Beyond Words

Morphology & Syntax

CS6120: Natural Language Processing
Northeastern University

David Smith

A Language

- Some sentences in the language
  - The man took the book.
  - Colorless green ideas sleep furiously.
  - This sentence is false.
- Some sentences not in the language
  - *The girl, the sidewalk, the chalk, drew.
  - *Backwards is sentence this.
  - *Je parle anglais.

Languages as Rewriting Systems

- Start with some "non-terminal" symbol **S**
- Expand that symbol, using a **rewrite rule**.
- Keep applying rules until all non-terminals are expanded to terminals.
- The string of terminals is a sentence of the language.

Chomsky Hierarchy

- Let Caps = nonterminals; lower = terminals; Greek = strings of terms/nonterms
- Recursively enumerable (Turing equivalent)
  - Rules: $\alpha \rightarrow \beta$
- Context-sensitive
  - Rules: $\alpha A \beta \rightarrow \alpha \gamma \beta$
- Context-free
  - Rules: $A \rightarrow \alpha$
- Regular (finite-state)
  - Rules: $A \rightarrow aB$; $A \rightarrow a$

Regular Language Example

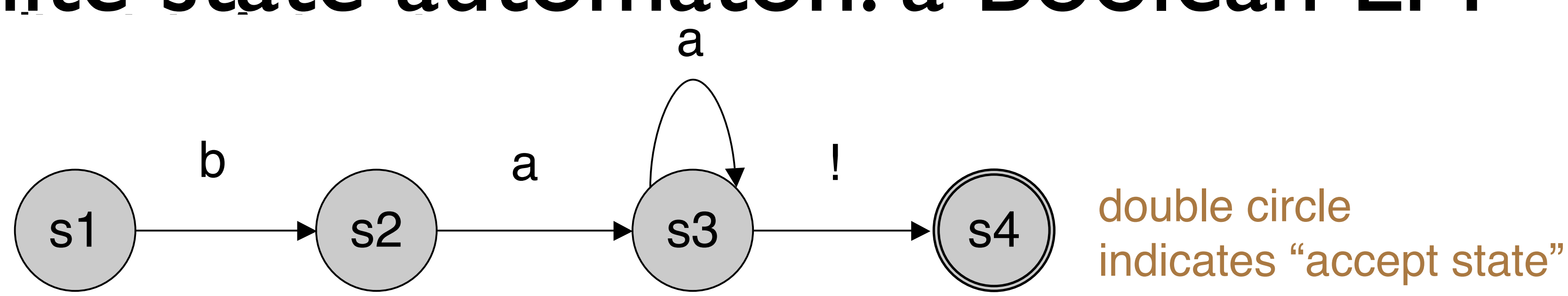
- Nonterminals: S, X
- Terminals: m, o
- Rules:
  - S→mX
  - X→oX
  - X→o
- Start symbol: S

One expansion

S
mX
moX
mooX
mooo

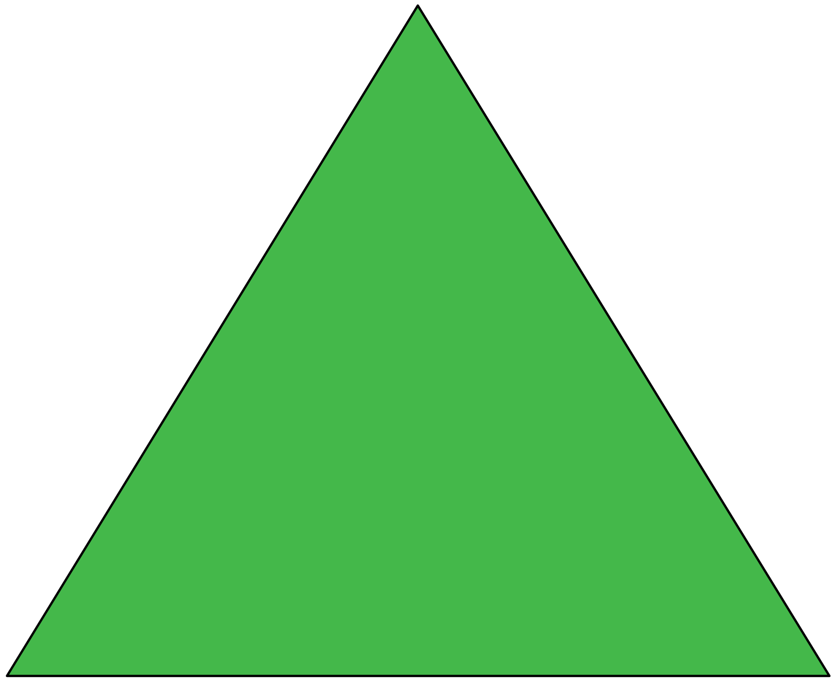
Another Regular Language

- Strings in and not in this language
  - In the language:
    - “ba!”, “baa!”, “baaaaaaa!”
  - Not in the language:
    - “ba”, “b!”, “ab!”, “bbaaa!”, “alibaba!”
- Regular expression: baa*!
- Finite state automaton: a Boolean LM



Regular Languages

Regular Languages
the accepted strings



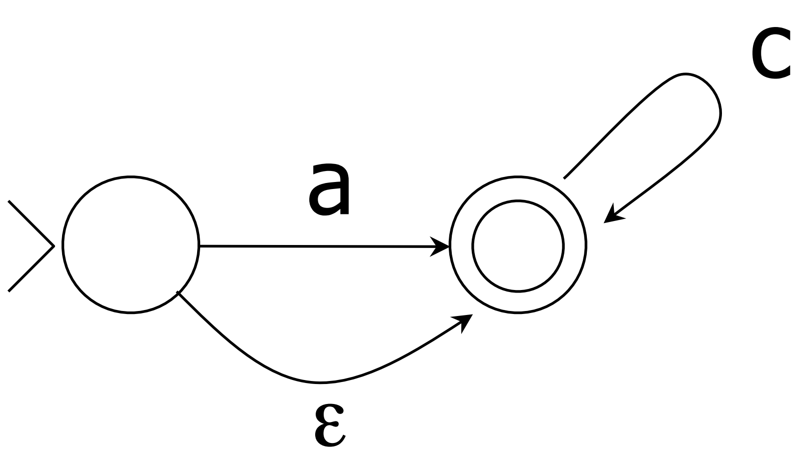
Finite-state Automata
machinery for accepting

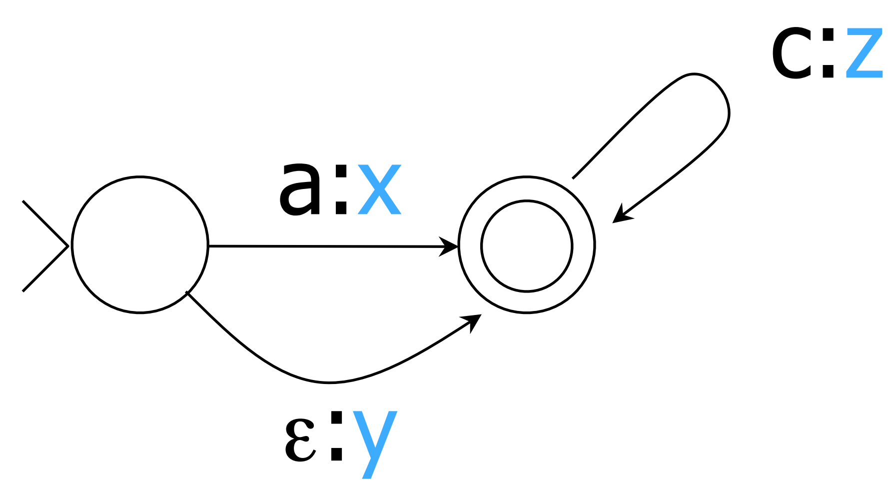
Regular Expressions
a way to type the automata

Function from strings to ...

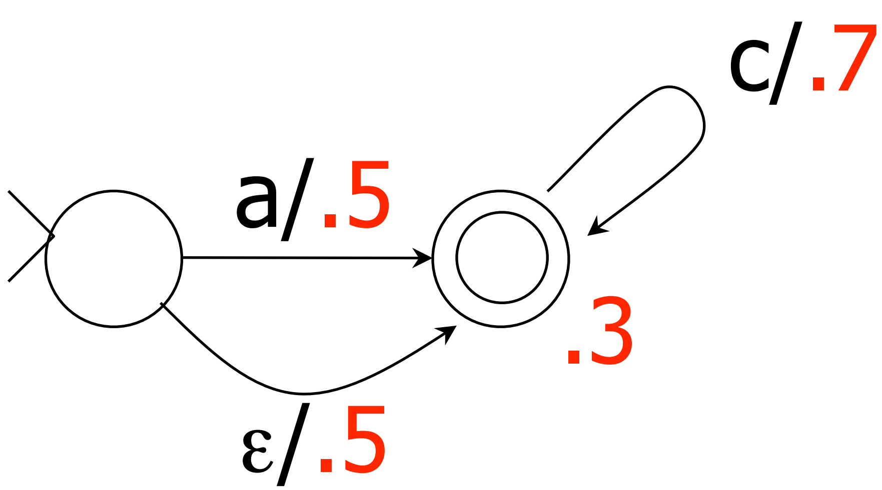
Acceptors (FSAs) | **Transducers (FSTs)**

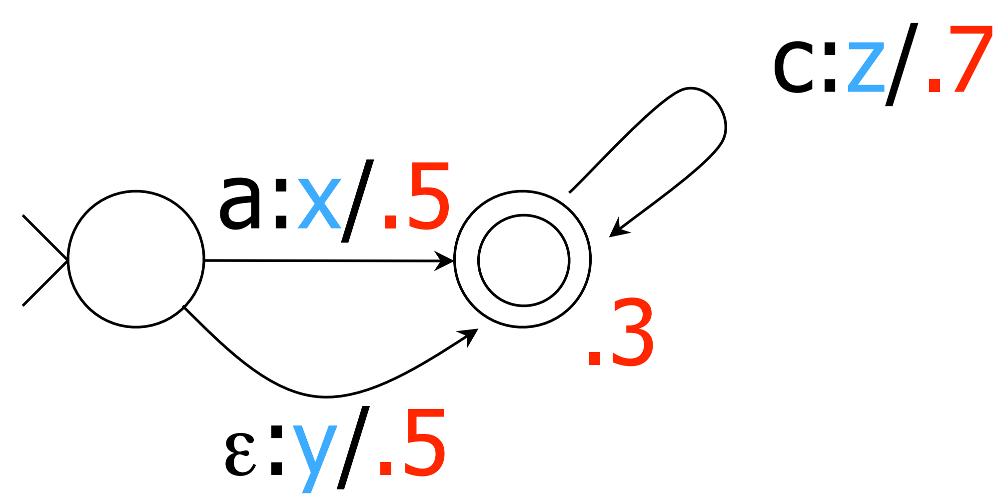
Unweighted





Weighted



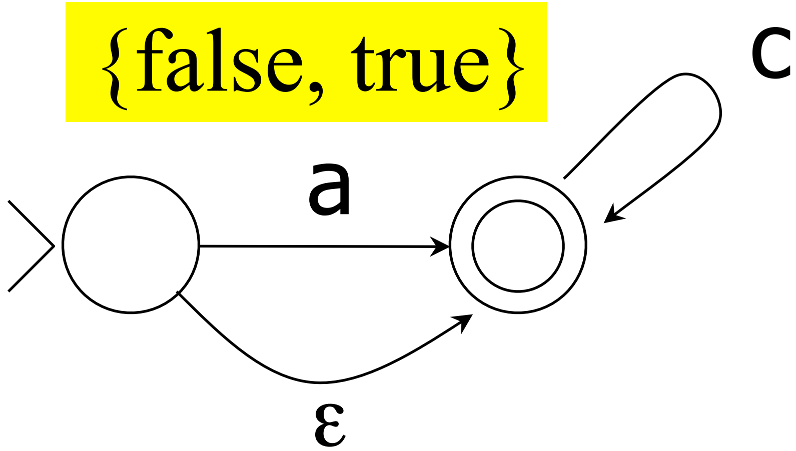


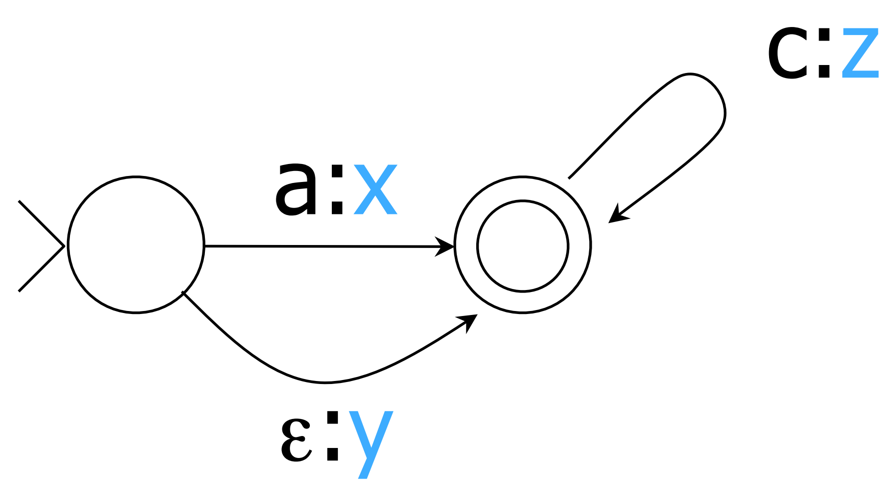
Function from strings to ...

Acceptors (FSAs) | **Transducers (FSTs)**

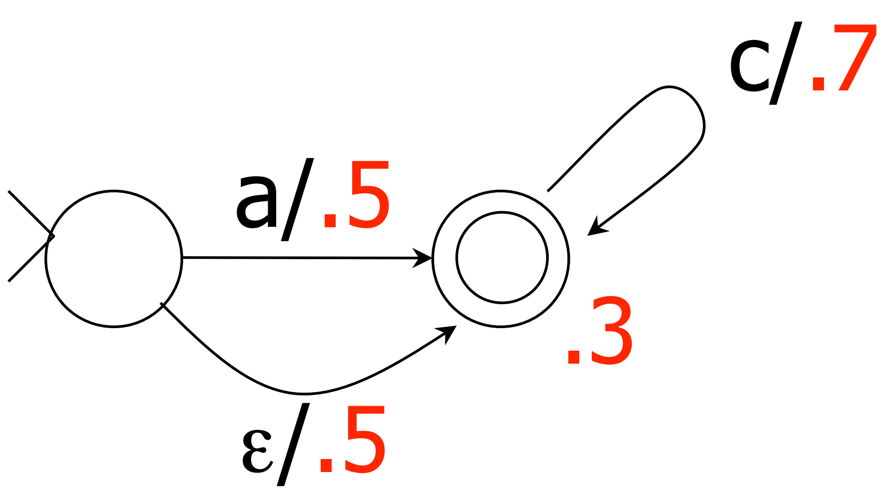
{false, true}

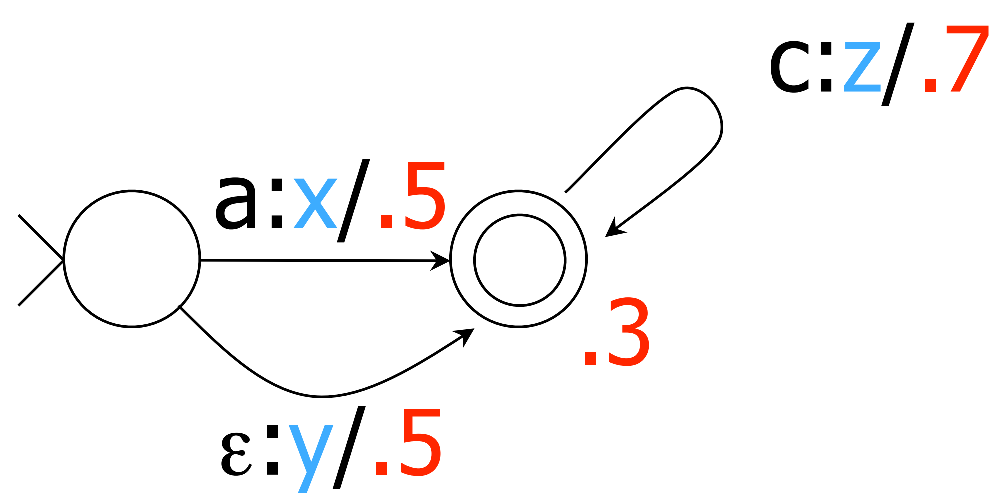
Unweighted





Weighted



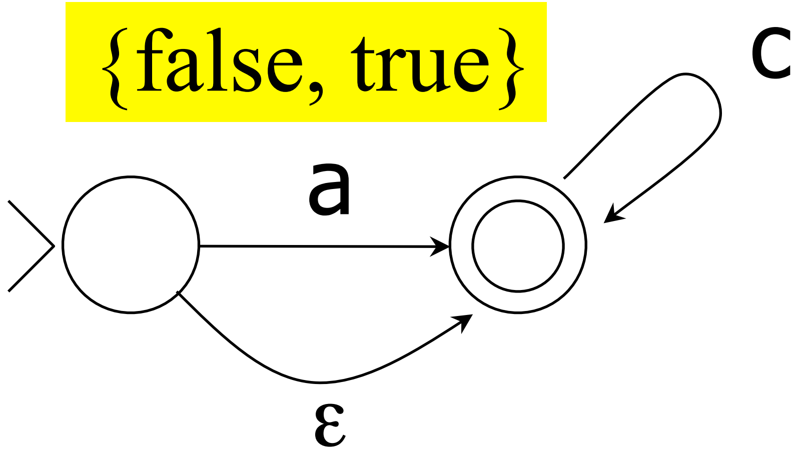


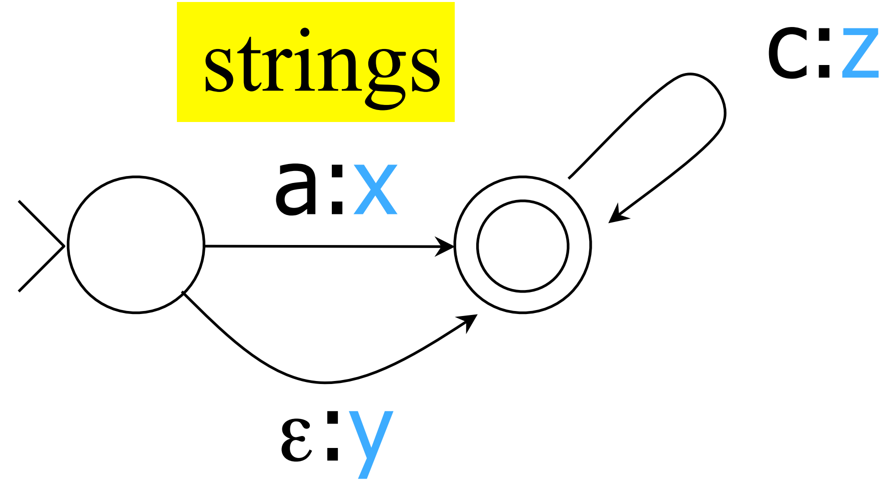
Function from strings to ...

Acceptors (FSAs) | **Transducers (FSTs)**

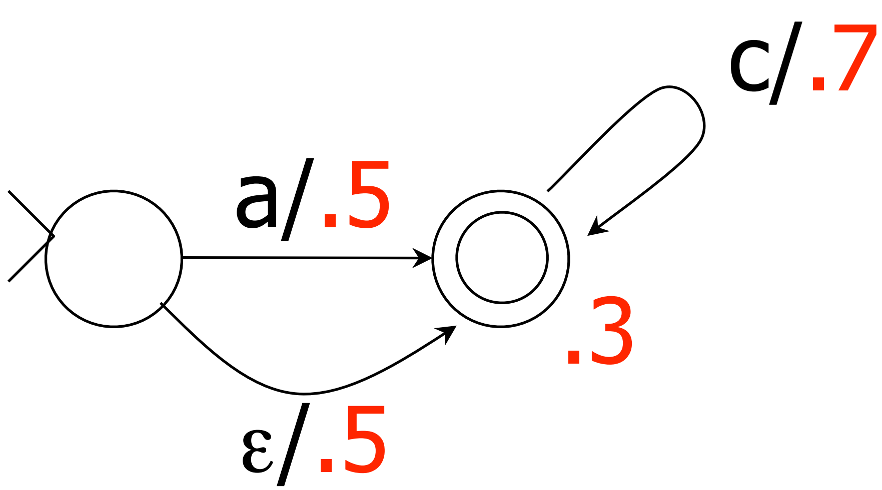
{false, true} | strings

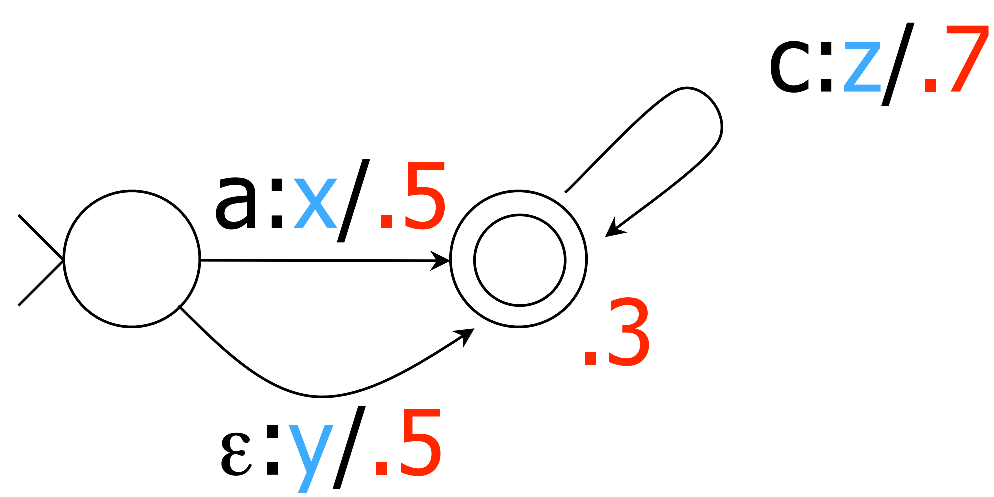
Unweighted



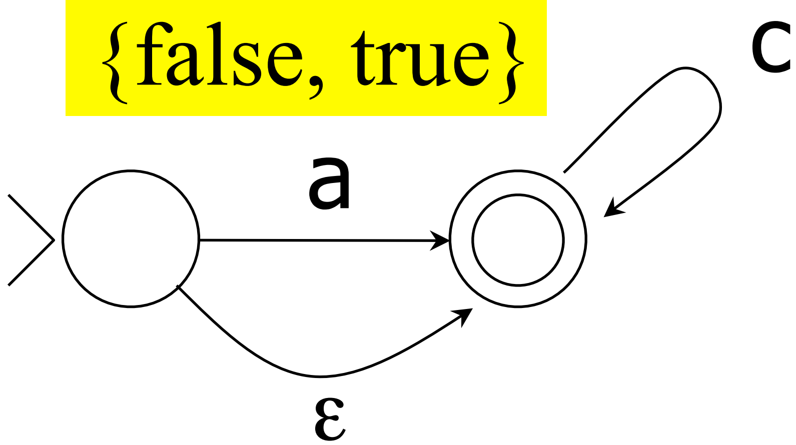
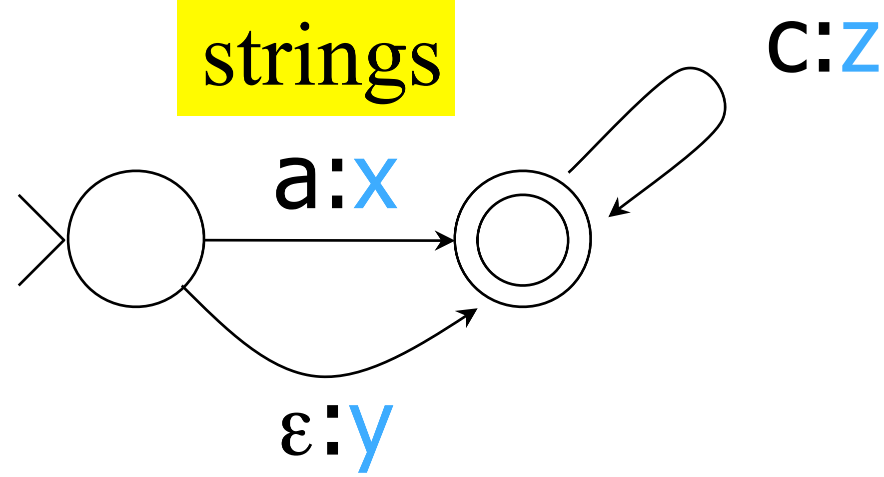
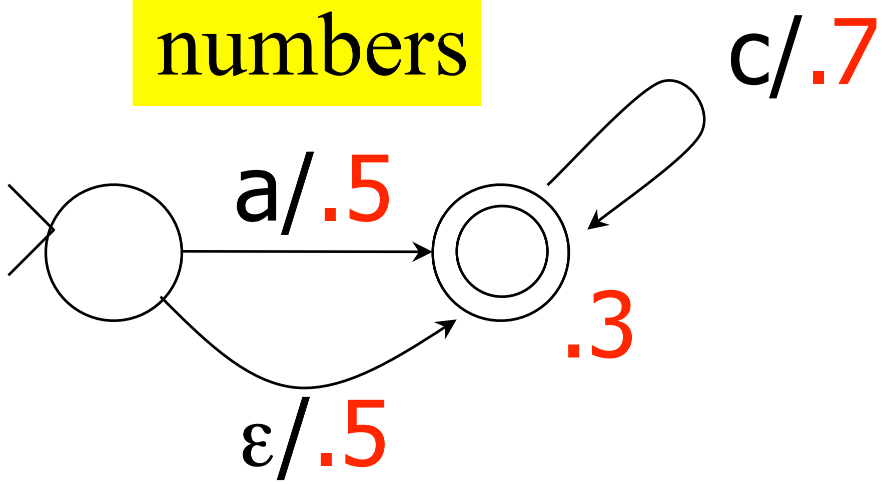
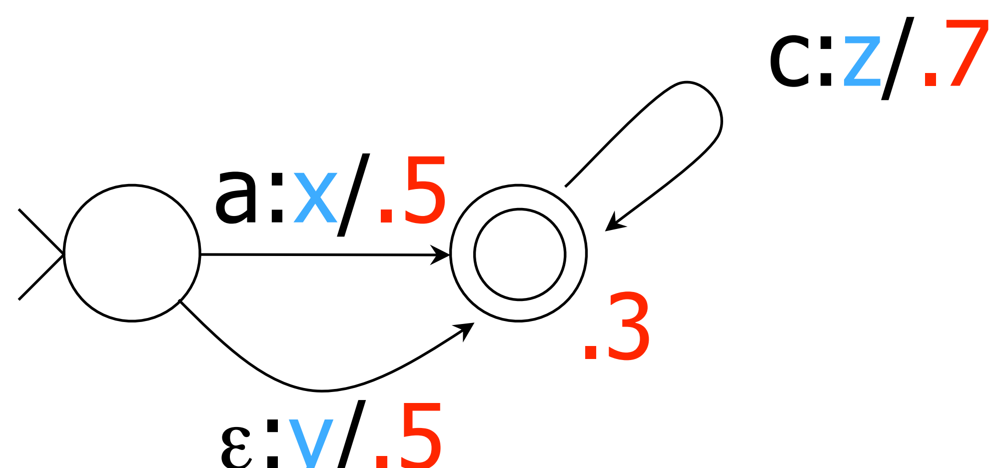


Weighted





Function from strings to ...

	Acceptors (FSAs)	Transducers (FSTs)
Unweighted	{false, true} 	strings 
Weighted	numbers 	

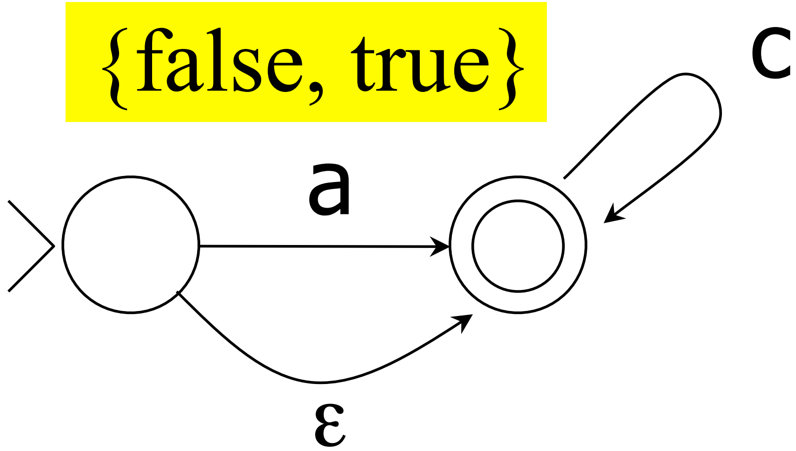
Function from strings to ...

