torpedoes, 4 June 1942

Whose blue dog does Mary think Bill believes John said he saw?



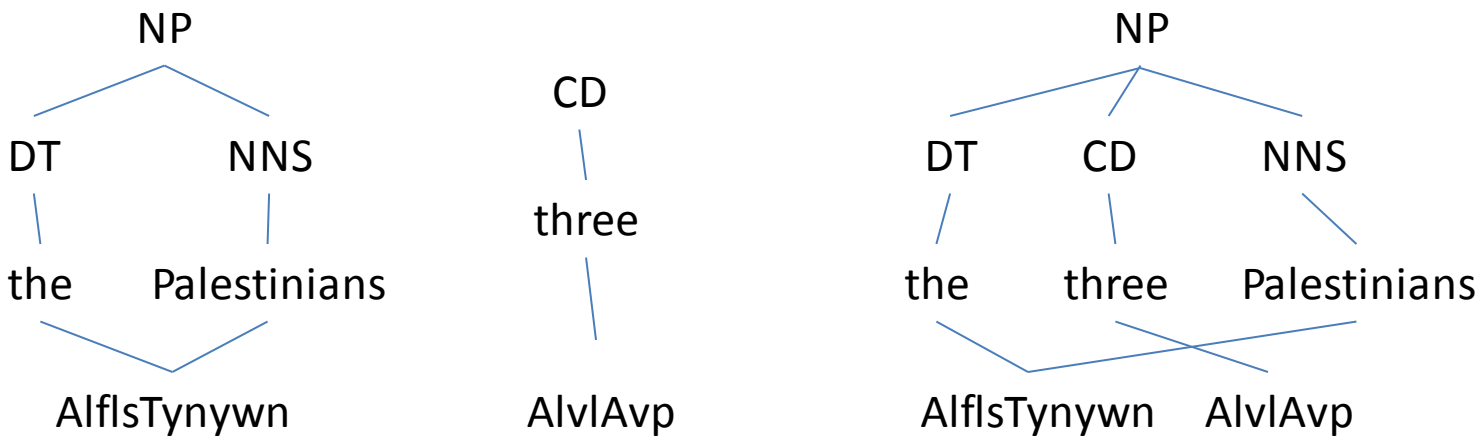
Mary thinks Bill believes John said he saw **whose blue dog**?
(Chinese)

Fit to Data #2: Linguistics Approach



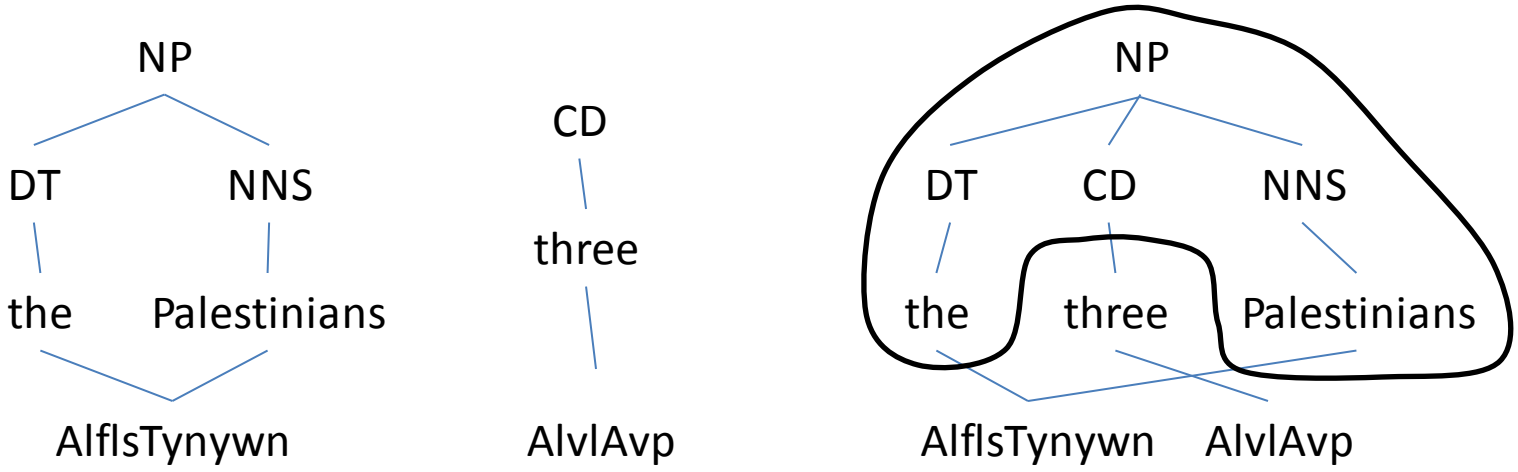
Synchronous TAG for MT

- [Abeille, Schabes, Joshi 90]
- [Nesson, Shieber, Rush 06]
- [Shieber 07]
- [DeNeefe 09, 10, 11]



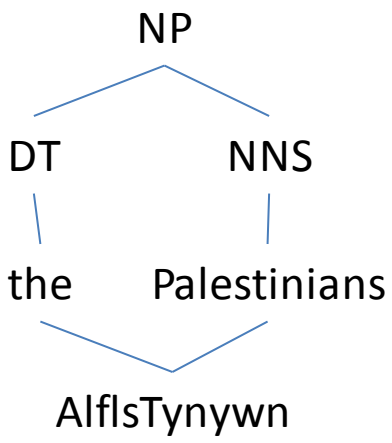
Synchronous TAG for MT

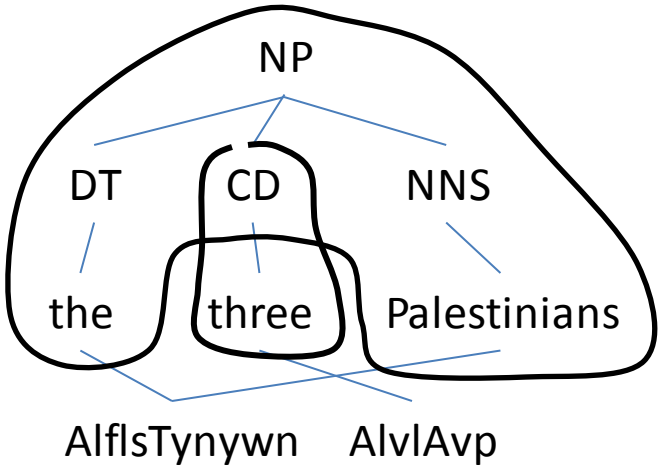
- [Abeille, Schabes, Joshi 90]
- [Nesson, Shieber, Rush 06]
- [Shieber 07]
- [DeNeefe 09, 10, 11]



Synchronous TAG for MT

- [Abeille, Schabes, Joshi 90]
- [Nesson, Shieber, Rush 06]
- [Shieber 07]
- [DeNeefe 09, 10, 11]





Fit to Data #2: Linguistics Approach

If you can explain data with good generalizations & limited exceptions, then you can simplify & compress that data ...

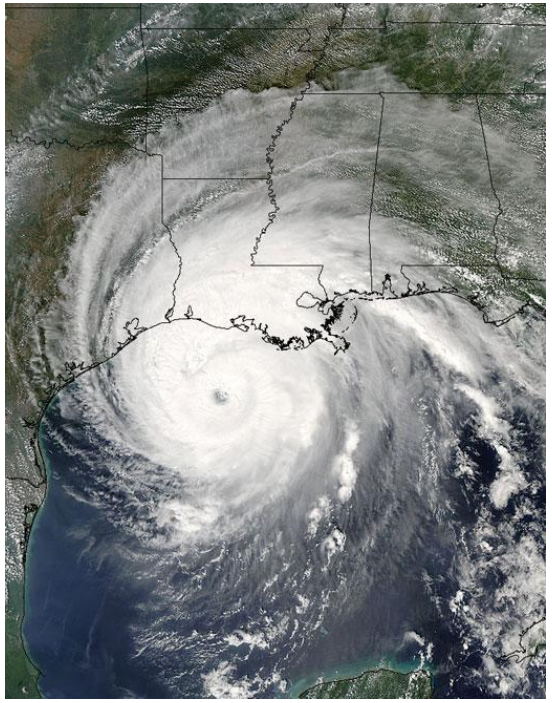
Um, doesn't data compression have something to do with ISO standards ... ?

Or worse ...

Fit to Data #3: Statistical Approach

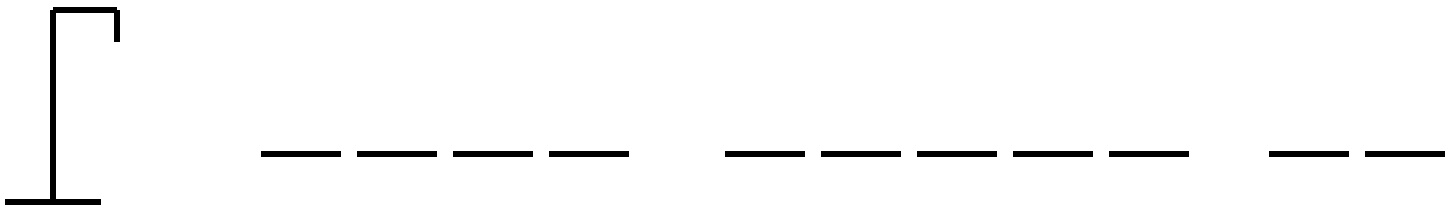
- Goal: Predict translation behavior
  - Model will always be wrong
  - Key: how to manage ignorance
  - Use past to predict future
  - Worry about the frequent stuff

  - If we could really predict human translator behavior well, we would be able to build good automatic translators ("do as a human would")



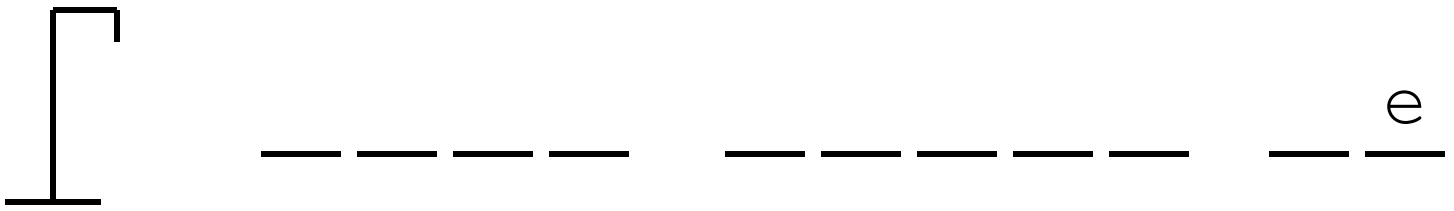
Shannon Hangman

- Hangman



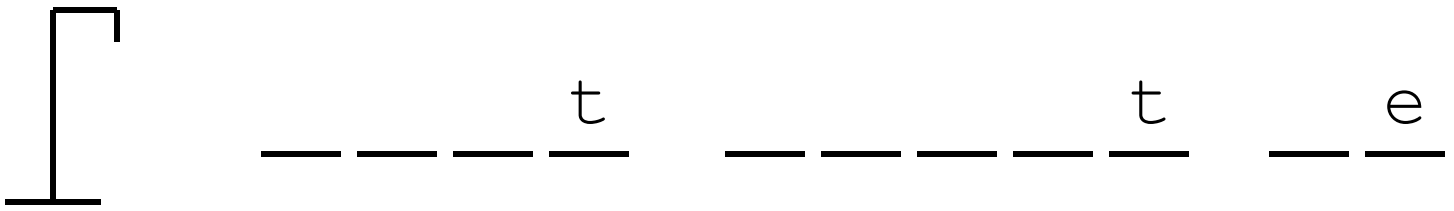
Shannon Hangman

- Hangman



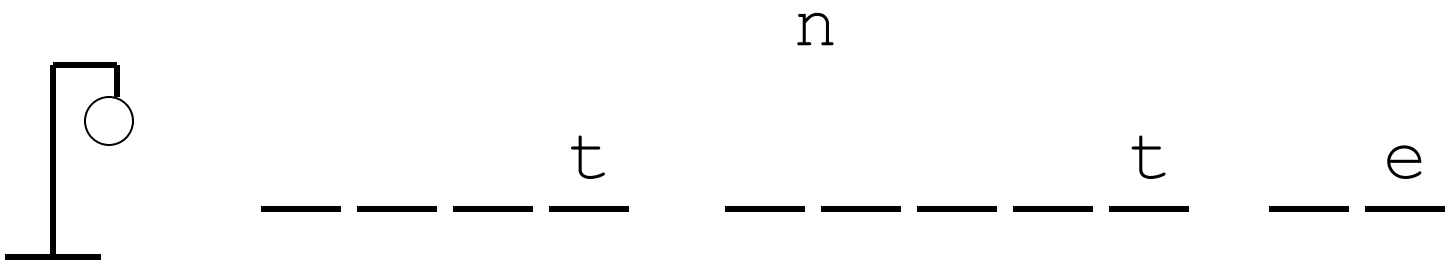
Shannon Hangman

- Hangman



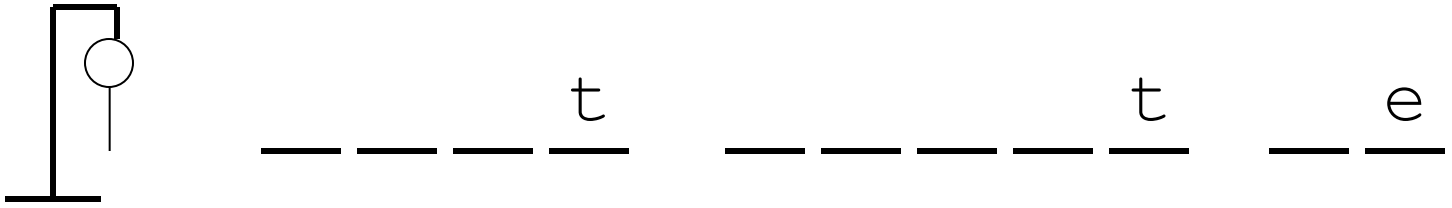
Shannon Hangman

- Hangman



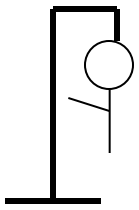
Shannon Hangman

- Hangman



Shannon Hangman

- Hangman

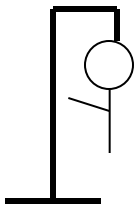


n i a

_ _ _ _ t _ _ _ _ _ _ _ t _ _ e

Shannon Hangman

- Hangman

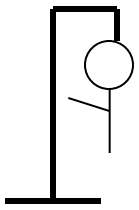


n i a

_ _ _ s t _ s _ _ _ _ t _ _ e

Shannon Hangman

- Hangman

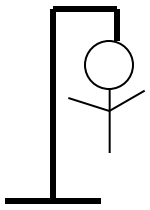


n i a

_ _ _ s t _ s h _ _ _ t _ e

Shannon Hangman

- Hangman



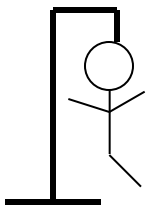
n i a b

_ _ _ s t _ s h _ _ _ t _ e

Shannon Hangman

- Hangman

n i a b r

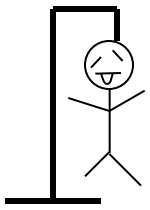


_ _ _ _ _ s t _ s h _ _ _ _ t _ _ e

Shannon Hangman

- Hangman

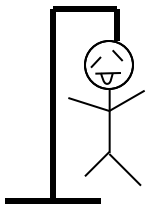
n i a b r q



_ _ _ _ _ s t _ s h _ _ _ _ t _ _ e

Shannon Hangman

- Hangman



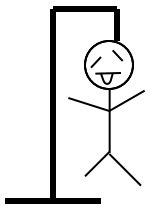
n i a b r q

_ _ _ s t _ s h _ _ _ t _ e

- Shannon Game is similar to hangman, except:
  - it's not fun
  - you go left to right
  - a correct guess only lights up one letter
  - they never hang you
  - you wish they would hang you

Shannon Hangman

- Hangman



n i a b r q

j u s t s h o o t m e

- Shannon Game is similar to hangman, except:
  - it's not fun
  - you go left to right
  - a correct guess only lights up one letter
  - they never hang you
  - you wish they would hang you

Shannon Hangman

- Estimates how well we can predict language.
- Guess sequence:

  1 4 16 1 1 1 74 …
- $P_{human}(guess) \sim P(1) \cdot P(4) \cdot P(16) \cdot P(1) \cdot P(1) \cdot P(1) \cdot P(74) \ldots$
- $H(guess) = - \log_2 P(guess) / N$
- Shannon's wife: 0.8-1.6

Shannon Hangman

English

model		H(guess)
human		1.9
char	1-gram	4.4
	2-gram	3.7
	3-gram	3.1
word	1-gram	2.8
	3-gram	1.8

Spain qualified for the World Cup Final, and will p_

?

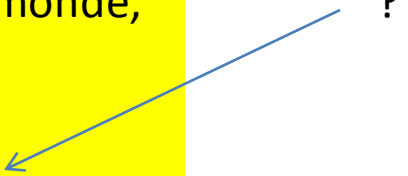
Experiments at ISI by Kyle Gorman and Kevin Knight

Shannon Hangman for Translation

		English	English given French
model		H(guess)	H(guess | f)
human		1.9	1.2
char	1-gram	4.4	
	2-gram	3.7	
	3-gram	3.1	
word	1-gram	2.8	2.2
	3-gram	1.8	

l'Espagne se qualifie pour la finale de la Coupe du monde, et affrontera dimanche les Pays-Bas.

