"There exists today a very elaborate system of formal logic, and specifically, of logic as applied to mathematics. This is a discipline with many good sides, but also with certain serious weaknesses. . . .

Everybody who has worked in formal logic will confirm that it is one of the technically most refractory parts of mathematics. The reason for this is that it deals with rigid, all-or-none concepts, and has very little contact with the continuous concept of the real or of complex number, that is, with mathematical analysis. Yet analysis is the technically most successful and best-elaborated part of mathematics. Thus formal logic is, by the nature of its approach, cut off from the best cultivated portions of mathematics, and forced onto the most difficult part of mathematical terrain, into combinatorics."

— John von Neumann

Collecting, err, Correcting Speech Errors

Mark Johnson

Brown University

TAG+8, July 2006

Joint work with Eugene Charniak and Matt Lease

Supported by NSF grants IIS0095940 and DARPA GALE

Talk outline

- What are speech repairs, and why are they interesting?
- A *noisy channel model* of speech repairs
  - combines two very different kinds of structures
  - a novel model of *interpreting ill-formed input*
- "Rough copy" dependencies, context free and *tree adjoining grammars*
- Reranking using machine-learning techniques
- Training and evaluating the model of speech errors
- RT04F evaluation

Speech errors in (transcribed) speech

- Restarts and repairs

  Why didn't he, why didn't she stay at home?

  I want a flight *to Boston, uh,* to Denver on Friday

- Filled pauses

  I think it's, *uh*, refreshing to see the, *uh*, support ...

- Parentheticals

  But, *you know*, I was reading the other day ...

- "Ungrammatical" constructions

Bear, Dowding and Schriberg (1992), Charniak and Johnson (2001), Heeman and Allen (1999), Nakatani and Hirschberg (1994), Stolcke and Schriberg (1996)

Why focus on speech repairs?

- *Filled pauses* are easy to recognize (in transcripts at least)
- *Parentheticals* are handled by current parsers fairly well
- *Filled pauses* and *parentheticals* improve constituent boundary identification (just as punctuation does)
  - parser performs slightly better with parentheticals and filled pauses than with them removed
- *Ungrammatical constructions* aren't necessarily fatal
  - Statistical parsers *learn constructions in training corpus*
- . . . but *speech repairs* warrant special treatment, since the best parsers badly misanalyse them . . . we will see why shortly

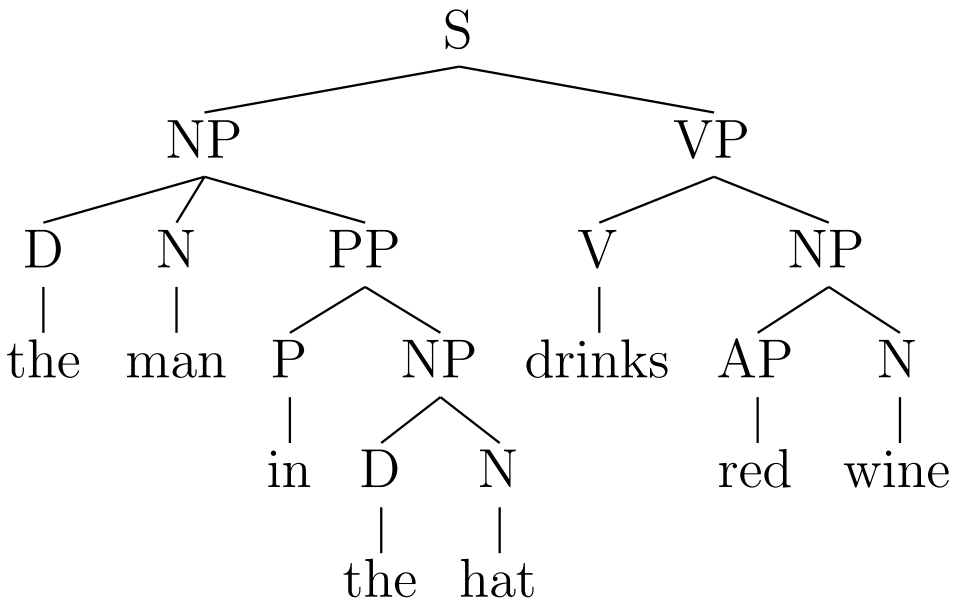
N-gram language models

- *n-gram* models capture dependencies between n adjacent words

  $\$ \rightarrow the \rightarrow man \rightarrow in \rightarrow the \rightarrow hat \rightarrow drinks \rightarrow red \rightarrow wine \rightarrow \$$