Acceptors (FSAs)

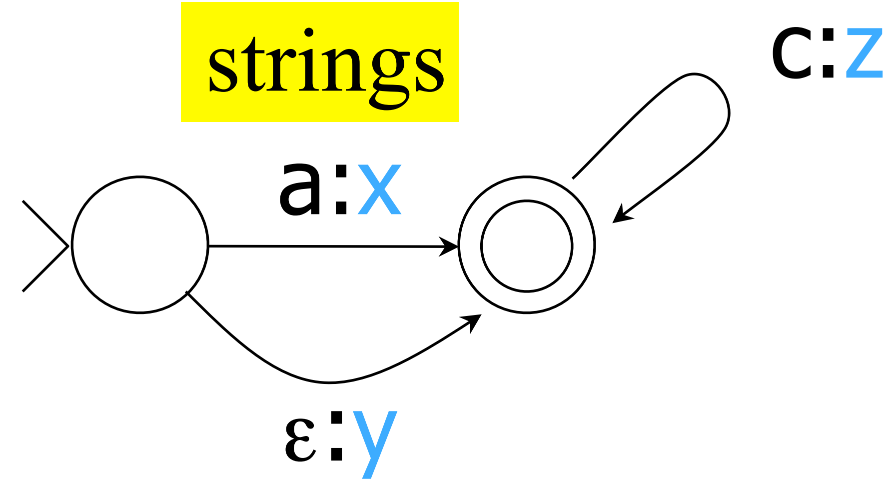
Transducers (FSTs)

Unweighted

{false, true}

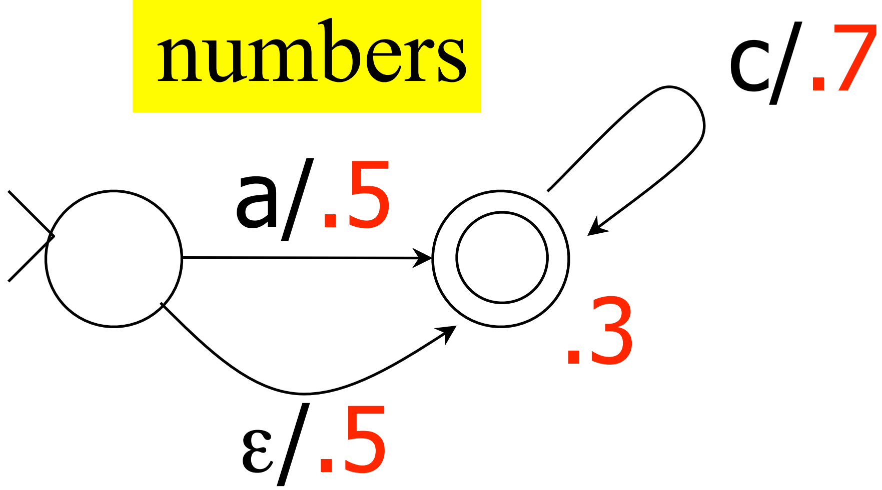


strings

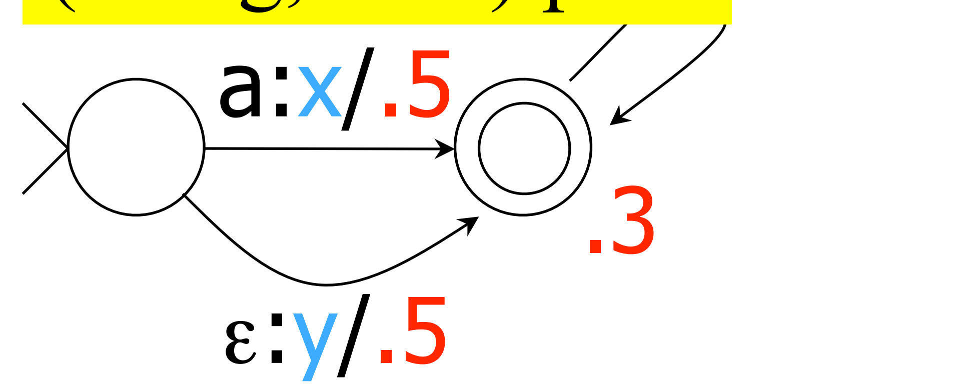


Weighted

numbers



(string, num) pairs



Sample functions

	Acceptors (FSAs)	Transducers (FSTs)
Unweighted	{false, true}	strings
Weighted	numbers	(string, num) pairs

Sample functions

	Acceptors (FSAs)	Transducers (FSTs)
Unweighted	{false, true} Grammatical?	strings
Weighted	numbers	(string, num) pairs

Sample functions

	Acceptors (FSAs)	Transducers (FSTs)
Unweighted	{false, true} Grammatical?	strings
Weighted	numbers How grammatical? Better, how likely?	(string, num) pairs

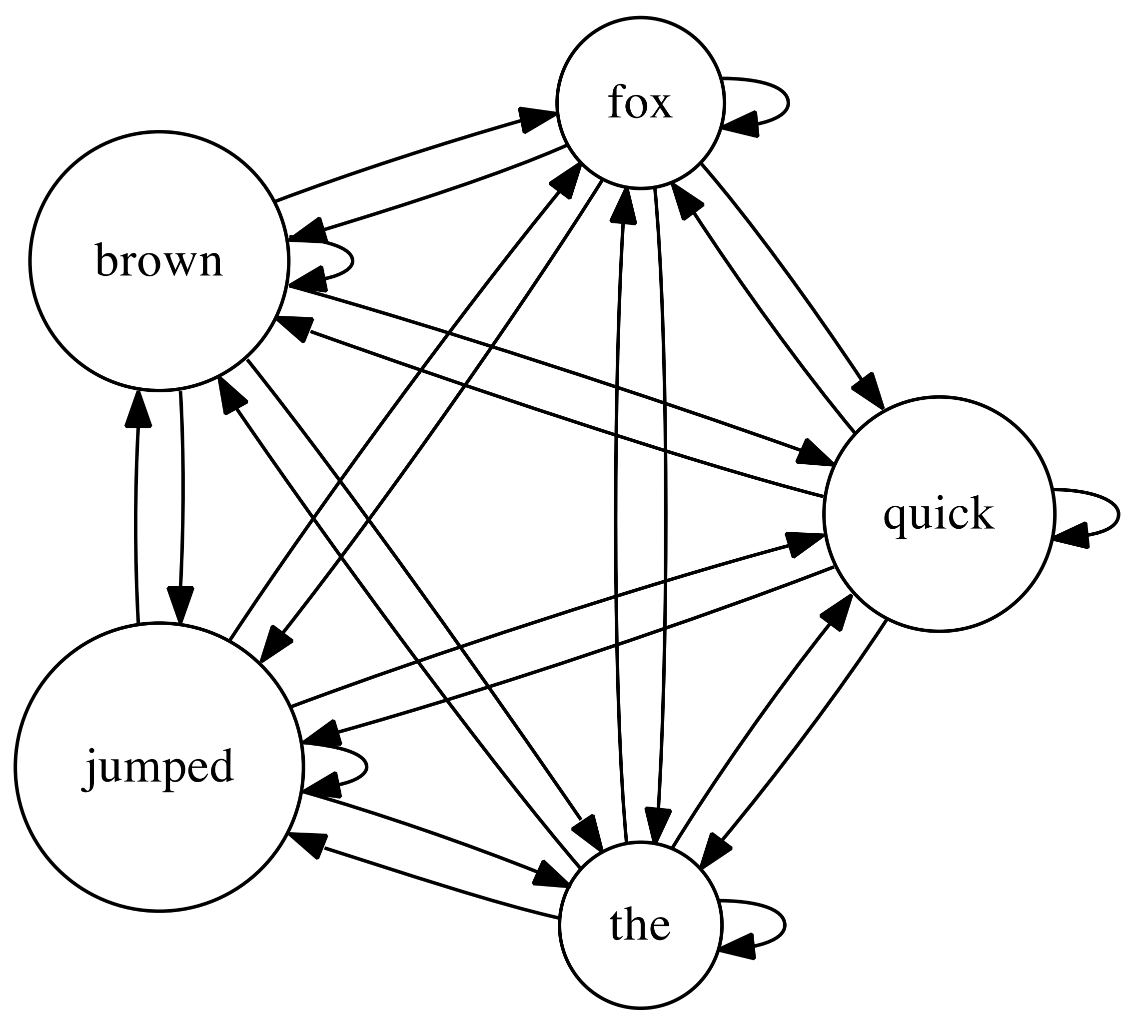
Sample functions

	Acceptors (FSAs)	Transducers (FSTs)
Unweighted	{false, true} Grammatical?	strings Markup Correction Translation
Weighted	numbers How grammatical? Better, how likely?	(string, num) pairs

Sample functions

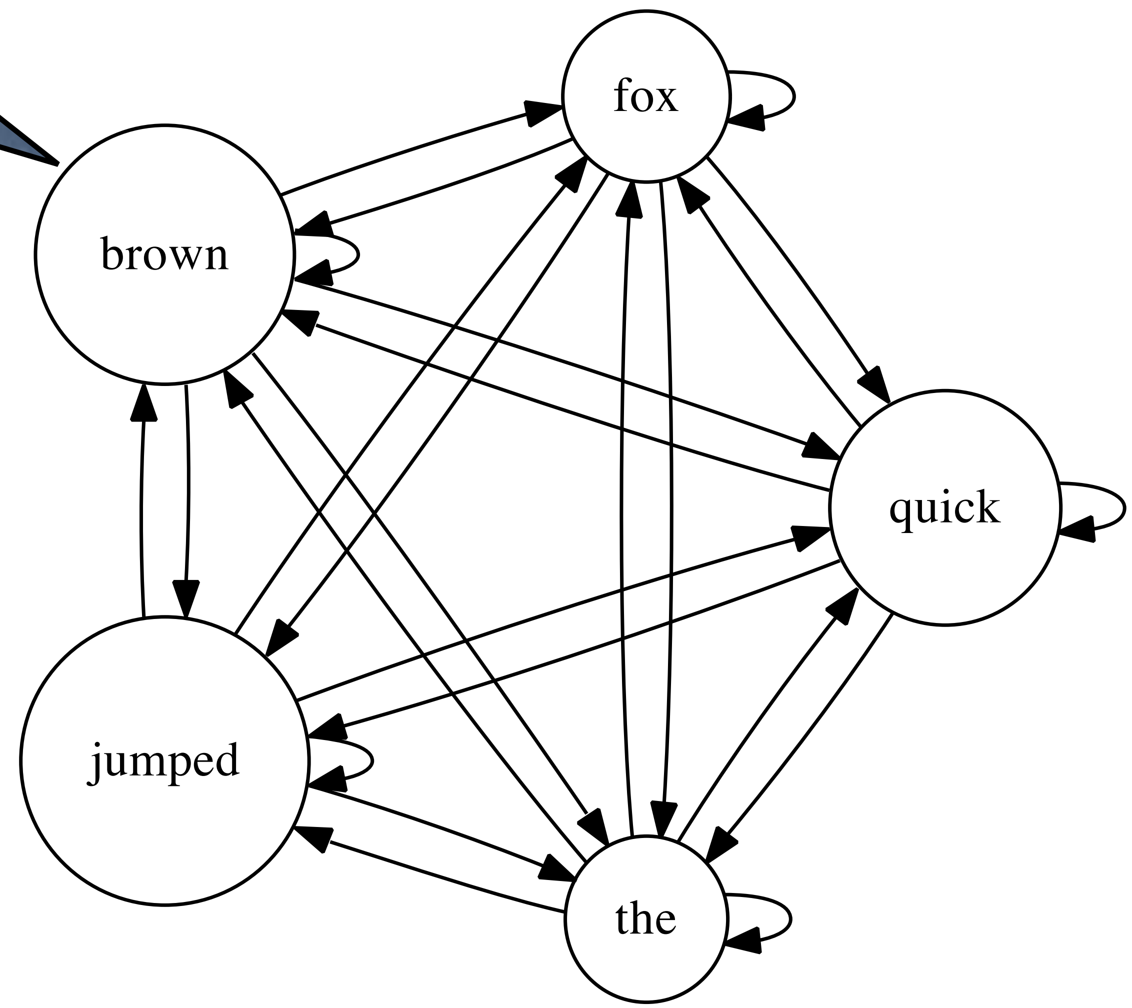
	Acceptors (FSAs)	Transducers (FSTs)
Unweighted	{false, true} Grammatical?	strings Markup Correction Translation
Weighted	numbers How grammatical? Better, how likely?	(string, num) pairs Good markups Good corrections Good translations

Bigram LM as WFSM



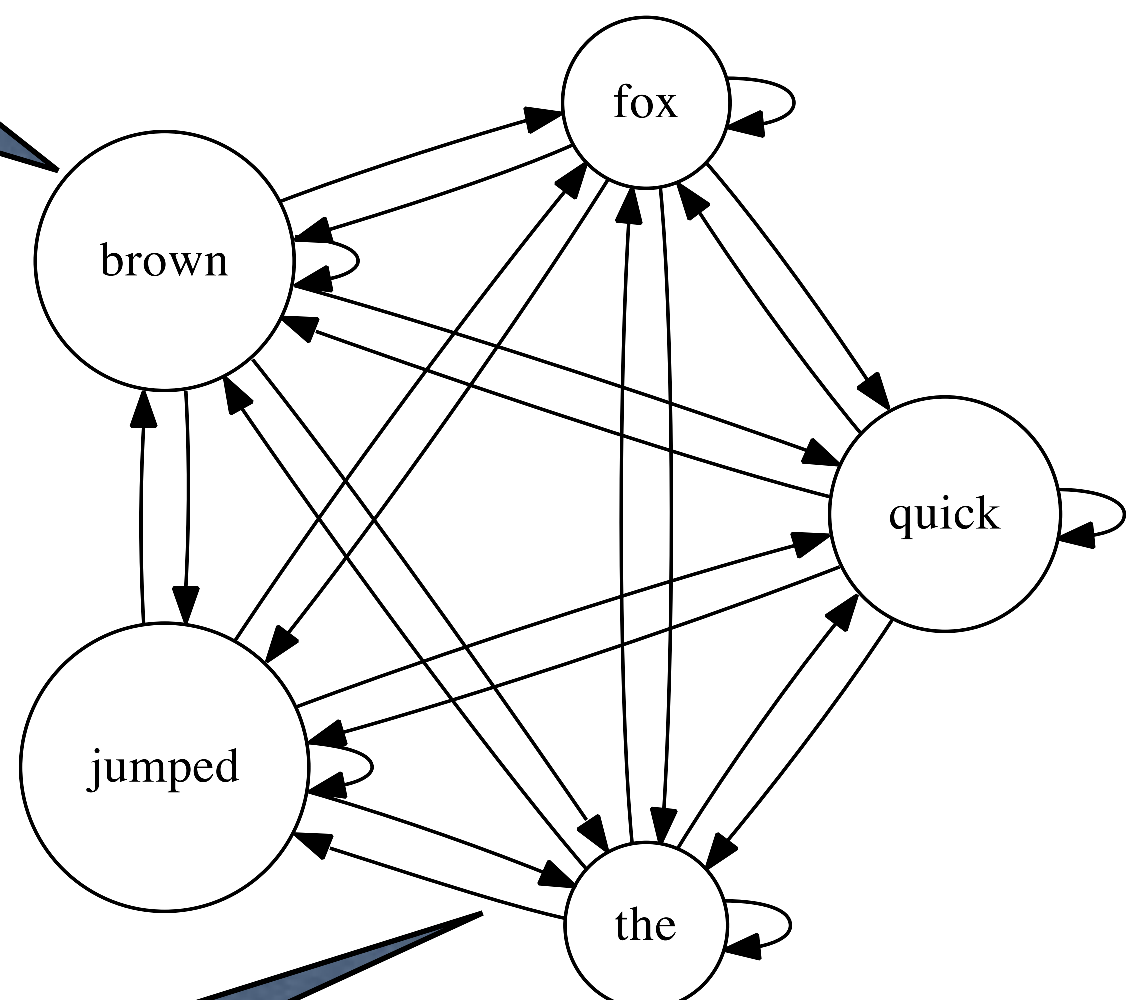
Bigram LM as WFSM

V states



Bigram LM as WFSM

V states

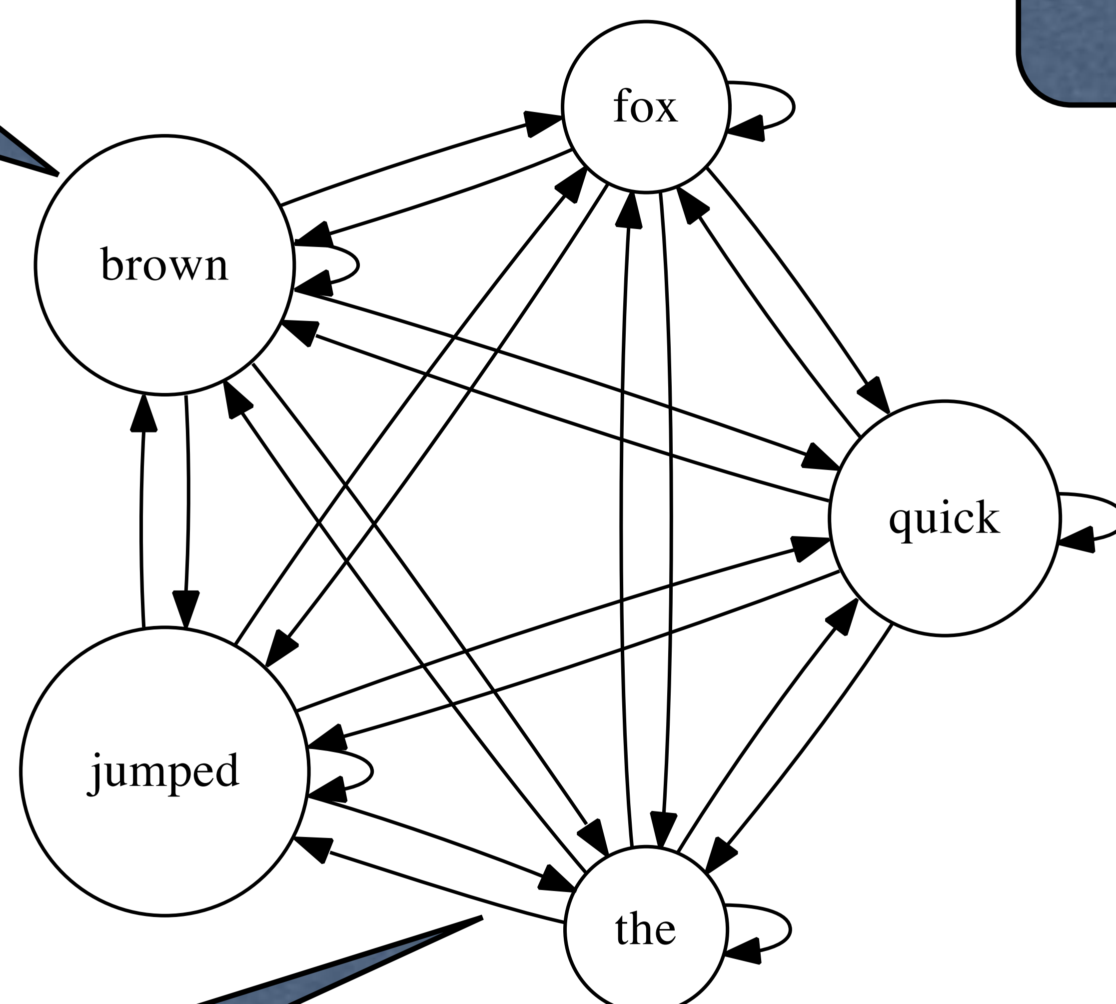


O(V²) arcs
(& parameters)

Bigram LM as WFSM

V states

What about a trigram model?

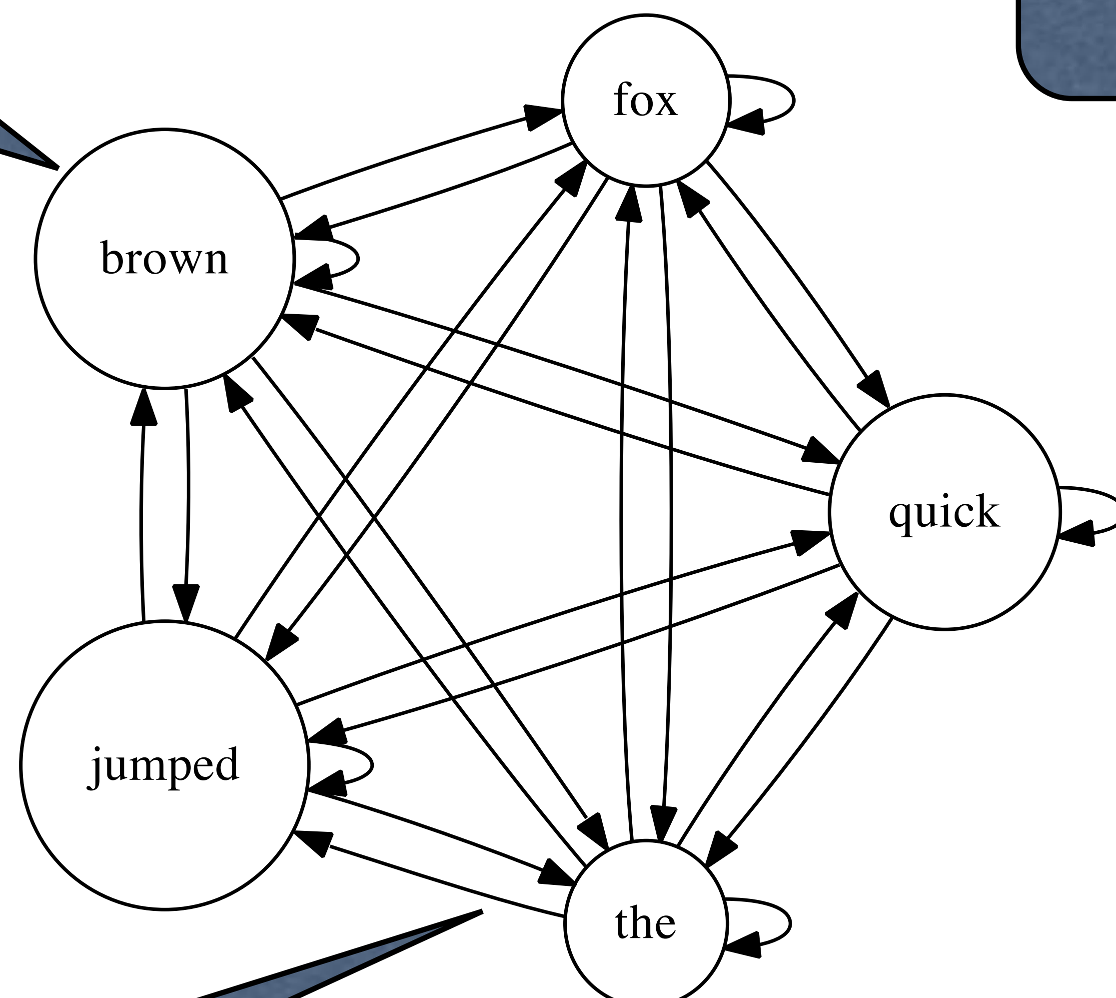


O(V²) arcs
(& parameters)

Bigram LM as WFSM

V states

What about a trigram model?



O(V²) arcs
(& parameters)

What about backoff?

Noisy Channels (Again)

Word Segmentation

theprophetsaidtothecity

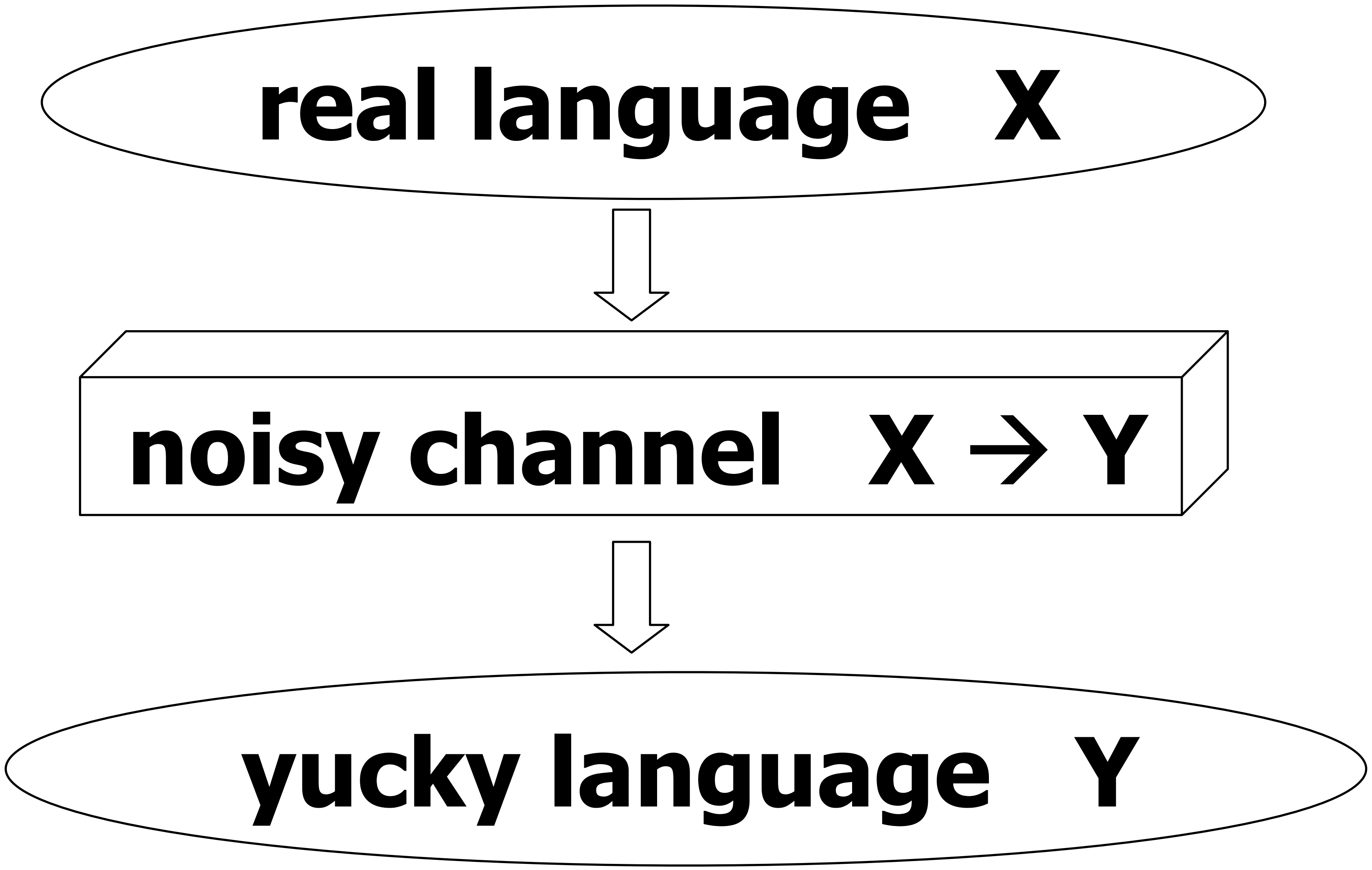
- What does this say?
  - And what other words are substrings?

- Given L = a “lexicon” FSA that matches all English words.
- How to apply to this problem?
- What if Lexicon is weighted?
- From unigrams to bigrams?
- Smooth L to include unseen words?

Spelling correction

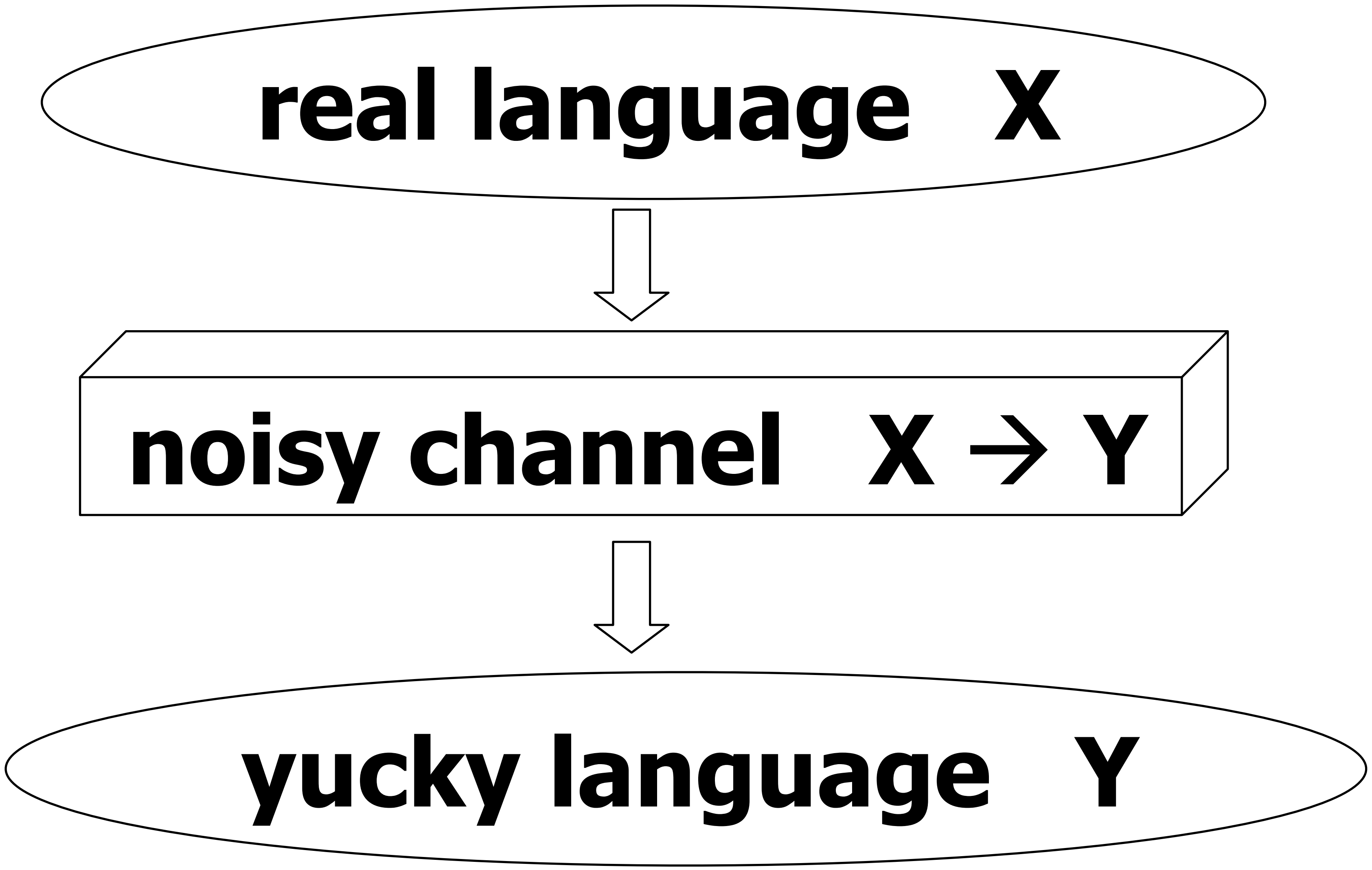
- Spelling correction also needs a lexicon L
- But there is distortion …
  - Let T be a transducer that models common typos and other spelling errors
    - ance (→) ence (deliverance, ...)
    - e → ε (deliverance, ...)
    - ε → e // Cons _ Cons (athlete, ...)
    - rr → r (embarrass, occurrence, …)
    - ge → dge (privilege, …)
    - etc.
  - Now what can you do with L .o. T ?
- Should T and L have probabilities?
- Want T to include "all possible" errors …