Spain qualified for the World Cup Final, and will p_

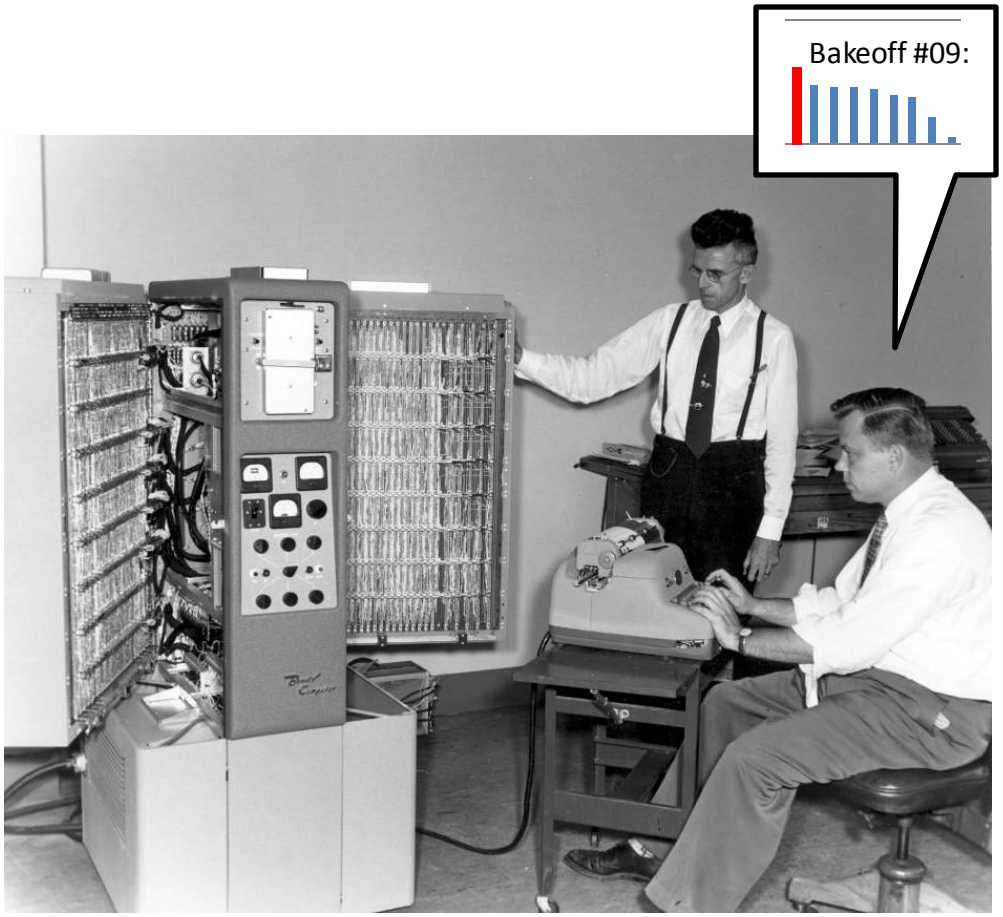


?

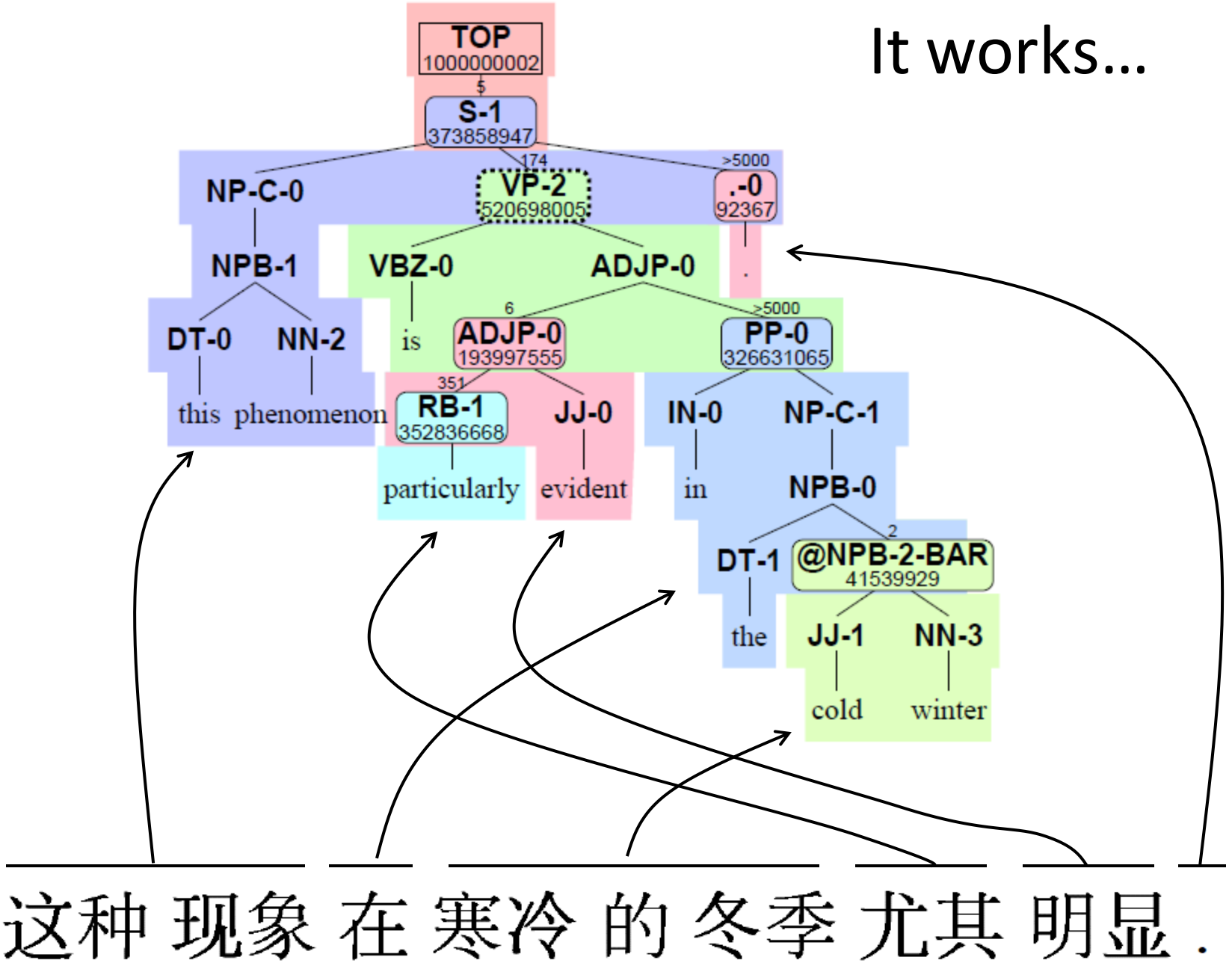
Experiments at ISI by Kyle Gorman and Kevin Knight

Fit to Data #4: Engineering Approach

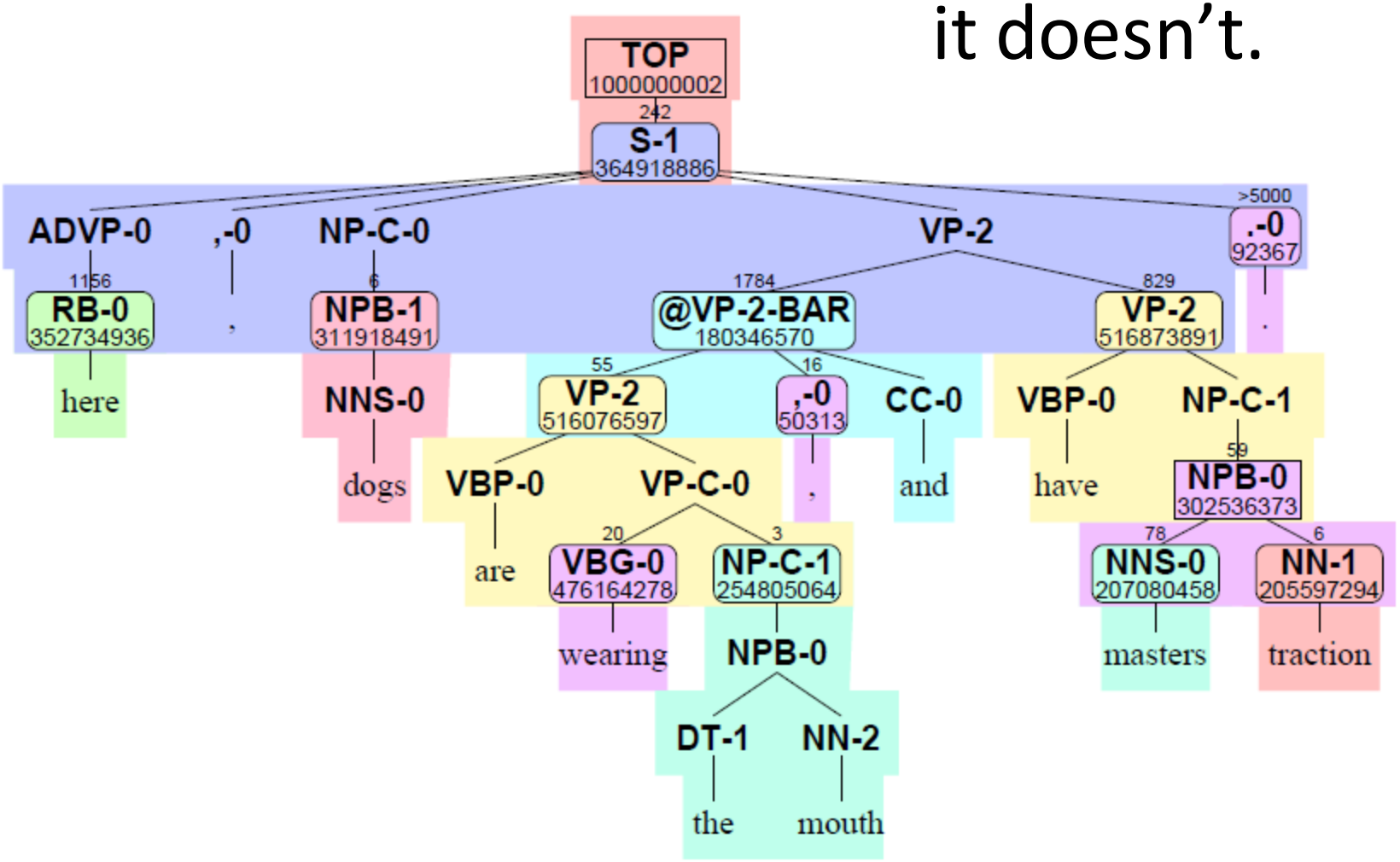
- Goal: Improve machine translation quality
  - Pick idea
  - Write program!
  - Get bugs out
  - Get more bugs out!
  - Evaluate!!
  - Add a feature
  - Clean the data
  - Evaluate
  - Iterate!



It works…

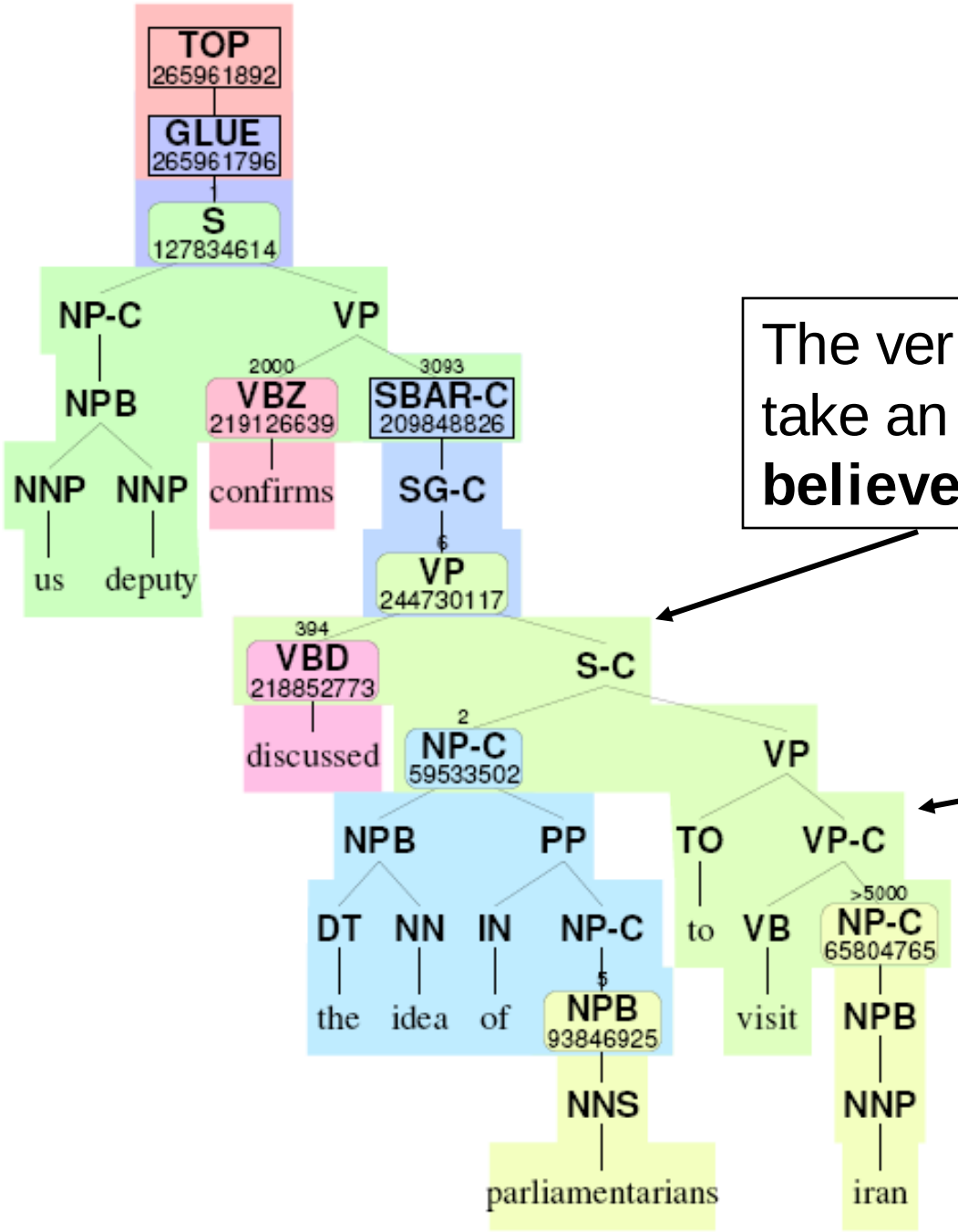

TOP
1000000002
5
S-1
373858947
NP-C-0
174
VP-2
520698005
>5000
.-0
92367
NPB-1
VBZ-0
ADJP-0
.
DT-0
NN-2
is
6
ADJP-0
193997555
>5000
PP-0
326631065
this
phenomenon
351
RB-1
352836668
JJ-0
IN-0
NP-C-1
particularly
evident
in
NPB-0
2
DT-1
@NPB-2-BAR
41539929
the
JJ-1
NN-3
cold
winter
这种 现象 在 寒冷 的 冬季 尤其 明显 .

…except when it doesn't.



在这里，狗都配戴嘴套，并有主人牵引。

…except when it doesn't.

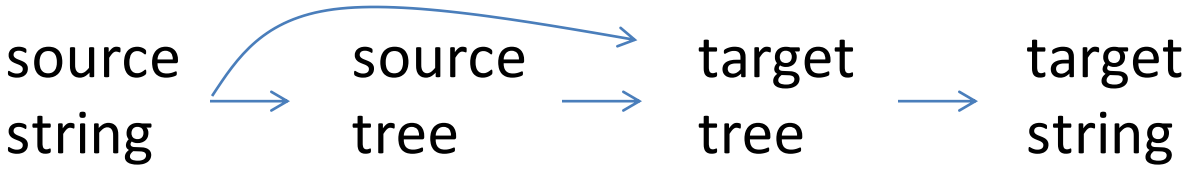


The verb **discuss** doesn't take an S argument (like **believe** and **realize** do)

An idea can't visit a place!