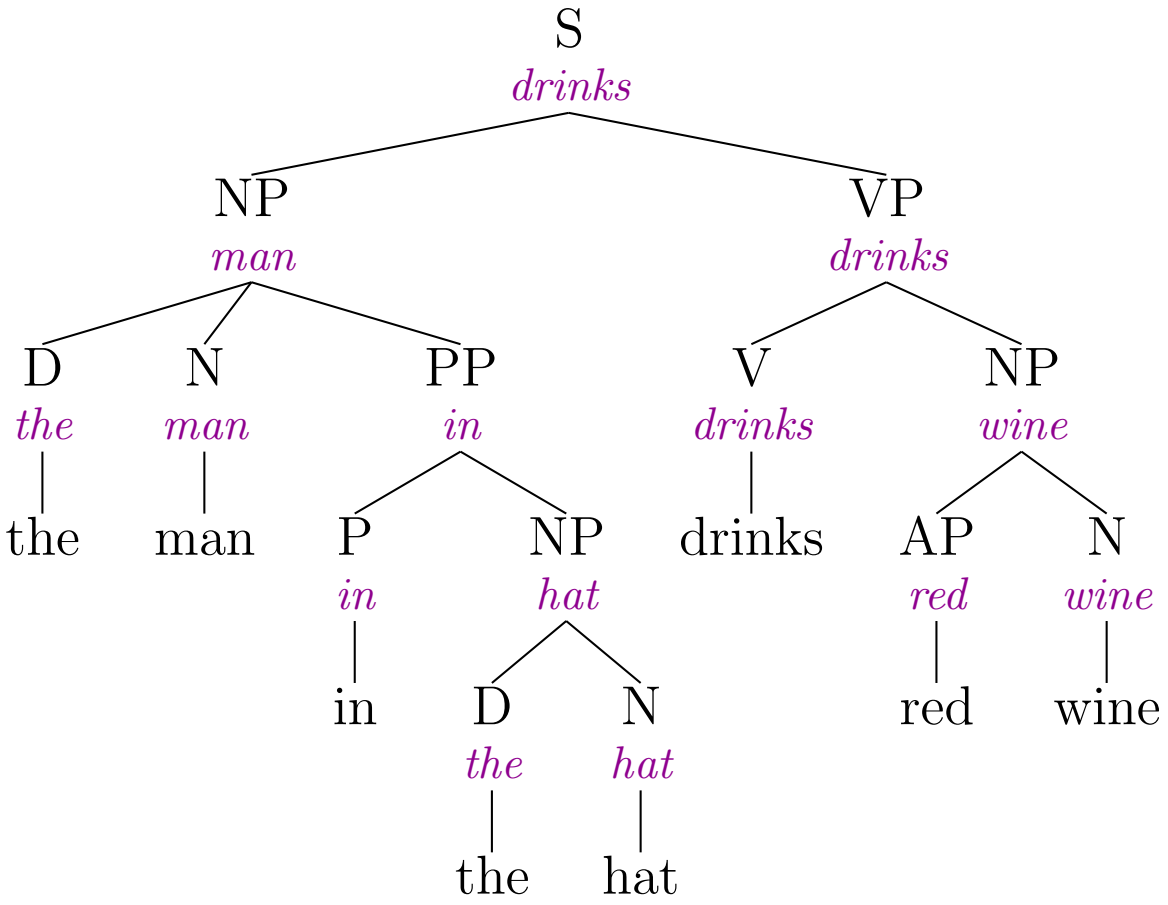
- Probabilities estimated from real *corpora*
- If model permits every word sequence to occur with non-zero probability $\Rightarrow$ model is *robust*
- Probability (rather than generativity) distinguishes "good" from "bad" sentences
- These simple models work surprisingly well because *they are lexicalized* (capture some semantic dependencies) and *most dependencies are local*

Probabilistic Context Free Grammars



- Rules are associated with *probabilities*
- Probability of a tree is the product of the probabilities of its rules
- *Most probable tree* is "best guess" at correct syntactic structure

Head to head dependencies



Rules:

$$\underset{drinks}{S} \rightarrow \underset{man}{NP}\ \underset{drinks}{VP}$$

$$\underset{drinks}{VP} \rightarrow \underset{drinks}{V}\ \underset{wine}{NP}$$

$$\underset{wine}{NP} \rightarrow \underset{red}{AP}\ \underset{wine}{N}$$

. . .

- *Lexicalization* captures many syntactic and semantic dependencies
- in right-branching structures, n-gram dependencies $\Leftrightarrow$ head-to-head dependencies