Noisy Channel Model



want to recover X from Y

Noisy Channel Model



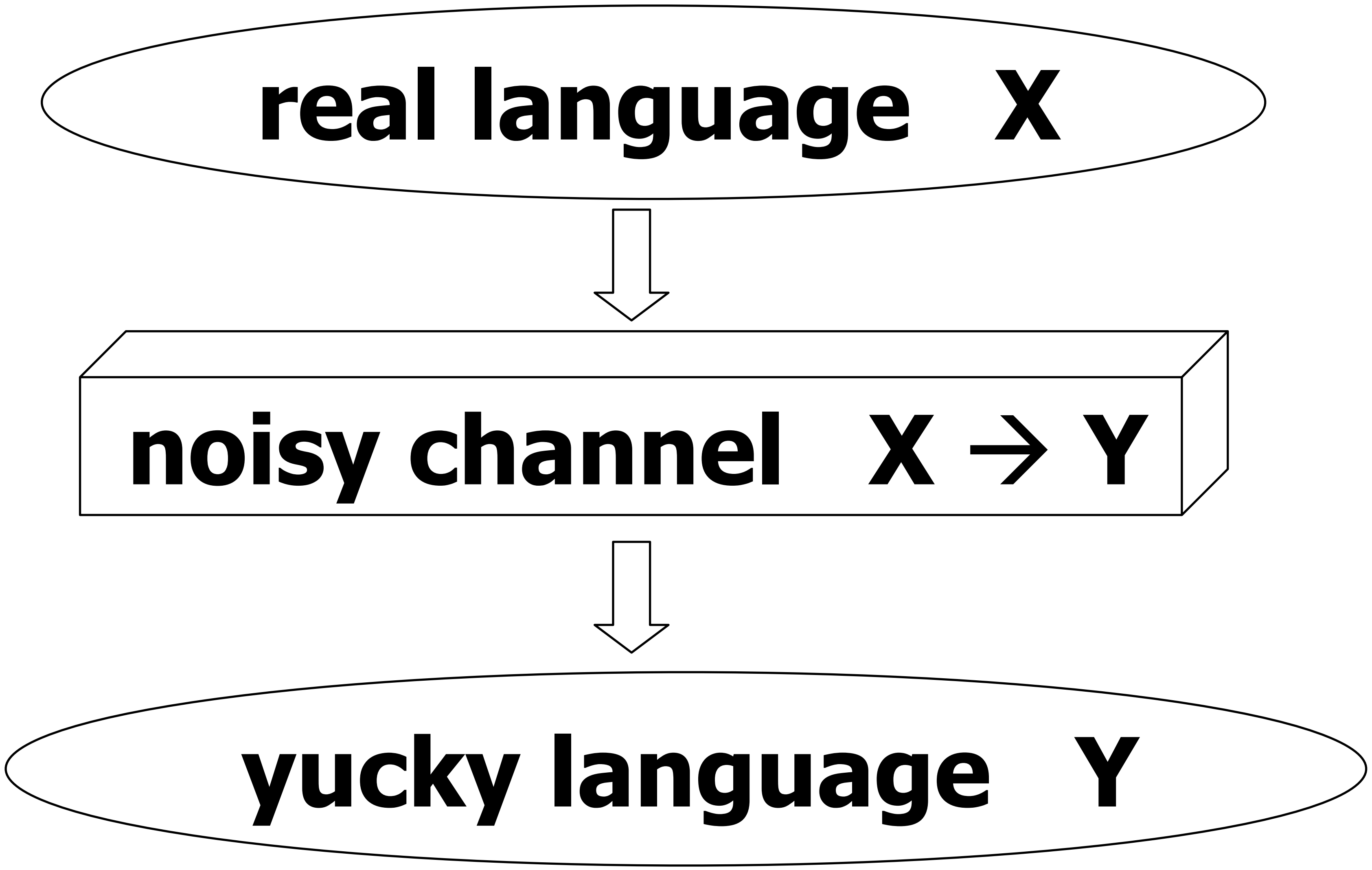
correct spelling

typos

misspelling

want to recover X from Y

Noisy Channel Model



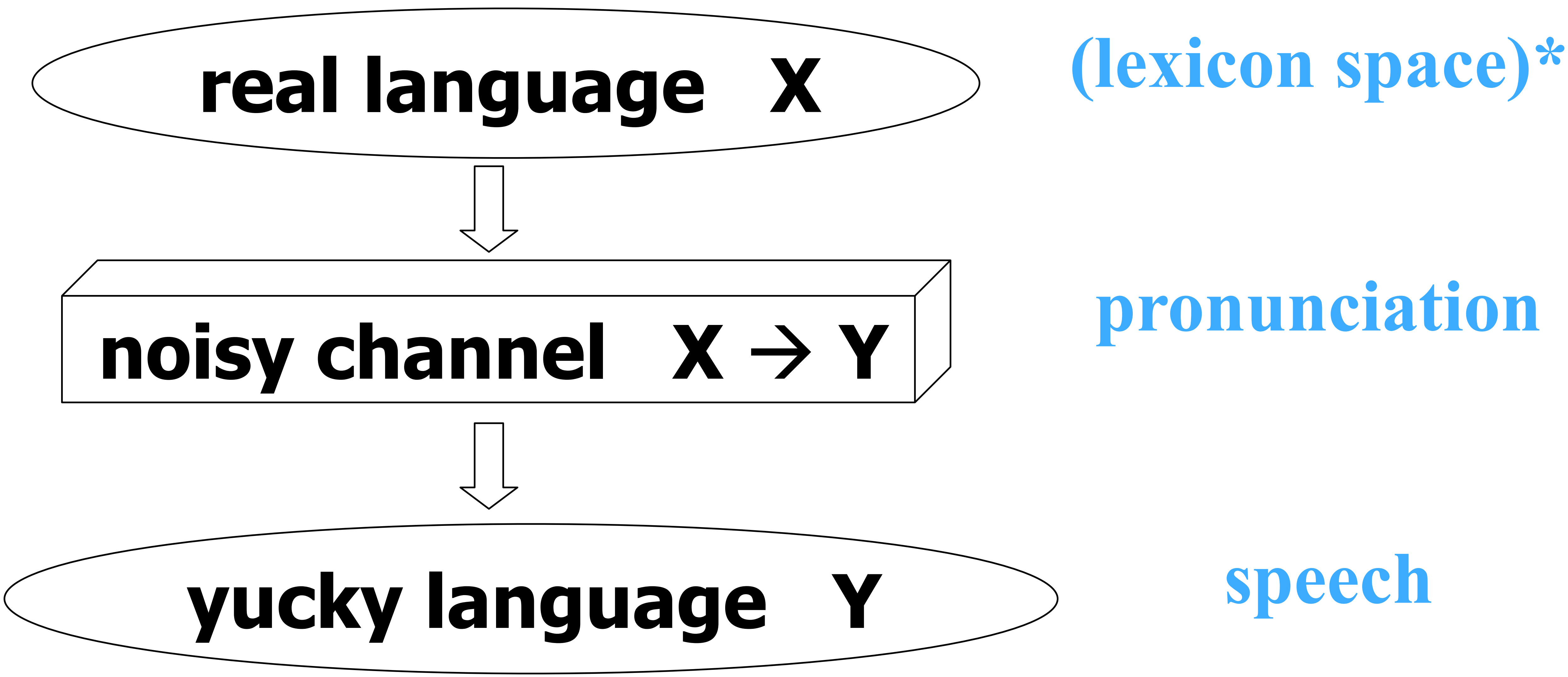
(lexicon space)*

delete spaces

text w/o spaces

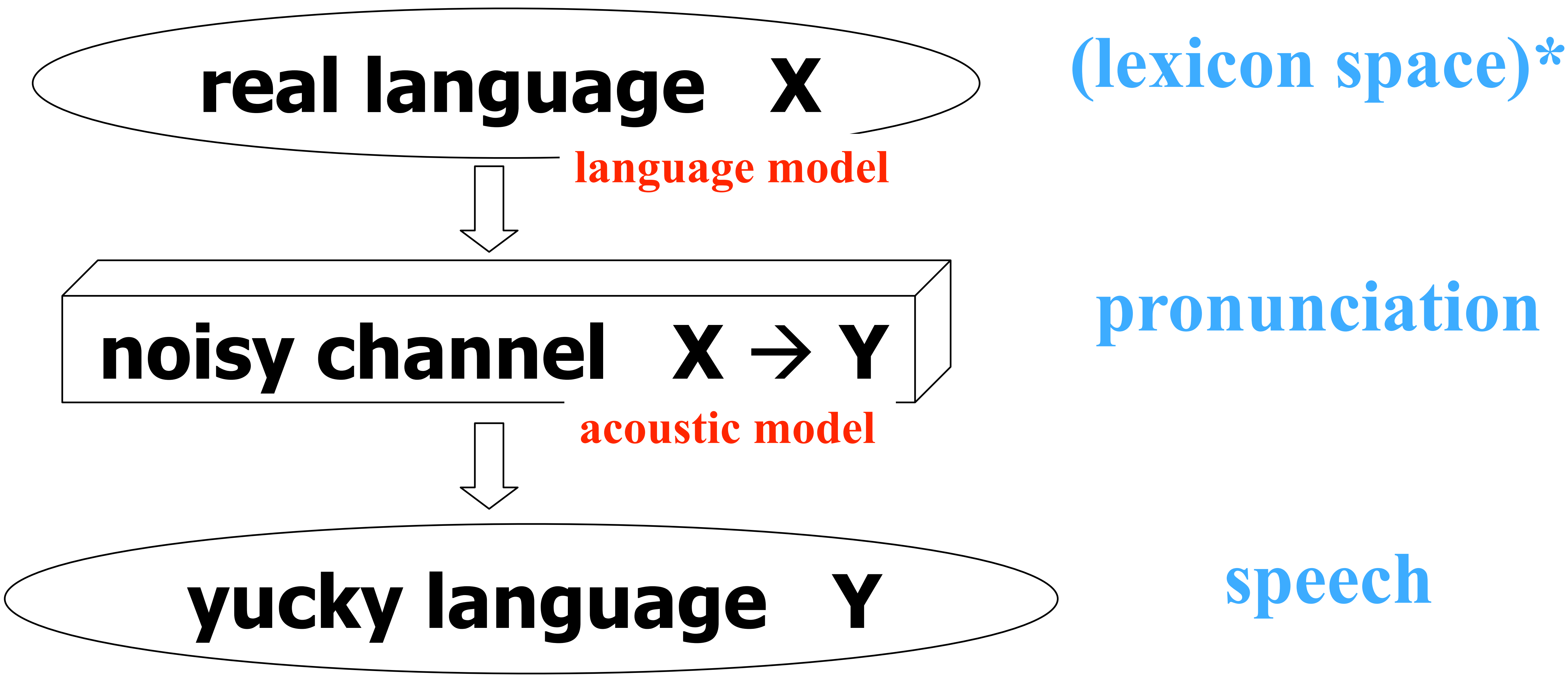
want to recover X from Y

Noisy Channel Model



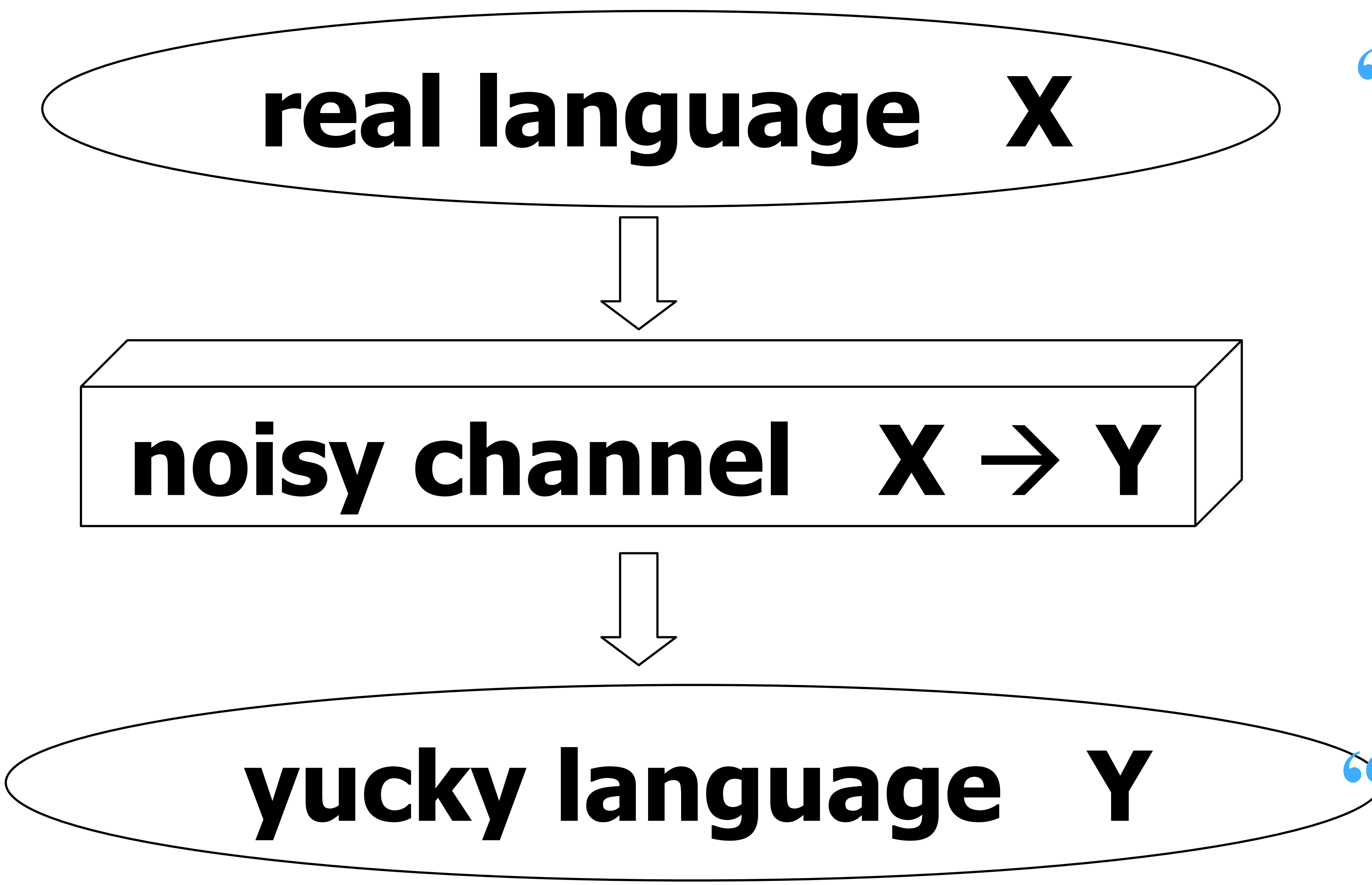
want to recover X from Y

Noisy Channel Model



want to recover X from Y

Noisy Channel Model



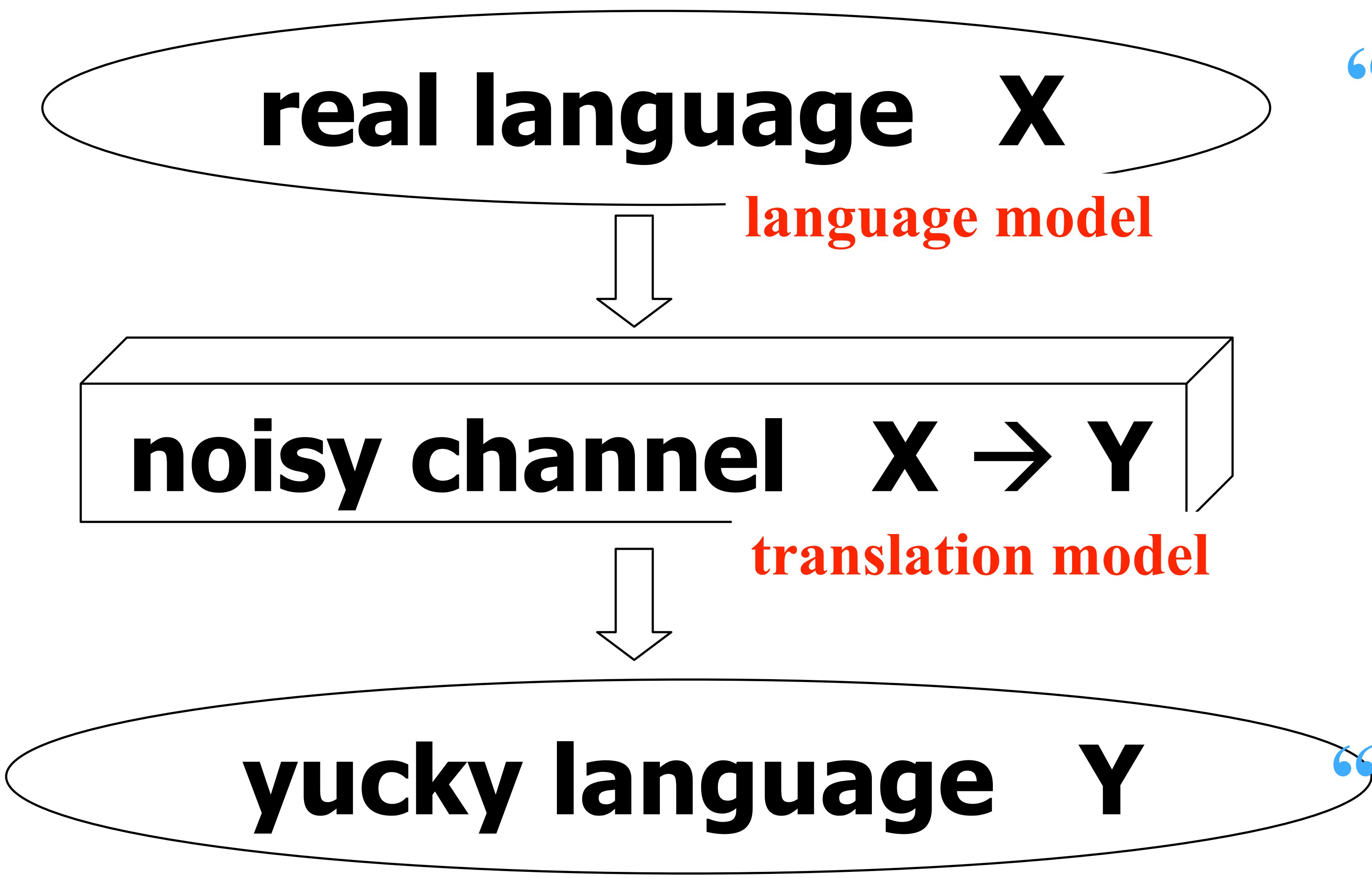
“target” language

translation

“source” language

want to recover X from Y

Noisy Channel Model



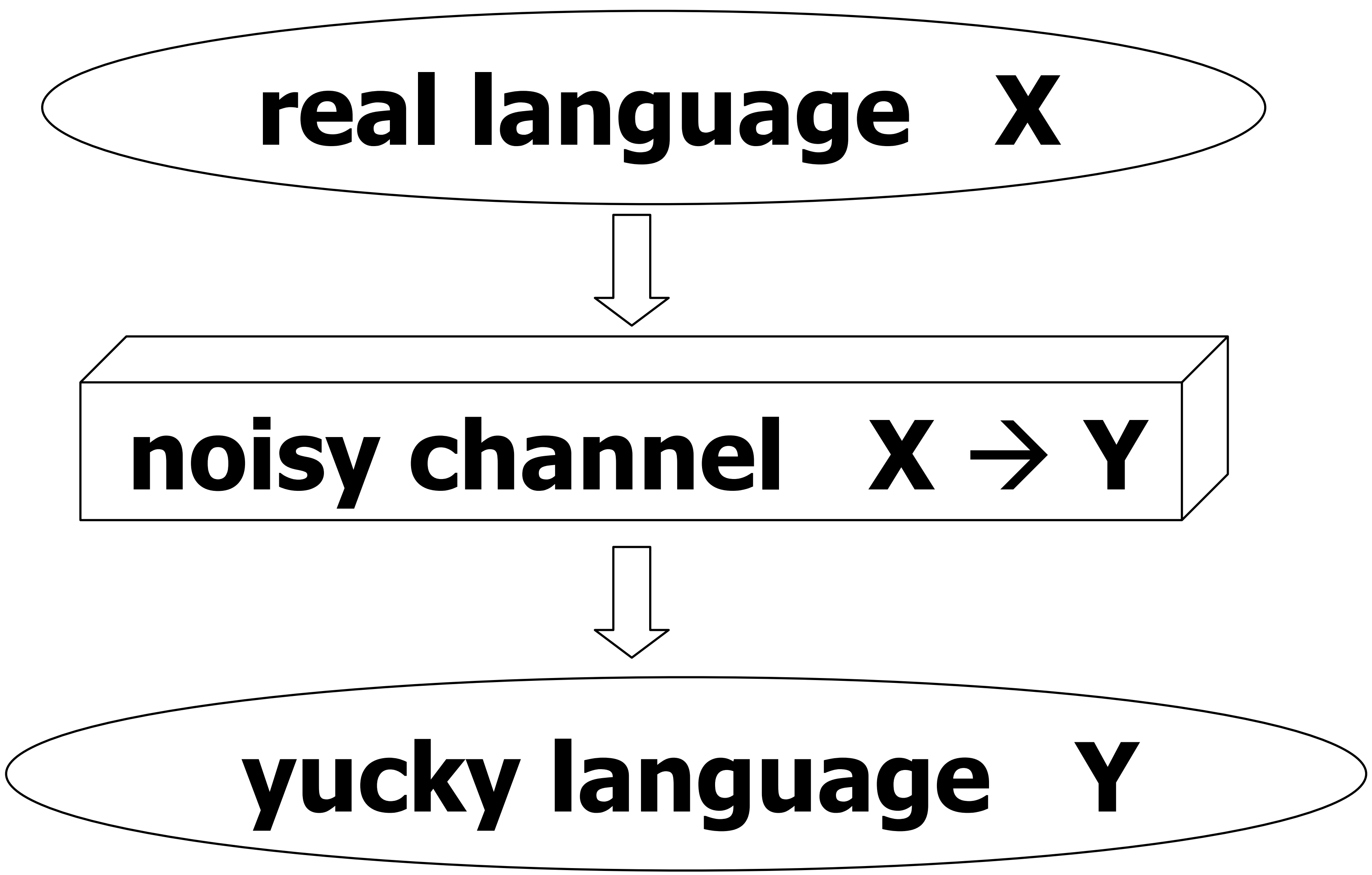
“target” language

translation

“source” language

want to recover X from Y

Noisy Channel Model



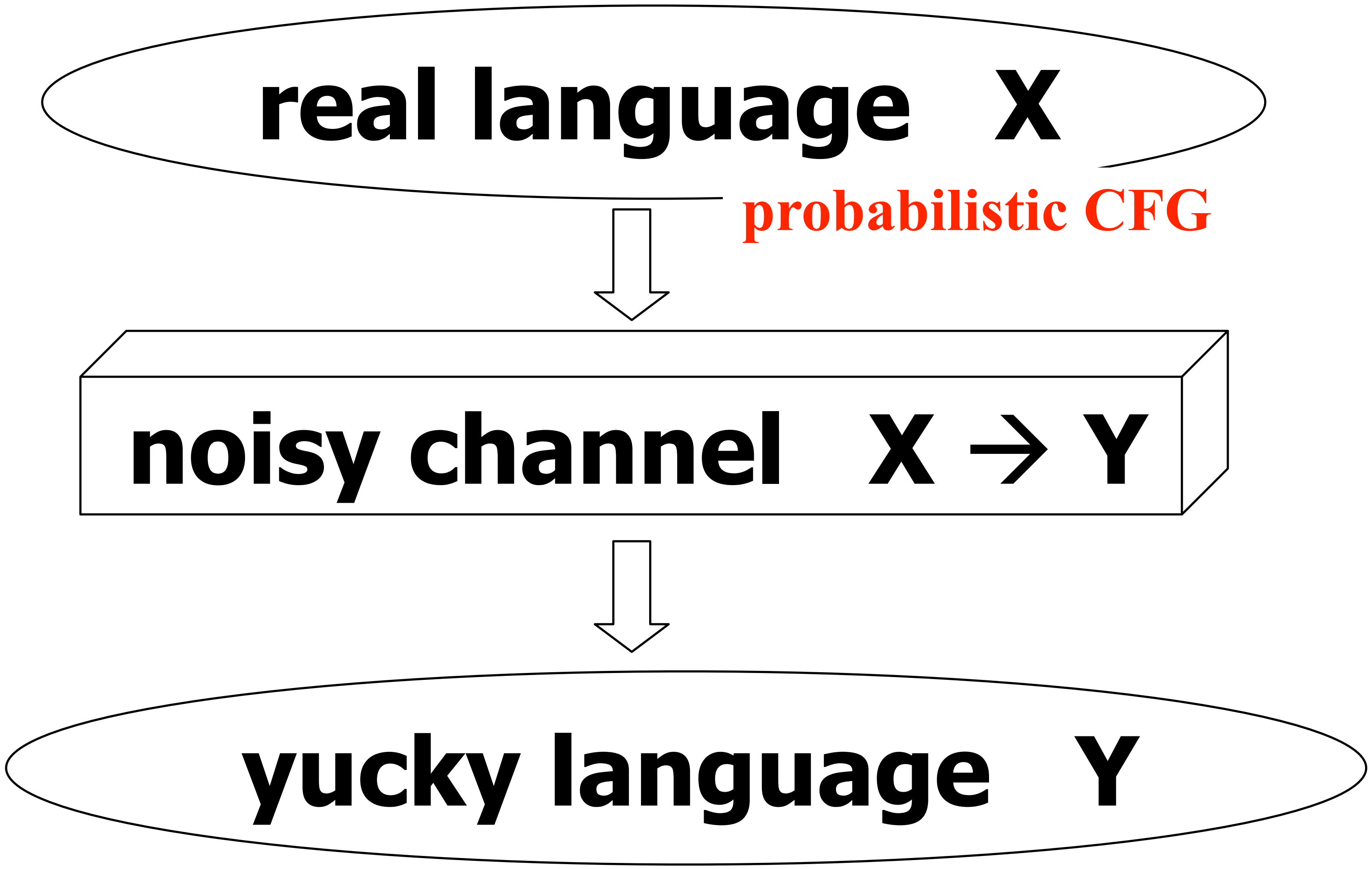
tree

delete everything but terminals

text

want to recover X from Y

Noisy Channel Model



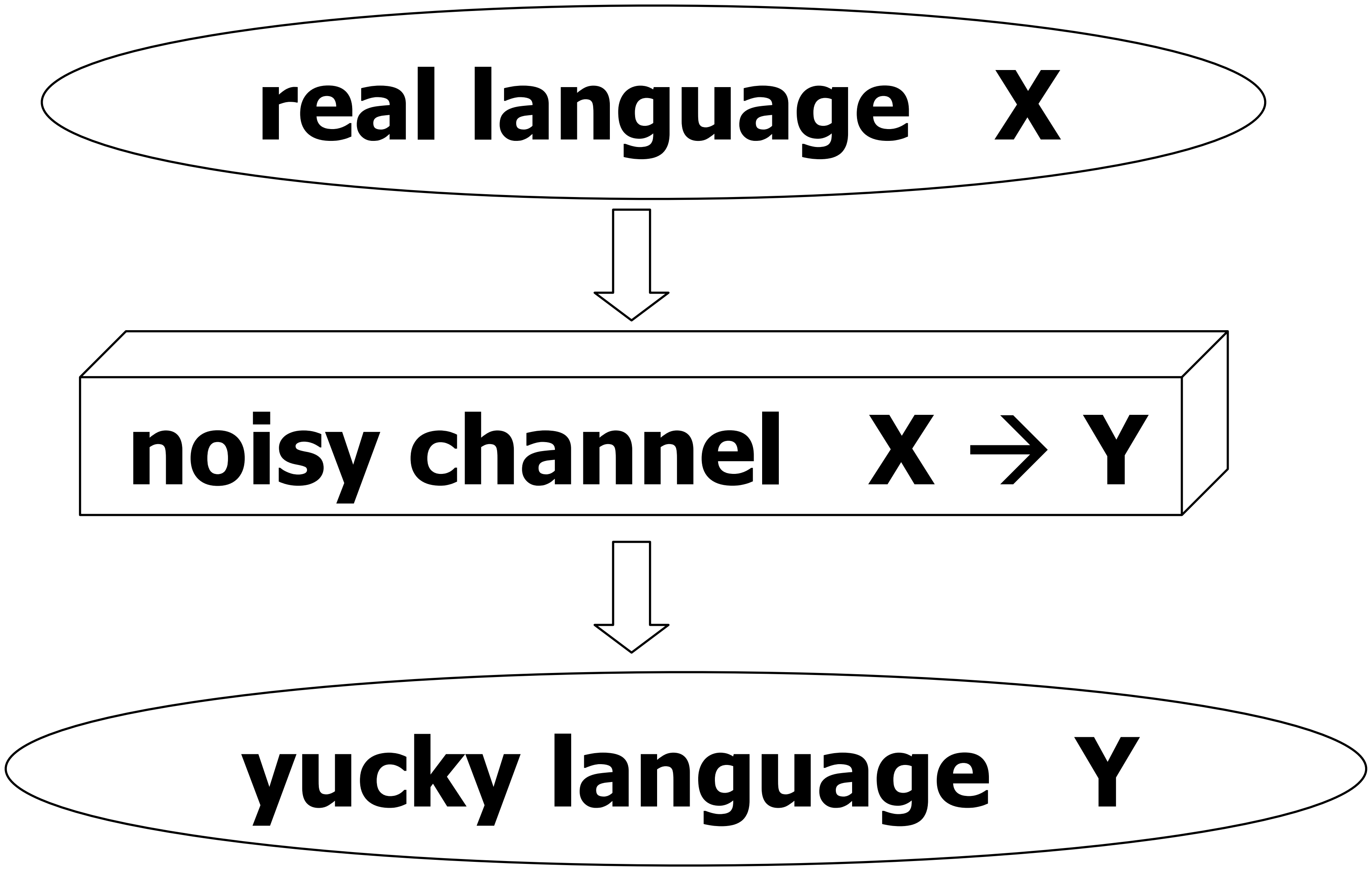
tree

delete everything
but terminals

text

want to recover X from Y

Noisy Channel Model



$$p(X)$$

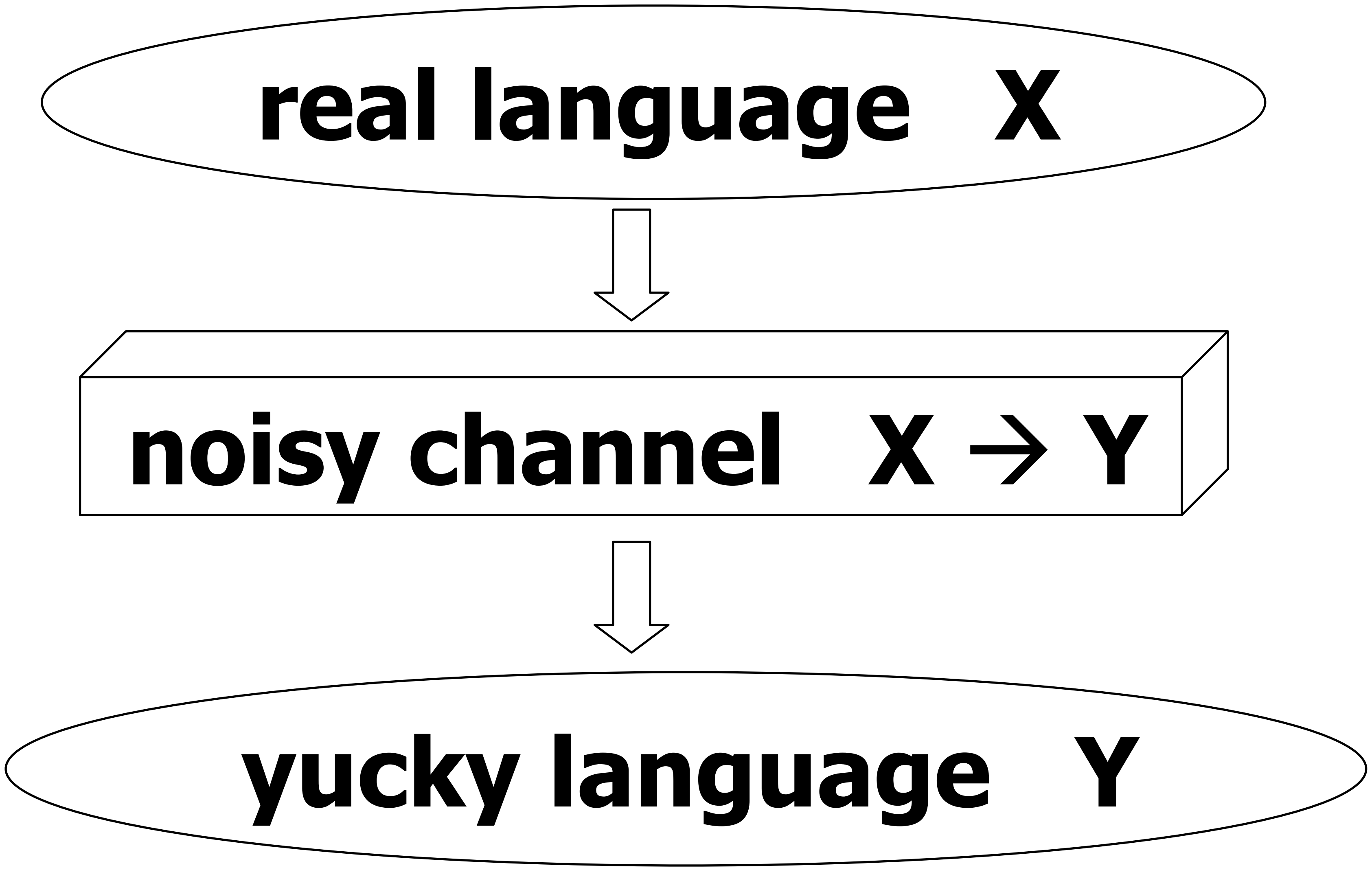
$$*$$

$$p(Y \mid X)$$

$$=$$