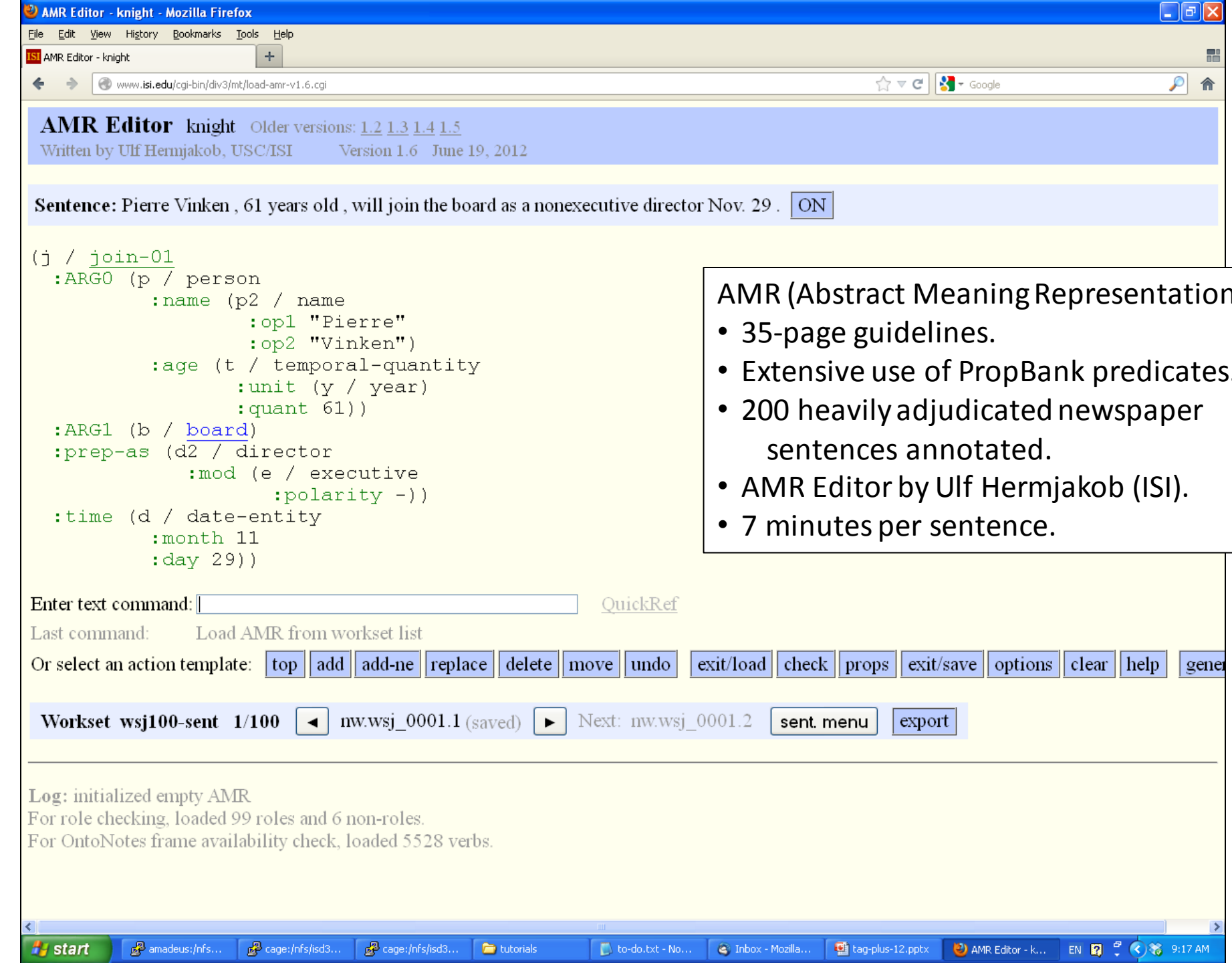
Machine Translation

Syntax-based MT



Towards Meaning-based MT

driven by ongoing large-scale semantic annotation

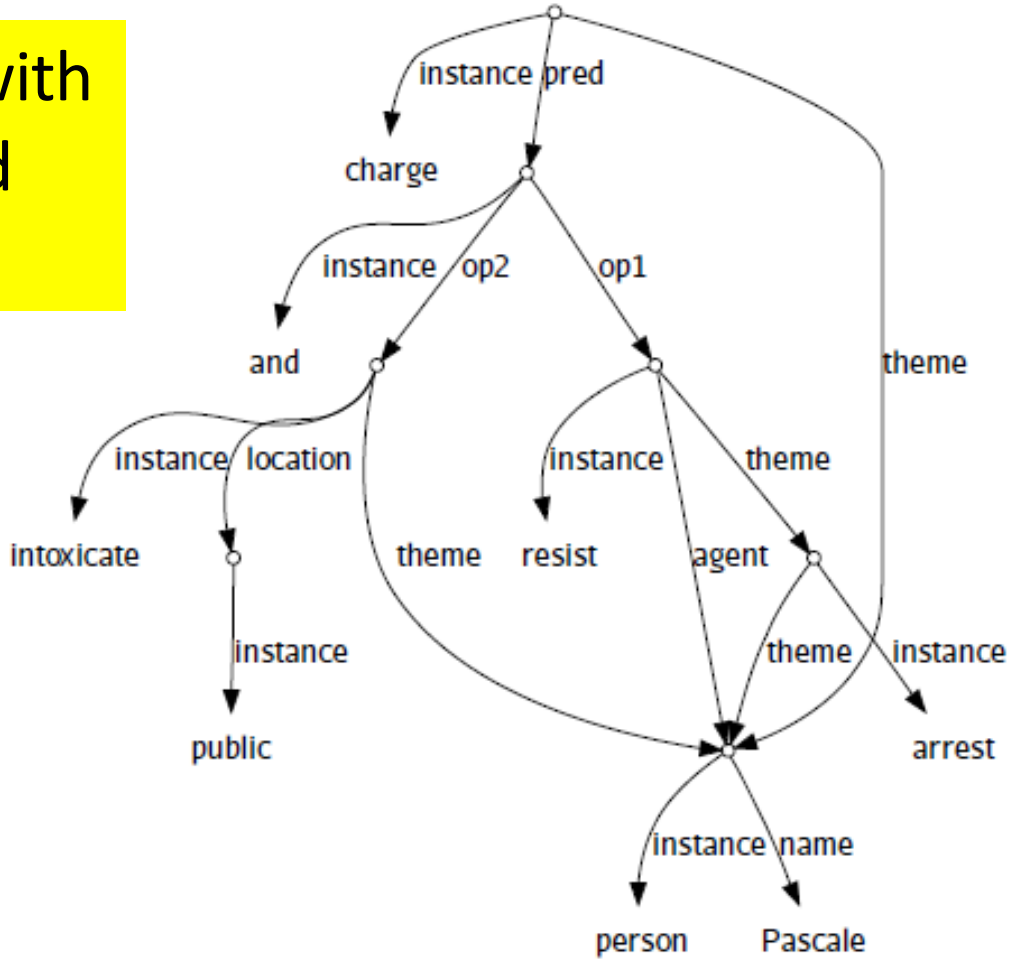


AMR (Abstract Meaning Representation)

- 35-page guidelines.
- Extensive use of PropBank predicates.
- 200 heavily adjudicated newspaper sentences annotated.
- AMR Editor by Ulf Hermjakob (ISI).
- 7 minutes per sentence.

Re-entrancy is Common

Pascale was charged with public intoxication and resisting arrest.