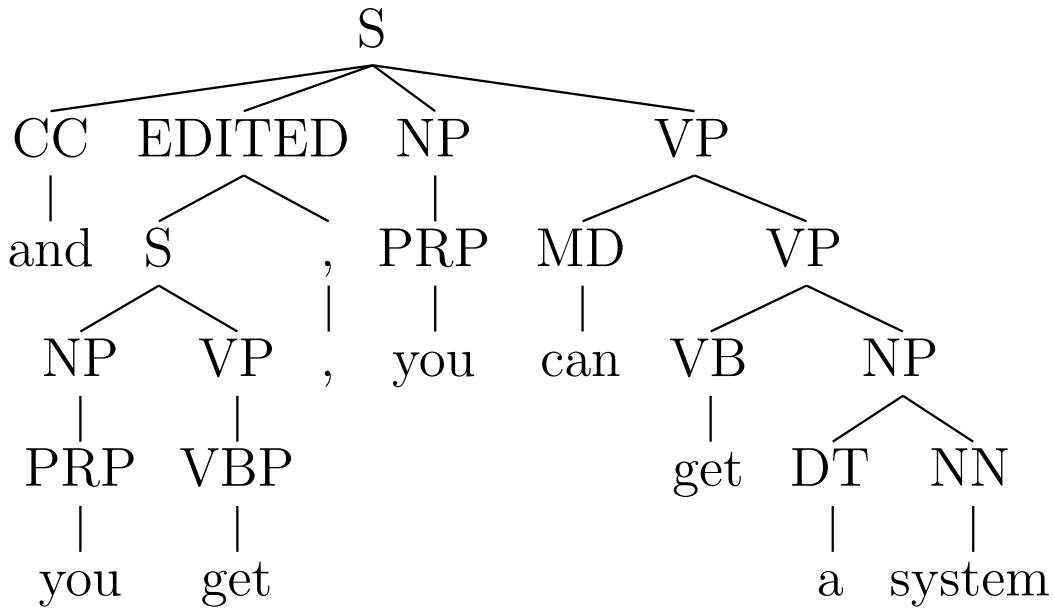
The structure of repairs

. . . and $\underbrace{\text{you get,}}_{\text{Reparandum}}$ $\underbrace{\text{uh,}}_{\text{Interregnum}}$ $\underbrace{\text{you can get}}_{\text{Repair}}$ a system . . .

- The Reparandum is *often not a syntactic phrase*
- The Interregnum is usually lexically and prosodically marked, but can be empty
- The Reparandum is often a *"rough copy"* of the Repair
  - Repairs are typically short
  - Repairs are not always copies

Shriberg 1994 "Preliminaries to a Theory of Speech Disfluencies"

Treebank representation of repairs



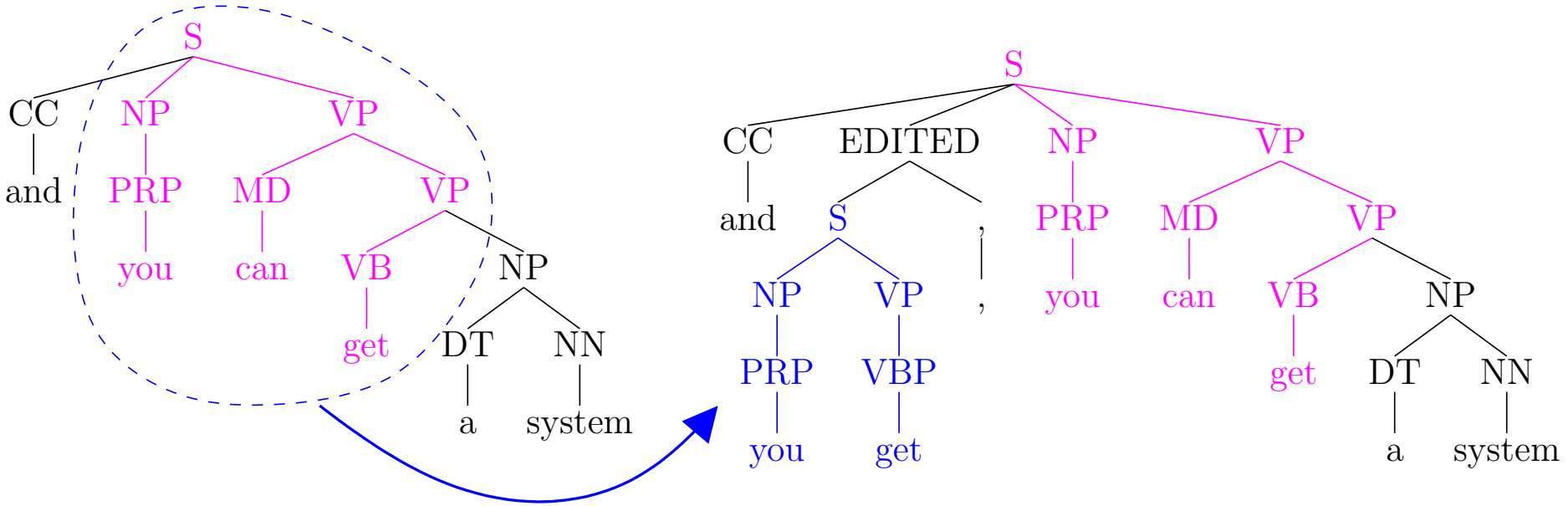
- The *Switchboard treebank* contains the parse trees for 1M words of spontaneous telephone conversations
- Each reparandum is indicated by an EDITED node (interregnum and repair are also annotated)
- But Charniak's parser finds very few EDITED nodes!

The "true model" of repairs (?)

. . . and $\underbrace{\text{you get,}}_{\text{Reparandum}}$ $\underbrace{\text{uh,}}_{\text{Interregnum}}$ $\underbrace{\text{you can get}}_{\text{Repair}}$ a system . . .