$$p(X,Y)$$

Noisy Channel Model



$$p(X)$$

$$*$$

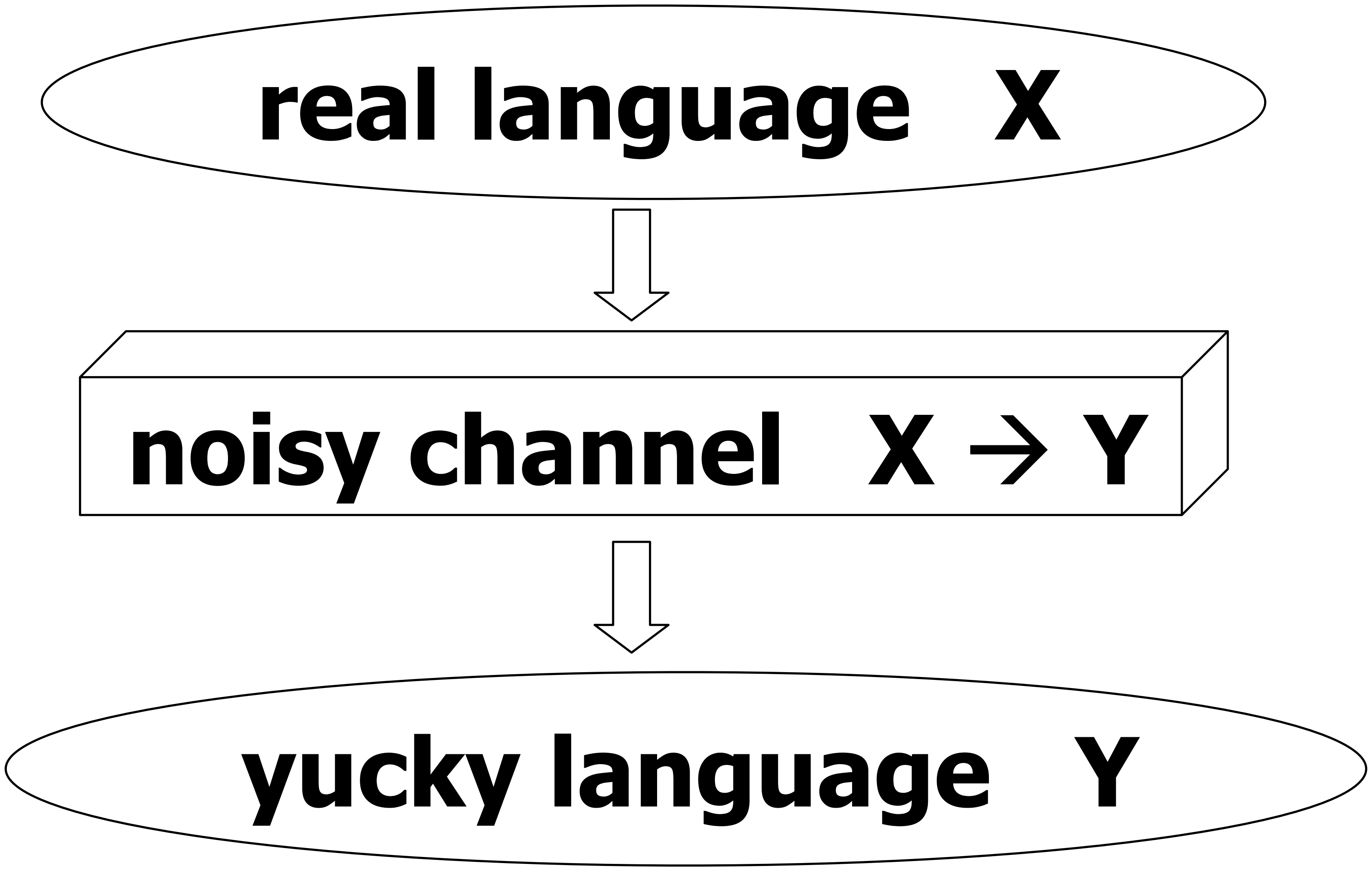
$$p(Y \mid X)$$

$$=$$

$$p(X,Y)$$

want to recover $x \in X$ from $y \in Y$

Noisy Channel Model



$p(X)$

$*$

$p(Y \mid X)$

$=$

$p(X,Y)$

want to recover $x \in X$ from $y \in Y$

choose x that maximizes $p(x \mid y)$ or equivalently $p(x,y)$

Noisy Channel Model

$$p(X)$$

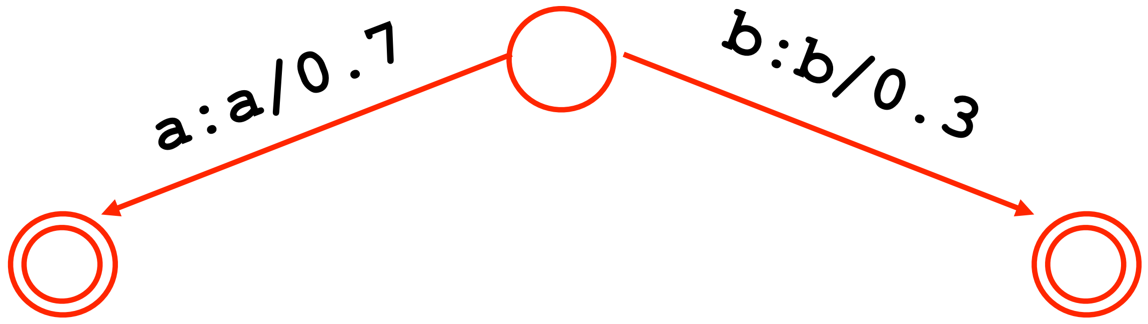
$$*$$

$$p(Y \mid X)$$

$$=$$

$$p(X,Y)$$

Noisy Channel Model



$$p(X)$$

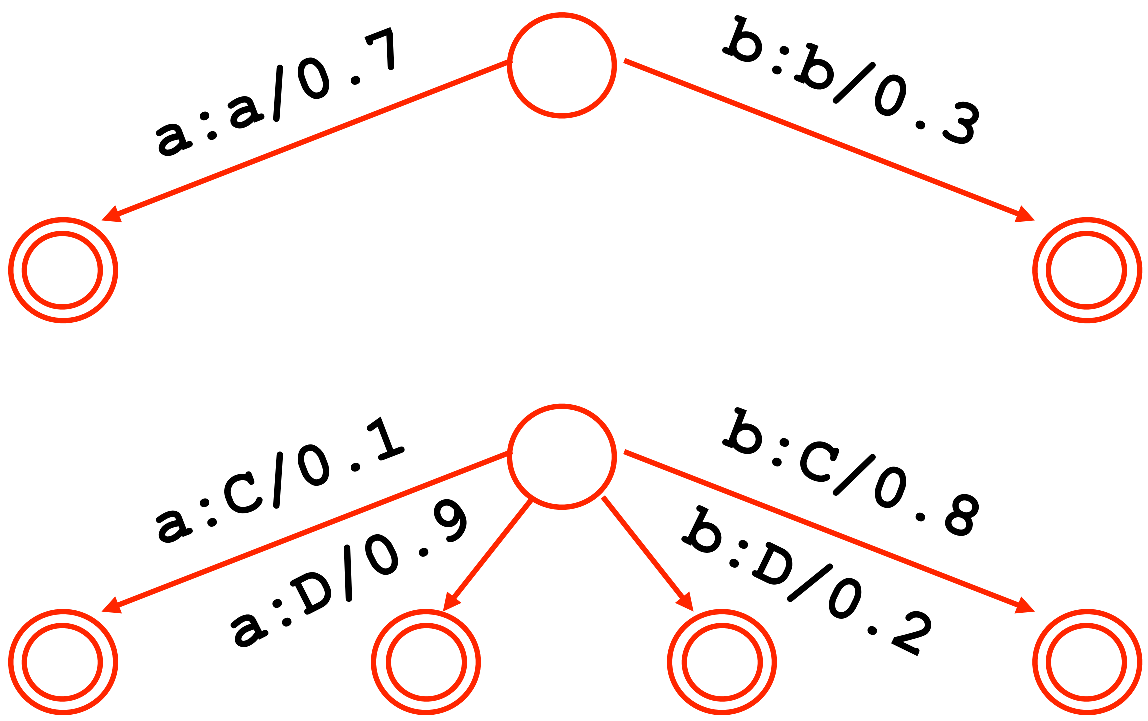
$$*$$

$$p(Y \mid X)$$

$$=$$

$$p(X,Y)$$

Noisy Channel Model



$$p(X)$$

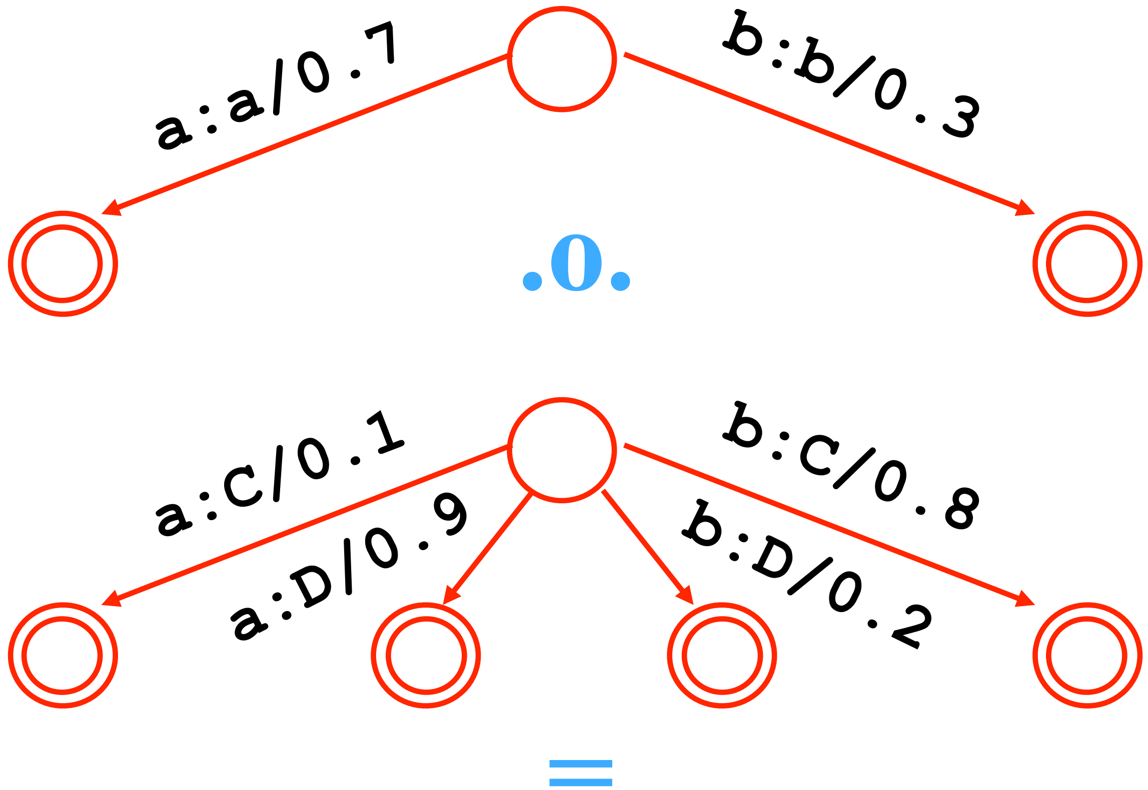
$$*$$

$$p(Y \mid X)$$

$$=$$

$$p(X,Y)$$

Noisy Channel Model



p(X)

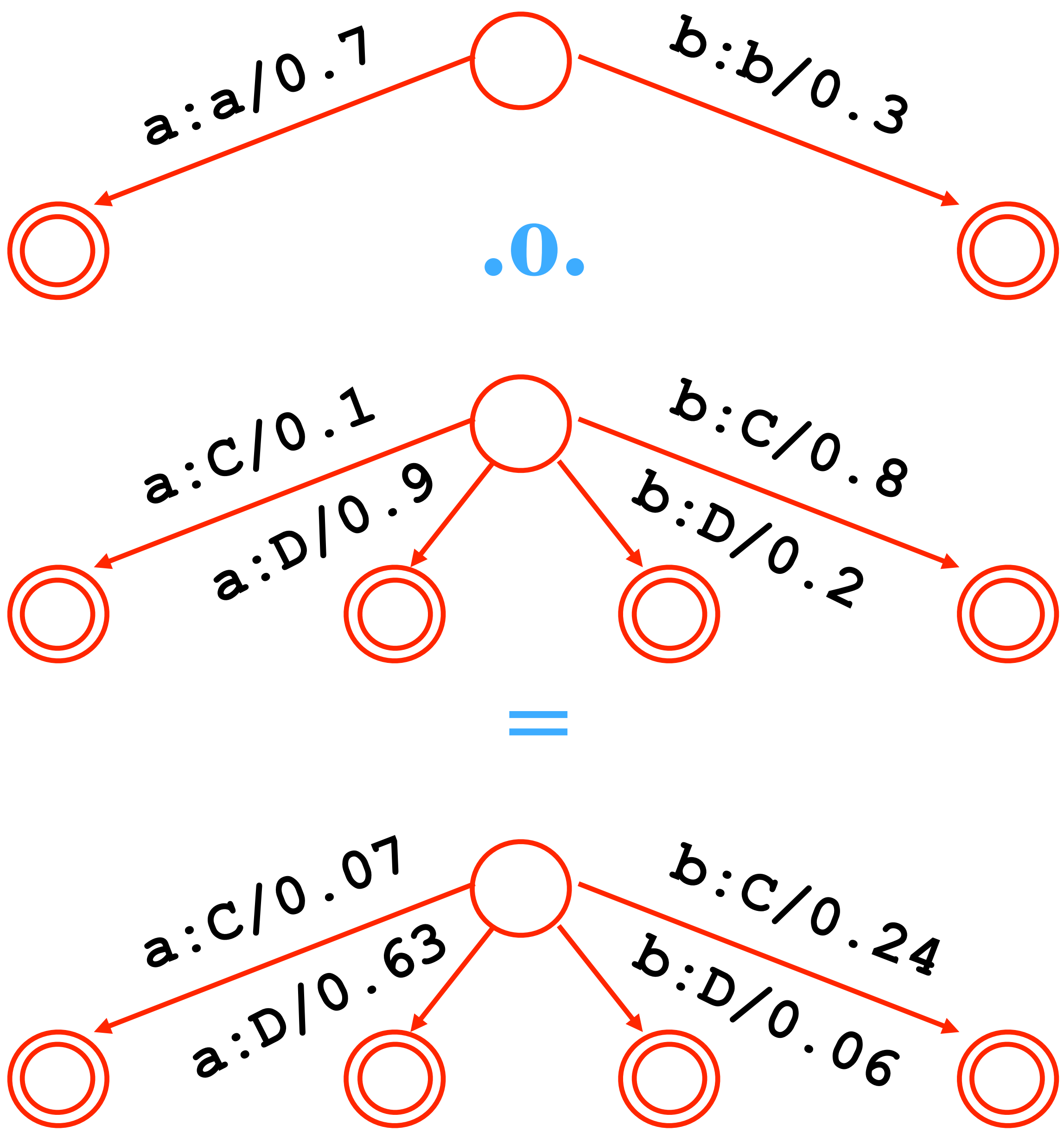
*

p(Y | X)

=

p(X,Y)

Noisy Channel Model



$$p(X)$$

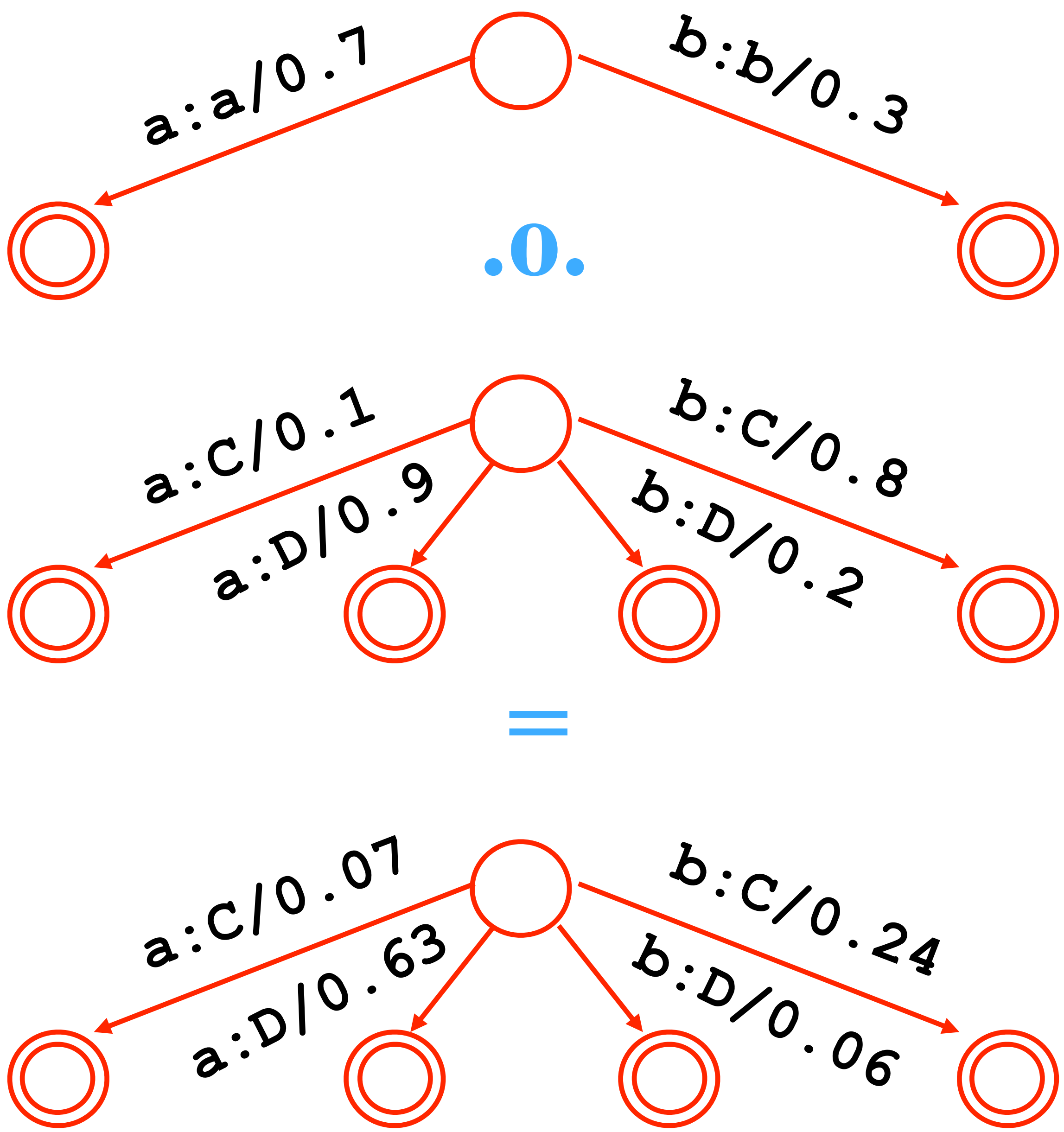
$$*$$

$$p(Y \mid X)$$

$$=$$

$$p(X,Y)$$

Noisy Channel Model



$p(X)$

$*$

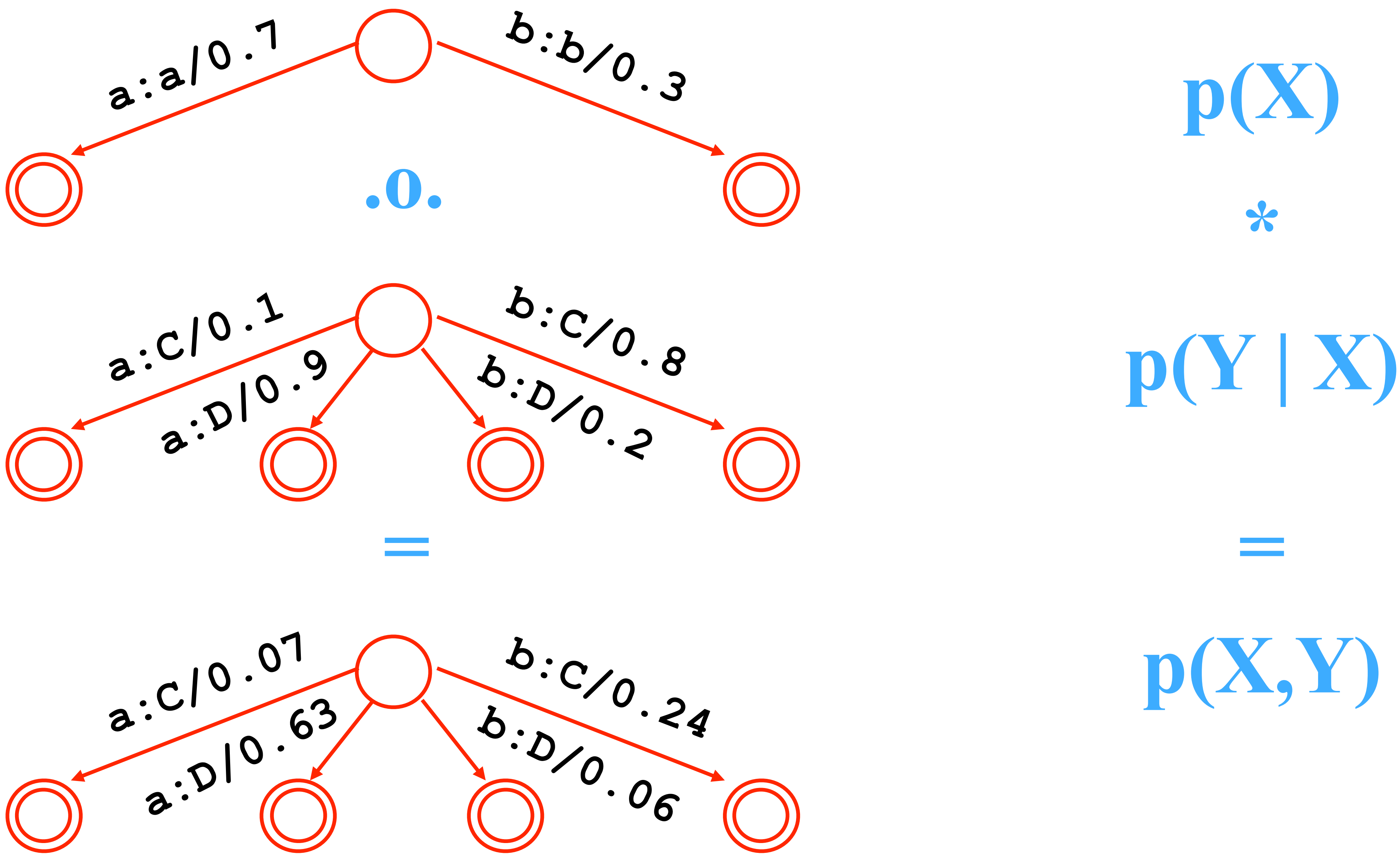
$p(Y \mid X)$

$=$

$p(X,Y)$

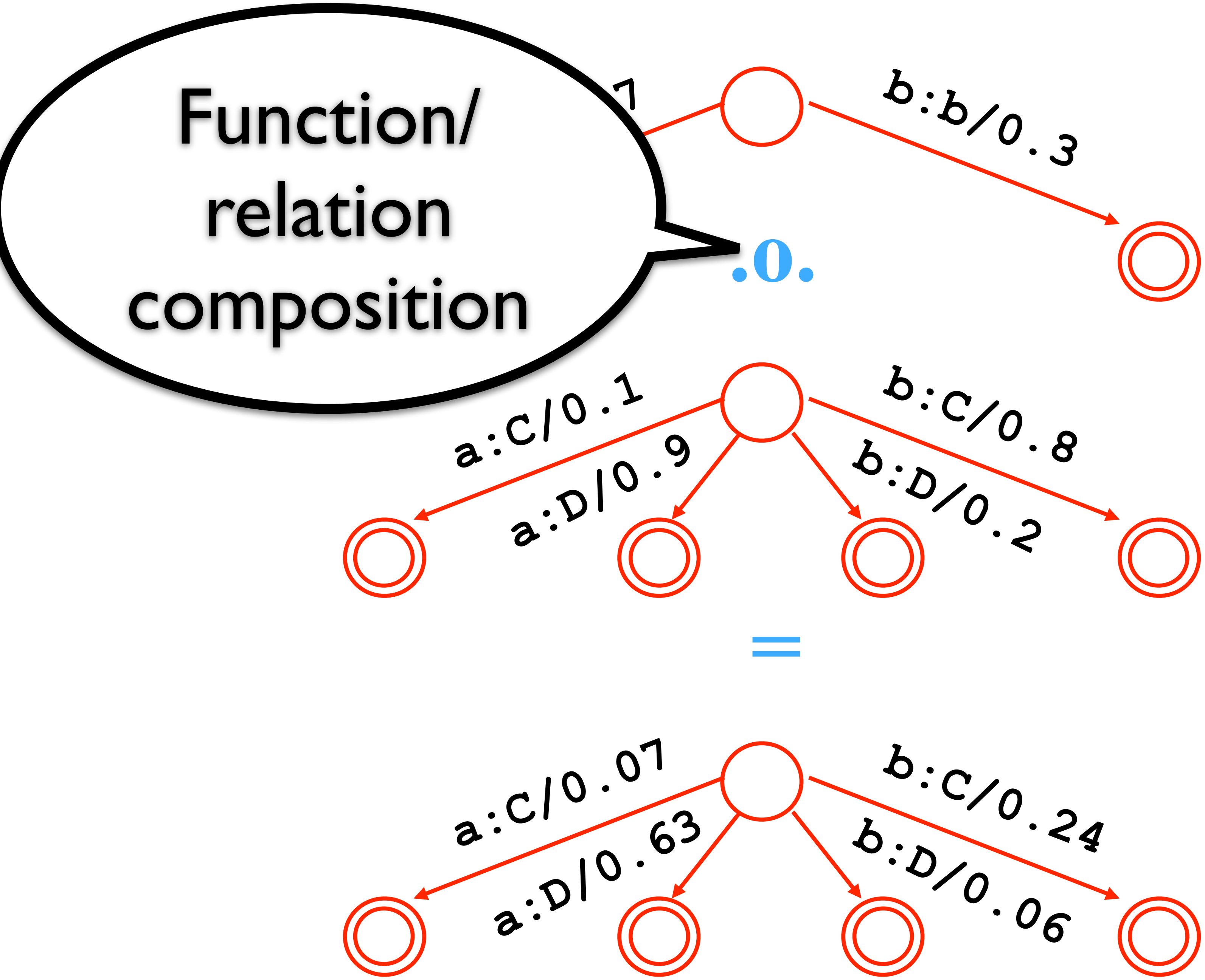
Note p(x,y) sums to 1.

Noisy Channel Model



Note p(x,y) sums to 1.
Suppose y="C"; what is best "x"?