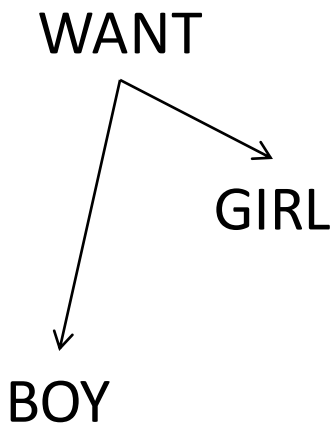


A Semantic Microworld

Node Labels: {WANT, BELIEVE, BOY, GIRL}

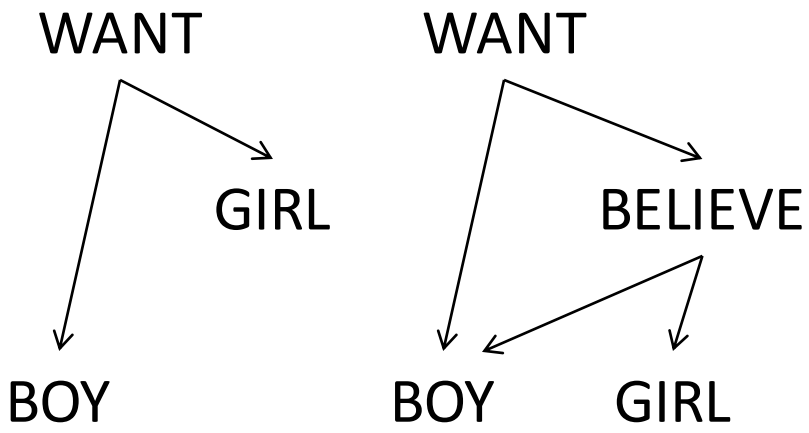
A Semantic Microworld

Node Labels: {WANT, BELIEVE, BOY, GIRL}



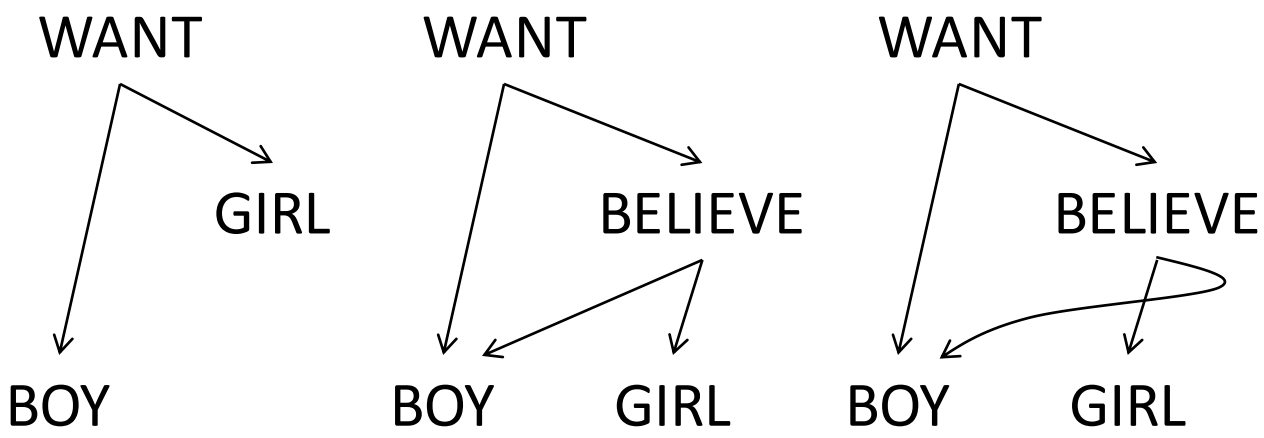
A Semantic Microworld

Node Labels: {WANT, BELIEVE, BOY, GIRL}



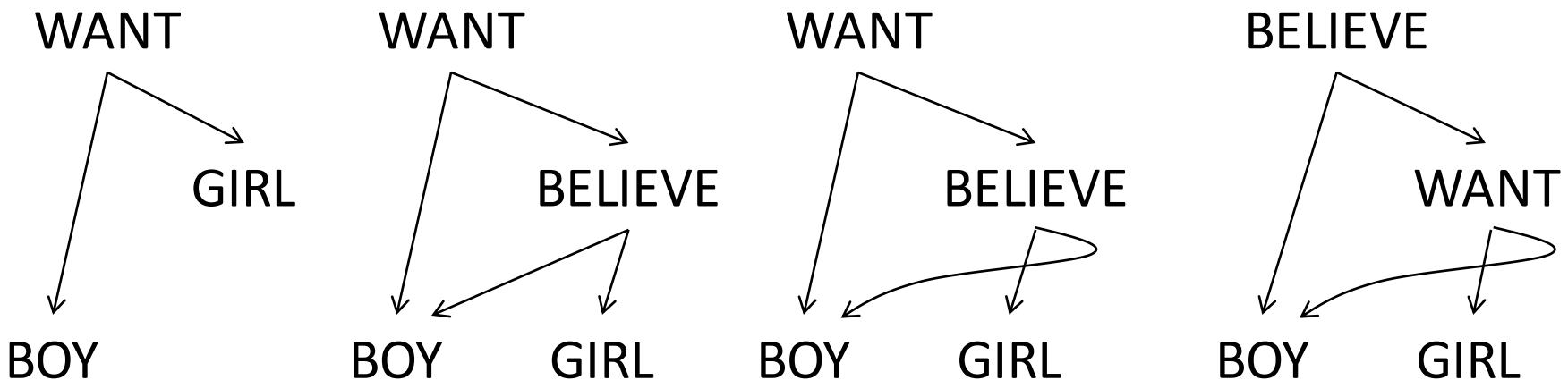
A Semantic Microworld

Node Labels: {WANT, BELIEVE, BOY, GIRL}



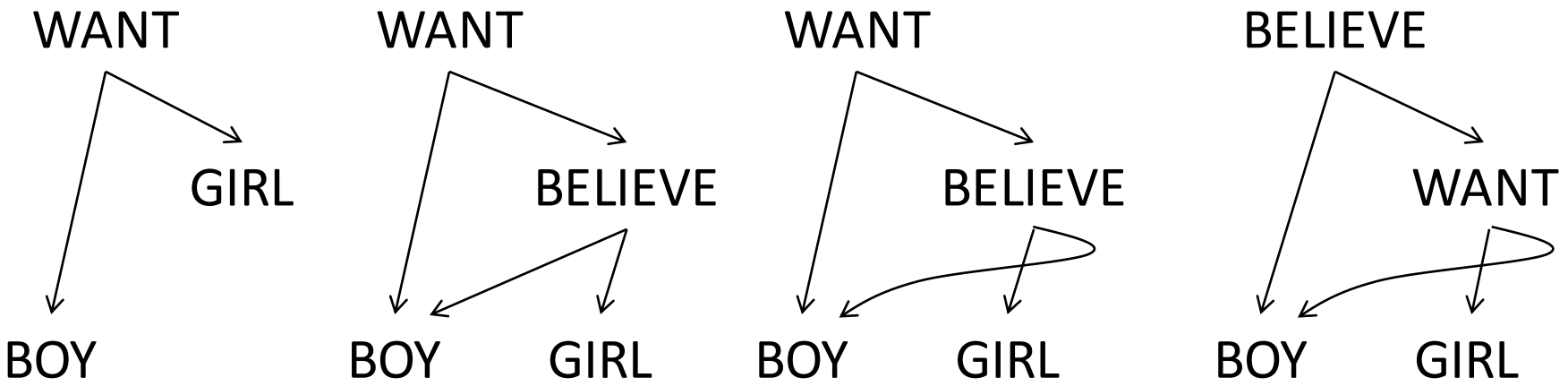
A Semantic Microworld

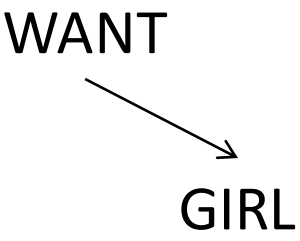
Node Labels: {WANT, BELIEVE, BOY, GIRL}



A Semantic Microworld

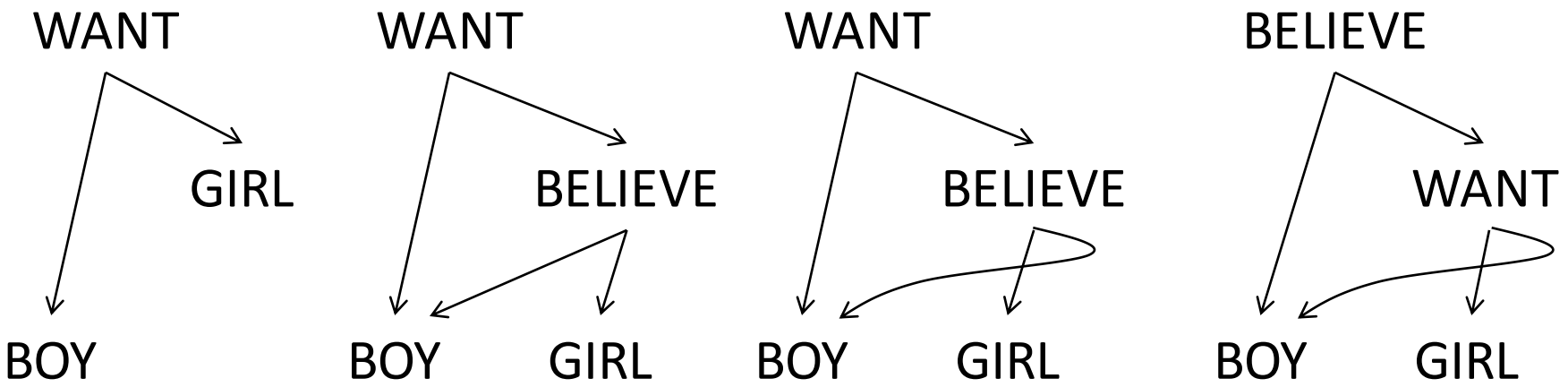
Node Labels: {WANT, BELIEVE, BOY, GIRL}

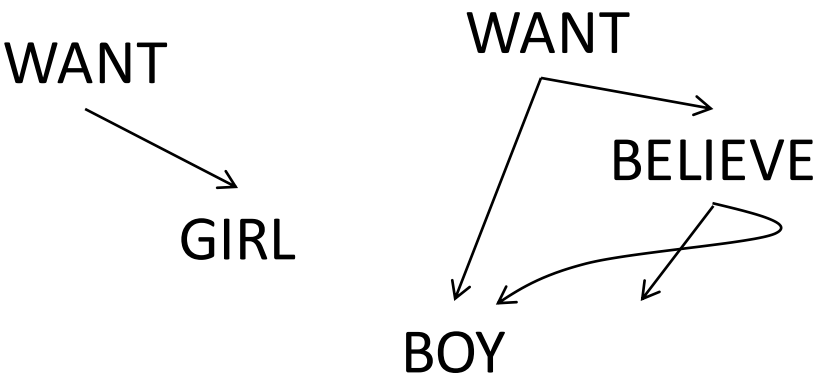




A Semantic Microworld

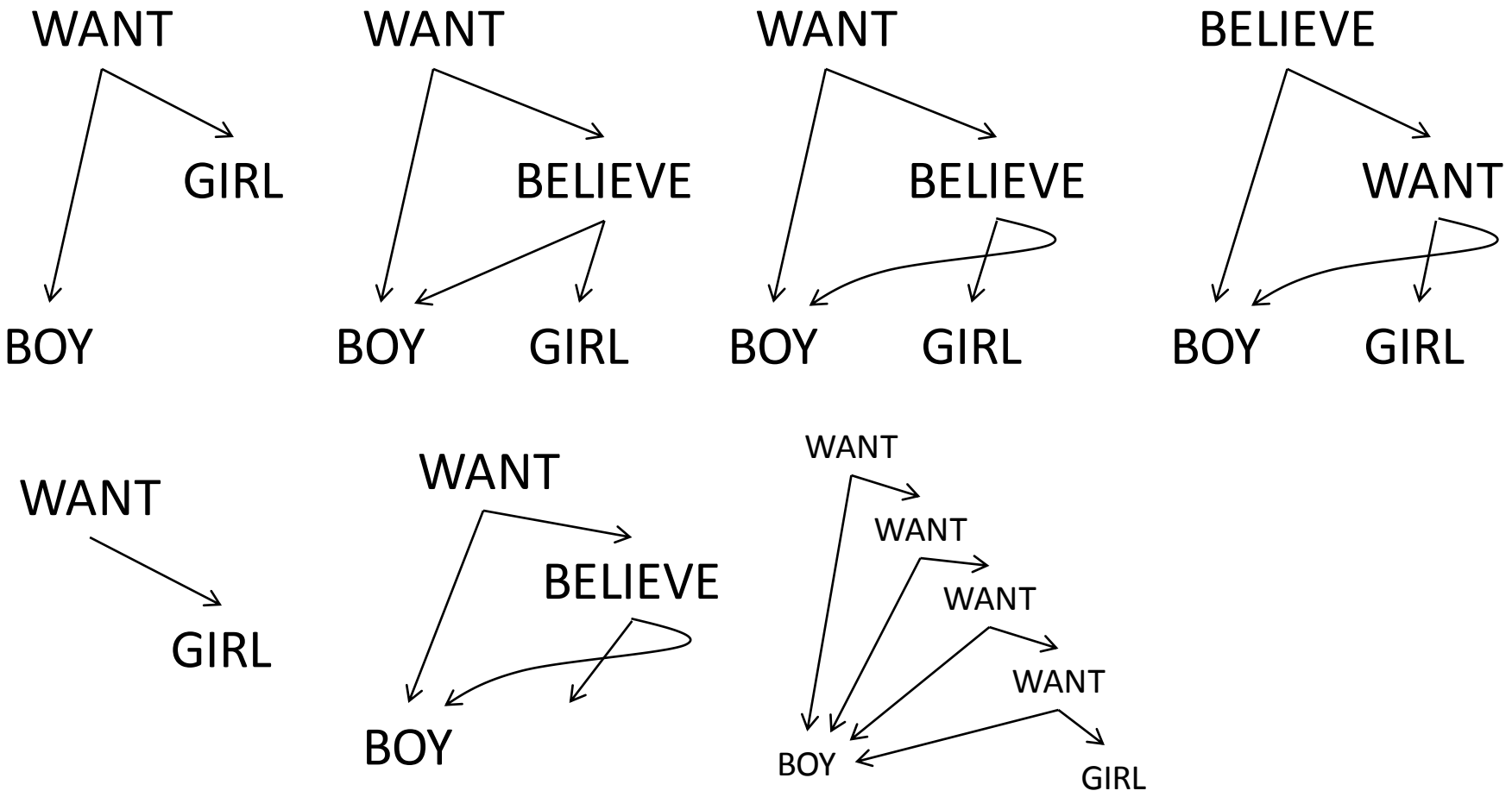
Node Labels: {WANT, BELIEVE, BOY, GIRL}





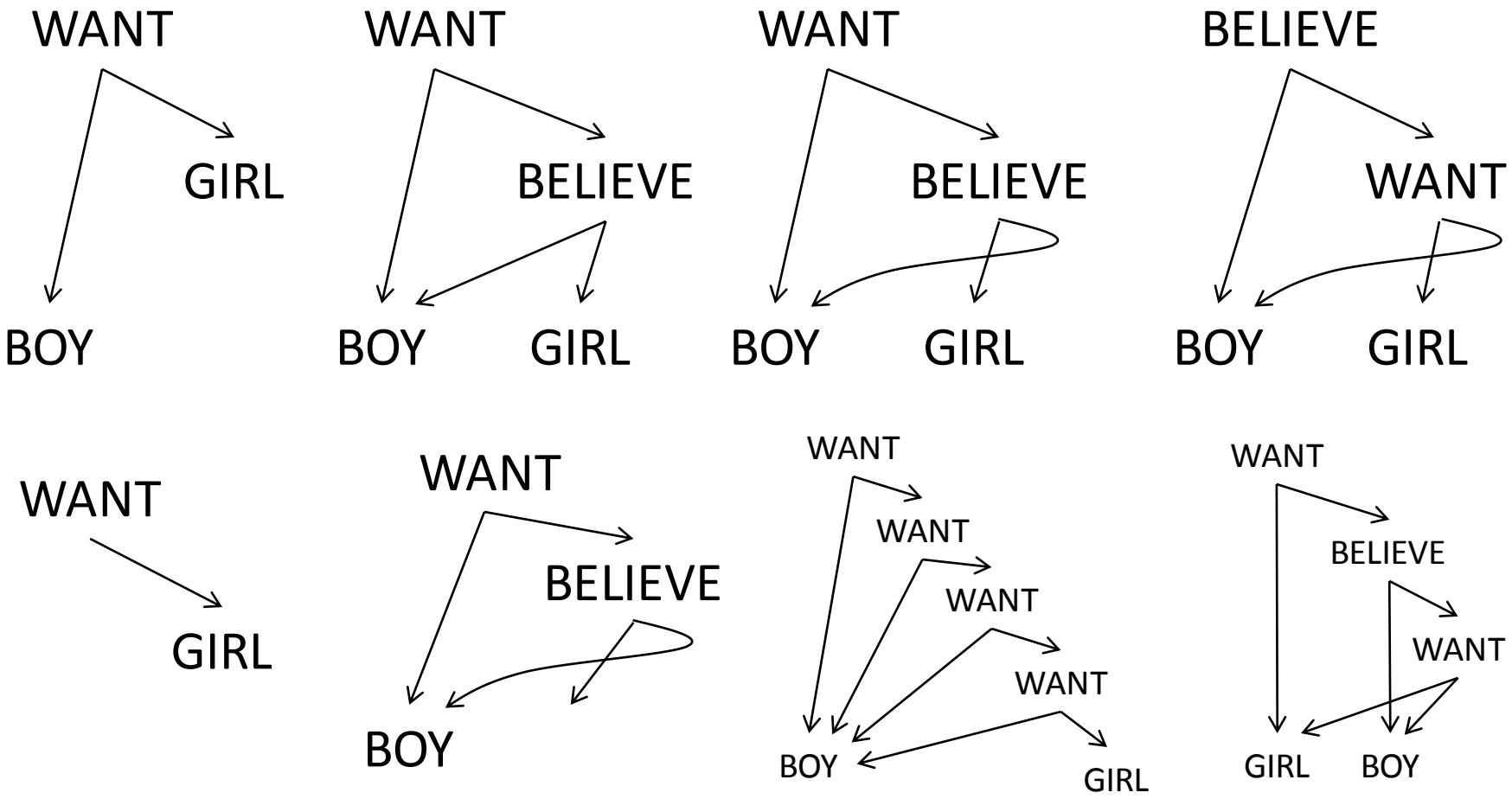
A Semantic Microworld

Node Labels: {WANT, BELIEVE, BOY, GIRL}

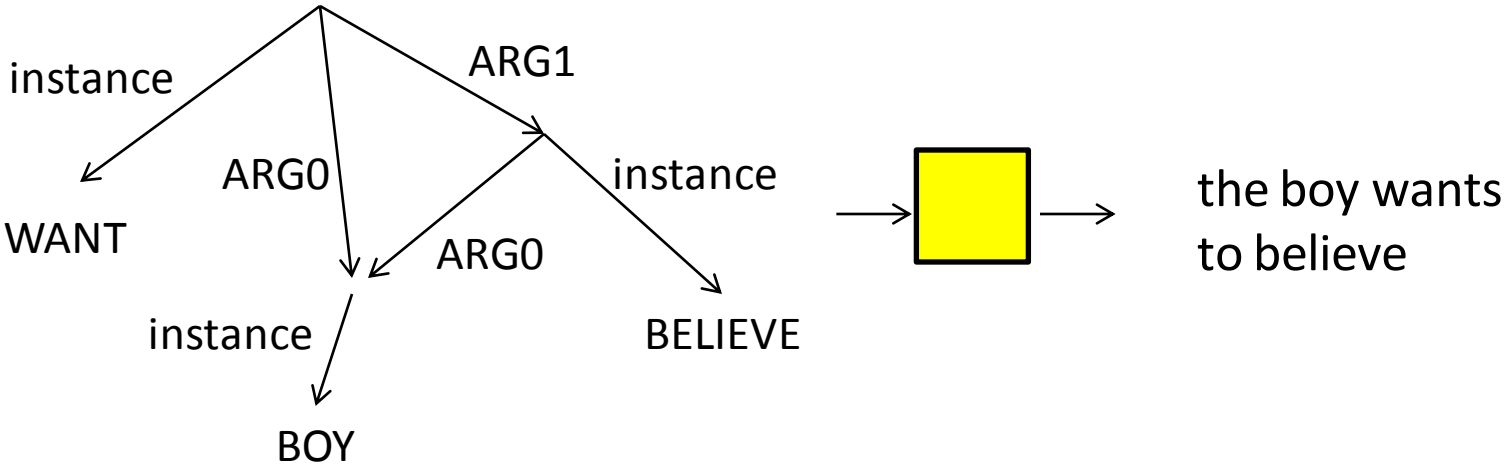


A Semantic Microworld

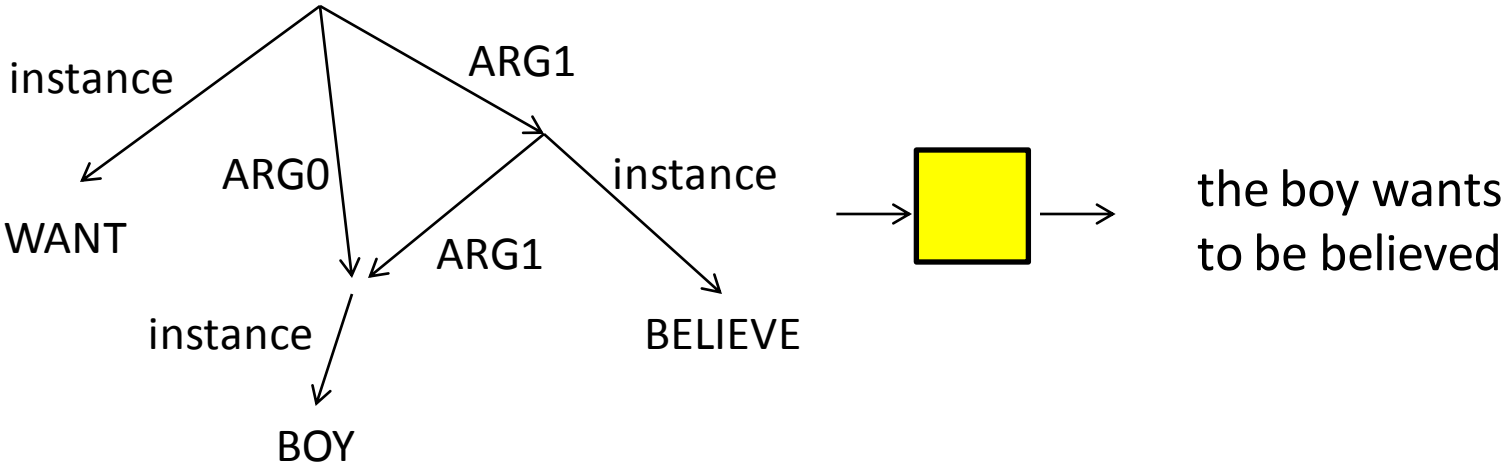
Node Labels: {WANT, BELIEVE, BOY, GIRL}



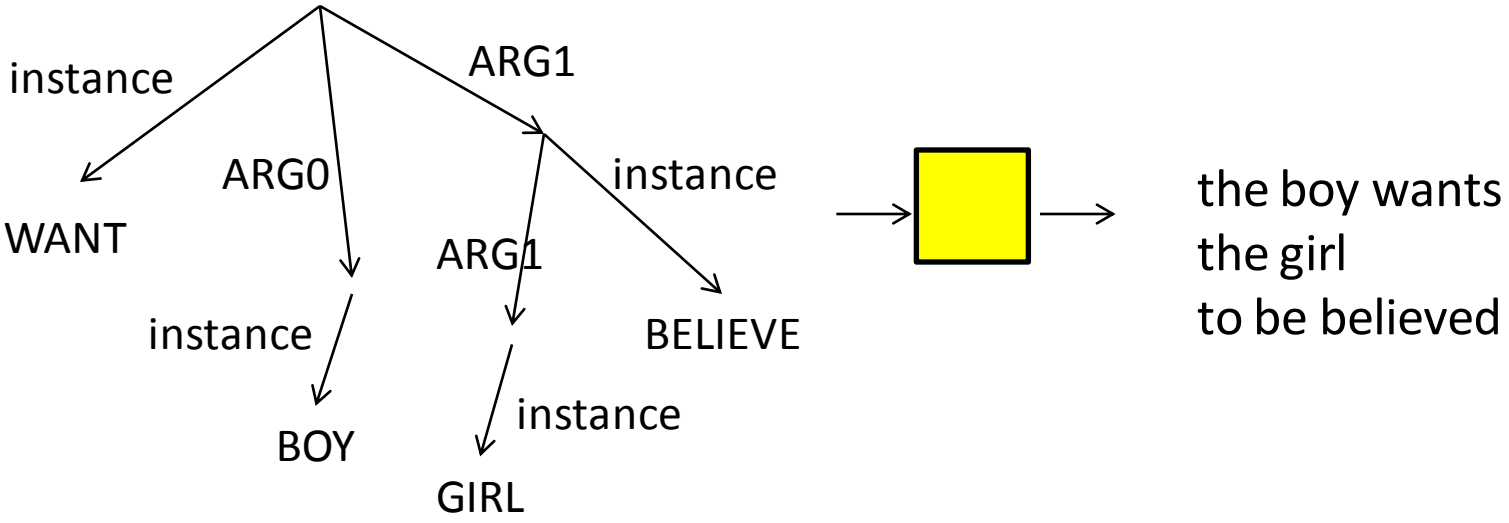
Mapping Between Meaning and Text



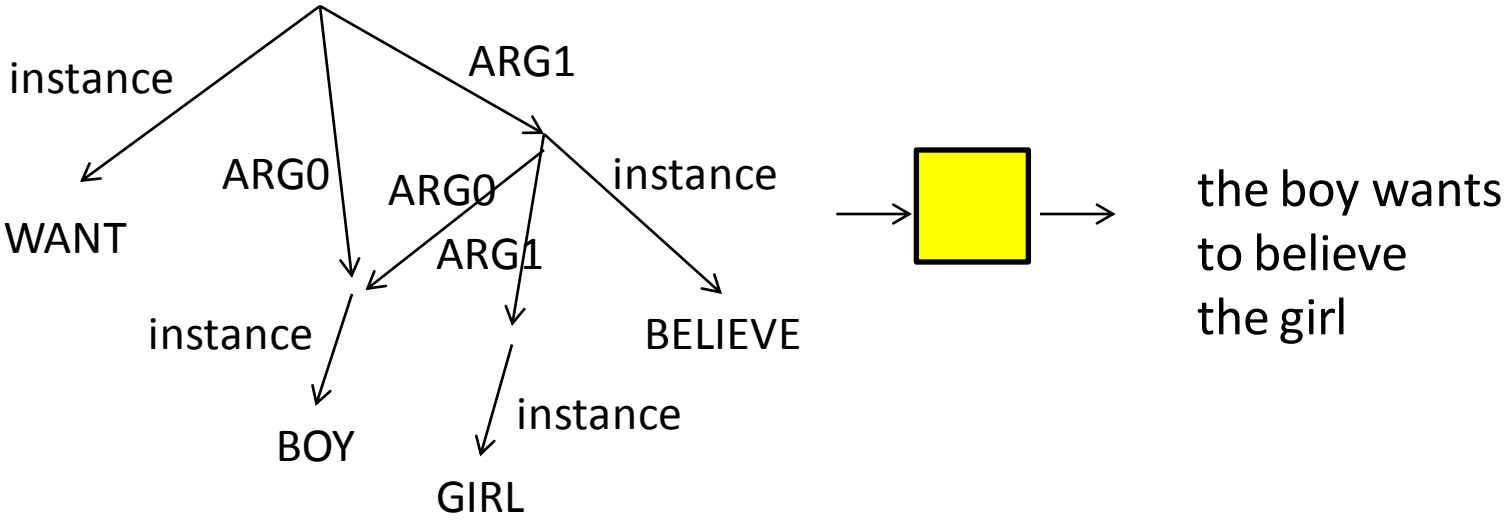
Mapping Between Meaning and Text



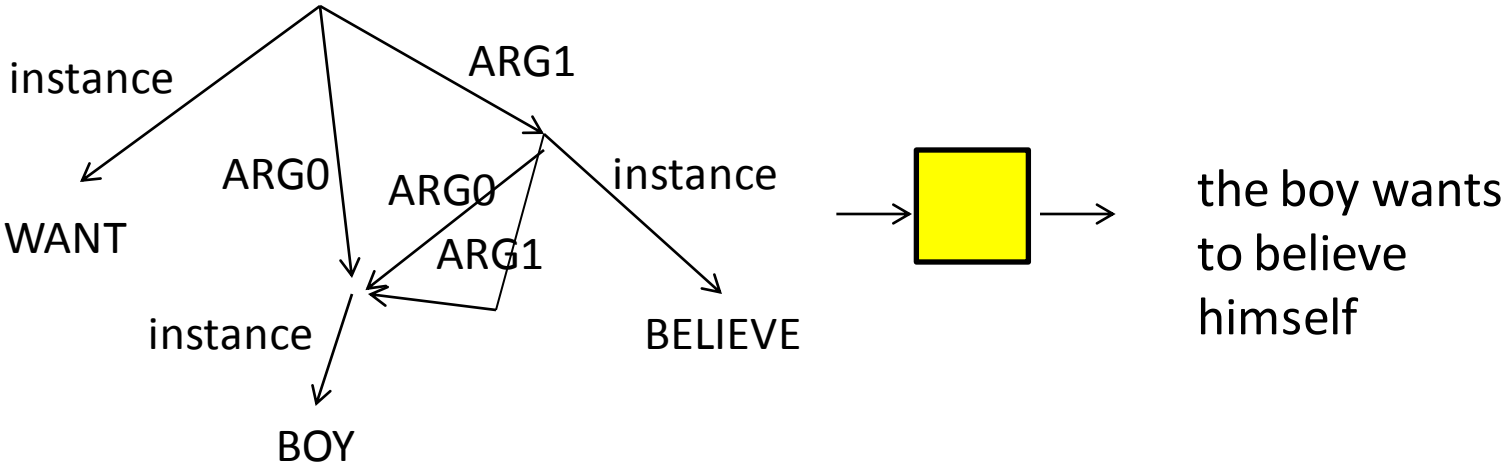
Mapping Between Meaning and Text



Mapping Between Meaning and Text



Mapping Between Meaning and Text



General-Purpose Algorithms

	String Automata Algorithms	Tree Automata Algorithms	Graph Automata Algorithms
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)	… trees in a weighted forest (Jiménez & Marzal, 2000; Huang & Chiang, 2005)	Graph Language Acceptors Graph Transducers Efficient operations
EM training	Forward-backward EM (Baum/Welch, 1971; Eisner 2003)	Tree transducer EM training (Graehl & Knight, 2004)	
Determinization…	… of weighted string acceptors (Mohri, 1997)	… of weighted tree acceptors (Borchardt & Vogler, 2003; May & Knight, 2005)	
Intersection	WFSA intersection	Tree acceptor intersection	
Applying transducers	string → WFST → WFSA	tree → TT → weighted tree acceptor	
Transducer composition	WFST composition (Pereira & Riley, 1996)	Many tree transducers not closed under composition (Maletti et al 09)	
General tools	FSM, Carmel, OpenFST	Tiburon (May & Knight 10)	