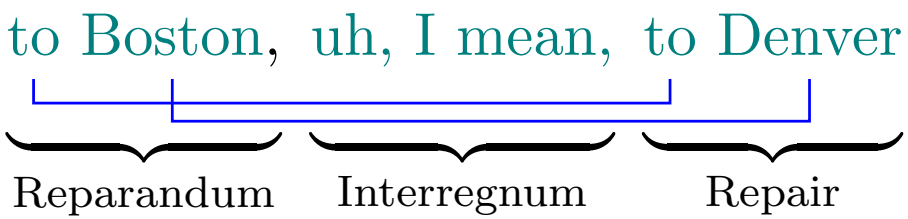
- Speaker generates intended "conceptual representation"
- Speaker incrementally generates syntax and phonology,
  - recognizes that what is said doesn't mean what was intended,
  - "backs up", i.e., partially deconstructs syntax and phonology, and
  - starts incrementally generating syntax and phonology again
- but without a good model of "conceptual representation", this may be hard to formalize . . .

Approximating the "true model" (1)



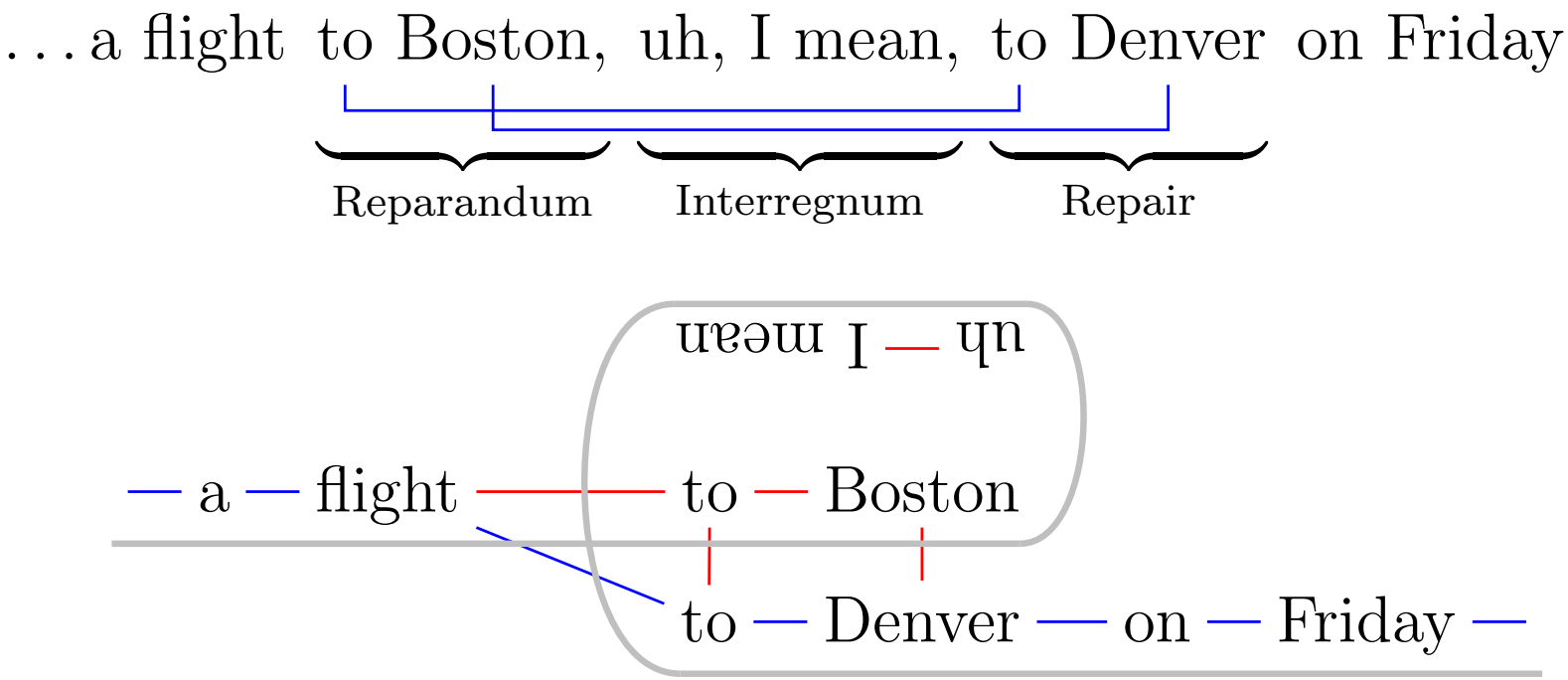
- Approximate semantic representation by *syntactic structure*
- Tree with reparandum and interregnum excised is what speaker intended to say
- Reparandum results from attempt to generate Repair structure
- Dependencies are *very different to those in "normal" language!*

Approximating the "true model" (2)