Noisy Channel Model



$p(X)$

$*$

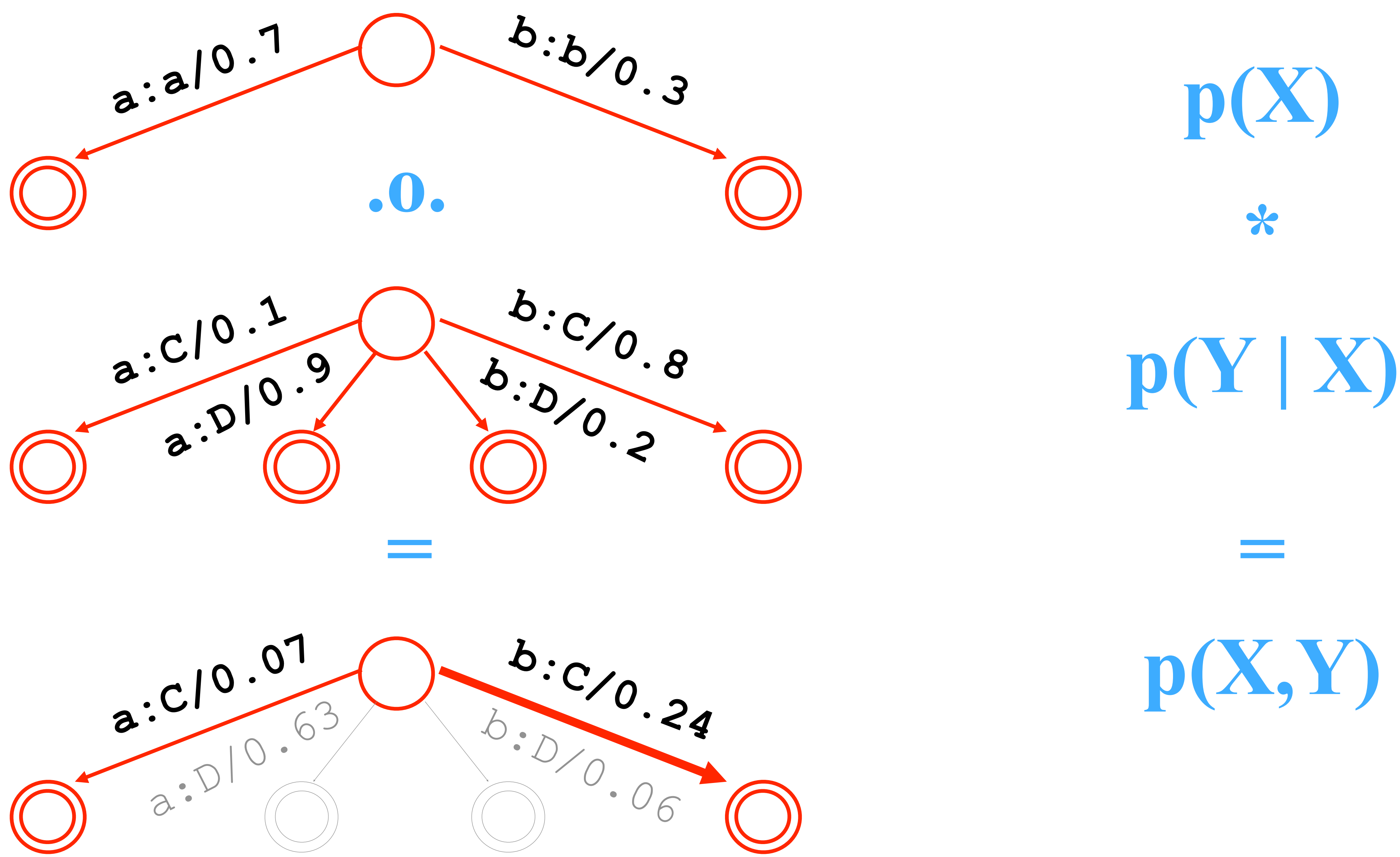
$p(Y \mid X)$

$=$

$p(X,Y)$

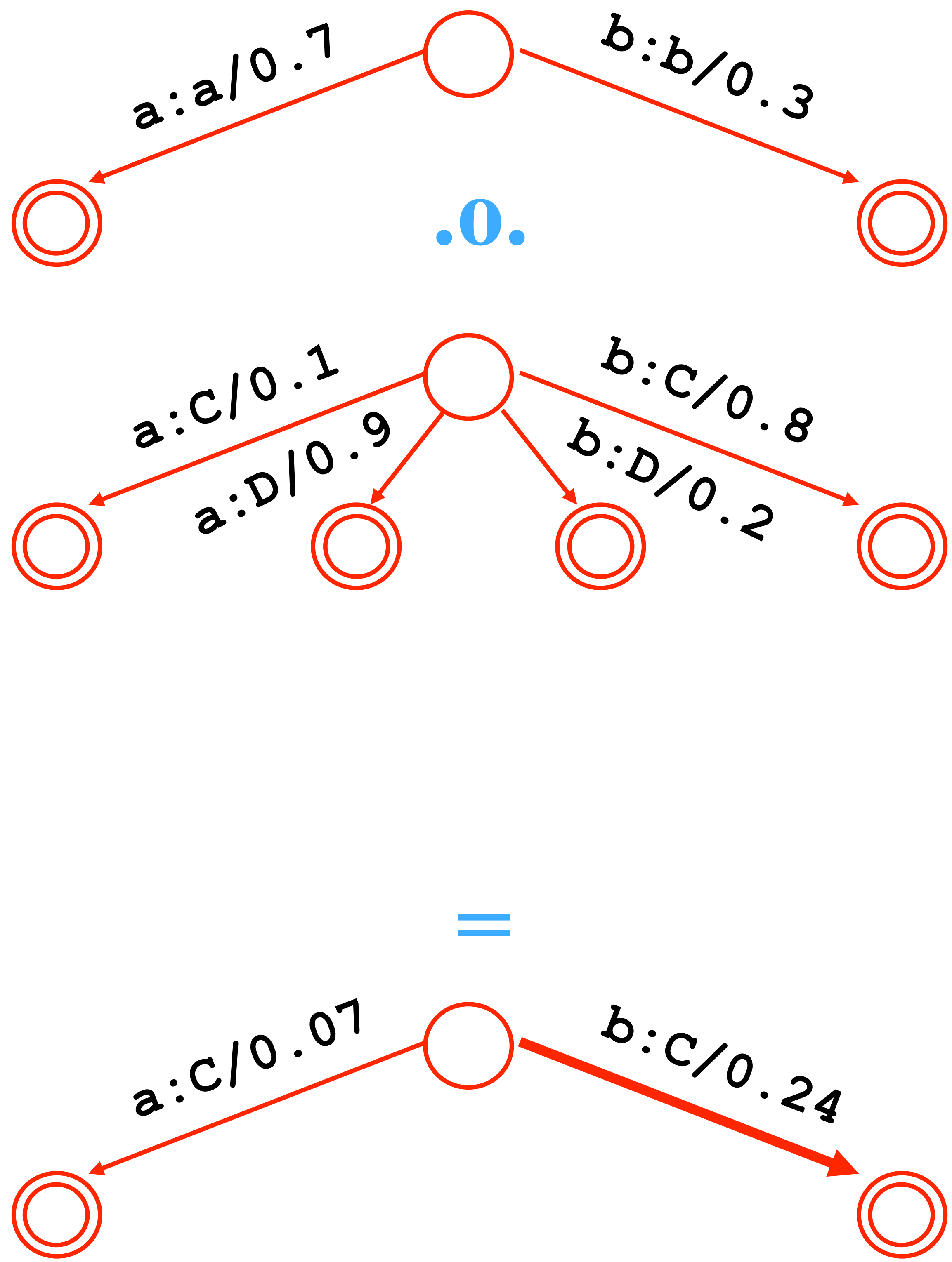
Note p(x,y) sums to 1.
Suppose y="C"; what is best "x"?

Noisy Channel Model



Suppose y="C"; what is best "x"?

Noisy Channel Model



$p(X)$

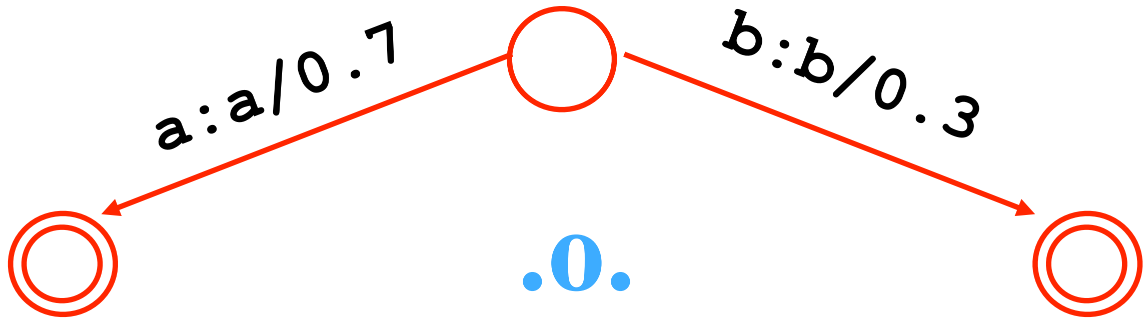
$*$

$p(Y \mid X)$

$=$

$p(X, y)$

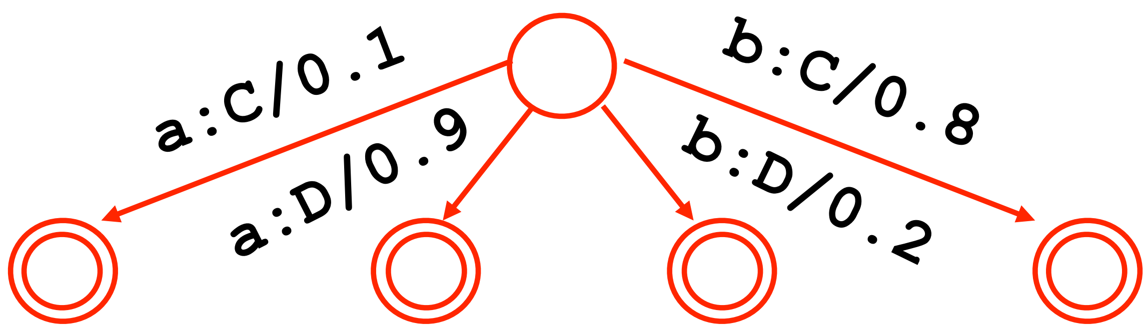
Noisy Channel Model



p(X)

.o.

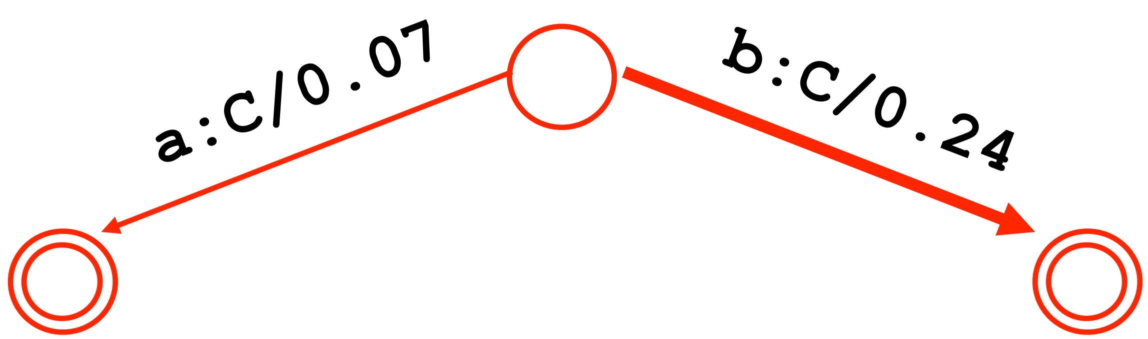
*



p(Y | X)

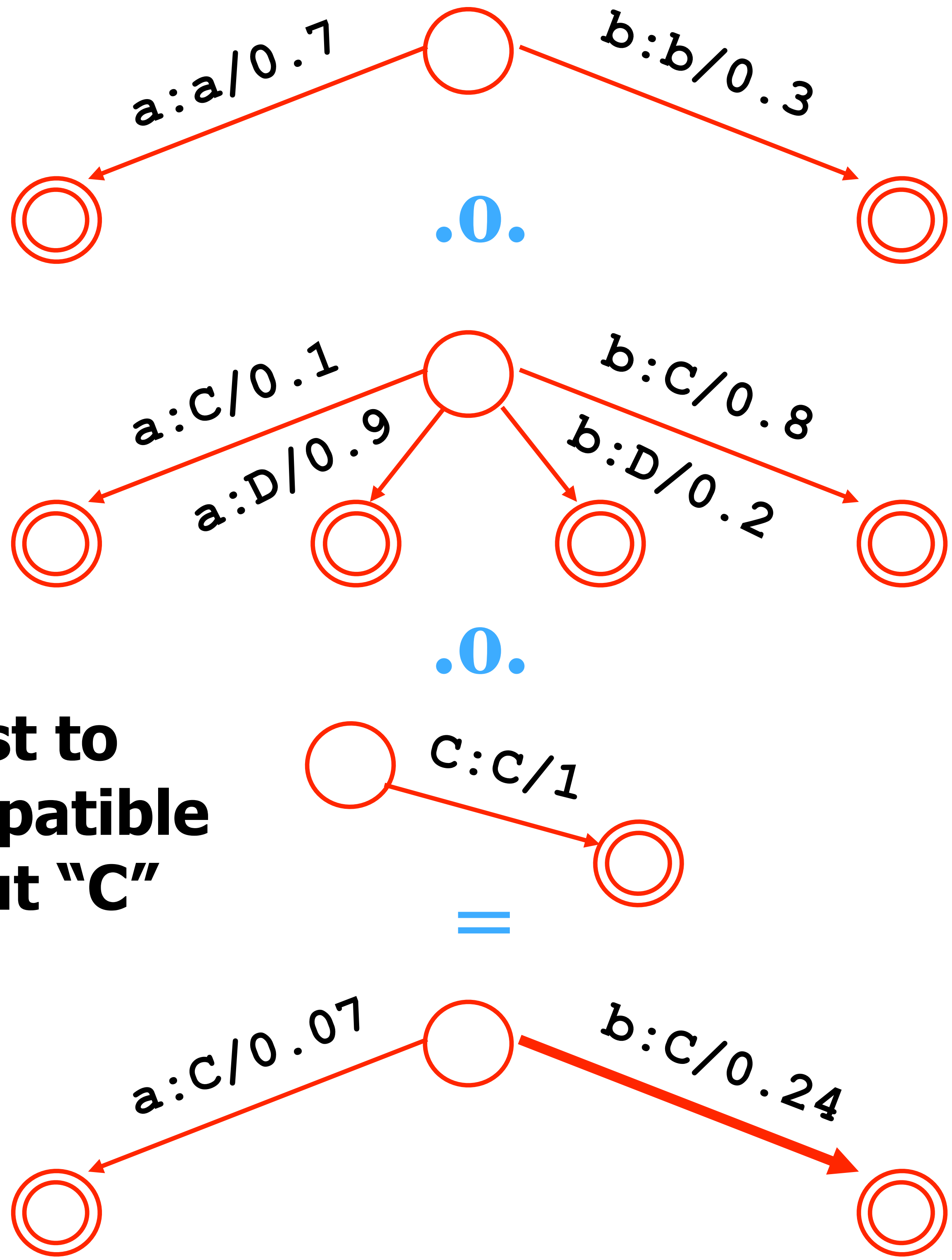
restrict just to paths compatible with output "C"

= =



p(X, y)

Noisy Channel Model



restrict just to paths compatible with output "C"

$p(X)$

$*$

$p(Y \mid X)$

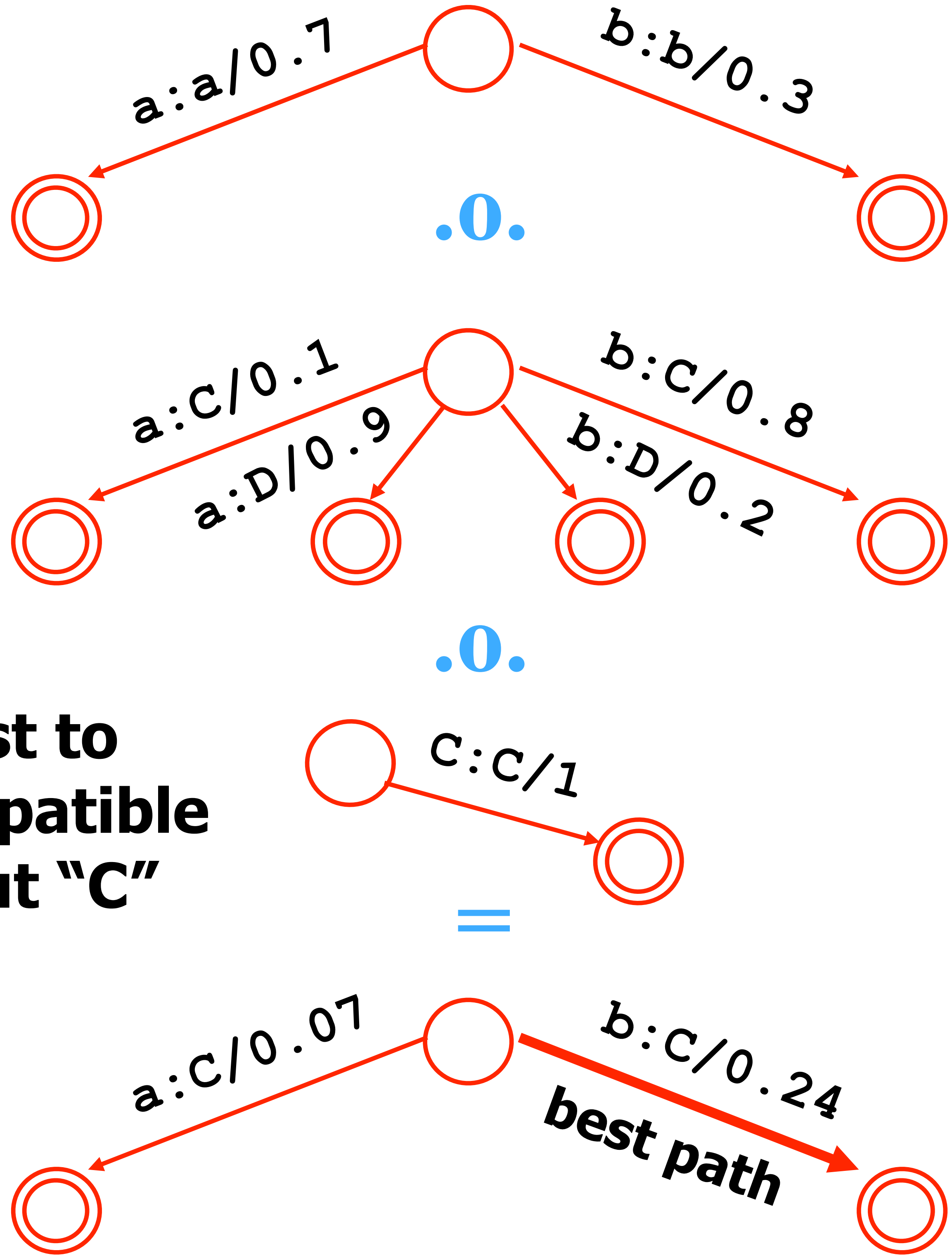
$*$

$(Y=y)?$

$=$

$p(X, y)$

Noisy Channel Model



restrict just to paths compatible with output "C"

$$p(X)$$

$$*$$

$$p(Y \mid X)$$

$$*$$

$$(Y=y)?$$

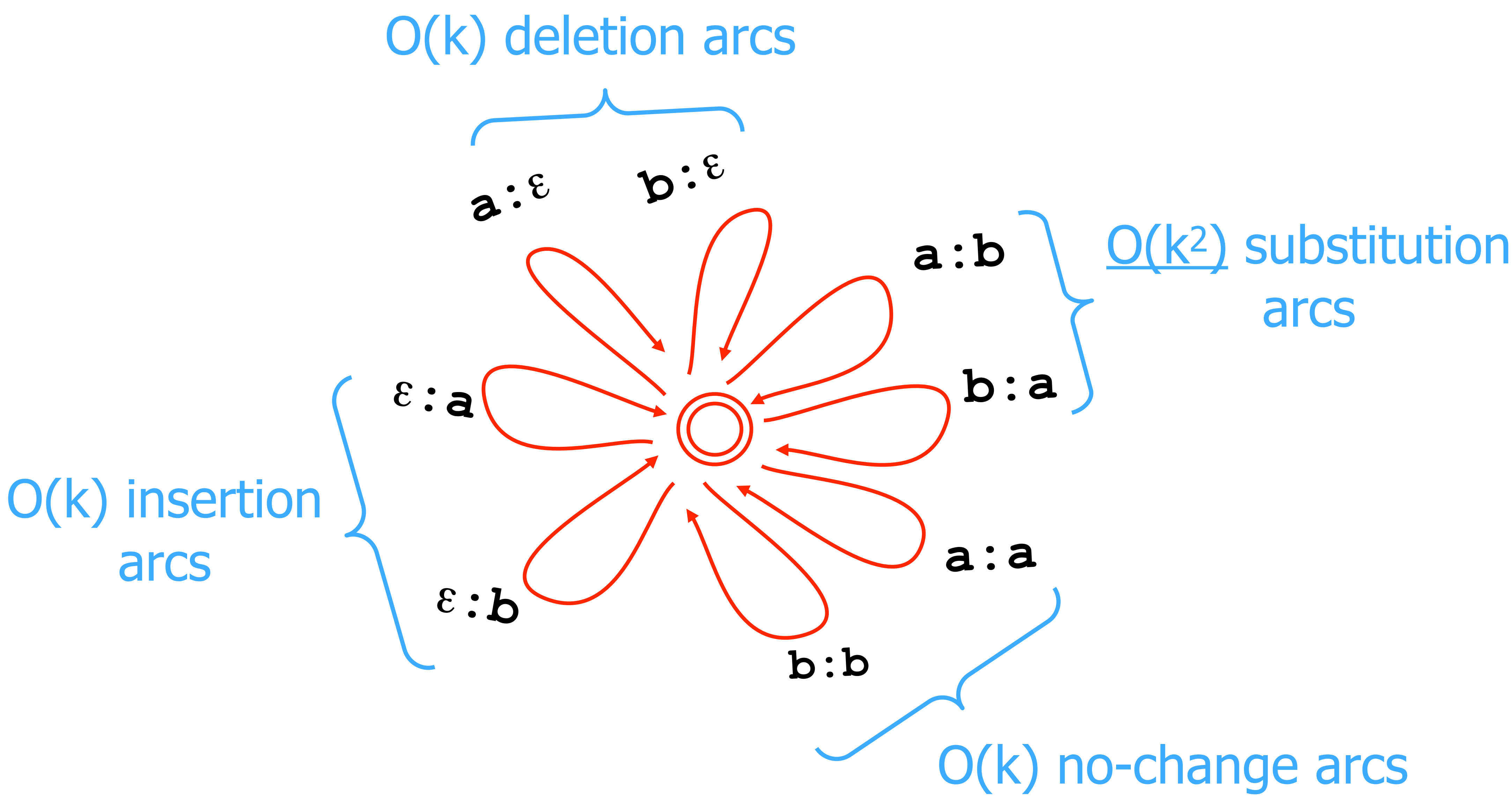
$$=$$

$$p(X, y)$$

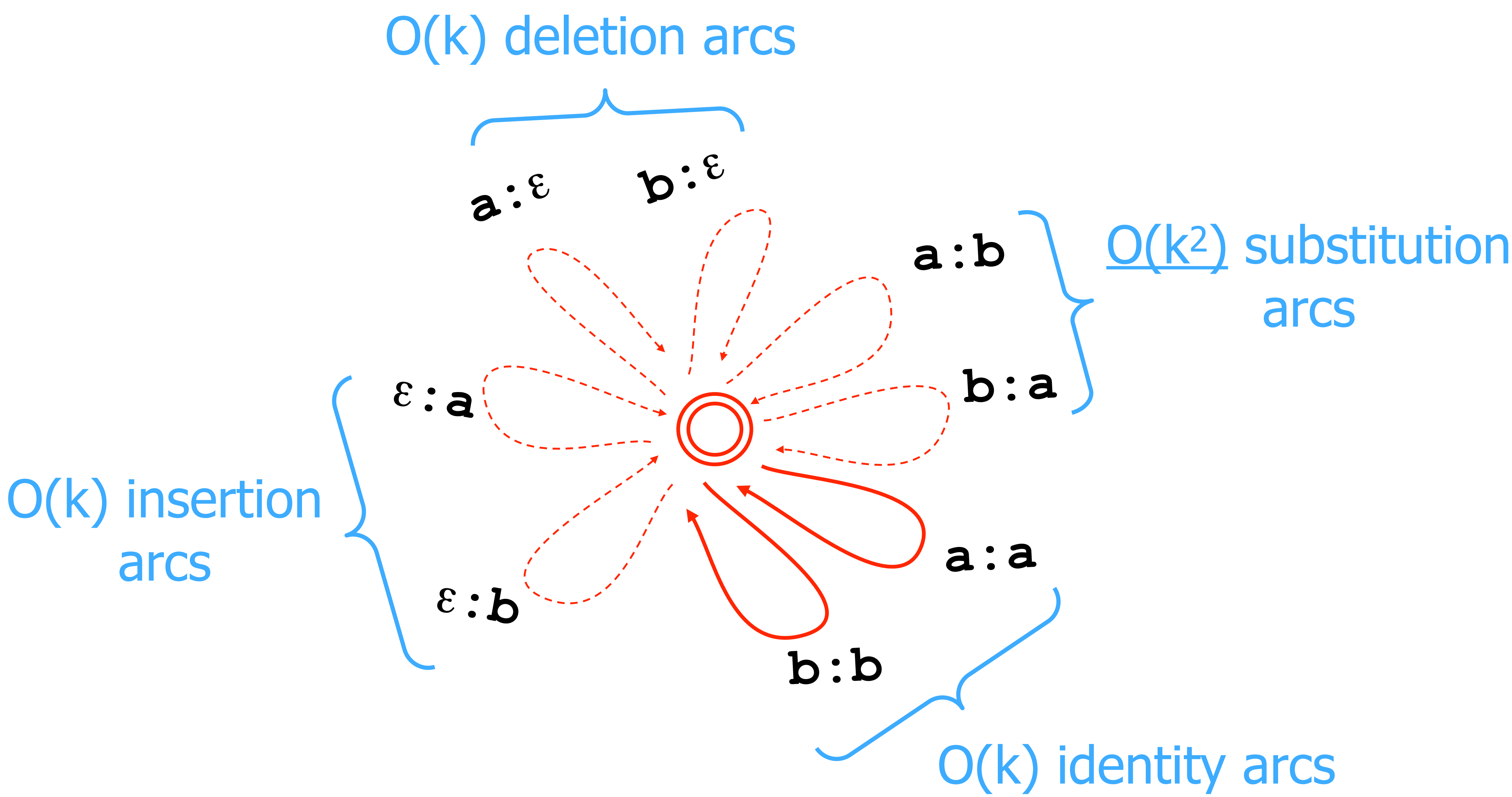
Morpheme Segmentation

- Let Lexicon be a machine that matches all <u>Turkish</u> words
  - Same problem as word segmentation (in, e.g., Chinese)
  - Just at a lower level: morpheme segmentation
  - Turkish word: uygarlaştıramadıklarımızdanmışsınızcasına
    = uygar+laş+tır+ma+dık+ları+mız+dan+mış+sınız+ca+sı+na
    (behaving) as if you are among those whom we could not cause to become civilized
  - Some constraints on morpheme sequence: bigram probs
  - Generative model – concatenate then fix up joints
    - stop + -ing = stopping, fly + -s = flies, vowel harmony
    - Use a cascade of transducers to handle all the fixups
  - But this is just morphology!
  - Can use probabilities here too (but people often don't)