Automata Frameworks

- Unification grammar: string-to-graph
  (Shieber 86, Moore 89)
- Hyperedge-replacement graph grammars
  (Drewes et al 97)
- DAG acceptors (Hart 75)
- DAG-to-tree transducers (Kamimura & Slutski 82)

DAG Acceptor
(Hart 75)

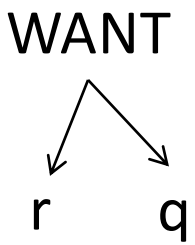
q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r

q

DAG Acceptor

(Hart 75)

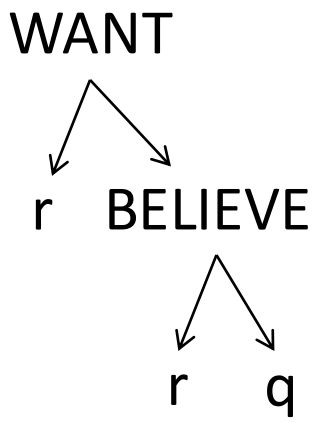
q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r



DAG Acceptor

(Hart 75)

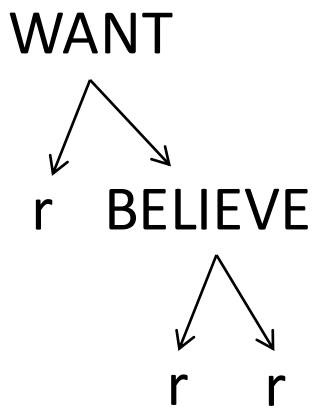
q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r



DAG Acceptor

(Hart 75)

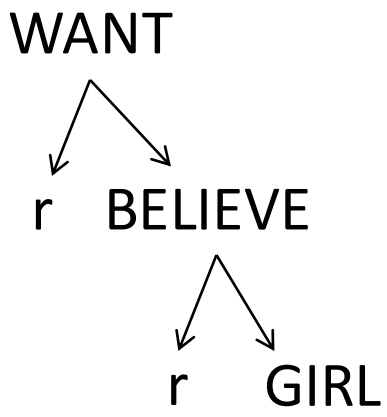
q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r



DAG Acceptor

(Hart 75)

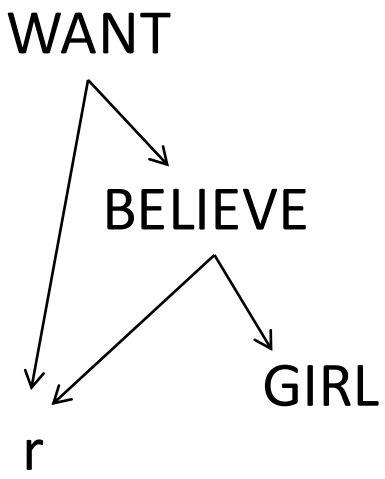
q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r



DAG Acceptor

(Hart 75)

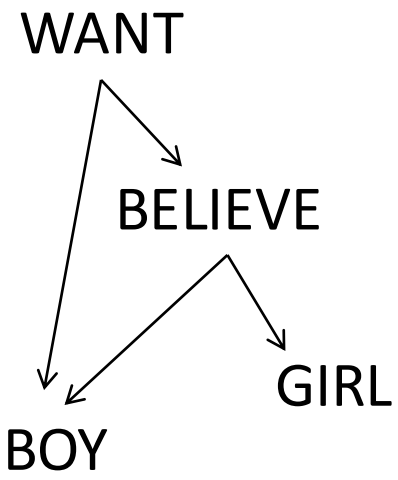
q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r



DAG Acceptor

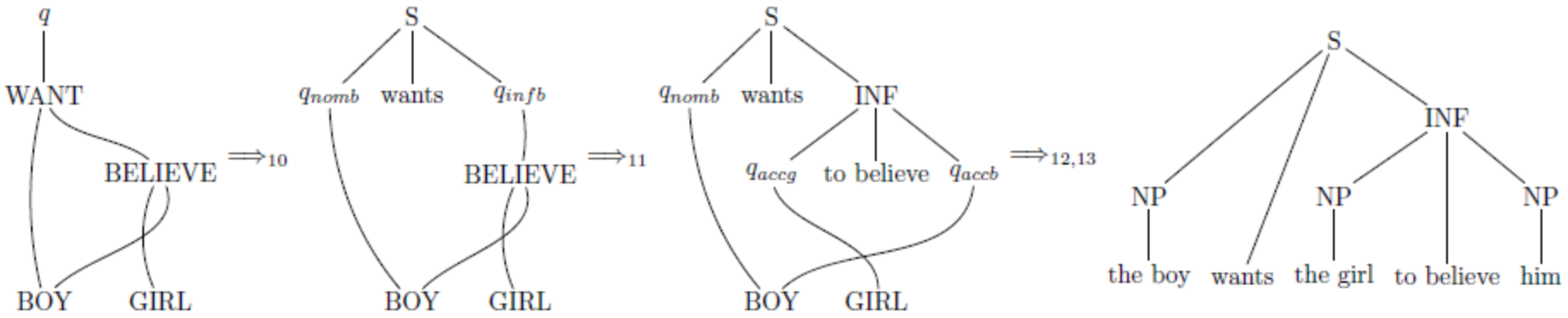
(Hart 75)

q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r



DAG-to-Tree Transducer

(Kamimura & Slutzki 82, Quernheim & Knight 12)



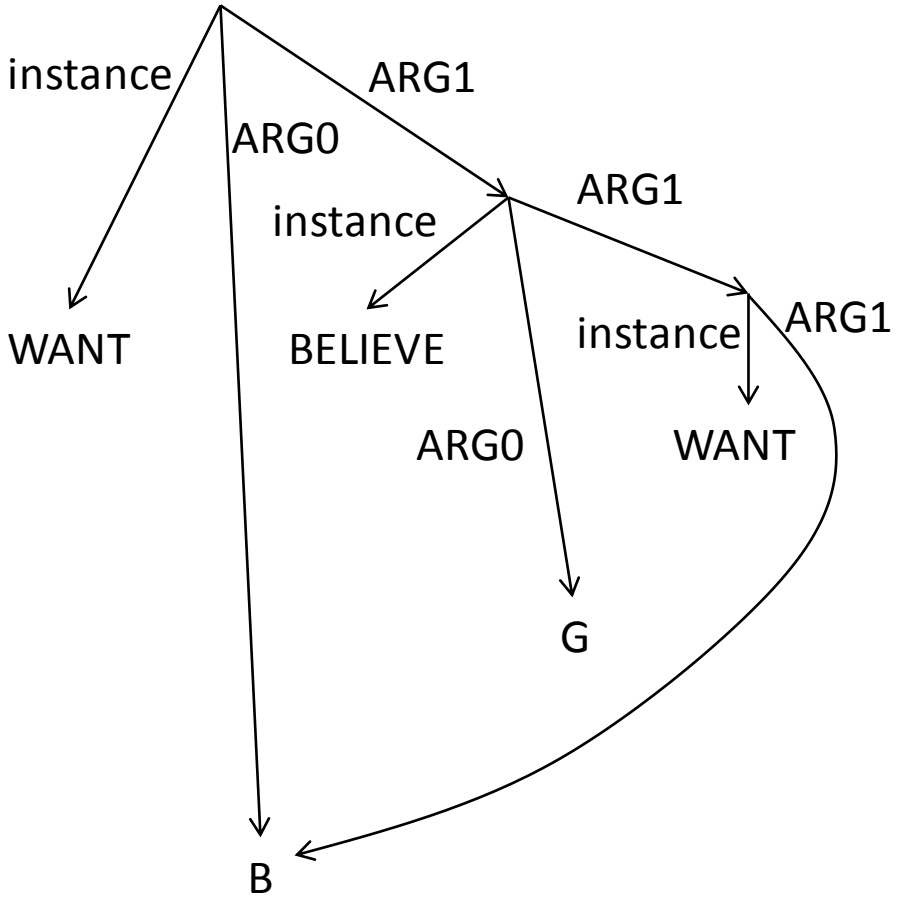
Hyperedge Replacement Grammars

- context-free derivation forests

- locality (multiple references generated in a single step, then “pushed apart”)

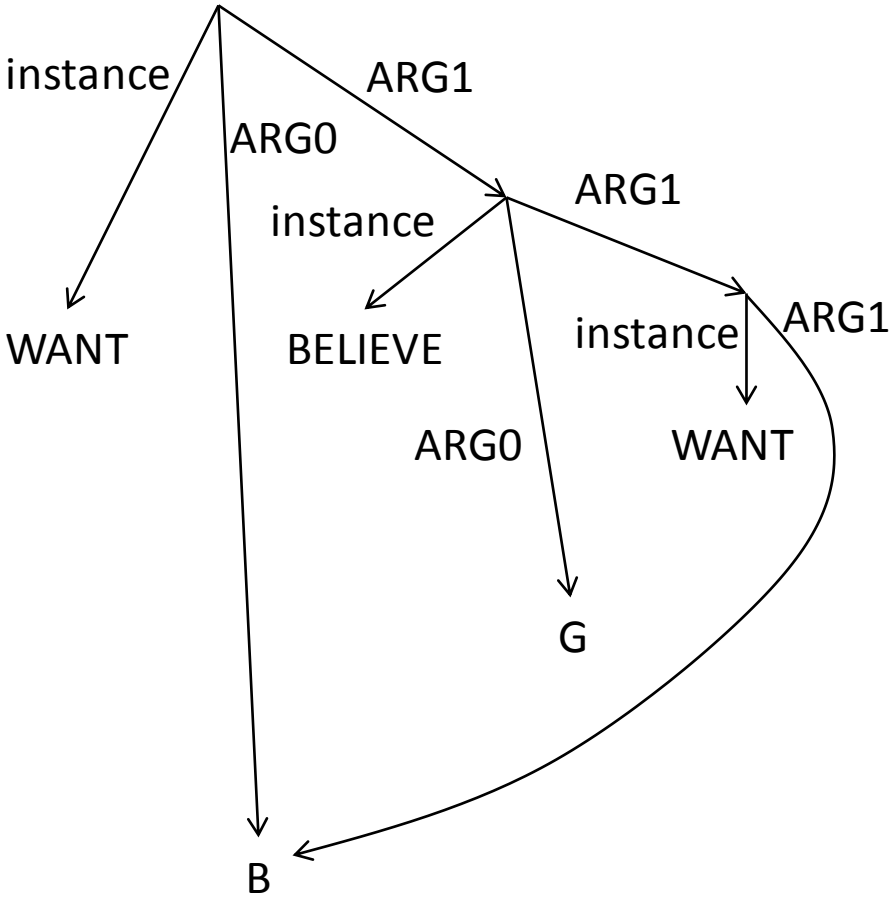
(should sound familiar ☺)

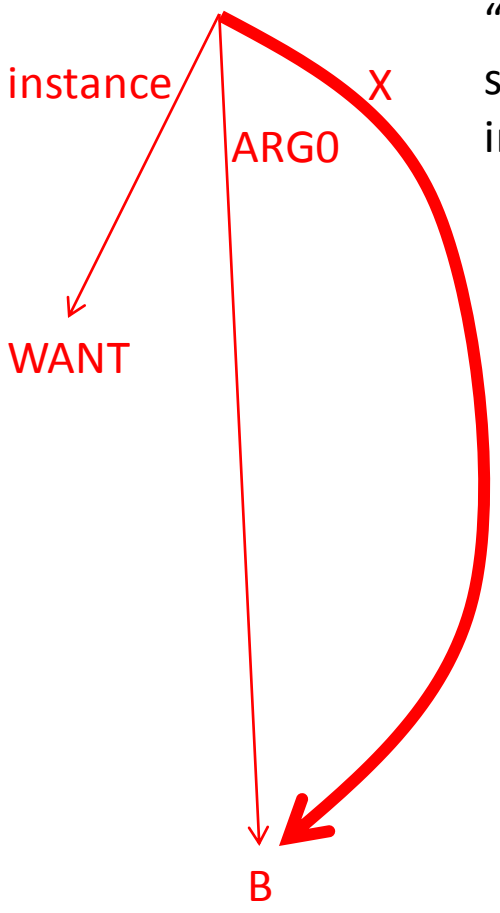
HRG Derivation 1



= boy wants girl to believe that he is wanted

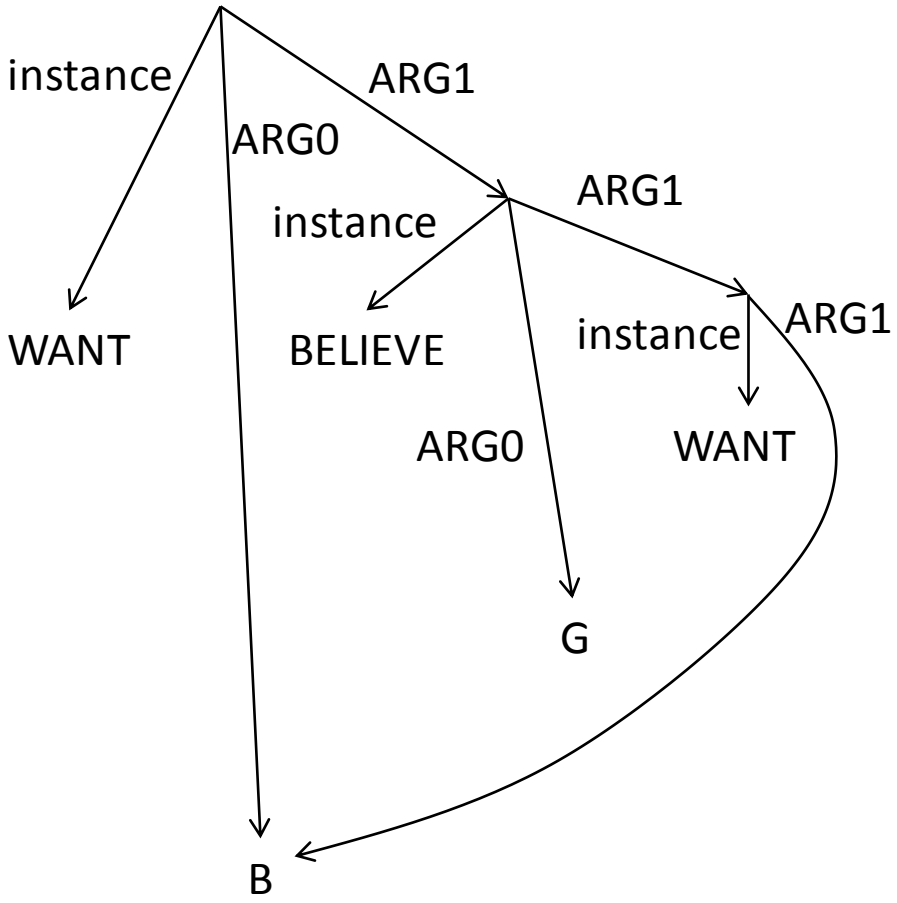
HRG Derivation 1

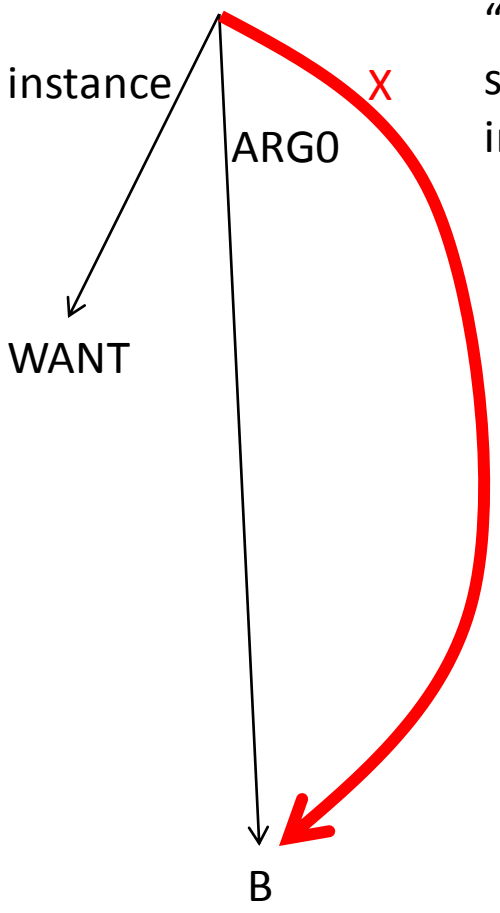




"the boy wants something involving himself"