- Use Repair string as approximation to intended meaning
- Reparandum string is "rough copy" of Repair string
  - involves *crossing* (rather than *nested*) dependencies
- String with reparandum and interregnum excised is well-formed
  - after correcting the error, what's left should have high probability
  - *use model of normal language to interpret ill-formed input*

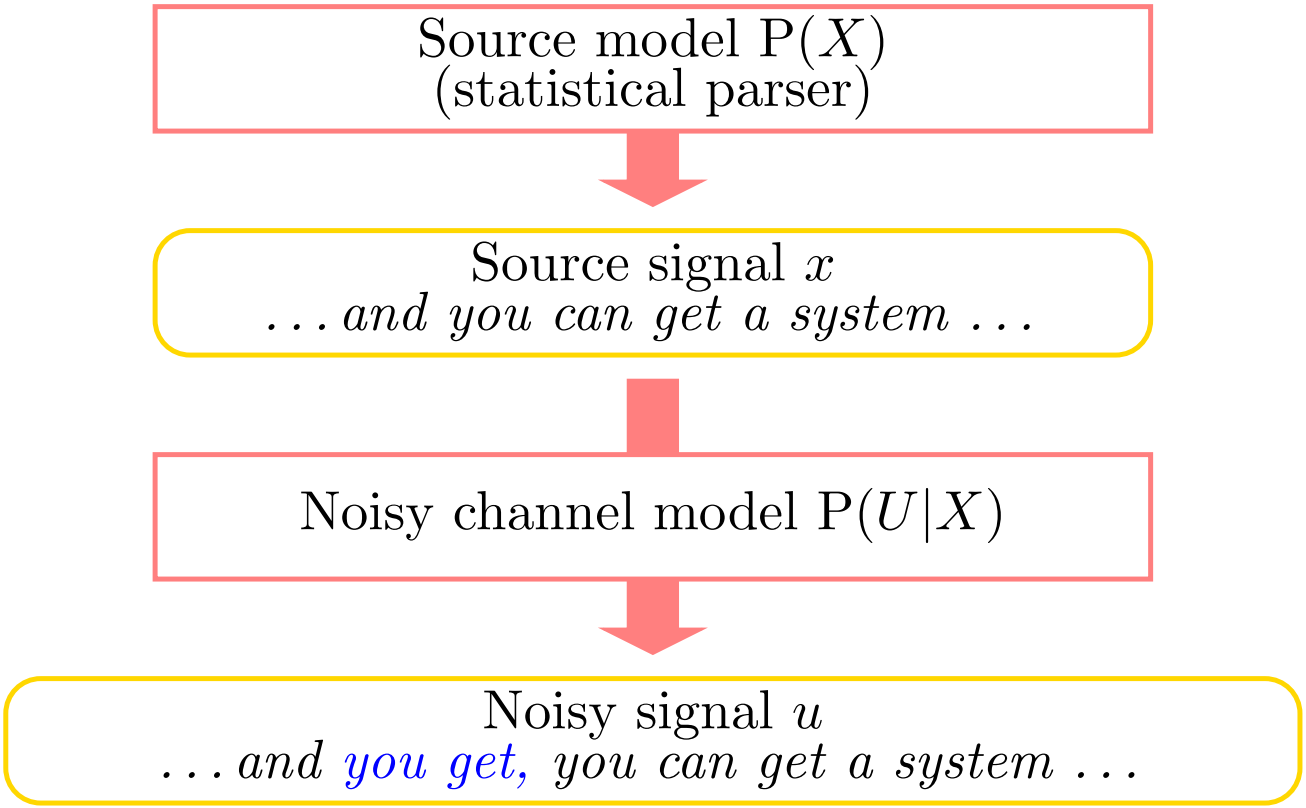
Helical structure of speech repairs



- Repair dependencies seem *incompatible* with standard syntactic structures
- *Can we have both syntactic structure and repair structure?*

Joshi (2002), ACL Lifetime achievement award talk

The Noisy Channel Model

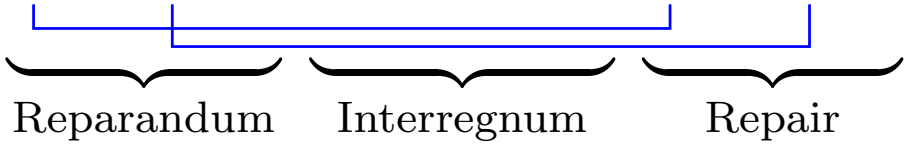


- Noisy channel models combines two different submodels
- *Bayes rule* describes how to invert the channel