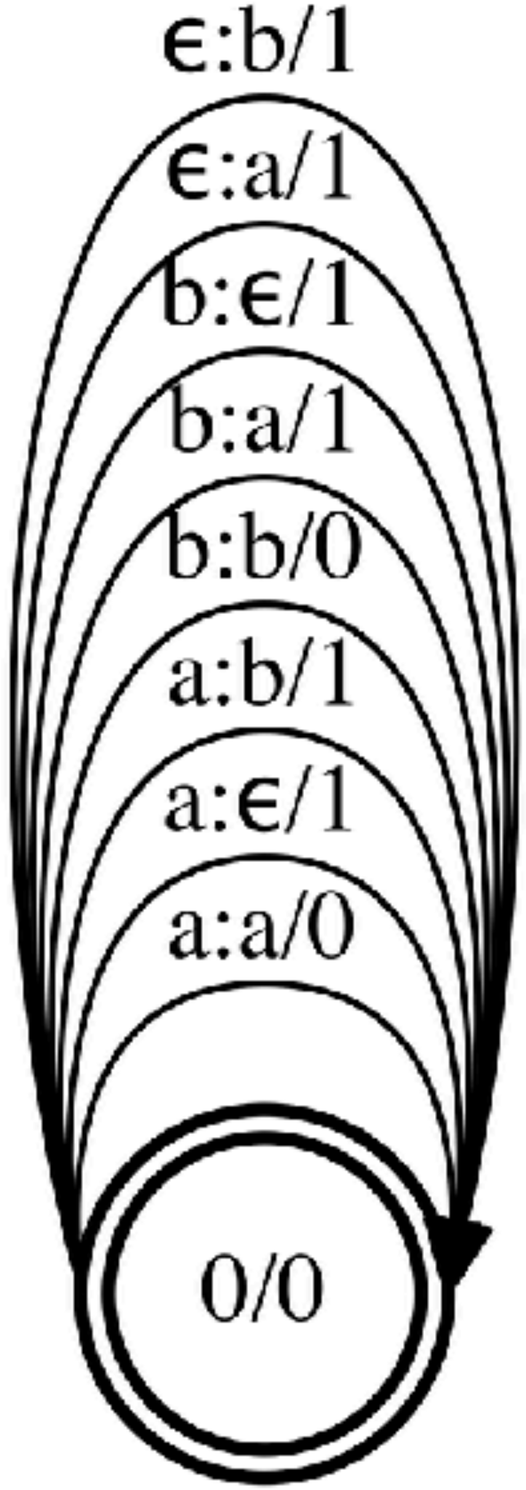
Edit Distance Transducer

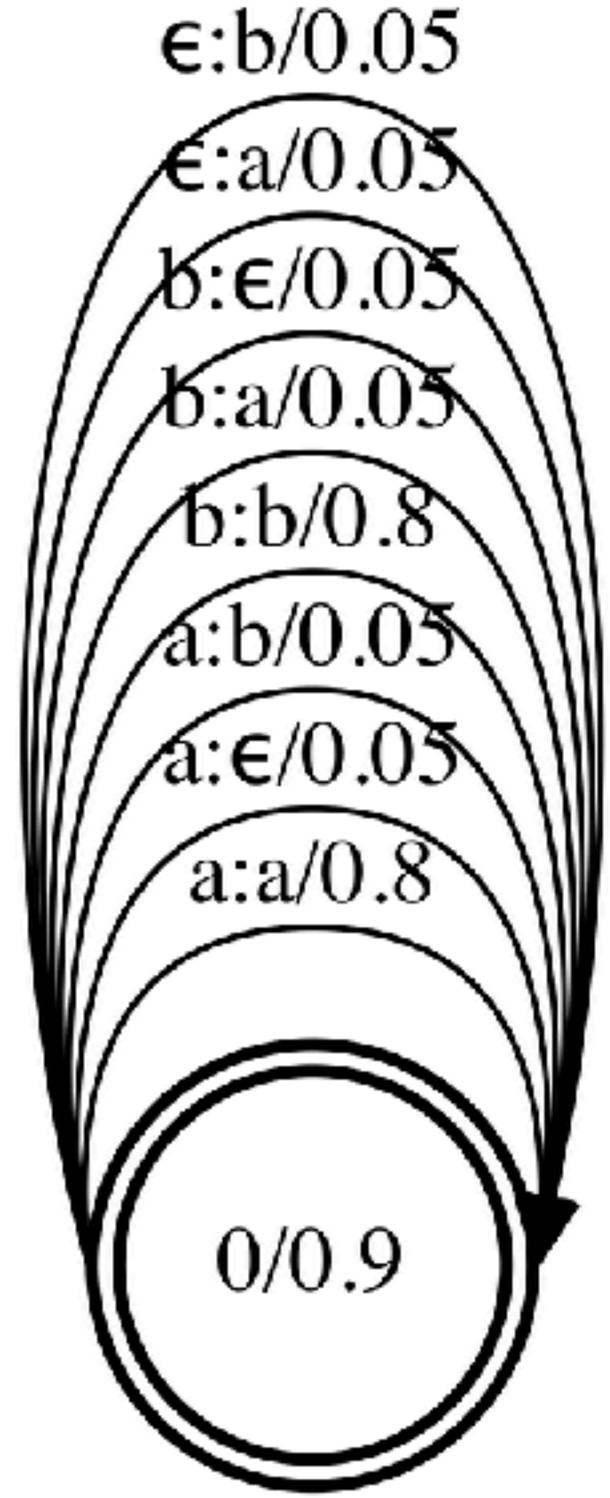


Stochastic Edit Distance Transducer



Likely edits = high-probability arcs





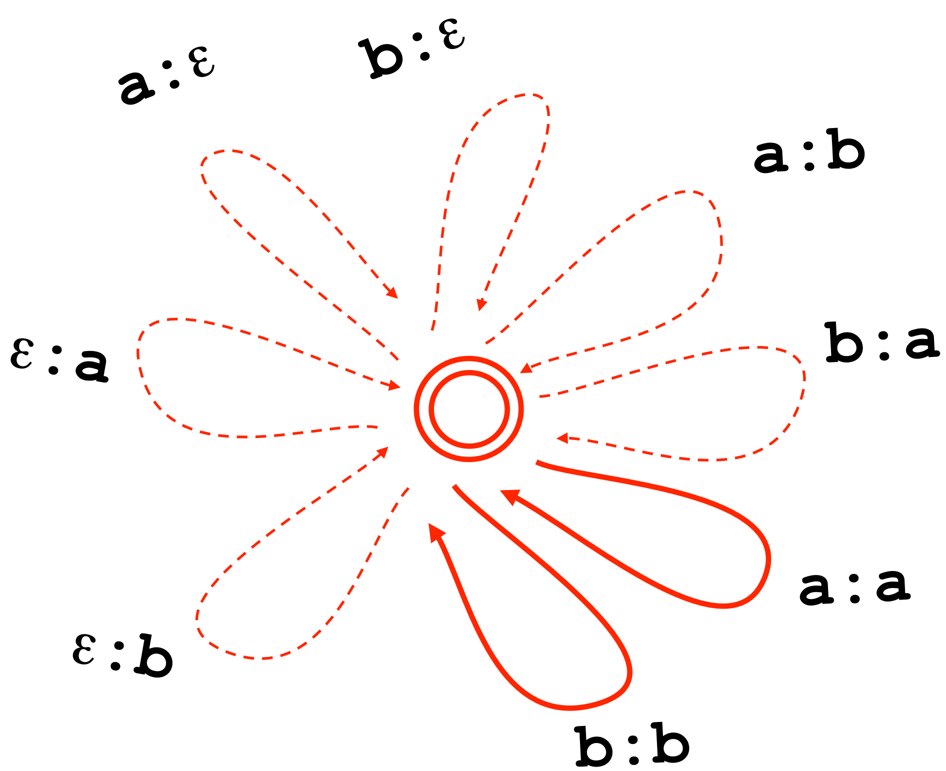
Edit transducer for Levenshtein distance
All edits have additive cost = 1

Edit transducer for **probabilistic** Levenshtein distance
with copy probability = 0.8

Stochastic
Edit Distance Transducer

clara

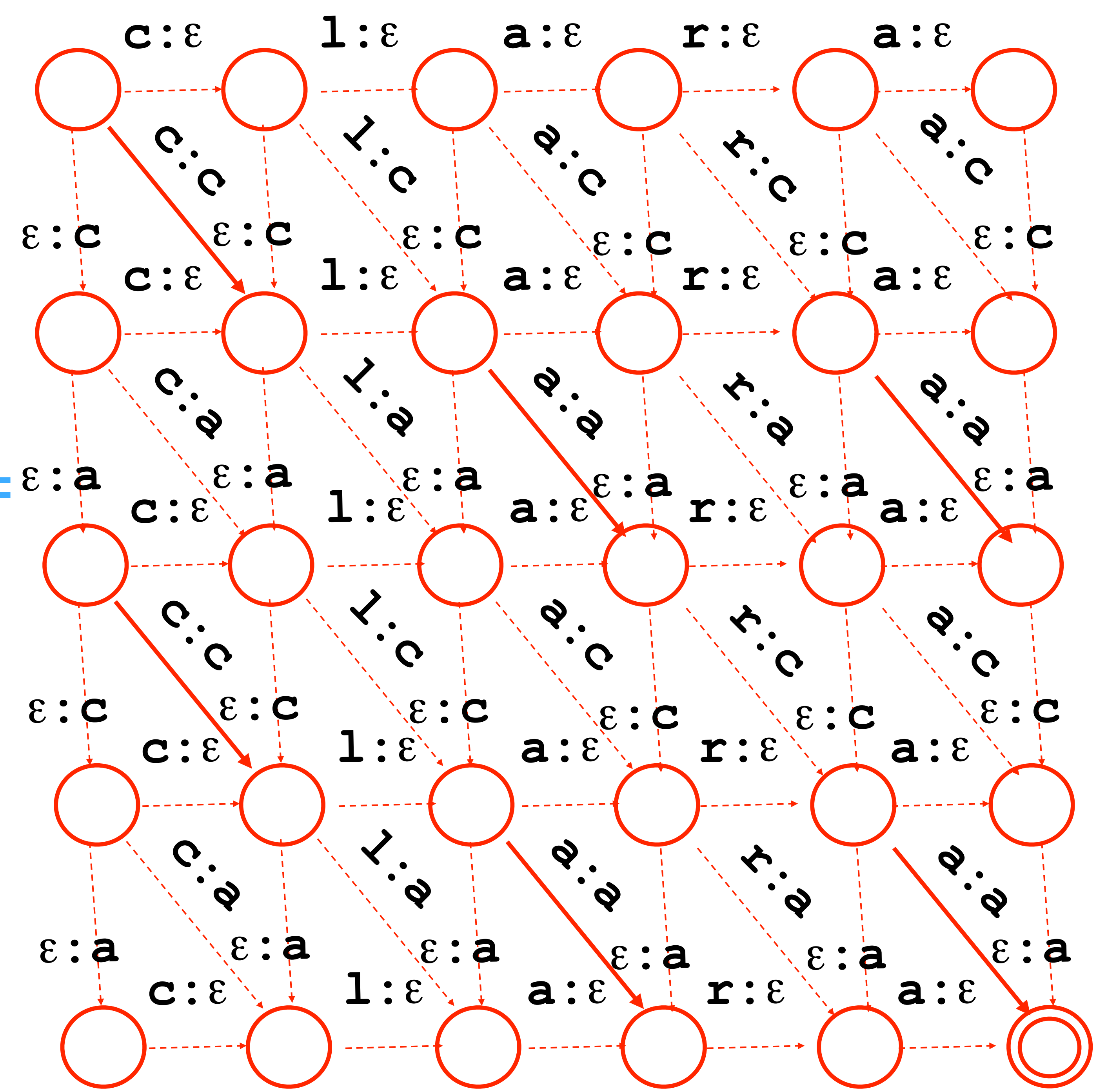
.o.



.o.

caca

=

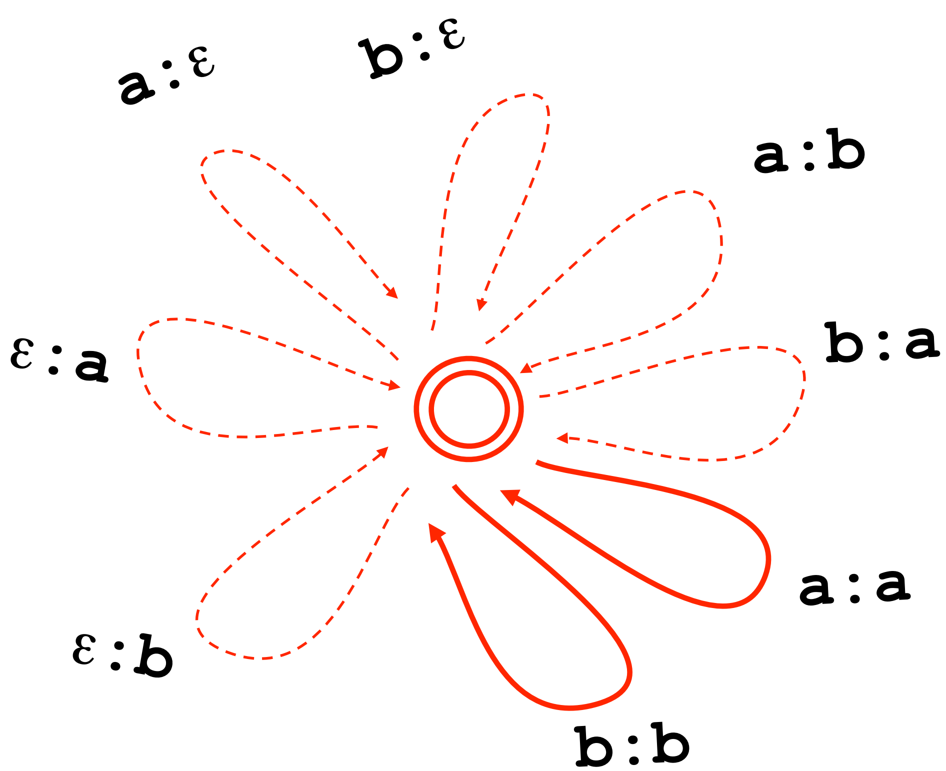


Stochastic

^ Edit Distance Transducer

clara

.o.

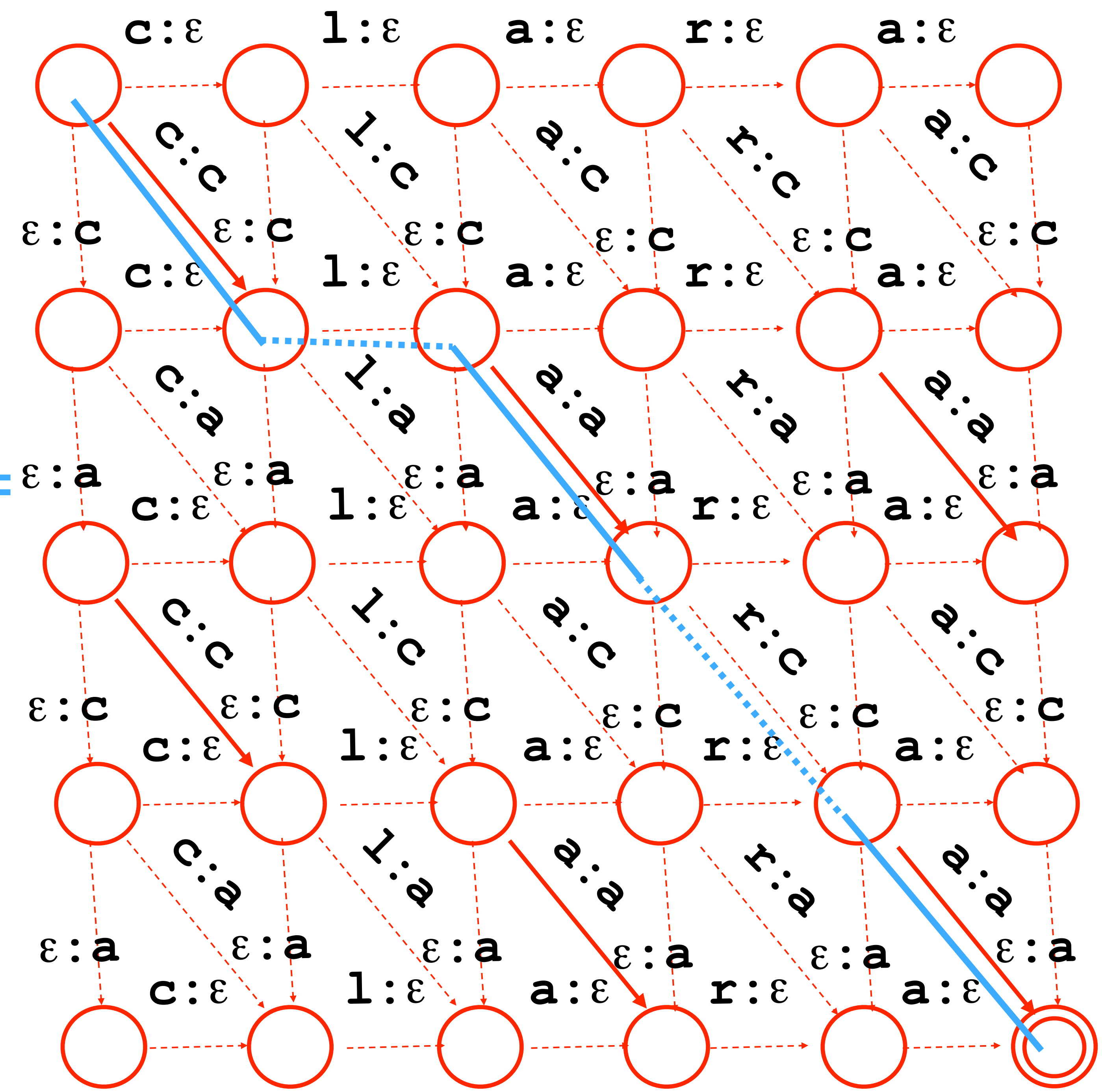


.o.

caca

=

Best path (by Dijkstra's algorithm)



Transliteration (Knight & Graehl, 1998)

Angela Johnson
アンジラ・ジョンソン
(a n jira jyo n so n)

New York Times
ニューヨーク・タイムズ
(nyu u yo o ku ta i mu zu)

ice cream
アイスクリーム
(a i su ku ri i mu)

Omaha Beach
オマハビーチ
(omahabiitchi)

pro soccer
プロサッカー
(purosakkaa)

Tonya Harding
トーニャ・ハーディング
(toonya haadingu)

ramp
ランプ
(ranpu)

lamp
ランプ
(ranpu)

casual fashion
カジュアルヒァッション
(kajyuaruhasshyon)

team leader
チームリーダー
(chiimuriidaa)

1. $P(w)$ — generates written English word sequences.
2. $P(e|w)$ — pronounces English word sequences.
3. $P(j|e)$ — converts English sounds into Japanese sounds.
4. $P(k|j)$ — converts Japanese sounds to katakana writing.
5. $P(o|k)$ — introduces misspellings caused by optical character recognition (OCR).

Sequence Labeling Applications

Parts of Speech

From the earliest linguistic traditions (Yaska and Panini 5th C. BCE, Aristotle 4th C. BCE), the idea that words can be classified into grammatical categories

- part of speech, word classes, POS, POS tags

8 parts of speech attributed to Dionysius Thrax of Alexandria (c. 1st C. BCE):

- noun, verb, pronoun, preposition, adverb, conjunction, participle, article
- These categories are relevant for NLP today.

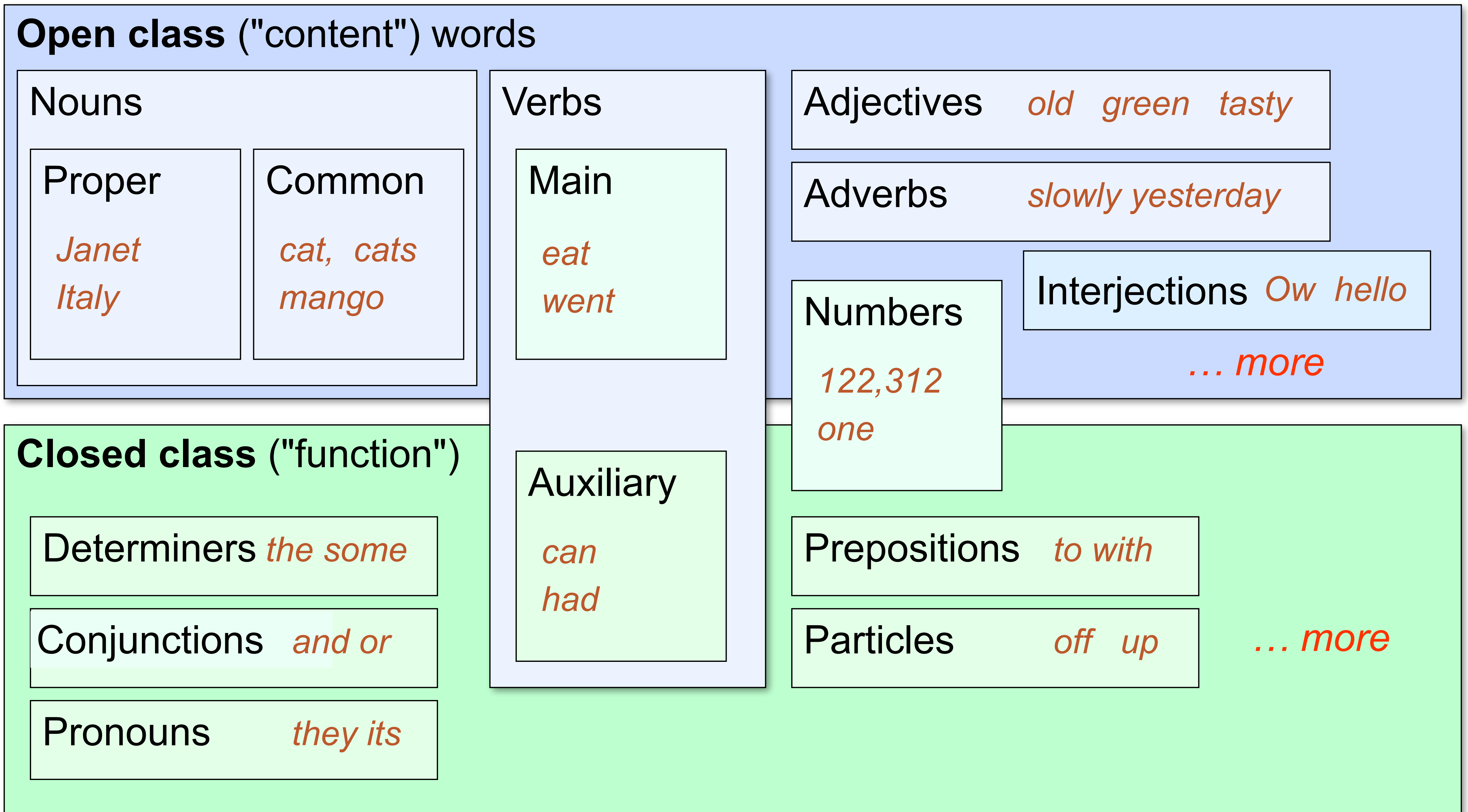
Two classes of words: Open vs. Closed

Closed class words

- Relatively fixed membership
- Usually **function** words: short, frequent words with grammatical function
  - determiners: ***a, an, the***
  - pronouns: ***she, he, I***
  - prepositions: ***on, under, over, near, by, …***
  - ***Very slow*** admission of new closed-class words, e.g. *regarding*

Open class words

- Usually **content** words: Nouns, Verbs, Adjectives, Adverbs
  - Plus interjections: **oh, ouch, uh-huh, yes, hello**
- New nouns and verbs like *iPhone* or *to fax*


Open class ("content") words
Nouns
Proper
Janet
Italy
Common
cat, cats
mango
Verbs
Main
eat
went
Adjectives old green tasty
Adverbs slowly yesterday
Interjections Ow hello
Numbers
122,312
one
… more
Closed class ("function")
Determiners the some
Conjunctions and or
Pronouns they its
Auxiliary
can
had
Prepositions to with
Particles off up
… more

Part-of-Speech Tagging

Assigning a part-of-speech to each word in a text.

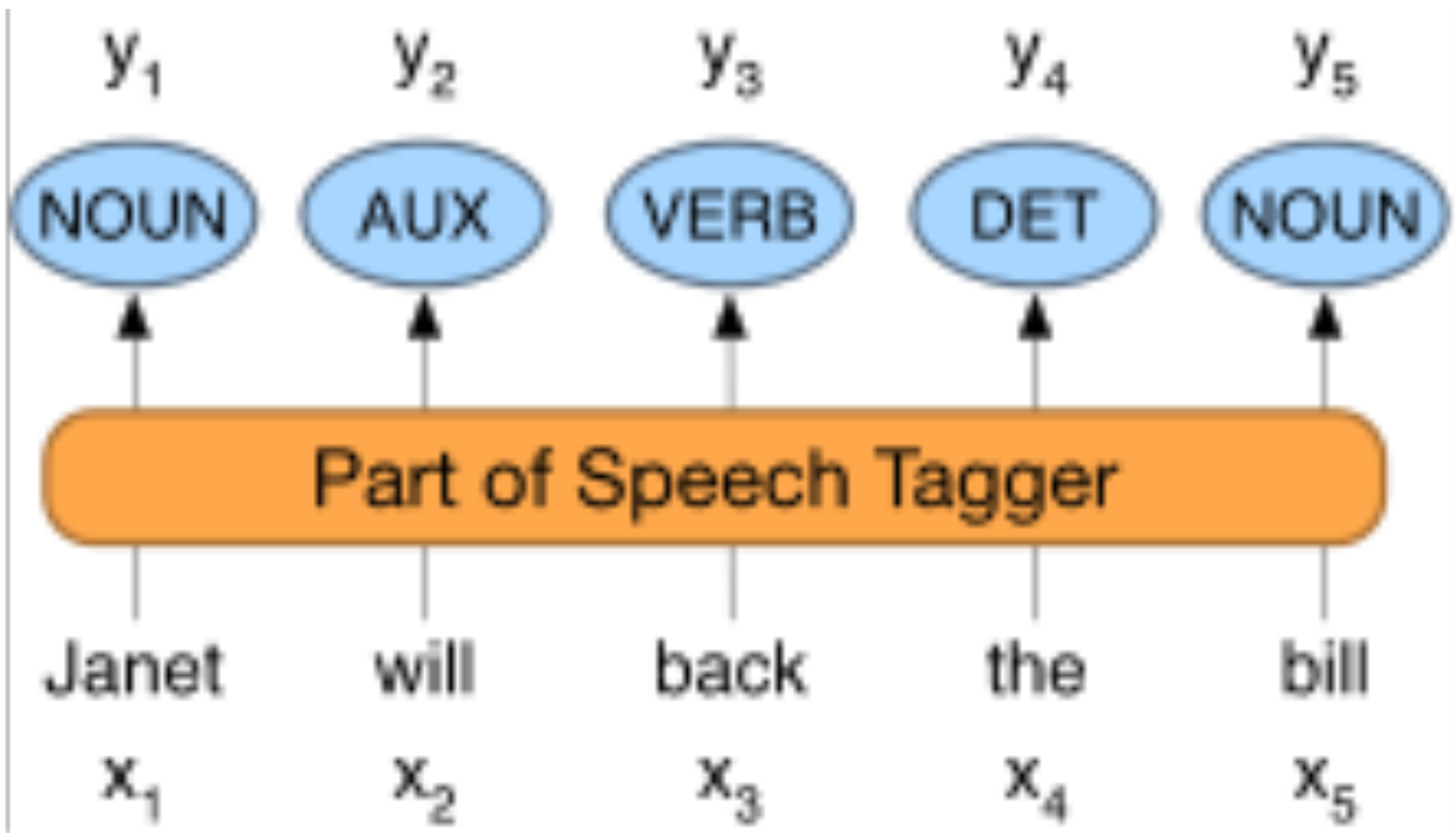
Words often have more than one POS.

book:

- VERB: (***Book** that flight*)
- NOUN: (*Hand me that **book***).

Part-of-Speech Tagging

Map from sequence $x_1,\ldots,x_n$ of words to $y_1,\ldots,y_n$ of POS tags



"Universal Dependencies" Tagset

Nivre et al. 20

	Tag	Description	Example
Open Class	**ADJ**	Adjective: noun modifiers describing properties	*red, young, awesome*
	ADV	Adverb: verb modifiers of time, place, manner	*very, slowly, home, yesterday*
	NOUN	words for persons, places, things, etc.	*algorithm, cat, mango, beauty*
	VERB	words for actions and processes	*draw, provide, go*
	PROPN	Proper noun: name of a person, organization, place, etc..	*Regina, IBM, Colorado*
	INTJ	Interjection: exclamation, greeting, yes/no response, etc.	*oh, um, yes, hello*
Closed Class Words	**ADP**	Adposition (Preposition/Postposition): marks a noun's spacial, temporal, or other relation	*in, on, by under*
	AUX	Auxiliary: helping verb marking tense, aspect, mood, etc.,	*can, may, should, are*
	CCONJ	Coordinating Conjunction: joins two phrases/clauses	*and, or, but*
	DET	Determiner: marks noun phrase properties	*a, an, the, this*
	NUM	Numeral	*one, two, first, second*
	PART	Particle: a preposition-like form used together with a verb	*up, down, on, off, in, out, at, by*
	PRON	Pronoun: a shorthand for referring to an entity or event	*she, who, I, others*
	SCONJ	Subordinating Conjunction: joins a main clause with a subordinate clause such as a sentential complement	*that, which*
Other	**PUNCT**	Punctuation	; , ()
	SYM	Symbols like $ or emoji	$, %
	X	Other	asdf, qwfg

Sample "Tagged" English sentences

There/PRO were/VERB 70/NUM children/NOUN there/ADV ./PUNC

Preliminary/ADJ findings/NOUN were/ AUX reported/VERB in/ADP today/NOUN ’s/PART New/PROPN England/PROPN Journal/PROPN of/ADP Medicine/PROPN

Why Part of Speech Tagging?

Why Part of Speech Tagging?

- Can be useful for other NLP tasks
  - Parsing: POS tagging can improve syntactic parsing
  - MT: reordering of adjectives and nouns (say from Spanish to English)
  - Sentiment or affective tasks: may want to distinguish adjectives or other POS
  - Text-to-speech (how do we pronounce “lead” or "object"?)

Why Part of Speech Tagging?

- Can be useful for other NLP tasks
  - Parsing: POS tagging can improve syntactic parsing
  - MT: reordering of adjectives and nouns (say from Spanish to English)
  - Sentiment or affective tasks: may want to distinguish adjectives or other POS
  - Text-to-speech (how do we pronounce “lead” or "object"?)
- Or linguistic or language-analytic computational tasks
  - Need to control for POS when studying linguistic change like creation of new words, or meaning shift
  - Or control for POS in measuring meaning similarity or difference

How difficult is POS tagging in English?

Roughly 15% of word types are ambiguous

- Hence 85% of word types are unambiguous
- *Janet* is always PROPN, *hesitantly* is always ADV

But those 15% tend to be very common.

So ~60% of word tokens are ambiguous

E.g., *back*

earnings growth took a back/ADJ seat
a small building in the back/NOUN
a clear majority of senators back/VERB the bill
enable the country to buy back/PART debt
I was twenty-one back/ADV then

POS tagging performance in English

How many tags are correct? (Tag accuracy)

POS tagging performance in English

How many tags are correct? (Tag accuracy)

- About 97%
  - Hasn't changed in the last 10+ years
  - HMMs, CRFs, BERT perform similarly .
  - Human accuracy about the same

POS tagging performance in English

How many tags are correct? (Tag accuracy)

- About 97%
  - Hasn't changed in the last 10+ years
  - HMMs, CRFs, BERT perform similarly .
  - Human accuracy about the same

But baseline is 92%!

POS tagging performance in English

How many tags are correct? (Tag accuracy)

- About 97%
  - Hasn't changed in the last 10+ years
  - HMMs, CRFs, BERT perform similarly .
  - Human accuracy about the same

But baseline is 92%!

- Baseline is performance of stupidest possible method
  - "Most frequent class baseline" is an important baseline for many tasks
    - Tag every word with its most frequent tag
    - (and tag unknown words as nouns)

POS tagging performance in English

How many tags are correct? (Tag accuracy)

- About 97%
  - Hasn't changed in the last 10+ years
  - HMMs, CRFs, BERT perform similarly .
  - Human accuracy about the same

But baseline is 92%!

- Baseline is performance of stupidest possible method
  - "Most frequent class baseline" is an important baseline for many tasks
    - Tag every word with its most frequent tag
    - (and tag unknown words as nouns)
- Partly easy because
  - Many words are unambiguous

Sources of information for POS tagging

Sources of information for POS tagging

```
Janet will back the bill
```

AUX/NOUN/VERB? NOUN/VERB?

Sources of information for POS tagging

`Janet will back the bill`

AUX/NOUN/VERB? NOUN/VERB?

Prior probabilities of word/tag

- "will" is usually an AUX

Sources of information for POS tagging

```
Janet will back the bill
```
AUX/NOUN/VERB? NOUN/VERB?

Prior probabilities of word/tag

- "will" is usually an AUX

Identity of neighboring words

- "the" means the next word is probably not a verb

Standard algorithms for POS tagging

Supervised Machine Learning Algorithms:

- Hidden Markov Models
- Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)
- Neural sequence models (RNNs or Transformers)
- Large Language Models (like BERT), finetuned

All required a hand-labeled training set, all about equal performance (97% on English)

All make use of information sources we discussed

- Via human created features: HMMs and CRFs
- Via representation learning: Neural LMs

Noisy Channel for Tagging

acceptor: p(tag sequence) $p(X)$

"Markov Model"

.o. $*$

transducer: tags → words $p(Y \mid X)$

"Unigram Replacement"

.o. $*$

acceptor: the observed words $p(Y = y)$?