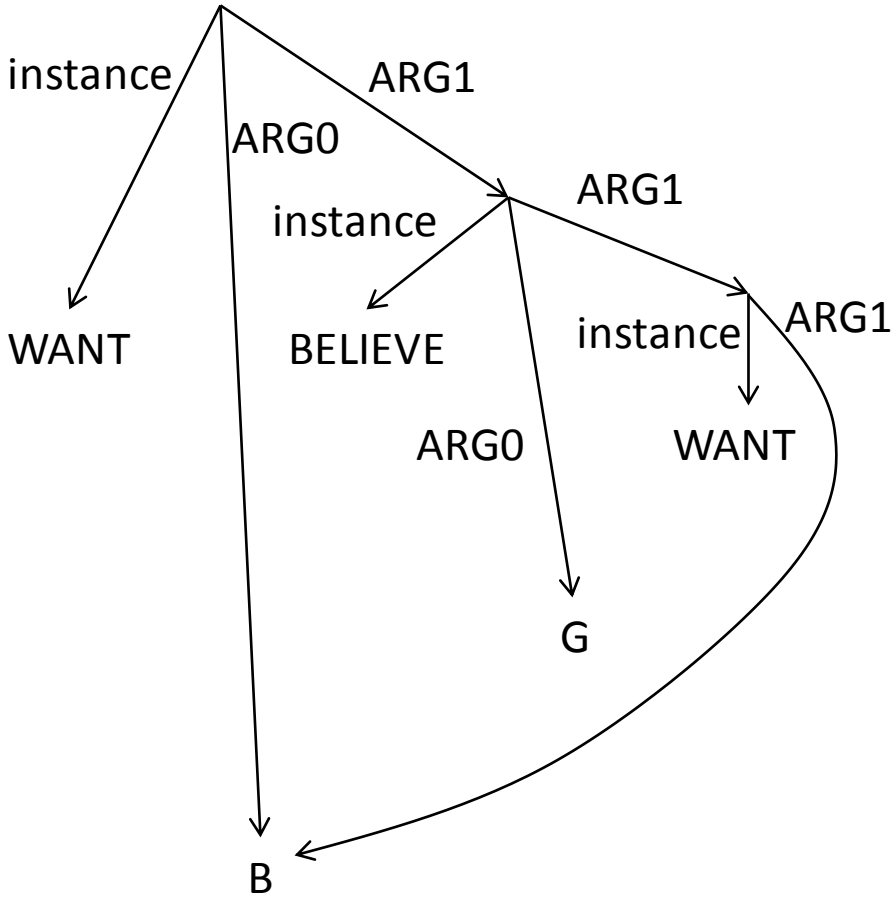
HRG Derivation 1

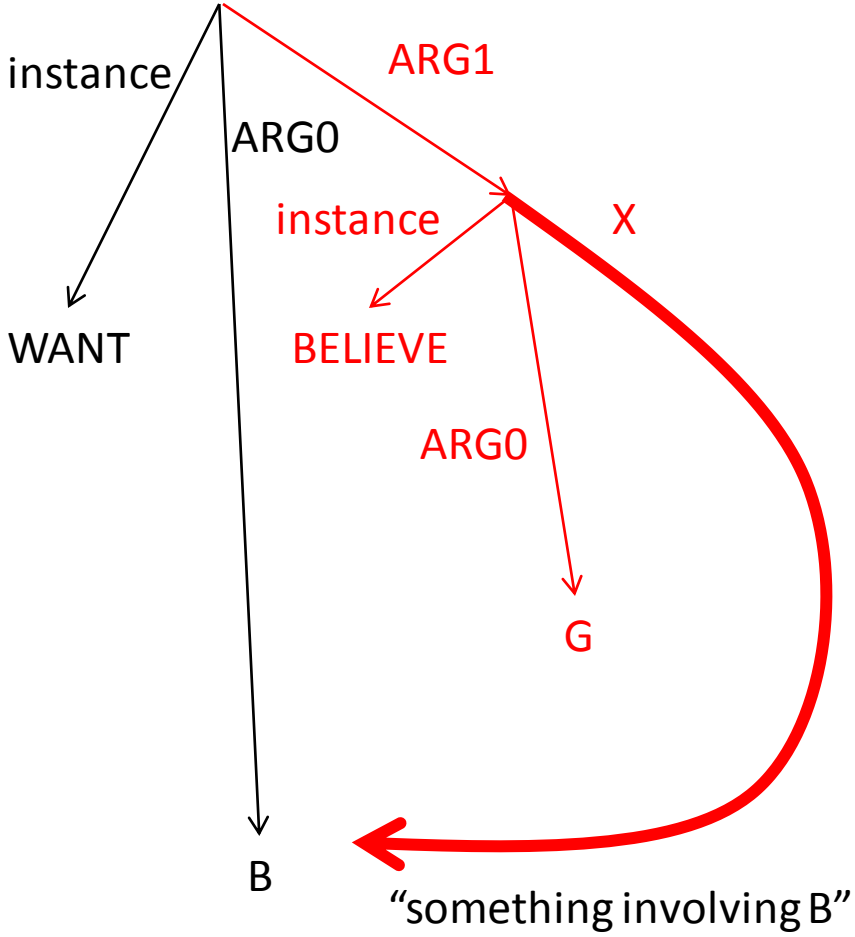




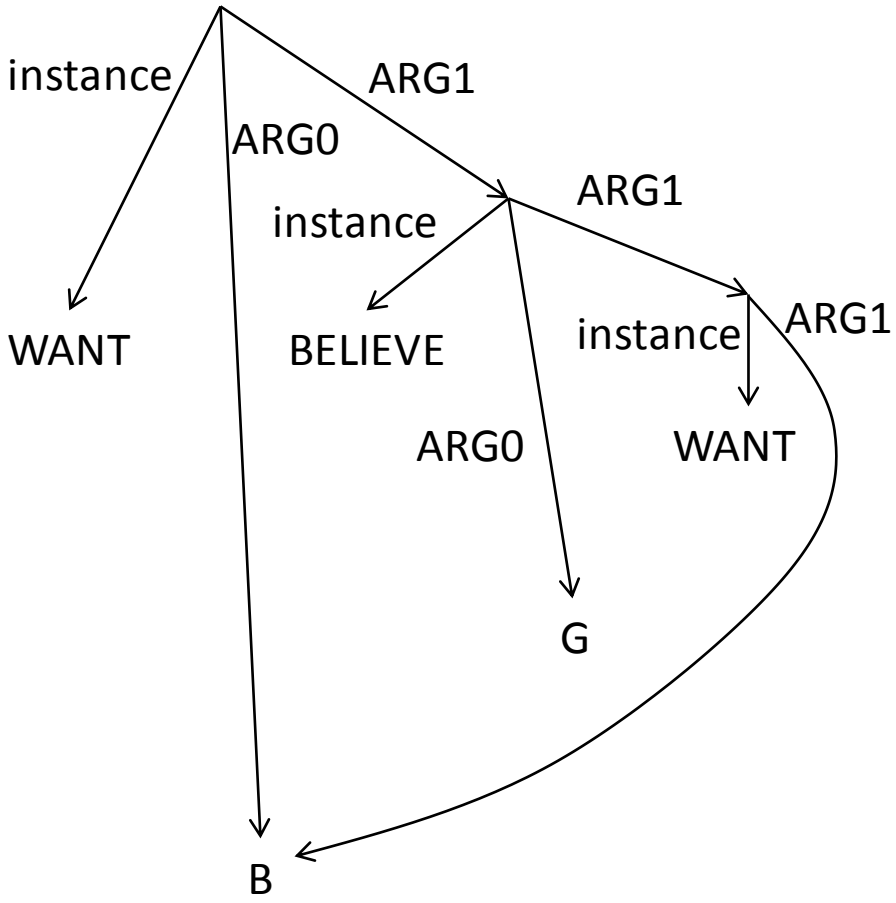
"the boy wants something involving himself"

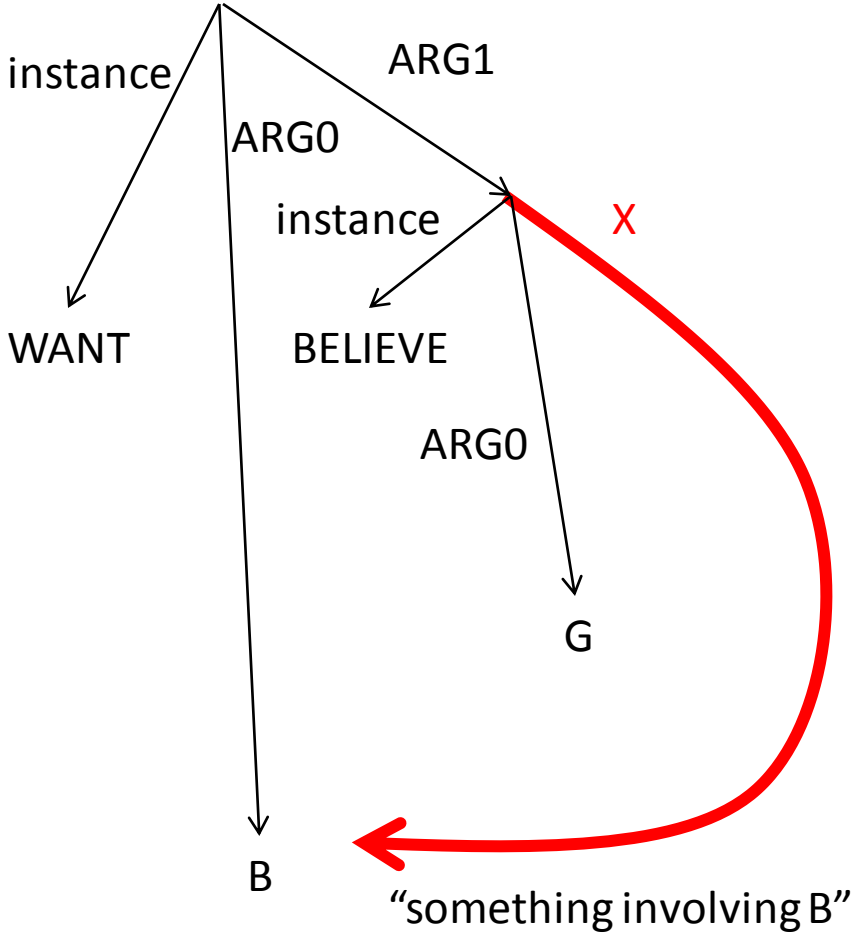
HRG Derivation 1



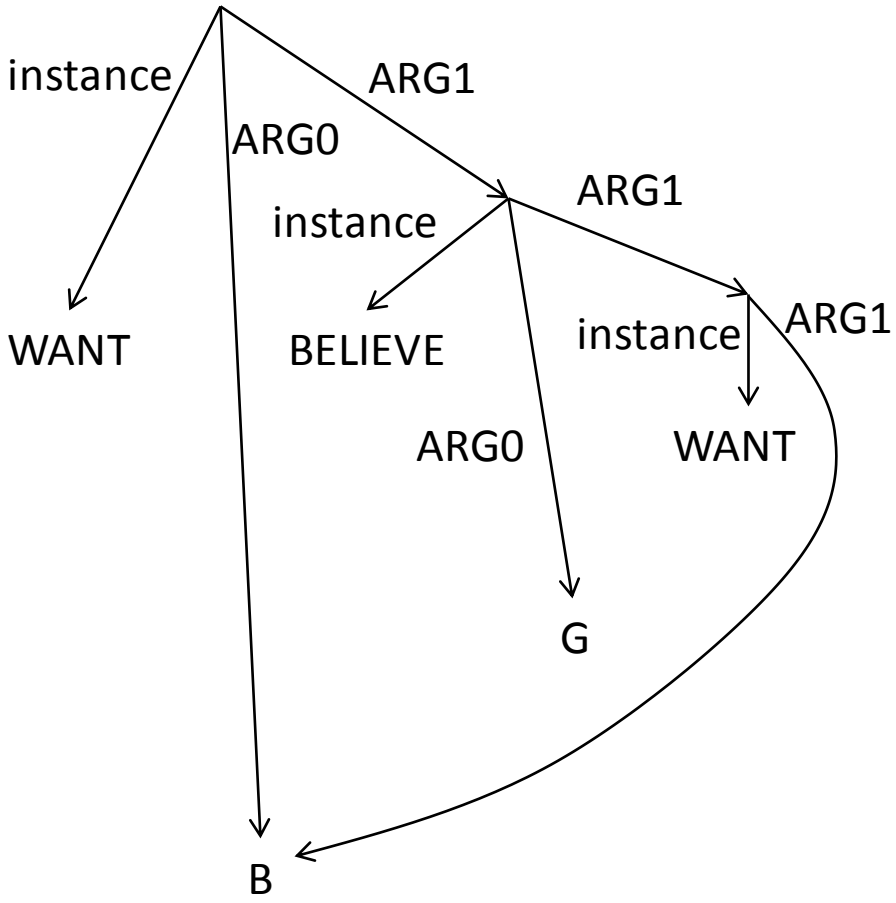


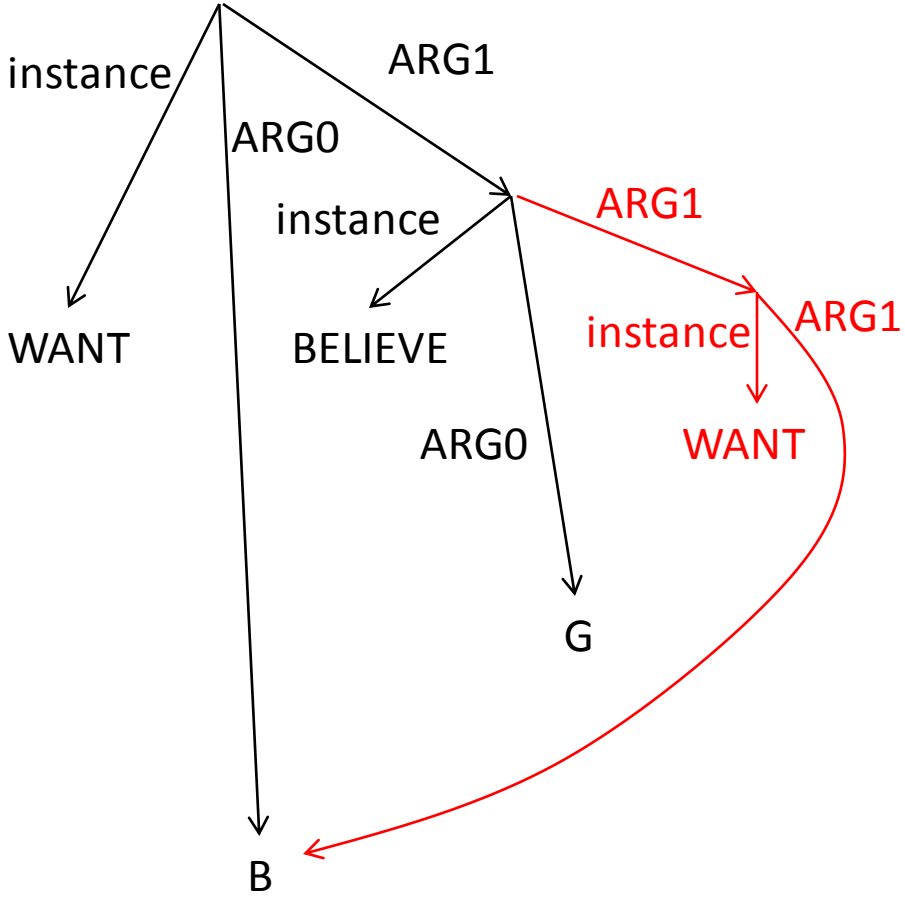
HRG Derivation 1



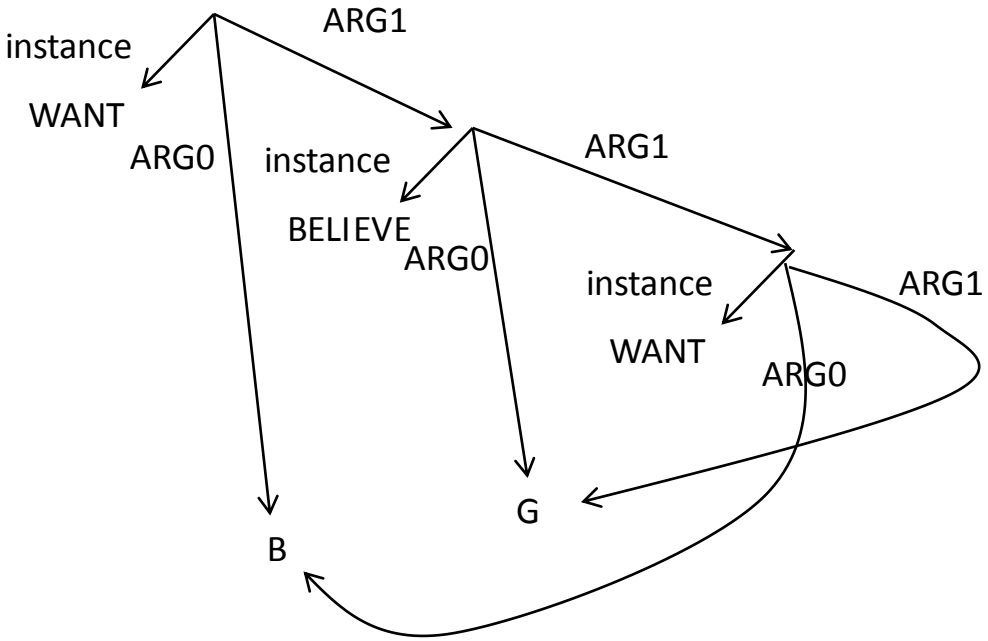


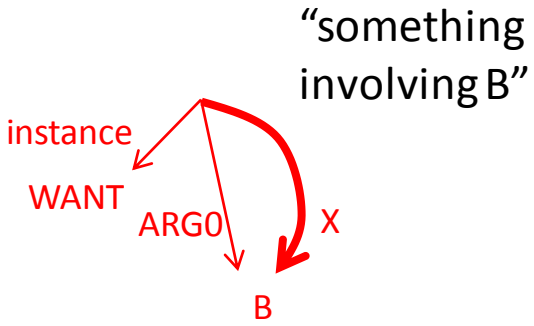
HRG Derivation 1





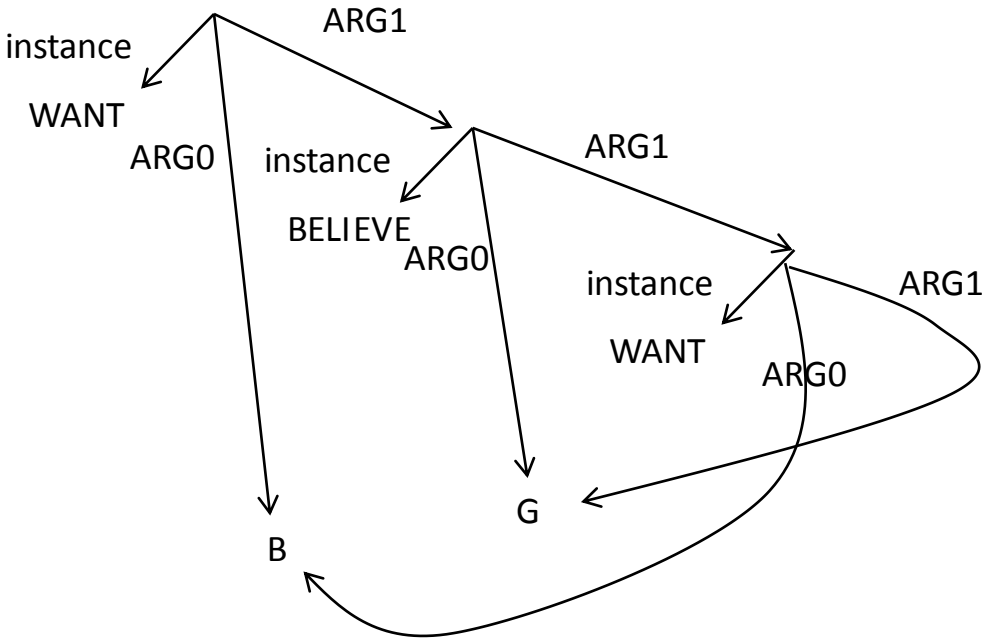
HRG Derivation 2

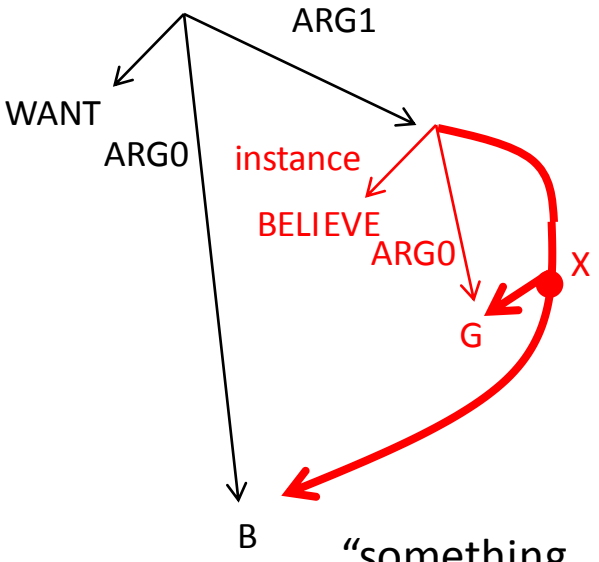




(= boy wants girl to believe that he wants her)