$$\mathrm{P}(x|u) = \frac{\mathrm{P}(u|x)\mathrm{P}(x)}{\mathrm{P}(u)}$$

The channel model

I want a flight to Boston, uh, I mean, to Denver on Friday



- Channel model is a *transducer* producing *source:output pairs*

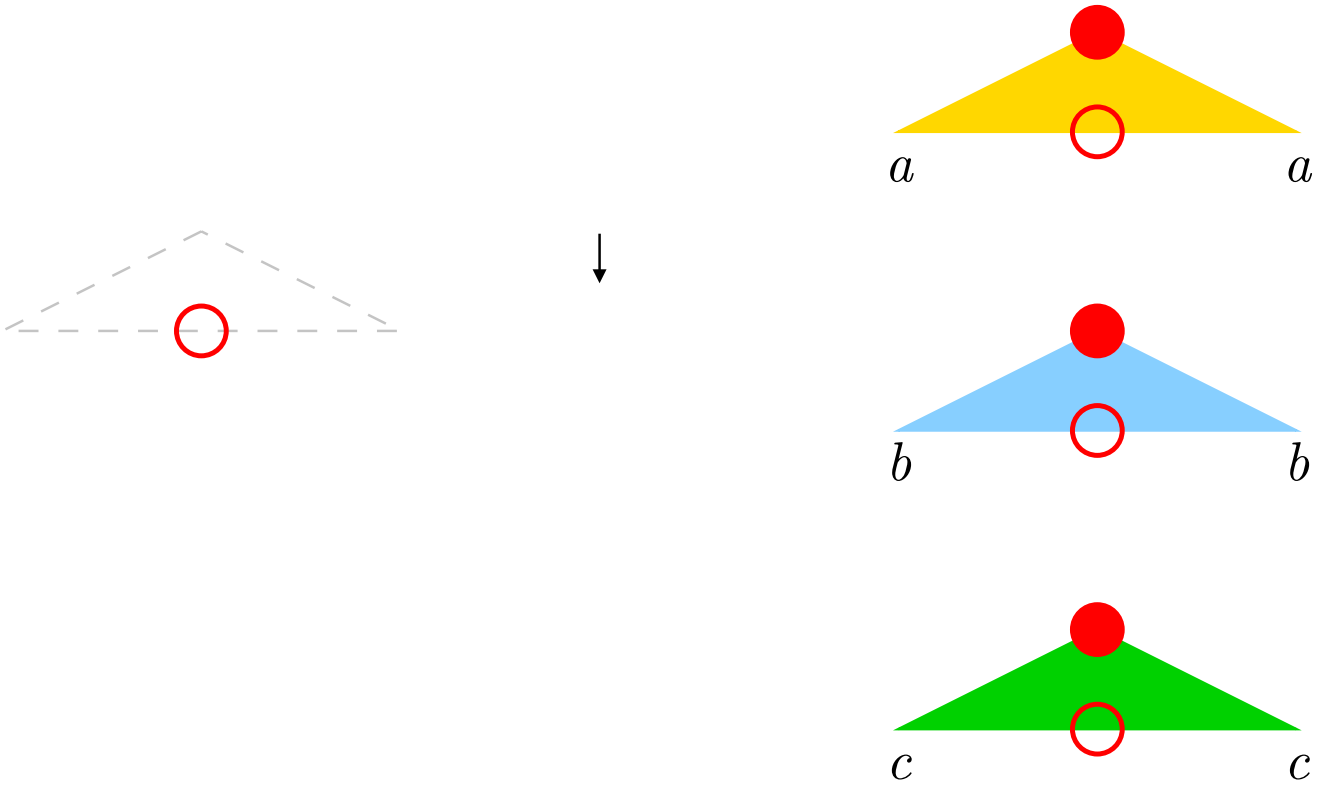
  . . . a:a flight:flight ∅:to ∅:Boston ∅:uh ∅:I ∅:mean to:to Denver:Denver . . .

- only 62 different phrases appear in *interregnum* (*uh*, *I mean*)