"straight line"

= =

transducer: scores candidate tag seqs on their joint probability with obs words, i.e. a Hidden Markov model $p(X, y)$

Named Entities

- **Named entity**, in its core usage, means anything that can be referred to with a proper name. Most common 4 tags:
  - PER (Person): “Marie Curie”
  - LOC (Location): “New York City”
  - ORG (Organization): “Stanford University”
  - GPE (Geo-Political Entity): "Boulder, Colorado"
- Often multi-word phrases
- But the term is also extended to things that aren't entities:
  - dates, times, prices

Named Entity tagging

The task of named entity recognition (NER):

- find spans of text that constitute proper names
- tag the type of the entity.

NER output

Citing high fuel prices, [ORG **United Airlines**] said [TIME **Friday**] it has increased fares by [MONEY **$6**] per round trip on flights to some cities also served by lower-cost carriers. [ORG **American Airlines**], a unit of [ORG **AMR Corp.**], immediately matched the move, spokesman [PER **Tim Wagner**] said. [ORG **United**], a unit of [ORG **UAL Corp.**], said the increase took effect [TIME **Thursday**] and applies to most routes where it competes against discount carriers, such as [LOC **Chicago**] to [LOC **Dallas**] and [LOC **Denver**] to [LOC **San Francisco**].

Why NER?

Sentiment analysis: consumer's sentiment toward a particular company or person?

Question Answering: answer questions about an entity?

Information Extraction: Extracting facts about entities from text.

Why NER is hard

1) Segmentation
   - In POS tagging, no segmentation problem since each word gets one tag.
   - In NER we have to find and segment the entities!

2

[$_{\text{PER}}$ Washington] was born into slavery on the farm of James Burroughs.
[$_{\text{ORG}}$ Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [$_{\text{LOC}}$ Washington] for what may well be his last state visit.
In June, [$_{\text{GPE}}$ Washington] passed a primary seatbelt law.

BIO Tagging

How can we turn this structured problem into a sequence problem like POS tagging, with one label per word?

[PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

BIO Tagging

[PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

Words	BIO Label
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

Now we have one tag per token!!!

BIO Tagging

B: token that *begins* a span

I: tokens *inside* a span

O: tokens outside of any span

# of tags (where n is #entity types):

1 O tag,

n B tags,

n I tags

total of *2n+1*

Words	BIO Label
Jane	B-PER
Villanueva	I-PER
of	O
United	B-ORG
Airlines	I-ORG
Holding	I-ORG
discussed	O
the	O
Chicago	B-LOC
route	O
.	O

BIO Tagging variants: IO and BIOES

[PER Jane Villanueva] of [ORG United] , a unit of [ORG United Airlines Holding] , said the fare applies to the [LOC Chicago] route.

Words	IO Label	BIO Label	BIOES Label
Jane	I-PER	B-PER	B-PER
Villanueva	I-PER	I-PER	E-PER
of	O	O	O
United	I-ORG	B-ORG	B-ORG
Airlines	I-ORG	I-ORG	I-ORG
Holding	I-ORG	I-ORG	E-ORG
discussed	O	O	O
the	O	O	O
Chicago	I-LOC	B-LOC	S-LOC
route	O	O	O
.	O	O	O

Standard algorithms for NER

Supervised Machine Learning given a human-labeled training set of text annotated with tags

- Hidden Markov Models
- Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)
- Neural sequence models (RNNs or Transformers)
- Large Language Models (like BERT), finetuned

Part-of-Speech Tagging

word	PTB tag	UD tag	UD attributes
The	DT	DET	DEFINITE=DEF PRONTYPE=ART
German	JJ	ADJ	DEGREE=POS
Expressionist	NN	NOUN	NUMBER=SING
movement	NN	NOUN	NUMBER=SING
was	VBD	AUX	MOOD=IND NUMBER=SING PERSON=3 TENSE=PAST VERBFORM=FIN
destroyed	VBN	VERB	TENSE=PAST VERBFORM=PART VOICE=PASS
as	IN	ADP	
a	DT	DET	DEFINITE=IND PRONTYPE=ART
result	NN	NOUN	NUMBER=SING
.	.	PUNCT	

Morphosyntactic Attributes

word	PTB tag	UD tag	UD attributes
The	DT	DET	DEFINITE=DEF PRONTYPE=ART
German	JJ	ADJ	DEGREE=POS
Expressionist	NN	NOUN	NUMBER=SING
movement	NN	NOUN	NUMBER=SING
was	VBD	AUX	MOOD=IND NUMBER=SING PERSON=3 TENSE=PAST VERBFORM=FIN
destroyed	VBN	VERB	TENSE=PAST VERBFORM=PART VOICE=PASS
as	IN	ADP	
a	DT	DET	DEFINITE=IND PRONTYPE=ART
result	NN	NOUN	NUMBER=SING
.	.	PUNCT	

Word Segmentation

theprophetsaidtothecity

(1)	日文	章魚	怎麼	說?
	Japanese	octopus	how	say

How to say octopus in Japanese?

(2)	日	文章	魚	怎麼	說?
	Japan	essay	fish	how	say

Figure 8.3: An example of tokenization ambiguity in Chinese (Sproat et al., 1996)

Code Switching

Although everything written on this site est disponible en anglais
is available in English
and in French, my personal videos seront bilingues
will be bilingual

Dialog Acts

Speaker	Dialogue Act	Utterance
A	YES-NO-QUESTION	*So do you go college right now?*
A	ABANDONED	*Are yo-*
B	YES-ANSWER	*Yeah,*
B	STATEMENT	*It's my last year [laughter].*
A	DECLARATIVE-QUESTION	*You're a, so you're a senior now.*
B	YES-ANSWER	*Yeah,*
B	STATEMENT	*I'm working on my projects trying to graduate [laughter]*
A	APPRECIATION	*Oh, good for you.*
B	BACKCHANNEL	*Yeah.*

Figure 8.4: An example of dialogue act labeling (Stolcke et al., 2000)

Beyond Token Labels: Syntax and Parsing

Chomsky Hierarchy

- Let Caps = nonterminals; lower = terminals; Greek = strings of terms/nonterms
- Recursively enumerable (Turing equivalent)
  - Rules: $\alpha \rightarrow \beta$
- Context-sensitive
  - Rules: $\alpha A \beta \rightarrow \alpha \gamma \beta$
- **Context-free**
  - **Rules: $A \rightarrow \alpha$**
- Regular (finite-state)
  - Rules: $A \rightarrow aB$; $A \rightarrow a$

Constituency Structure

Phrase structure organizes words into nested constituents

Starting unit: words

the, cat, cuddly, by, door

Words combine into phrases

the cuddly cat, by the door

Phrases can combine into bigger phrases

the cuddly cat by the door

Constituency Structure

Phrase structure organizes words into nested constituents.

the cat

a dog

large in a crate

barking on the table

cuddly by the door

large barking

talk to

walked behind

Dependency Structure

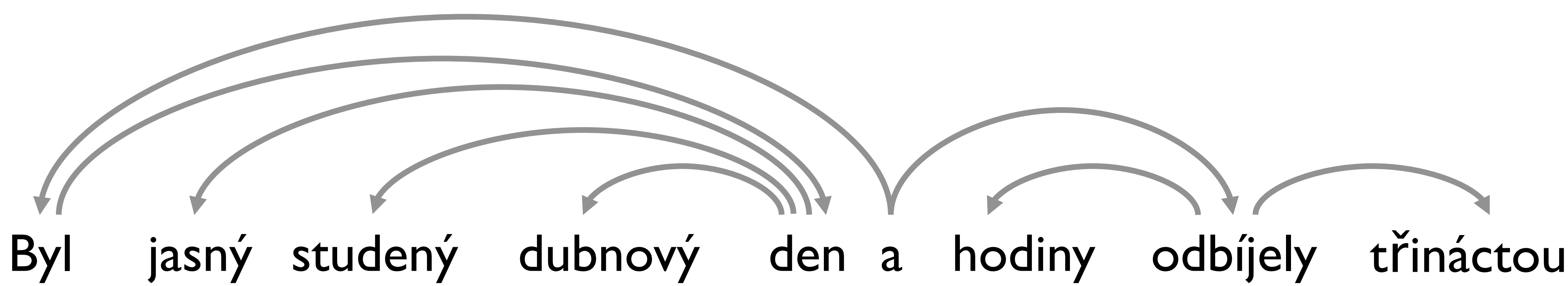
- Dependency structure shows which words depend on (modify, attach to, or are arguments of) which other words.

Byl jasný studený dubnový den a hodiny odbíjely třináctou

"It was a bright cold day in April and the clocks were striking thirteen"

Dependency Structure

- Dependency structure shows which words depend on (modify, attach to, or are arguments of) which other words.



"It was a bright cold day in April and the clocks were striking thirteen"

Why Syntax?

- Humans communicate complex ideas by composing words together into bigger units to convey complex meanings.
- Human listeners need to work out what modifies [attaches to] what. Explain human processing speed and errors.
- A model needs to understand sentence structure in order to be able to interpret language correctly, but it may not structure it in the same way as linguistic theories.
- Most usefully for NLP, linguistics gives us a vocabulary for describing phenomena and makes predictions about data.

Prepositional Attachment

San Jose cops kill man with knife

Close

Text Paper Translate Listen

San Jose cops kill man with knife

Ex-college football player, 23, shot 9 times allegedly charged police at fiancee's home

shortly after she called a suicide intervention hotline in hopes of get-

ed help from police."
She said Watkins was on the sidewalk in front

ing for their safety and defense of their life, fired at the suspect."

By Hamed Aleaziz and Vivian Ho

A man fatally shot by San Jose police officers while allegedly charging at them with a knife was a 23-year-old former football player at De Anza College in Cupertino who was distraught and de- pressed, his family said

Thursday.
Police officials said two officers opened fire Wed- nesday afternoon on Phillip Watkins outside his fiancee's home be- cause they feared for their lives. The officers had been drawn to the home, officials said, by a 911 call reporting an armed home invasion

BBC Sign in News Sport Weather Shop Reel Travel

NEWS

Home Video World US & Canada UK Business Tech Science Stories

Science & Environment

Scientists count whales from space

By Jonathan Amos
BBC Science Correspondent

Prepositional Attachment

Scientists count whales from space

✓

Scientists count whales from space

✗

Ambiguities Multiply

- A key parsing decision is how we 'attach' various constituents
  - PPs, adverbial or participial phrases, infinitives, coordinations, etc.

The board approved [its acquisition] [by Royal Trustco Ltd.]

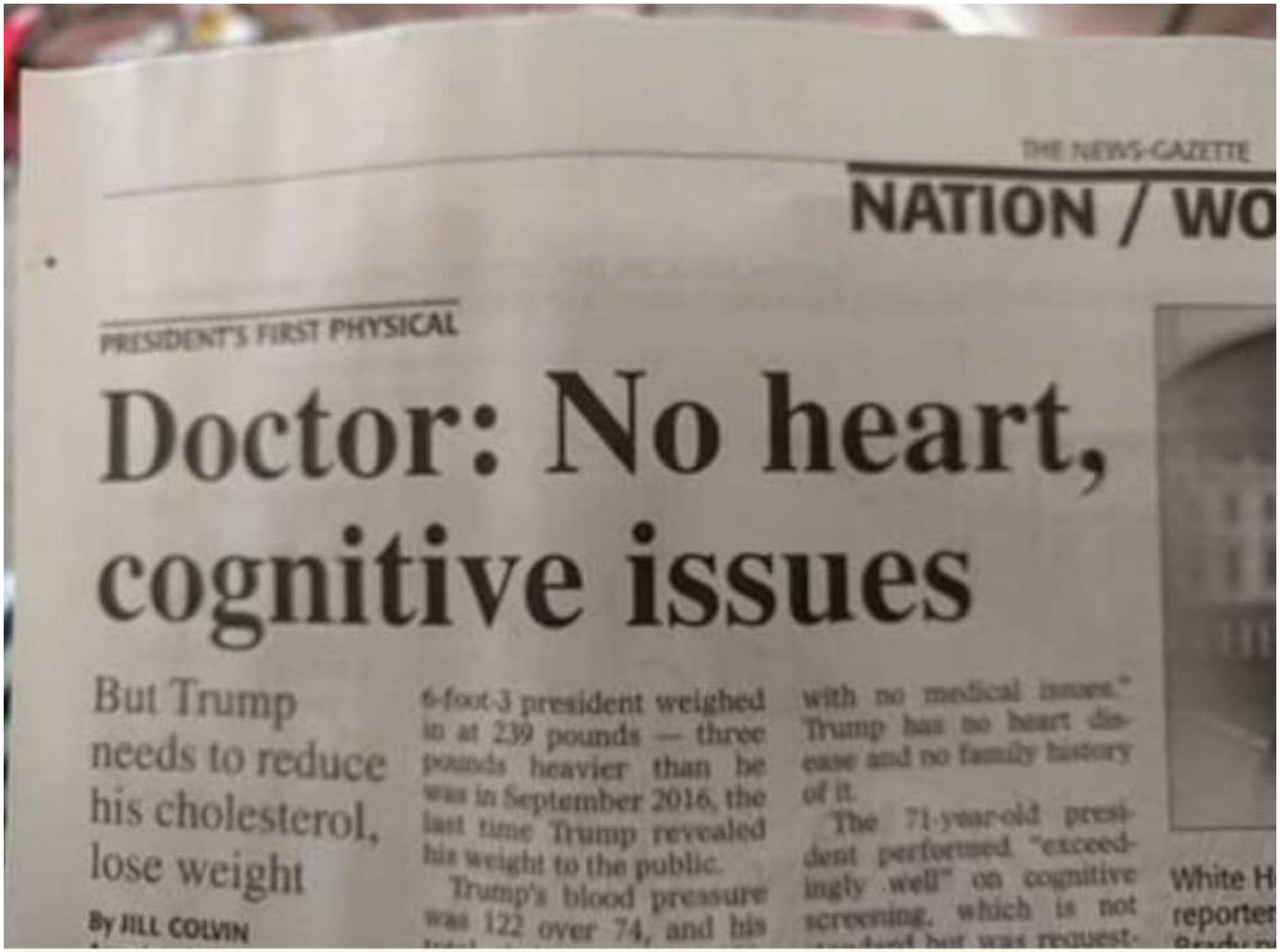
[of Toronto]

[for $27 a share]

[at its monthly meeting].

- Catalan numbers: $C_n = (2n)!/[(n+1)!n!]$
- An exponentially growing series, which arises in many tree-like contexts:
  - E.g., the number of possible triangulations of a polygon with $n+2$ sides

Coordination Scope Ambiguity



THE NEWS-GAZETTE

NATION / WO

PRESIDENT'S FIRST PHYSICAL

Doctor: No heart, cognitive issues

But Trump needs to reduce his cholesterol, lose weight

By JILL COLVIN

6-foot-3 president weighed in at 239 pounds — three pounds heavier than he was in September 2016, the last time Trump revealed his weight to the public.

Trump's blood pressure was 122 over 74, and his

with no medical issues." Trump has no heart disease and no family history of it.

The 71-year-old president performed "exceedingly well" on cognitive screening, which is not

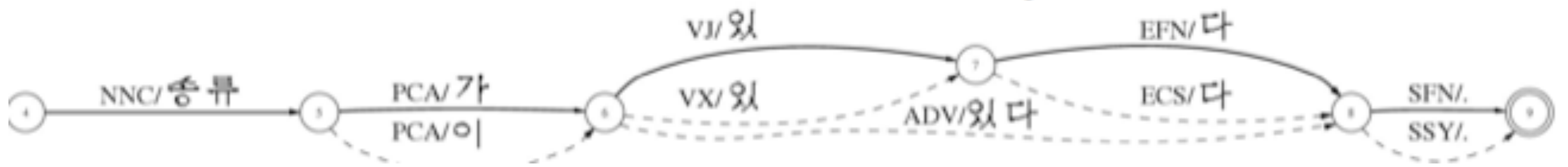
White H

reporter

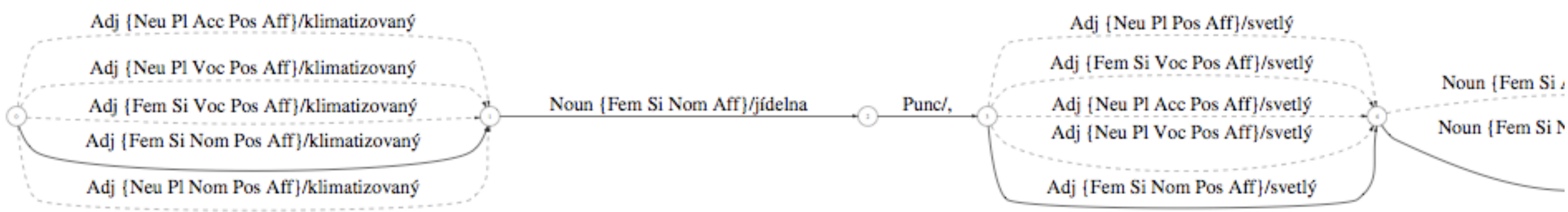
VP Attachment Ambiguity

Morphological Ambiguity

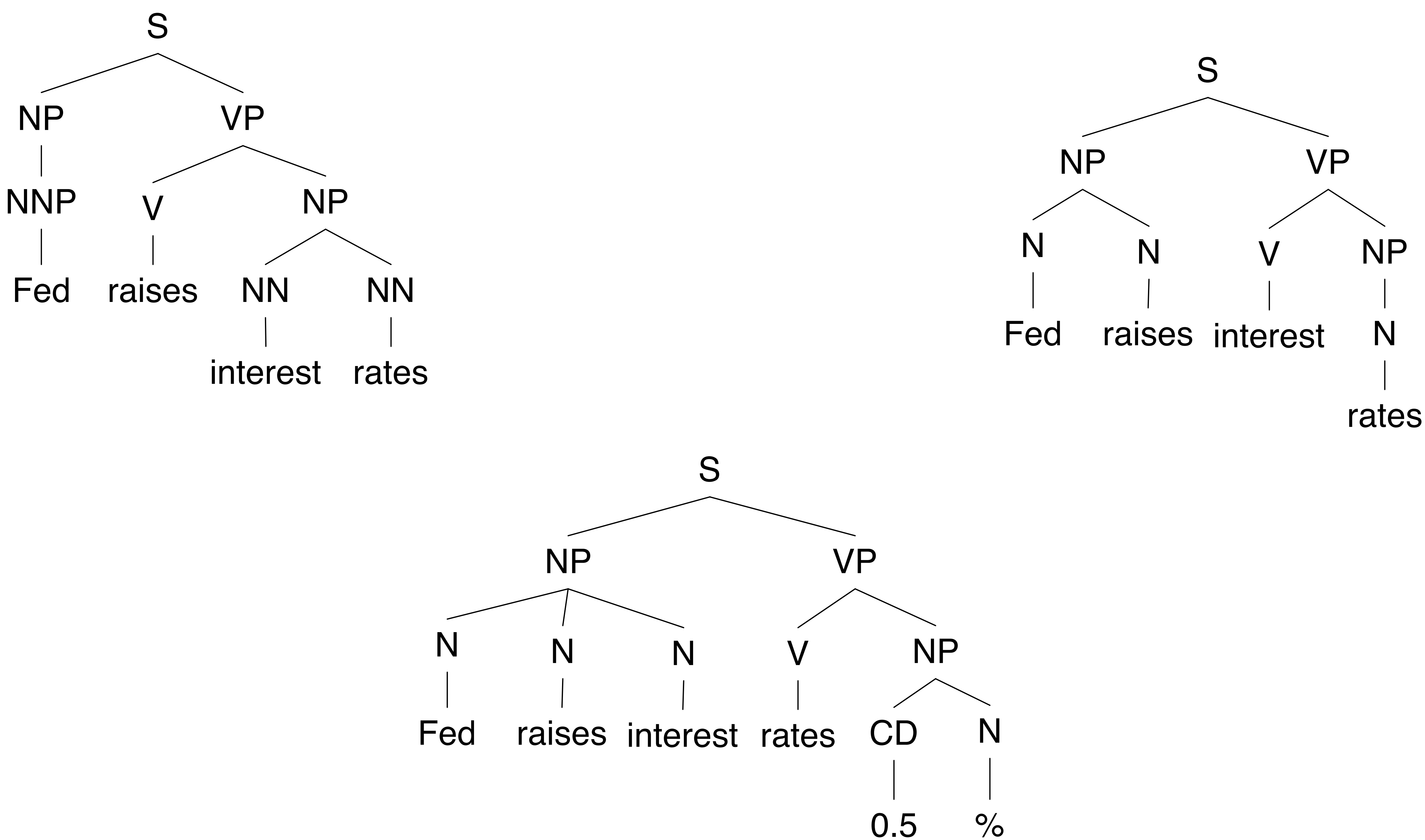
There are many kinds of trench mortars.



c. Klimatizovaná jídelna, světlá místnost pro snídaně. Air-conditioned dining room,



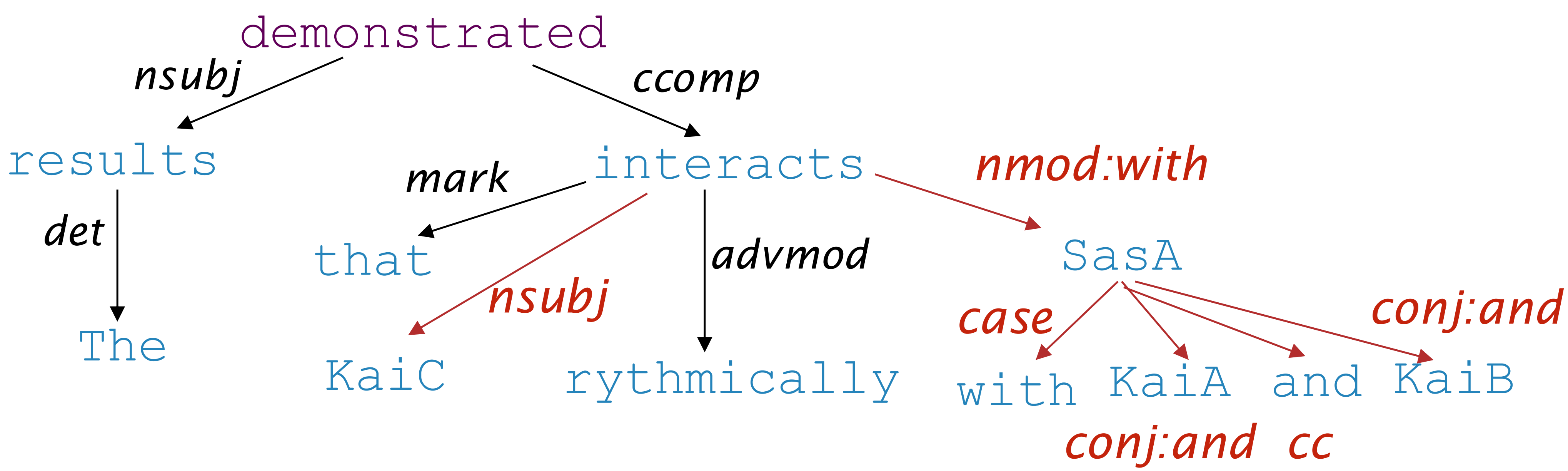
Syntactic Ambiguity



More Ambiguity

- Iraqi Head Seeks Arms
- Juvenile Court to Try Shooting Defendant
- Teacher Strikes Idle Kids
- Stolen Painting Found by Tree
- Kids Make Nutritious Snacks
- Local HS Dropouts Cut in Half
- British Left Waffles on Falkland Islands
- Red Tape Holds Up New Bridges
- Clinton Wins on Budget, but More Lies Ahead
- Ban on Nude Dancing on Governor's Desk

Dependencies Mapping to Semantics



KaiC 🡐nsubj interacts nmod:with 🡒 SasA

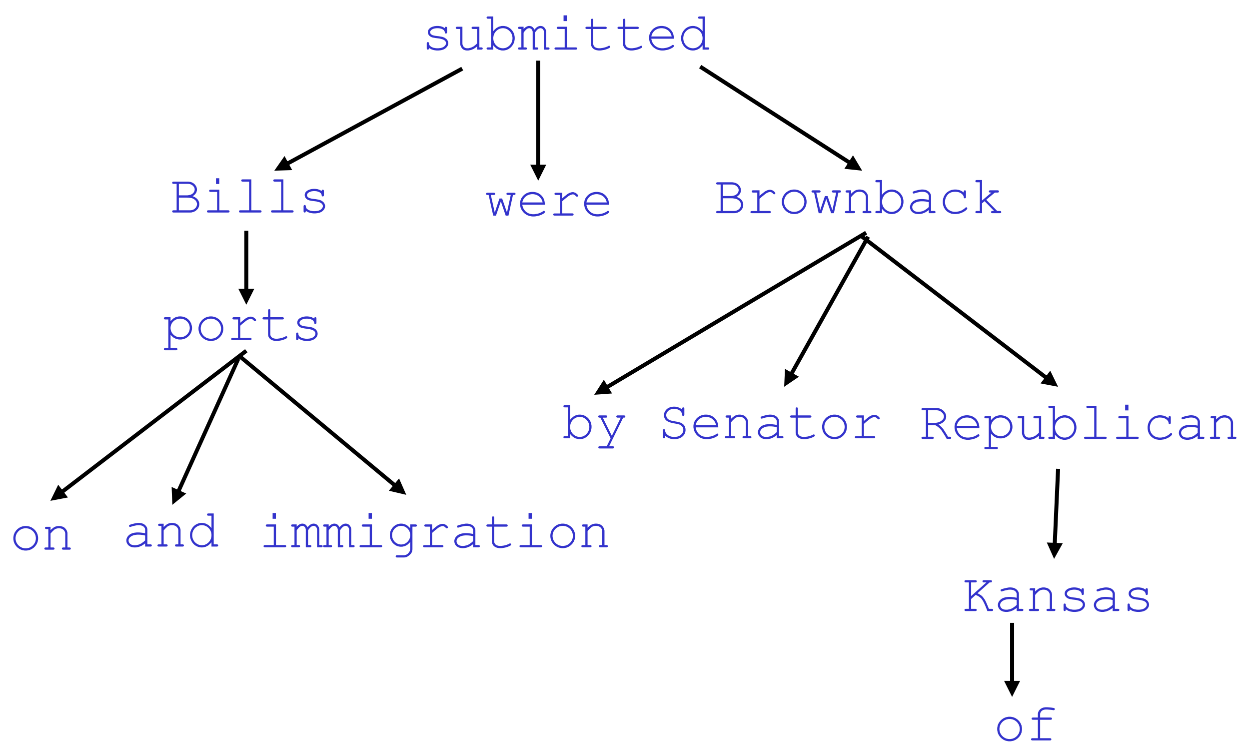
KaiC 🡐nsubj interacts nmod:with 🡒 SasA conj:and🡒 KaiA

KaiC 🡐nsubj interacts nmod:with 🡒 SasA conj:and🡒 KaiB

[Erkan et al. EMNLP 07, Fundel et al. 2007, etc.]

Dependency Structure

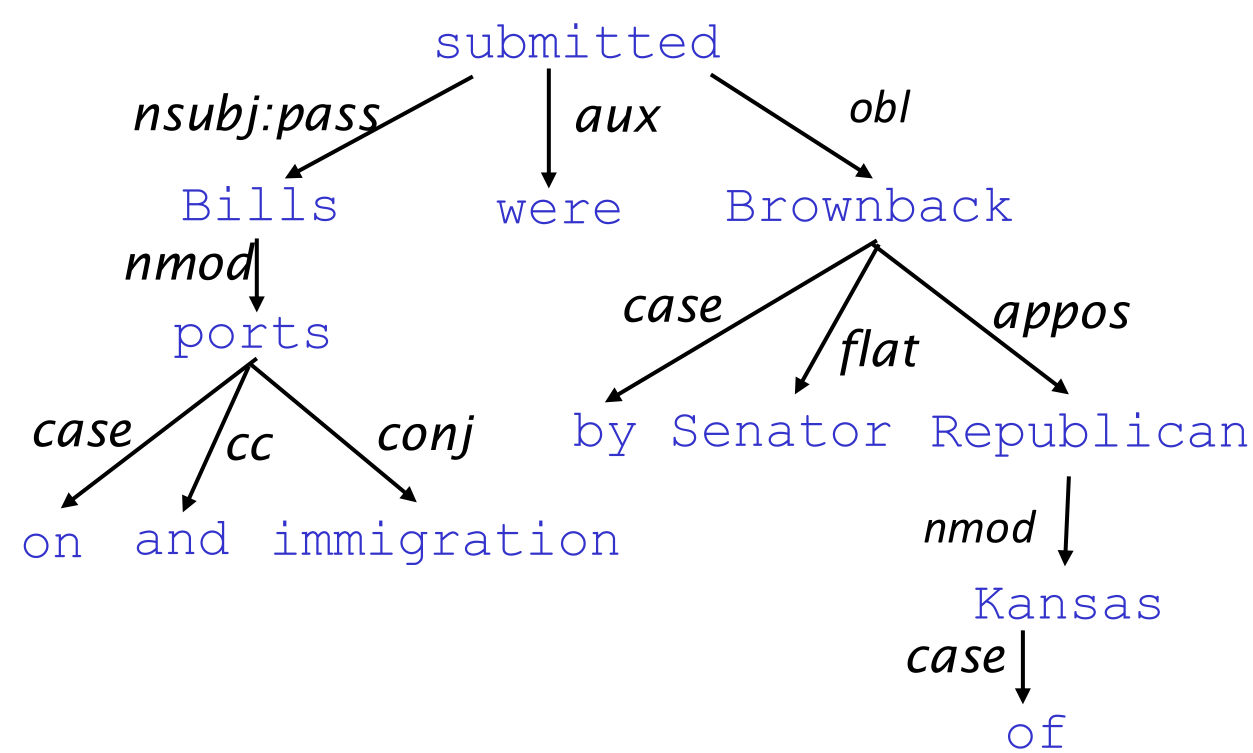
Dependency syntax postulates that syntactic structure consists of relations between lexical items, normally binary asymmetric relations ("arrows") called dependencies



Dependency Structure

Dependency syntax postulates that syntactic structure consists of relations between lexical items, normally binary asymmetric relations ("arrows") called dependencies

The arrows are commonly typed with the name of grammatical relations (subject, prepositional object, apposition, etc.)