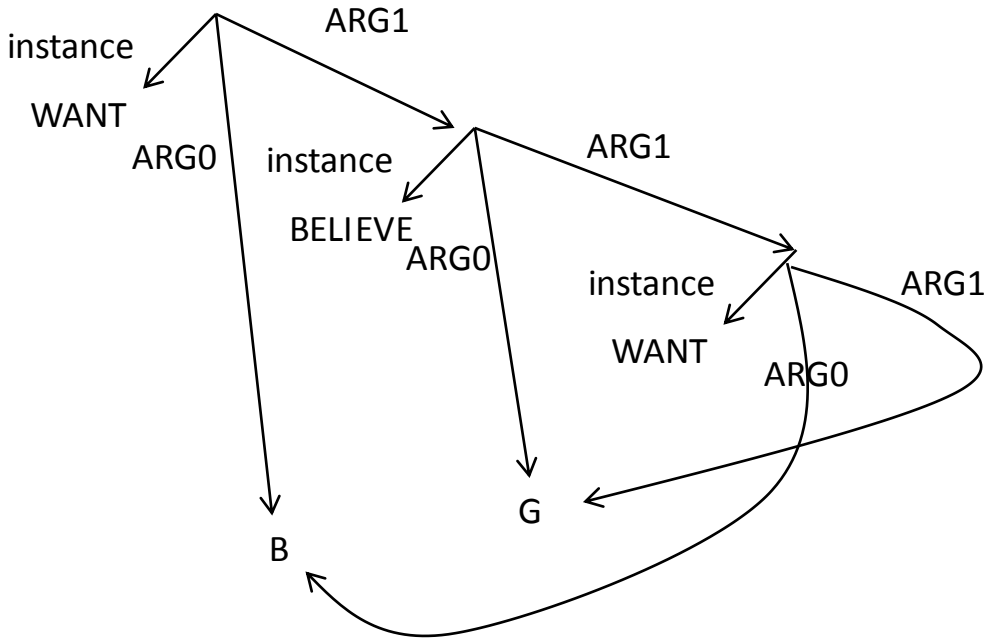
HRG Derivation 2

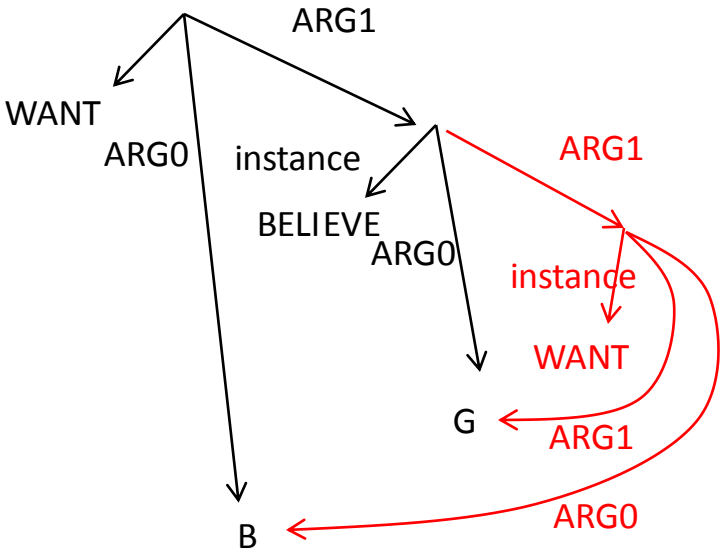




"something involving B and G"

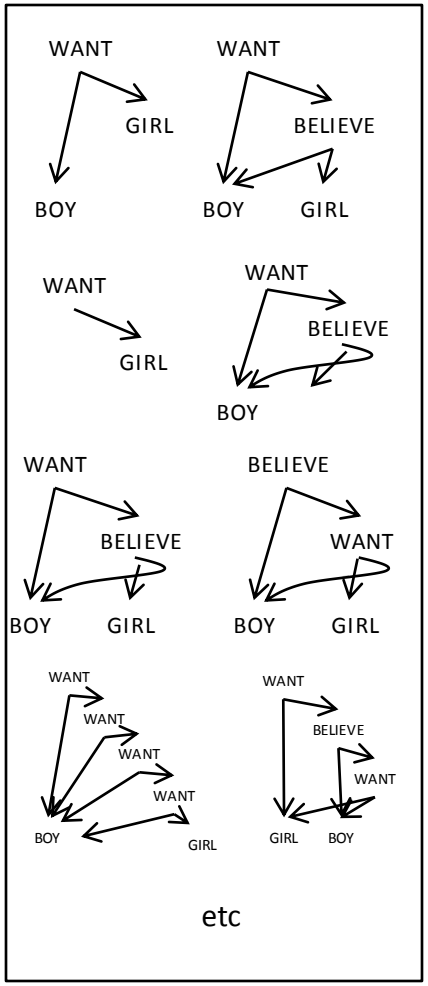
HRG Derivation 2





Graph Acceptors

Infinite graph language



DAG acceptor

```
q
q → WANT(r q)
q → BELIEVE(r q)
q → r | 0
r → BOY | GIRL | 0
[r r] → r
```

HRG acceptor

```
P ||| (n :instance want)
P ||| (n :instance believe)
E ||| (n :instance boy)
E ||| (n :instance girl)
S ||| (n :P[1])
S ||| (n :P[1] :arg1 (n :E[2]))
S ||| (n :P[1] :arg1 (n :S[2]))
S ||| (n :P[1] :arg0 (n :E[2]))
S ||| (n :P[1] :arg0 (n :E[2]) :arg1 (n :E[3]))
S ||| (n :P[1] :arg0 (n :E[2]) :arg1 (n :S[3]))
S ||| (n :P[1] :arg0 #0=(n :E[2]) :arg1 #0)
S ||| (n :P[1] :arg0 #0=(n :E[2]) :arg1 (n :S1[3] #0))
S1 ||| (n :P[1] :arg1 n*1)
S1 ||| (n :P[1] :arg1 (n :S1[2] n*1))
S1 ||| (n :P[1] :arg0 (n :E[2]) :arg1 n*1)
S1 ||| (n :P[1] :arg0 (n :E[2]) :arg1 (n :S1[3] n*1))
S1 ||| (n :P[1] :arg0 #0=(n :E[2]) :arg1 (n :S2[3] n*1 #0))
S1 ||| (n :P[1] :arg0 n*1)
S1 ||| (n :P[1] :arg0 n*1 :arg1 (n :E[2]))
S1 ||| (n :P[1] :arg0 n*1 :arg1 (n :S[2]))
S1 ||| (n :P[1] :arg0 #0=n*1 :arg1 (n :S1[2] #0))
S1 ||| (n :P[1] :arg0 #0=n*1 :arg1 #0)
S2 ||| (n :P[1] :arg0 (n :E[2]) :arg1 (n :S2[3] n*1 n*2))
S2 ||| (n :P[1] :arg1 (n :S2[3] n*1 n*2))
S2 ||| (n :P[1] :arg0 n*1 :arg1 n*2)
S2 ||| (n :P[1] :arg0 n*2 :arg1 n*1)
S2 ||| (n :P[1] :arg0 #0=n*1 :arg1 (n :S2[2] #0 n*2))
S2 ||| (n :P[1] :arg0 #0=n*1 :arg1 (n :S1[2] n*2))
S2 ||| (n :P[1] :arg0 #0=n*2 :arg1 (n :S2[2] n*1 #0))
S2 ||| (n :P[1] :arg0 #0=n*2 :arg1 (n :S1[2] n*1))
```

locality

no fixed arity

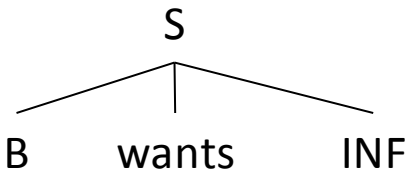
Synchronous HRG Derivation

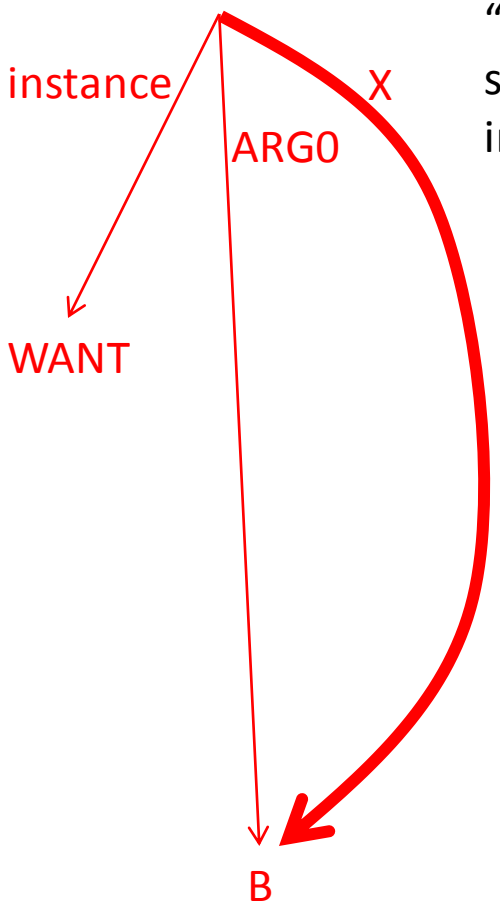
syntax
tree

semantic
graph

can be used to map in either direction

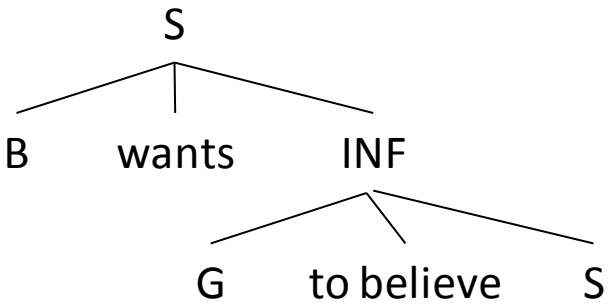
HRG Derivation

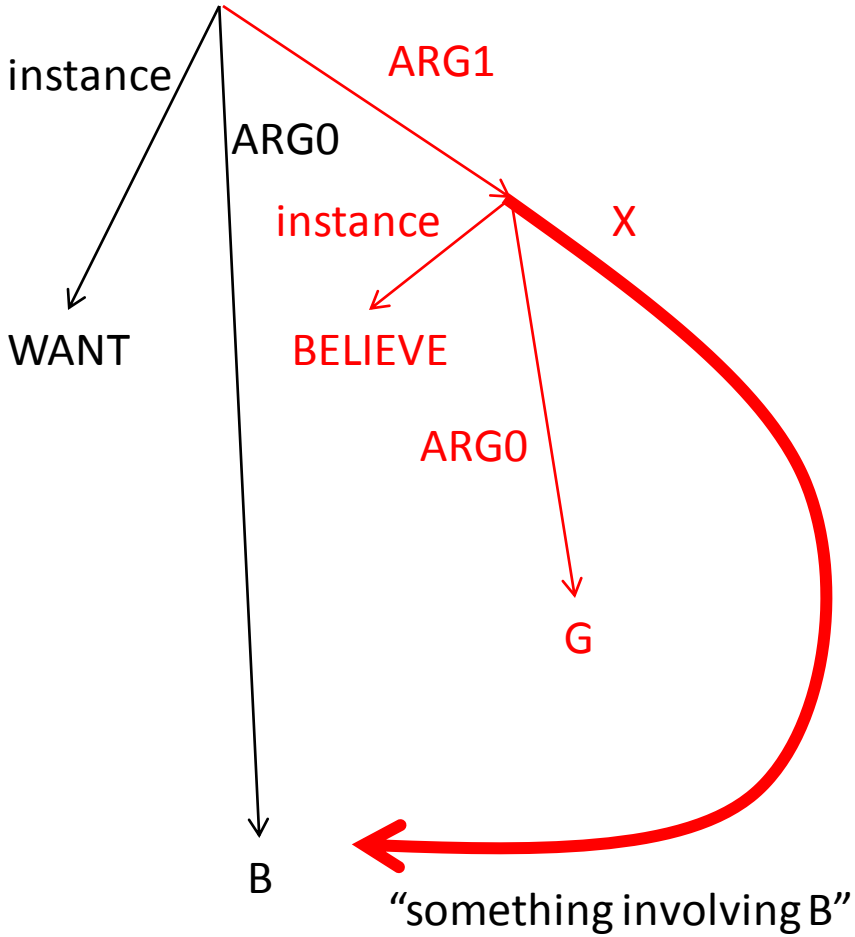




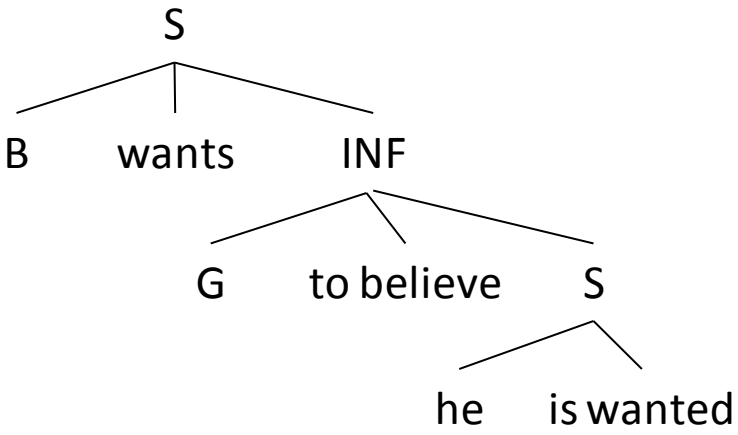
"the boy wants something involving himself"

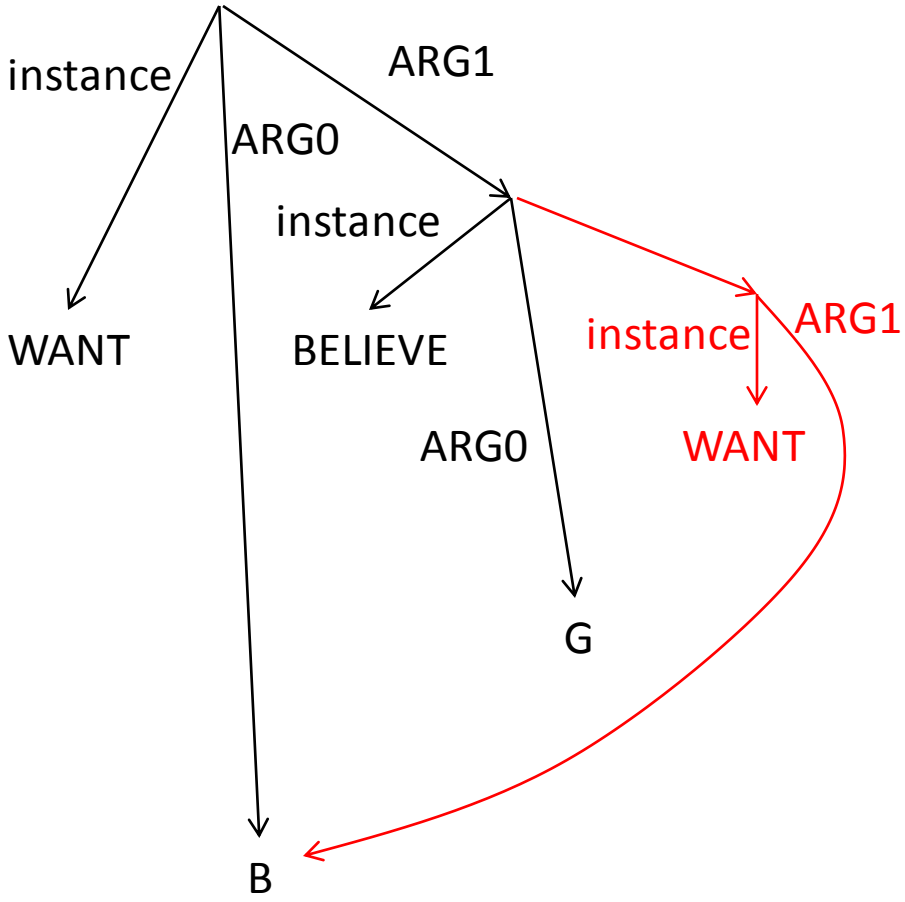
HRG Derivation





HRG Derivation





General-Purpose Algorithms

	String Automata Algorithms	Tree Automata Algorithms	Graph Automata Algorithms
N-best …	… paths through an WFSA (Viterbi, 1967; Eppstein, 1998)	… trees in a weighted forest (Jiménez & Marzal, 2000; Huang & Chiang, 2005)	Graph Language Acceptors Graph Transducers Efficient operations
EM training	Forward-backward EM (Baum/Welch, 1971; Eisner 2003)	Tree transducer EM training (Graehl & Knight, 2004)	
Determinization…	… of weighted string acceptors (Mohri, 1997)	… of weighted tree acceptors (Borchardt & Vogler, 2003; May & Knight, 2005)	
Intersection	WFSA intersection	Tree acceptor intersection	
Applying transducers	string → WFST → WFSA	tree → TT → weighted tree acceptor	
Transducer composition	WFST composition (Pereira & Riley, 1996)	Many tree transducers not closed under composition (Maletti et al 09)	
General tools	FSM, Carmel, OpenFST	Tiburon (May & Knight 10)	