  ⇒ *unigram model* of interregnum phrases

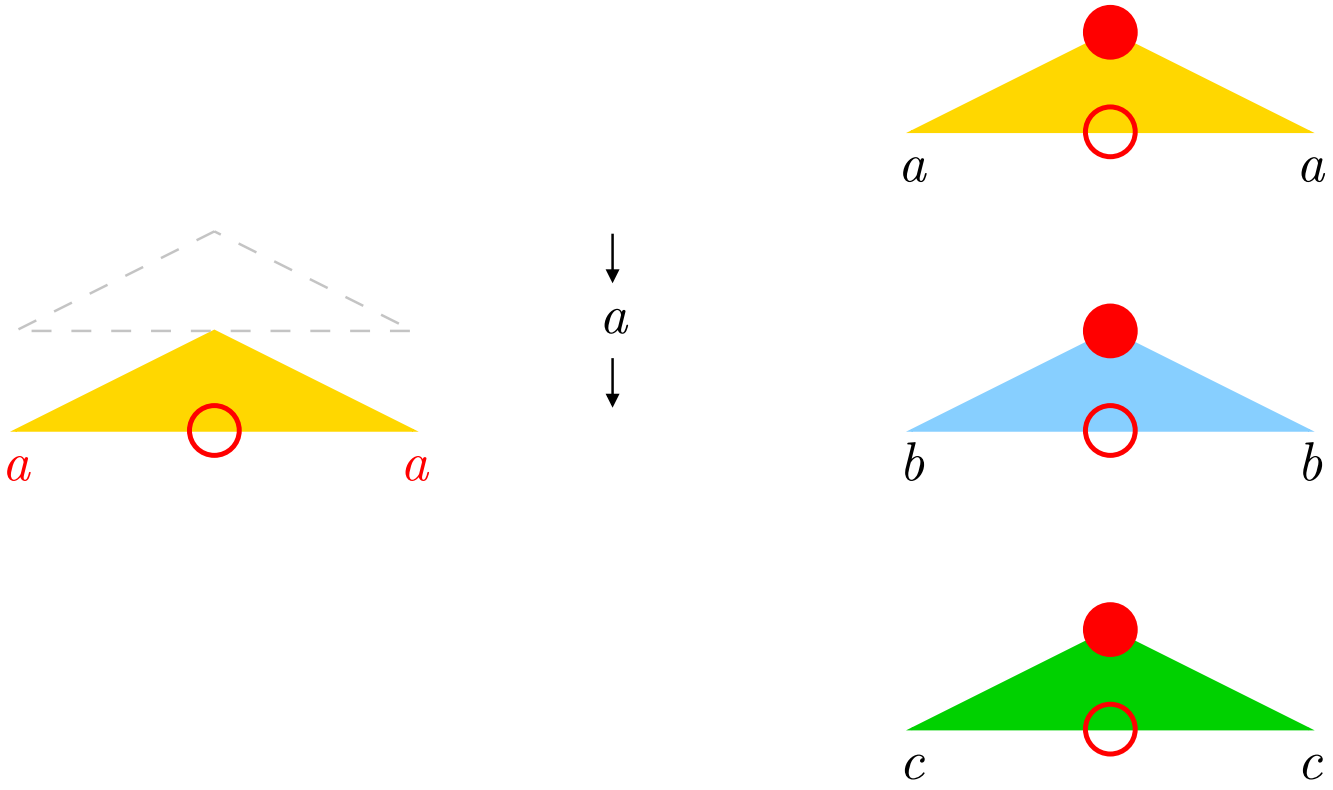
- *Reparandum* is "rough copy" of repair
  - We need a probabilistic model of rough copies
  - FSMs and CFGs *can't generate copy dependencies* . . .
  - but *Tree Adjoining Grammars* can

CFGs generate ww^R dependencies (1)



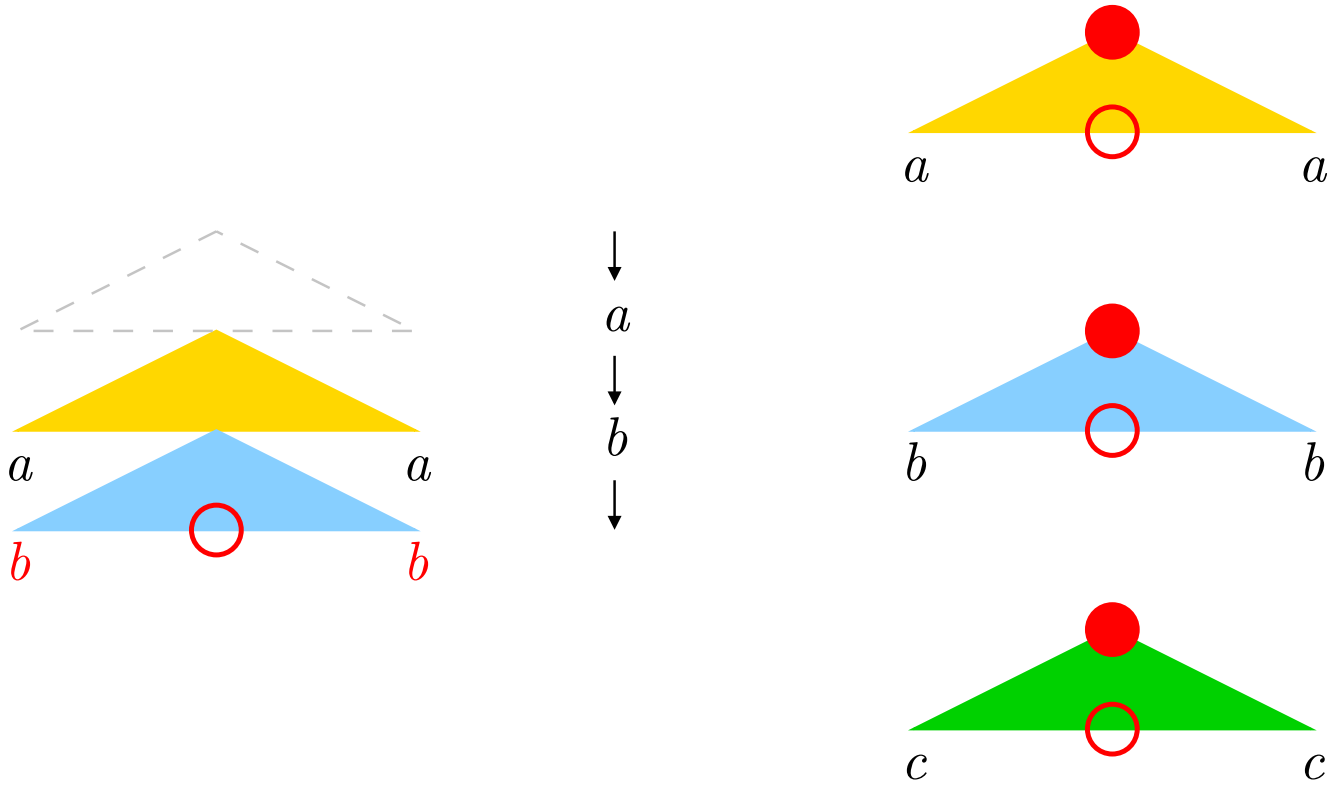
- CFGs generate *nested dependencies* between a string w and its reverse w^R