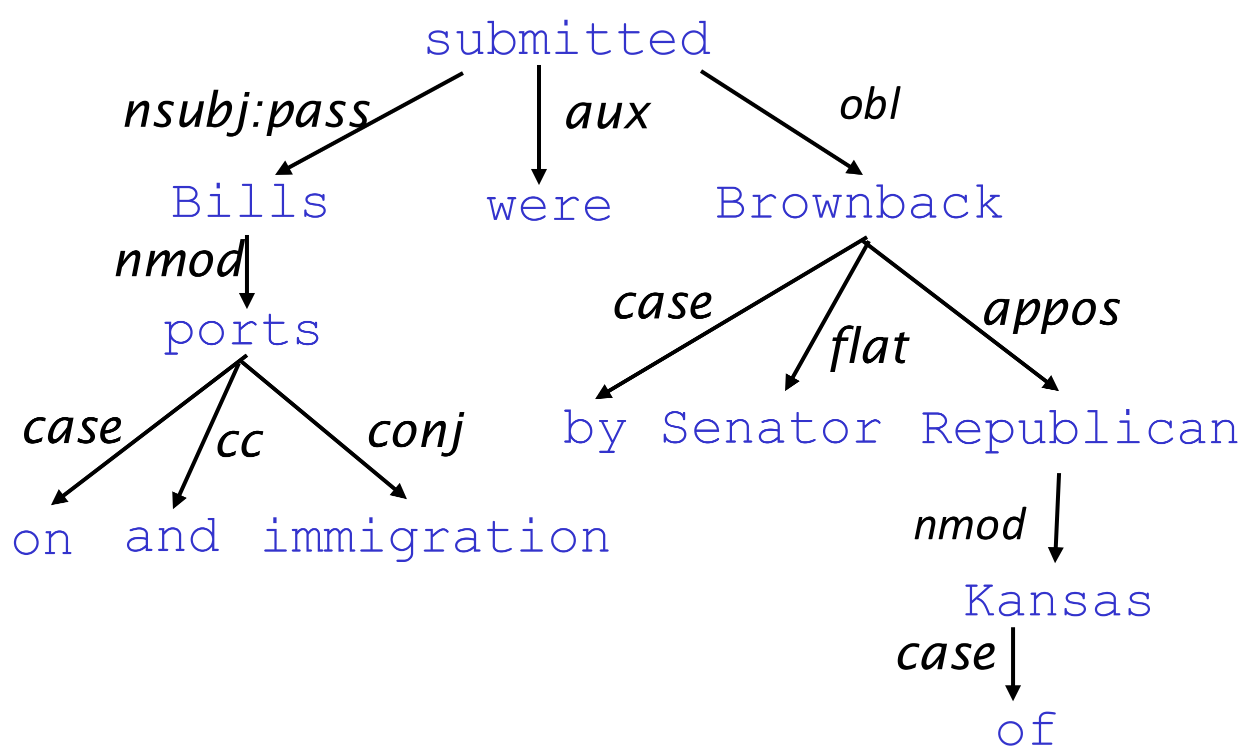


Dependency Structure

Dependency syntax postulates that syntactic structure consists of relations between lexical items, normally binary asymmetric relations ("arrows") called dependencies

An arrow connects a head with a dependent

Usually, dependencies form a tree (a connected, acyclic, single-root graph)



Panini's Grammar (5c BCE)

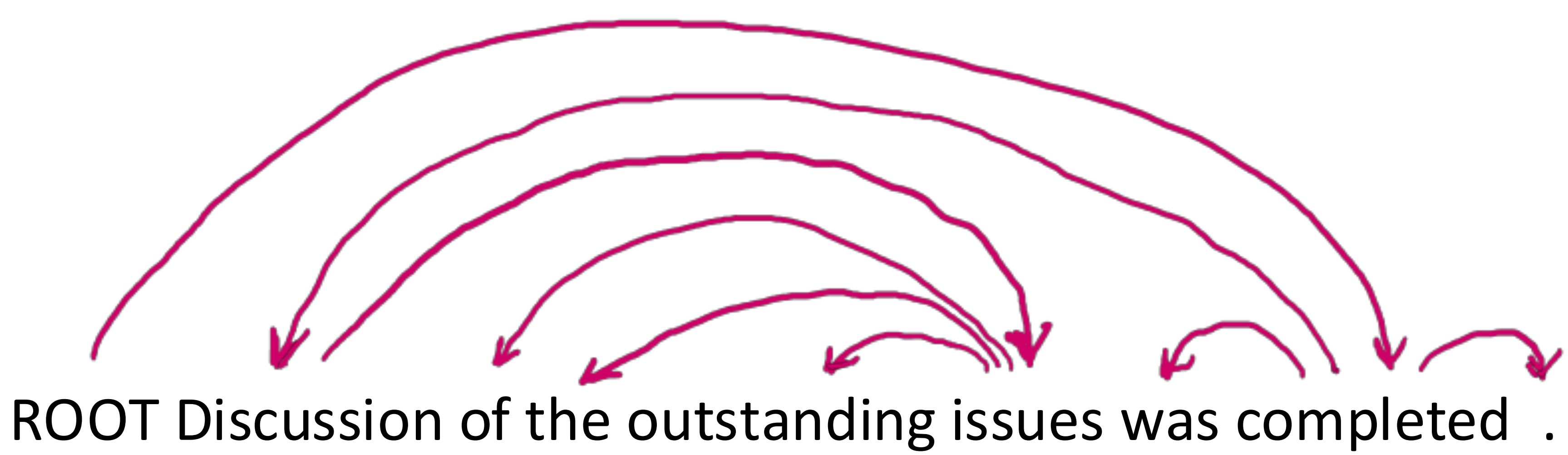
Gallery: http://wellcomeimages.org/indexplus/image/L0032691.html

But this comes from much later – originally the grammar was **oral**

Parsing History

- The idea of dependency structure goes back a long way
  - To Pāṇini's grammar (c. 5th century BCE)
  - Basic approach of 1st millennium Arabic grammarians
- Constituency/context-free grammar is a new-fangled invention
  - 20th century invention (R.S. Wells, 1947; then Chomsky 1953, etc.)
- Modern dependency work is often sourced to Lucien Tesnière (1959)
  - Was dominant approach in "East" in 20th Century (Russia, China, …)
    - Good for free-er word order, inflected languages like Russian (or Latin!)
- Used in some of the earliest parsers in NLP, even in the US:
  - David Hays, one of the founders of U.S. computational linguistics, built early (first?) dependency parser (Hays 1962) and published on dependency grammar in *Language*

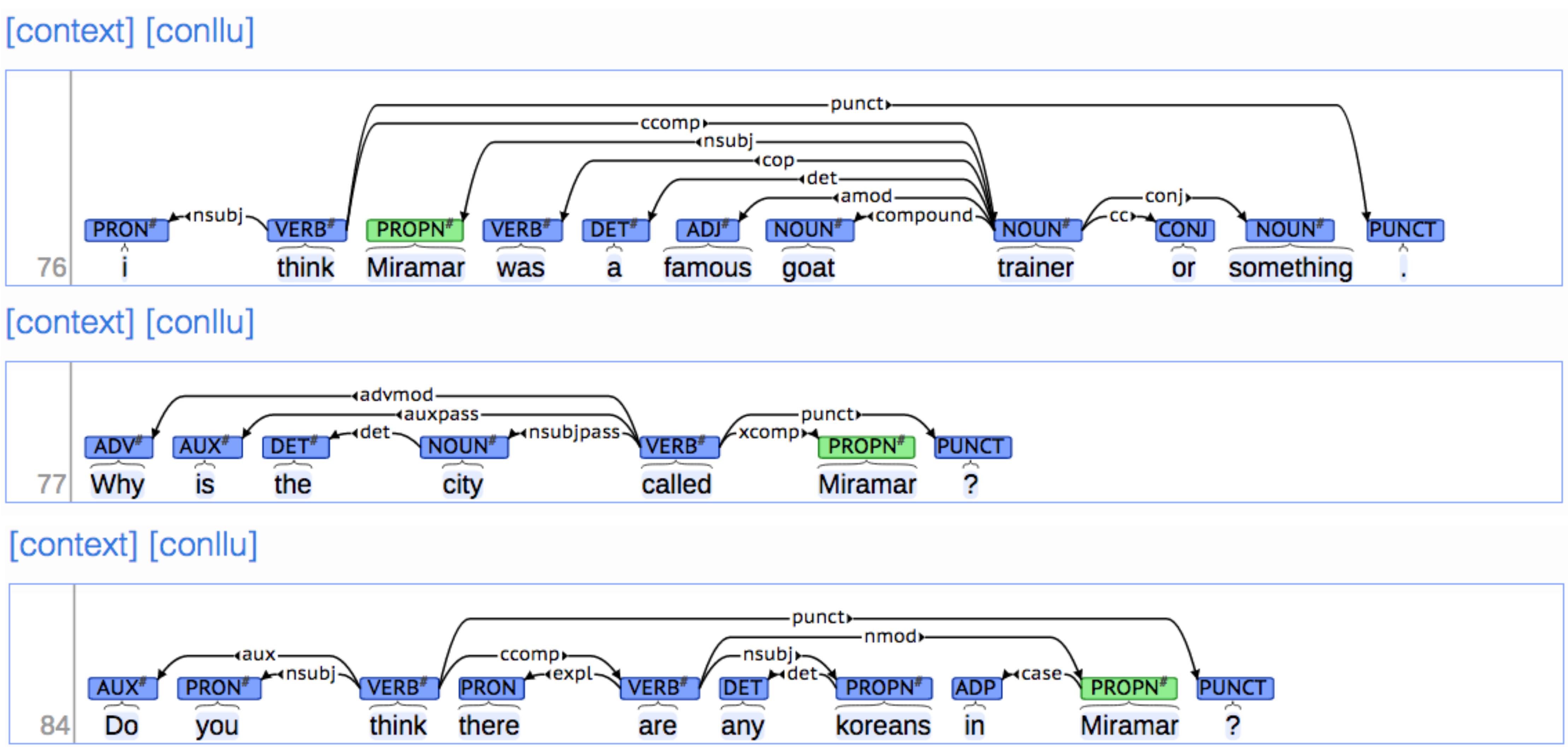
Dependency Parsing



- Some people draw the arrows one way; some the other way!
  - Tesnière had them point from head to dependent – we follow that convention
- We usually add a fake ROOT so every word is a dependent of precisely 1 other node

Dependency Treebanks

Brown corpus (1967; PoS tagged 1979); Lancaster-IBM Treebank (starting late 1980s);
Marcus et al. 1993, The Penn Treebank, *Computational Linguistics;*
Universal Dependencies: http://universaldependencies.org/



Dependency Treebanks

Starting off, building a treebank seems a lot slower and less useful than writing a grammar (by hand)

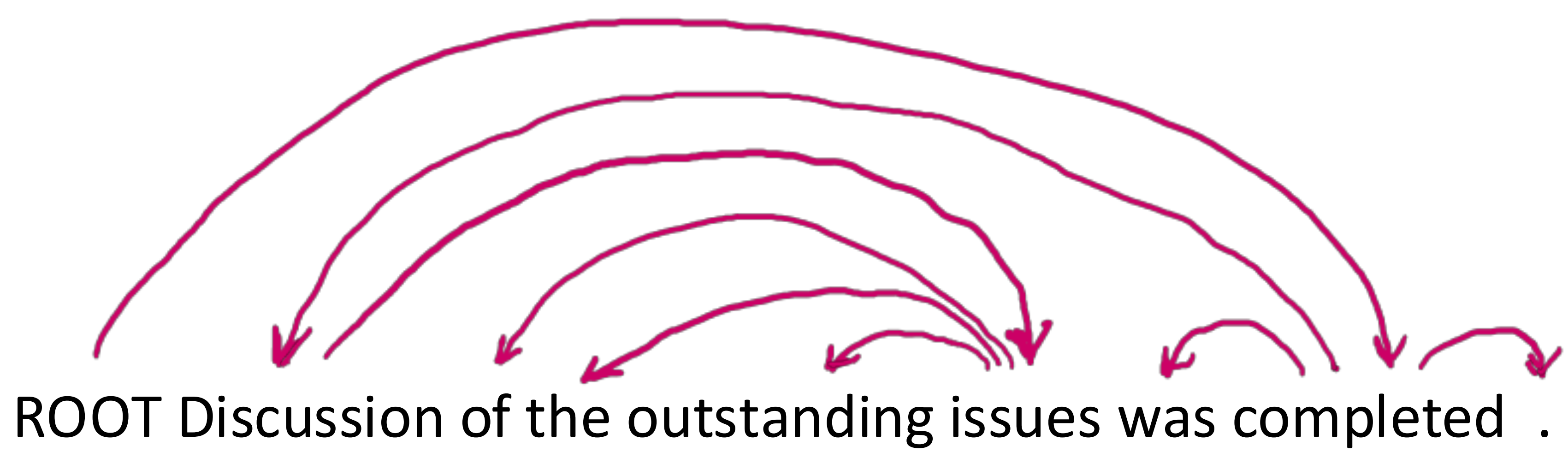
But a treebank gives us many things

- Reusability of the labor
  - Many parsers, part-of-speech taggers, etc. can be built on it
  - Valuable resource for linguistics
- Broad coverage, not just a few intuitions
- Frequencies and distributional information
- A way to evaluate NLP systems

Dependency Features

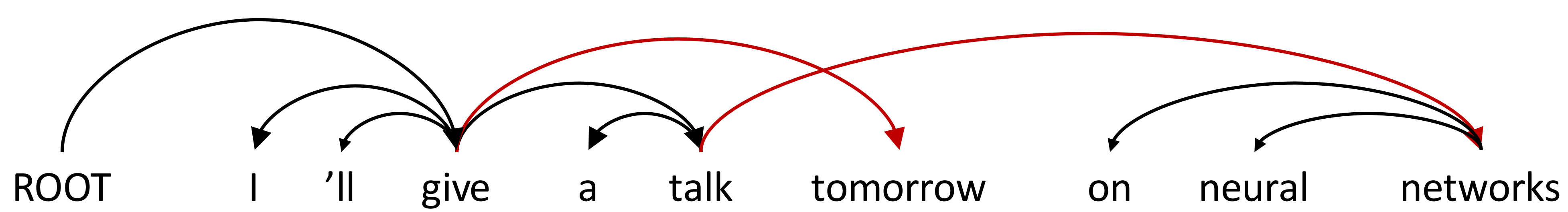
What are the straightforward sources of information for dependency parsing?

1. Bilexical affinities — The dependency [discussion → issues] is plausible
2. Dependency distance — Most dependencies are between nearby words
3. Intervening material — Dependencies rarely span intervening verbs or punctuation
4. Valency of heads — How many dependents on which side are usual for a head?



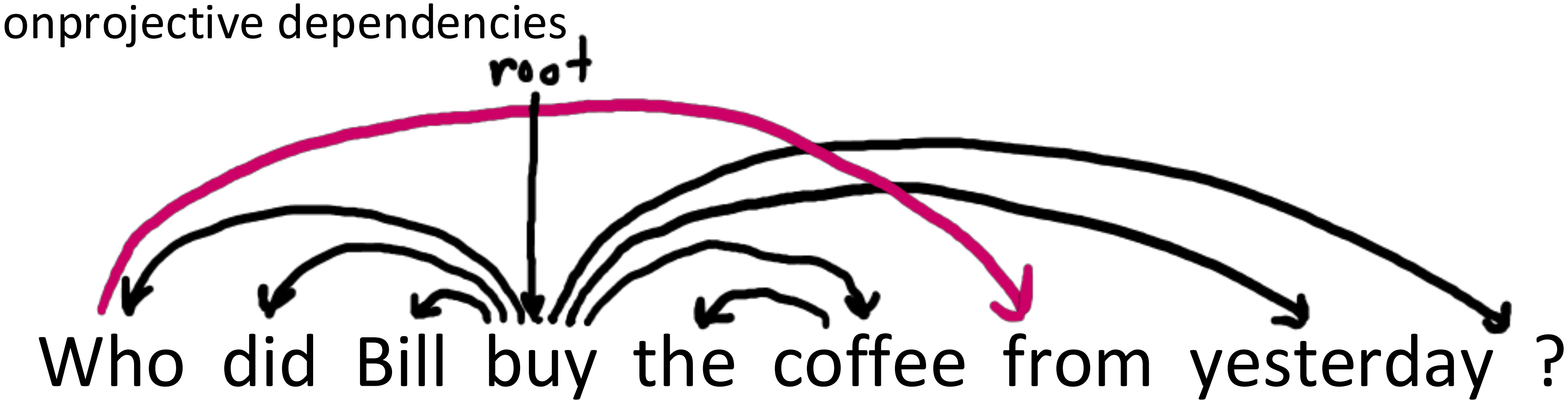
Dependency Parsing

- A sentence is parsed by choosing for each word what other word (including ROOT) it is a dependent of
- Usually some constraints:
  - Only one word is a dependent of ROOT
  - Don’t want cycles A → B, B → A
- This makes the dependencies a tree
- Final issue is whether arrows can cross (be non-projective) or not



Projectivity

- Definition of a projective parse: There are no crossing dependency arcs when the words are laid out in their linear order, with all arcs above the words
- Dependencies corresponding to a CFG tree must be projective
  - I.e., by forming dependencies by taking 1 child of each category as head
- Most syntactic structure is projective like this, but dependency theory normally does allow non-projective structures to account for displaced constituents
  - You can't easily get the semantics of certain constructions right without these nonprojective dependencies



Dependency Parsing

1. Dynamic programming

   Eisner (1996) gives a clever algorithm with complexity $O(n^3)$, by producing parse items with heads at the ends rather than in the middle

2. Graph algorithms

   You create a Minimum Spanning Tree for a sentence

   McDonald et al.'s (2005) $O(n^2)$ MSTParser scores dependencies independently using an ML classifier (he uses MIRA, for online learning, but it can be something else)

   Neural graph-based parser: Dozat and Manning (2017) et seq. – very successful!

3. Constraint Satisfaction

   Edges are eliminated that don't satisfy hard constraints. Karlsson (1990), etc.

4. "Transition-based parsing" or "deterministic dependency parsing"

   Greedy choice of attachments guided by good machine learning classifiers

   E.g., MaltParser (Nivre et al. 2008). Has proven highly effective. And fast.

Greedy Transition-Based Parsing

Nivre 2003

- A simple form of a greedy discriminative dependency parser
- The parser does a sequence of bottom-up actions
  - Roughly like "shift" or "reduce" in a shift-reduce parser – CS143, anyone?? – but the "reduce" actions are specialized to create dependencies with head on left or right
- The parser has:
  - a stack σ, written with top to the right
    - which starts with the ROOT symbol
  - a buffer β, written with top to the left
    - which starts with the input sentence
  - a set of dependency arcs A
    - which starts off empty
  - a set of actions

Transition-Based Parsing

Start: σ = [ROOT], $\beta = w_1, \ldots, w_n$, A = $\emptyset$

1. Shift $\quad \sigma, w_i|\beta, A \rightarrow \sigma|w_i, \beta, A$
2. Left-Arc$_r$ $\quad \sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_j, \beta, A \cup \{r(w_j,w_i)\}$
3. Right-Arc$_r$ $\quad \sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_i, \beta, A \cup \{r(w_i,w_j)\}$

Finish: $\sigma = [w]$, $\beta = \emptyset$

Transition-Based Parsing

Analysis of "I ate fish"

Start

[root] | I ate fish

Shift

[root] I | ate fish

Shift

[root] I ate | fish

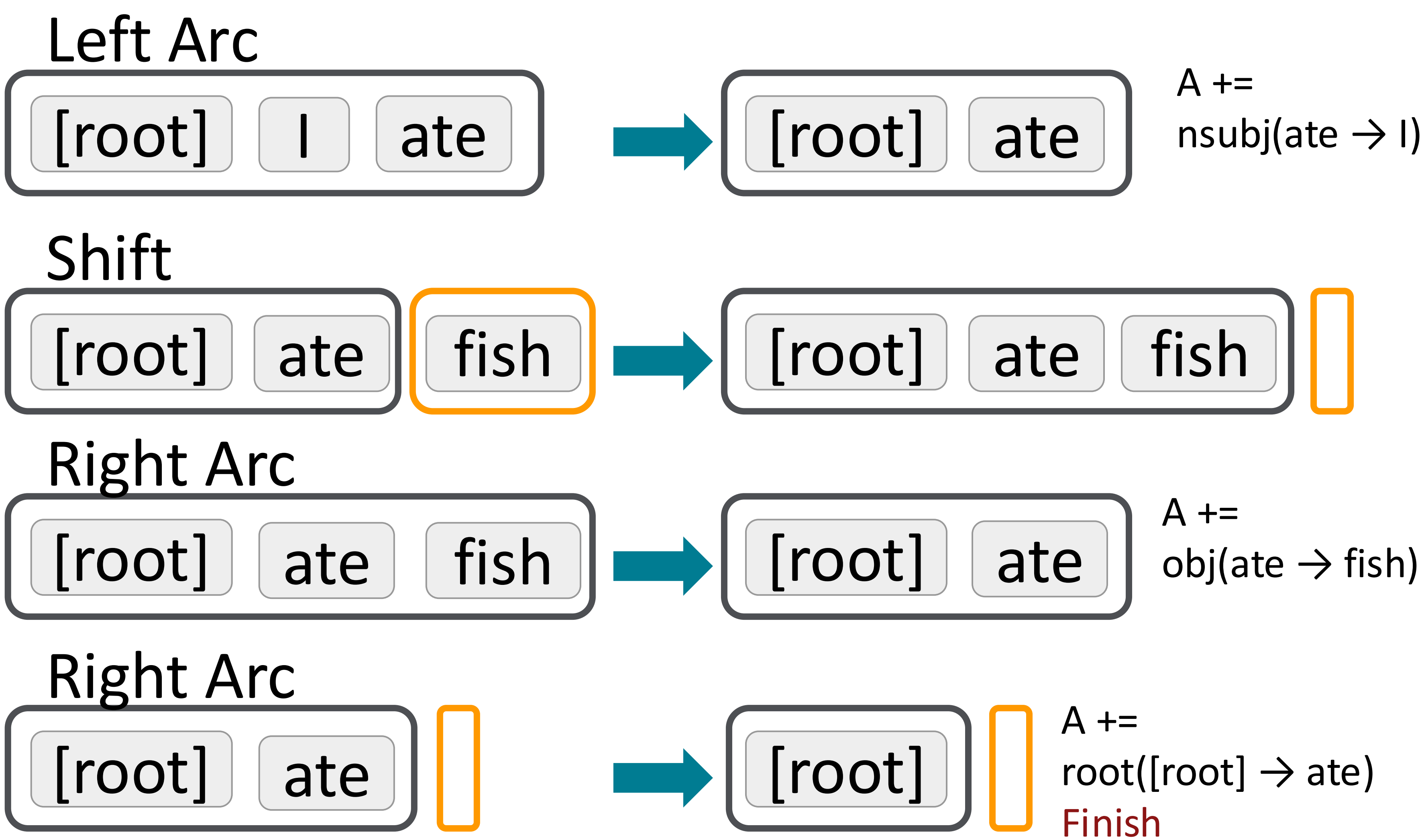
Start: σ = [ROOT], $\beta = w_1, \ldots, w_n$, A = $\emptyset$

1. Shift $\sigma, w_i|\beta, A \rightarrow \sigma|w_i, \beta, A$
2. Left-Arc$_r$ $\sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_j, \beta, A \cup \{r(w_j,w_i)\}$
3. Right-Arc$_r$ $\sigma|w_i|w_j, \beta, A \rightarrow \sigma|w_i, \beta, A \cup \{r(w_i,w_j)\}$

Finish: $\sigma = [w]$, $\beta = \emptyset$

Transition-Based Parsing

Analysis of “I ate fish”



Nota bene:
In this example I’ve at each step made the “correct” next transition.
But a parser has to work this out – by exploring or inferring!

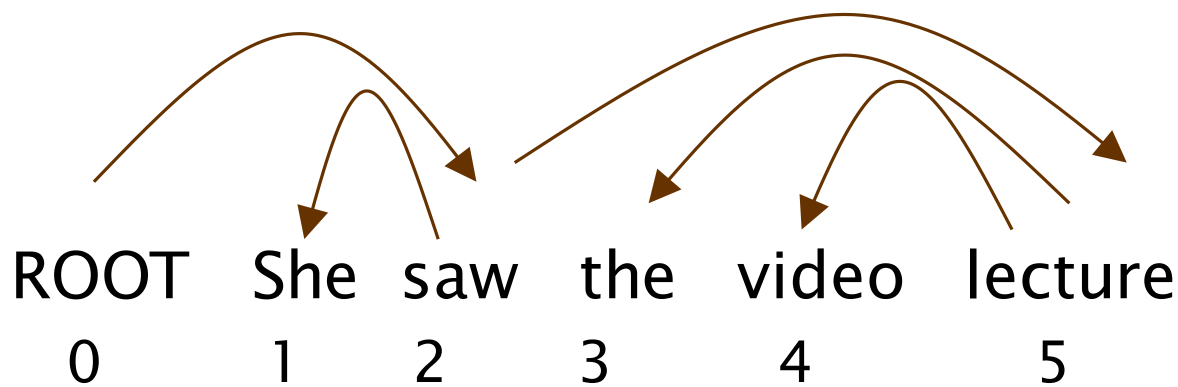
A = { nsubj(ate → I),
obj(ate → fish)
root([root] → ate) }

MaltParser

Nivre & Hall 2005

- We have left to explain how we choose the next action 🤷‍♀️
  - Answer: Stand back, I know machine learning!
- Each action is predicted by a discriminative classifier (e.g., softmax classifier) over each legal move
  - Max of 3 untyped choices (max of $|R| \times 2 + 1$ when typed)
  - Features: top of stack word, POS; first in buffer word, POS; etc.
- There is NO search (in the simplest form)
  - But you can profitably do a beam search if you wish (slower but better):
    - You keep *k* good parse prefixes at each time step
- The model's accuracy is *fractionally* below the state of the art in dependency parsing, but
- It provides **very fast linear time parsing**, with high accuracy – great for parsing the web

Evaluating Dependencies



$$\text{Acc} = \frac{\text{\# correct deps}}{\text{\# of deps}}$$

UAS = 4 / 5 = 80%

LAS = 2 / 5 = 40%

Gold

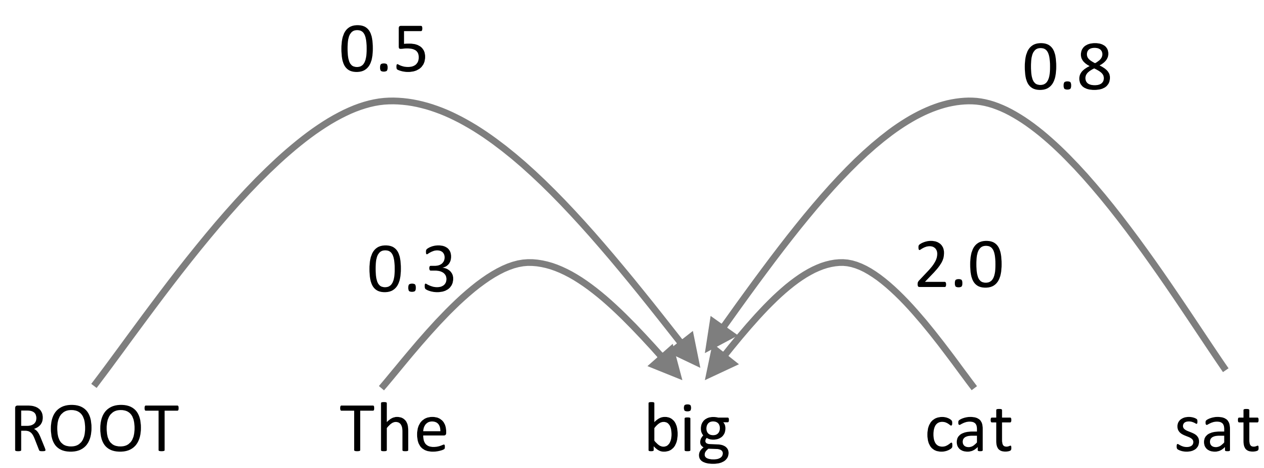
1	2	She	nsubj
2	0	saw	root
3	5	the	det
4	5	video	nn
5	2	lecture	obj

Parsed

1	2	She	nsubj
2	0	saw	root
3	4	the	det
4	5	video	nsubj
5	2	lecture	ccomp

Graph-Based Parsing

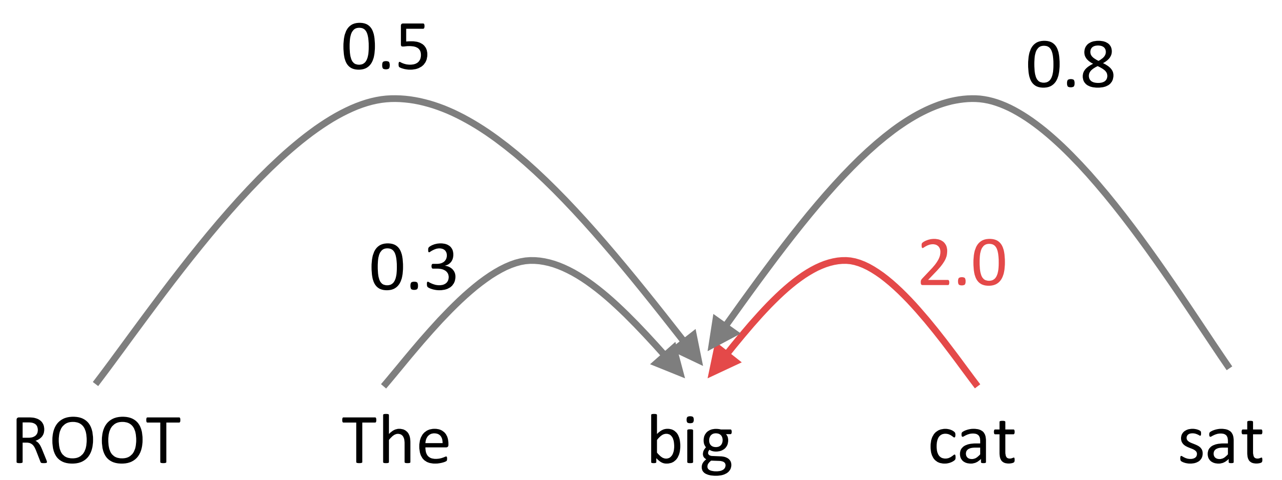
- Compute a score for every possible dependency for each word
  - Doing this well requires good “contextual” representations of each word token, which we will develop in coming lectures



e.g., picking the head for “big”

Graph-Based Parsing

- Compute a score for every possible dependency (choice of head) for each word
  - Doing this well requires more than just knowing the two words
  - We need **good "contextual" representations** of each word token, which we will develop in the coming lectures
- Repeat the same process for each other word; find the best parse (MST algorithm)



e.g., picking the head for "big"