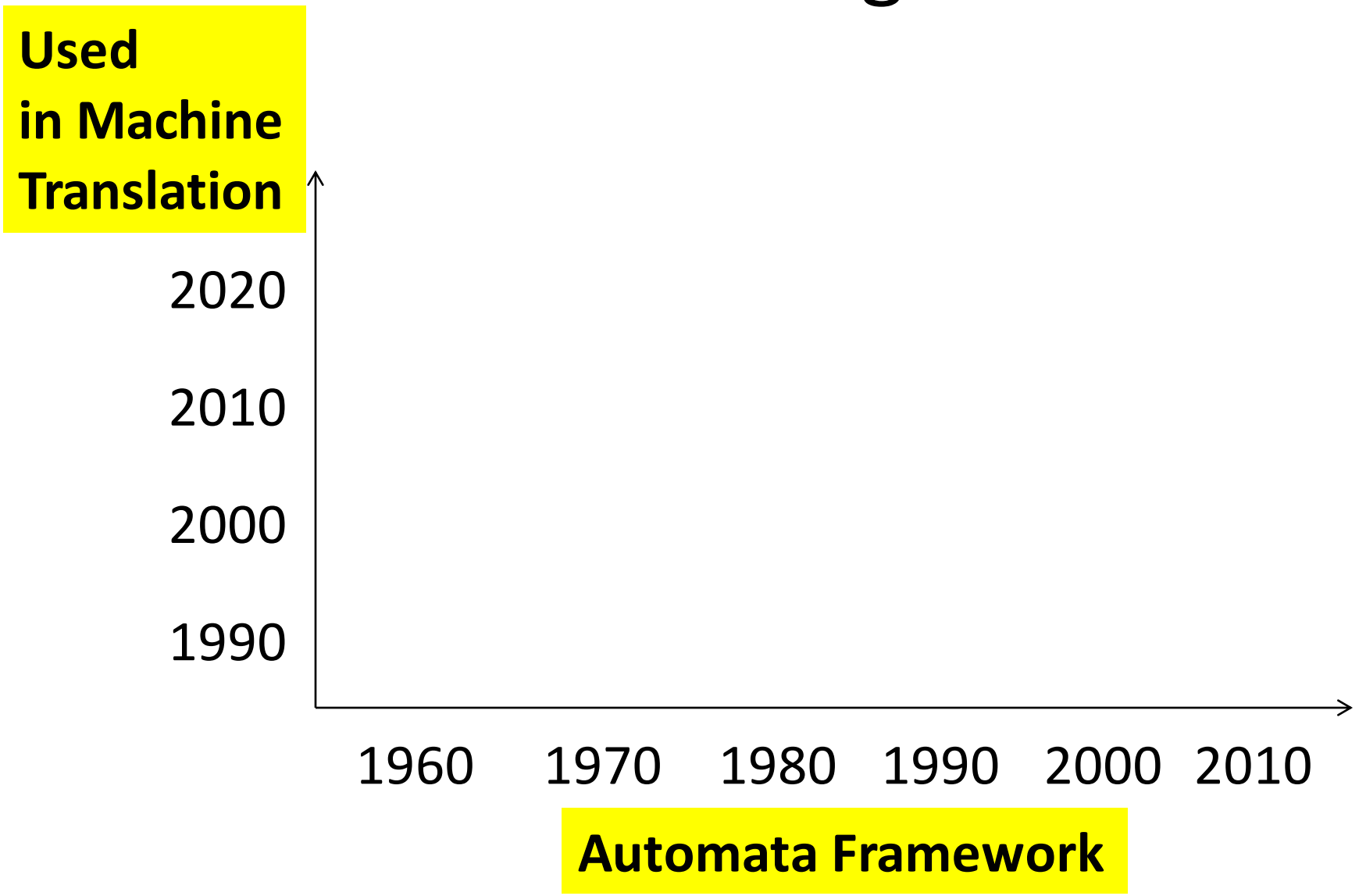
Very Interesting Area

- as more graph/string data becomes available…
- want to fit models to that data…
- algorithmic efficiency is important…

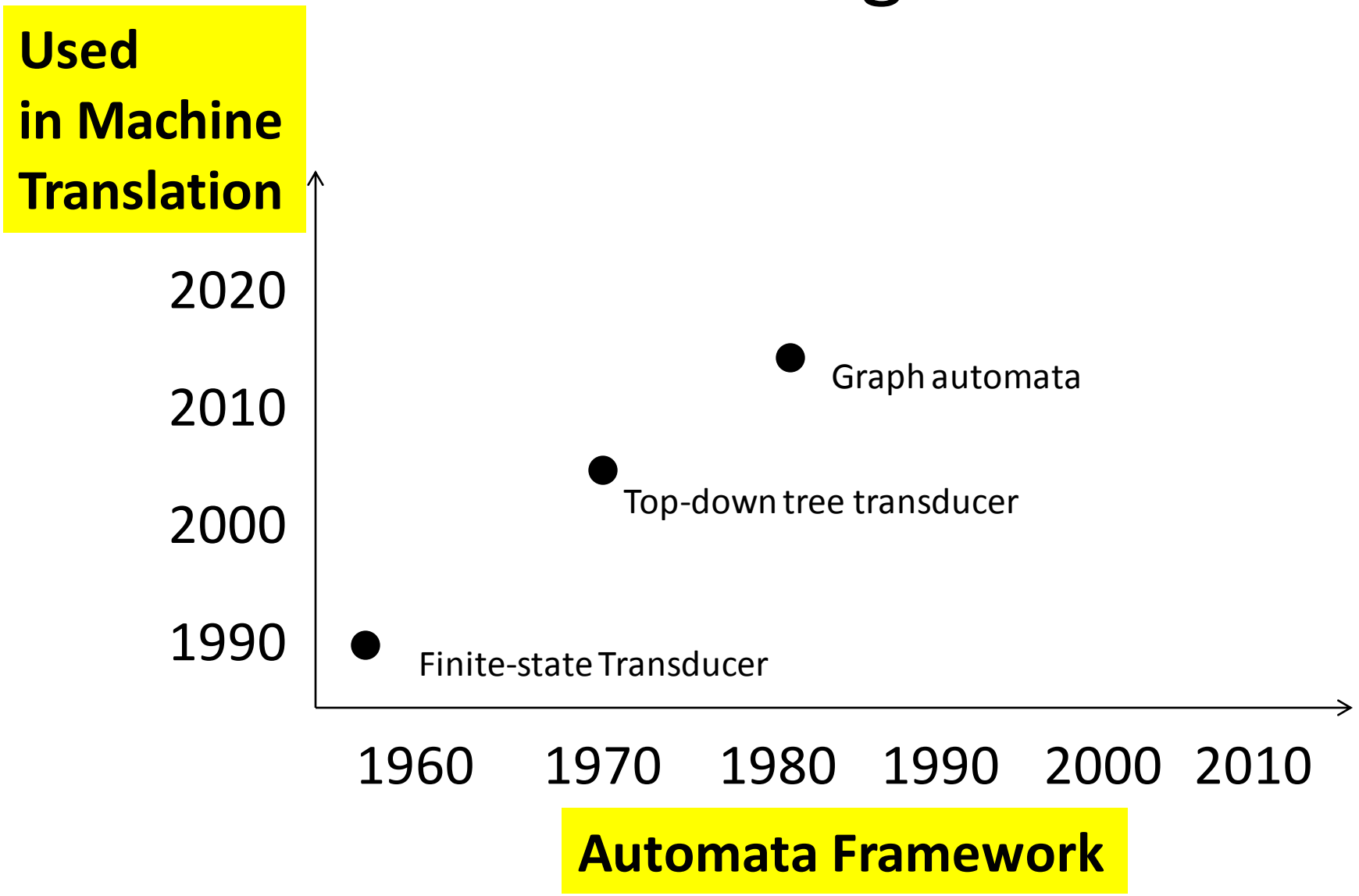
- but, very important to see if the models really capture what is happening in translation data!

thanks

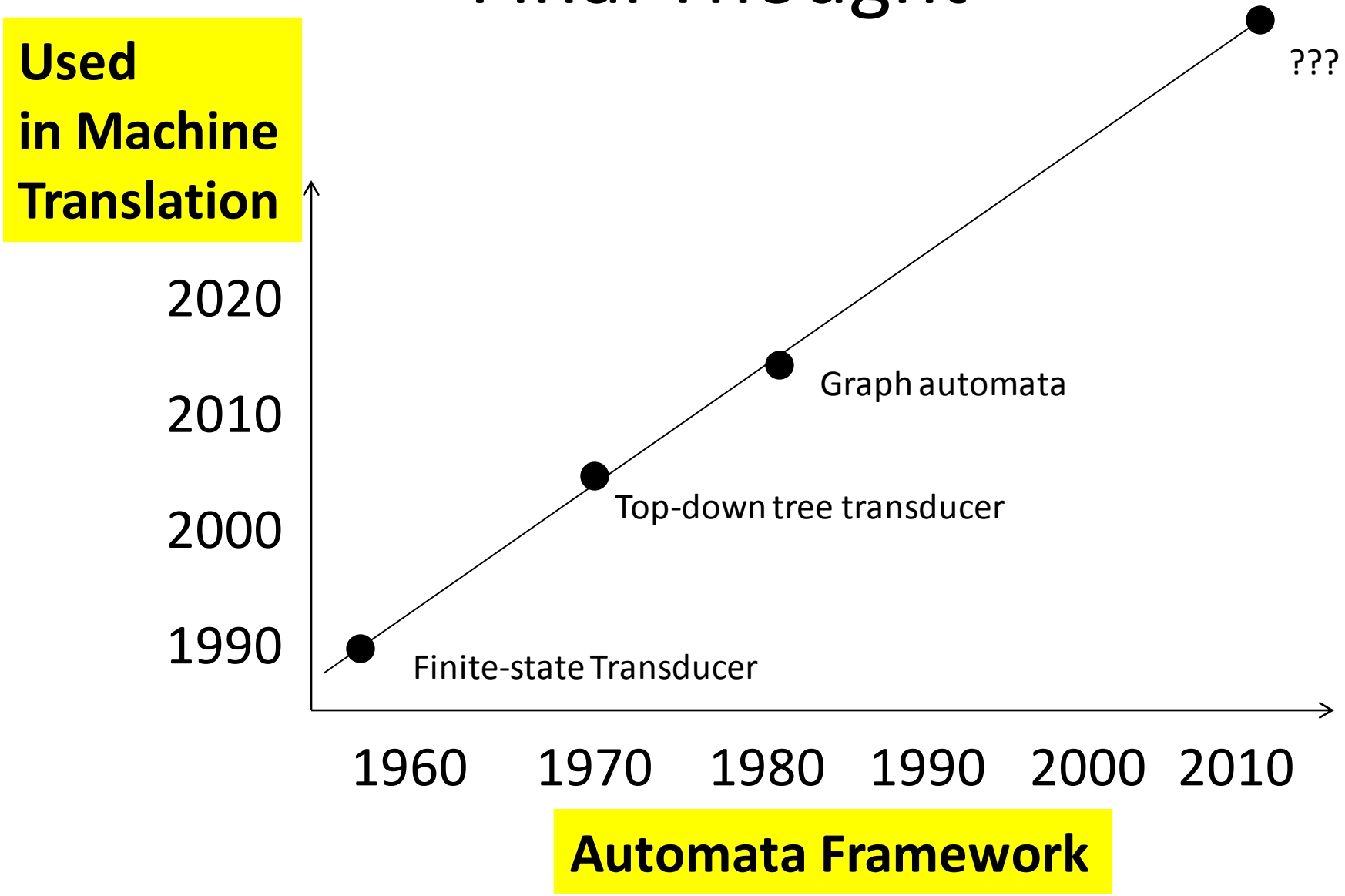
Final Thought



Final Thought



Final Thought



2005

In *Proc. of the Sixth International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*, Lecture Notes in Computer Science, ©Springer Verlag, 2005.

An Overview of Probabilistic Tree Transducers for Natural Language Processing

Kevin Knight and Jonathan Graehl

Information Sciences Institute (ISI) and Computer Science Department
University of Southern California
knight@isi.edu, graehl@isi.edu

Abstract. Probabilistic finite-state string transducers (FSTs) are extremely popular in natural language processing, due to powerful generic methods for applying, composing, and learning them. Unfortunately, FSTs are not a good fit for much of the current work on probabilistic modeling for machine translation, summarization, paraphrasing, and language modeling. These methods operate directly on trees, rather than strings. We show that tree acceptors and tree transducers subsume most of this work, and we discuss algorithms for realizing the same benefits found in probabilistic string transduction.

1 Strings

Many natural language problems have been successfully attacked with finite-state machines. It has been possible to break down very complex problems, both conceptually and literally, into cascades of simpler probabilistic **finite-state transducers** (FSTs). These transducers are bidirectional, and they can be trained on sample input/output string data. By adding a probabilistic **finite-state acceptor** (FSAs) language model to one end of the cascade, we can implement probabilistic **noisy-channel** models.[1] Figure 1 shows a cascade of FSAs and FSTs for the problem of transliterating names and technical terms across languages with different sounds and writing systems [1].

The finite-state framework is popular because it offers powerful, generic operations for statistical reasoning and learning. There are standard algorithms for:

- **intersection** of FSAs
- **forward application** of strings and FSAs through FSTs
- **backward application** of strings and FSAs through FSTs
- **composition** of FSTs
- **k-best** path extraction
- supervised and unsupervised **training** of FST transition probabilities from data

Goals:

- introduce tree automata to NLP practitioners
- make connections between MT and theory
- list some open issues

Open Issues from [Knight & Graehl 05]

1. What is the most efficient algorithm for selecting the k-best trees from a probabilistic regular tree grammar (RTG)?

   [Jiménez and Marzal 00; Huang & Chiang 05; Pauls & Klein 09].
   Still no separation of k and n, as in [Eppstein 94] for FSA.

8. What is the linguistically most appropriate tree transducer class for machine translation? For summarization? Which classes best handle the most common linguistic constructions, and which classes best handle the most difficult ones?

   Unclear.

10. What are the theoretical and computational properties of extended left-hand-side transducers (**x**)? E.g., is xRLN closed under composition?

    [Maletti, Graehl, Hopkins & Knight 09]

11. Where do synchronous grammars [50,17] and tree cloning [15] fit into the tree transducer hierarchy?

[Shieber 06]

13. Are there tree transducers that can move unbounded material over unbounded distances, while maintaining efficient computational properties?

Unclear.

16. Can we build useful, generic tree-transducer toolkits, and what sorts of programming interfaces will be most effective?

[May & Knight 06; May 10]

14. In analogy with extended context-free grammars [35], are there types of tree transducers that can process tree sets which are not limited to a finite set of rewrites (e.g., $S \rightarrow NP\ VP\ PP^*$)?

"Horizontal" processing of input trees

12. As many syntactic and semantic theories generate acyclic graphs rather than trees, can graph transducers adequately capture the desired transformations?

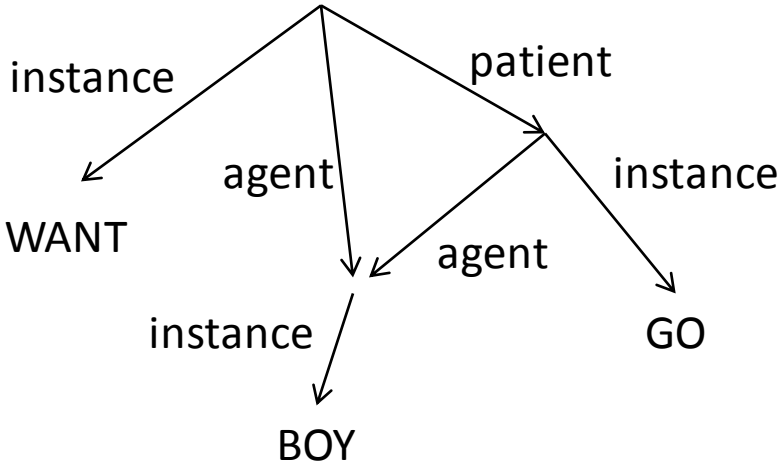
Hmm...

Semantic Structure

"The boy wants to go."

DIRECTED ACYCLIC GRAPH

Rooted, edge-labeled, leaf-labeled graphs



FEATURE STRUCTURE

instance: WANT
agent: [1] (instance: BOY)
patient: (instance: GO
agent: [1])

LOGICAL FORM

∃ w, b, g : instance(w, WANT) ^
instance(g, GO) ^
instance(b, BOY) ^
agent(w, b) ^
patient(w, g) ^
agent(g, b)

PENMAN

```
(w / WANT
  :agent (b / BOY)
  :patient (g / GO
             :agent b)))
```