CFGs generate ww^R dependencies (2)



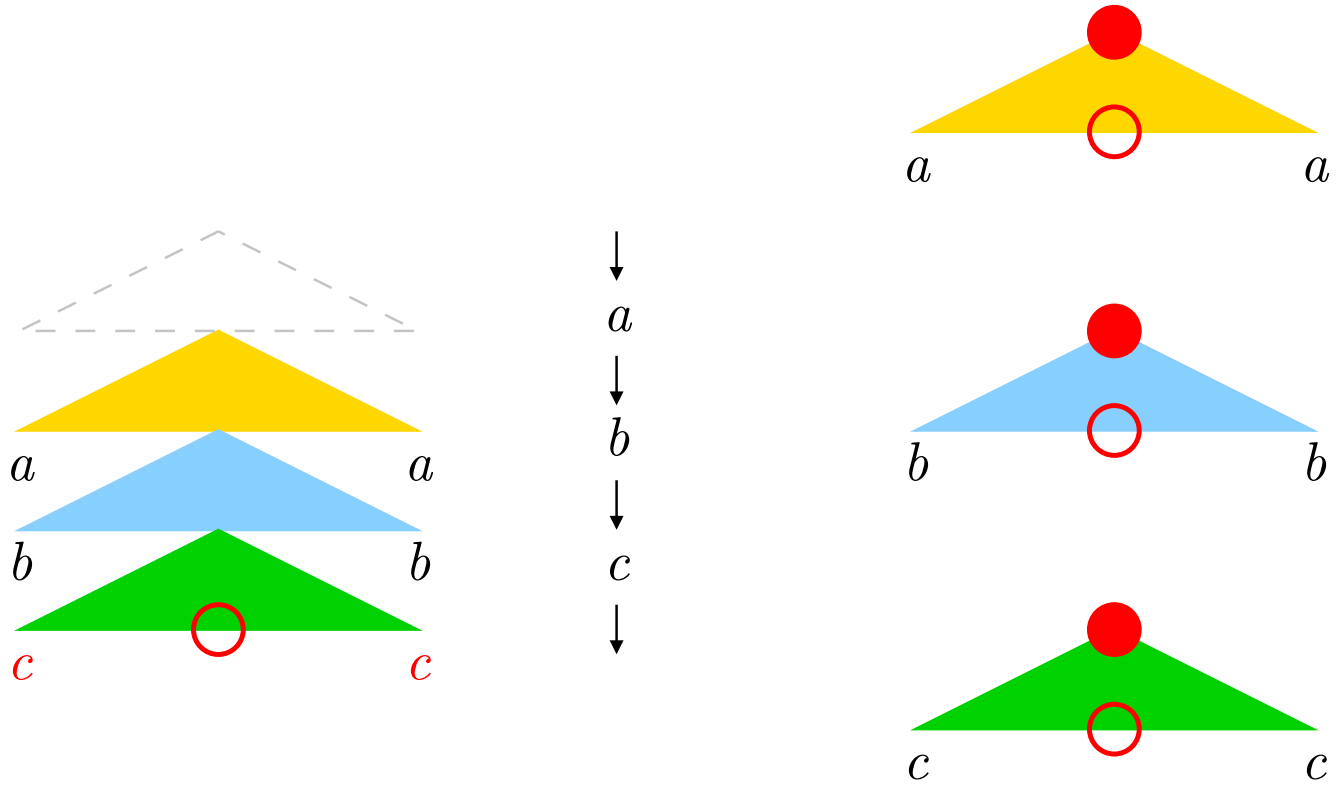
- CFGs generate *nested dependencies* between a string w and its reverse w^R

CFGs generate ww^R dependencies (3)



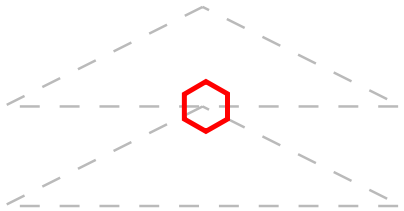
- CFGs generate *nested dependencies* between a string w and its reverse w^R

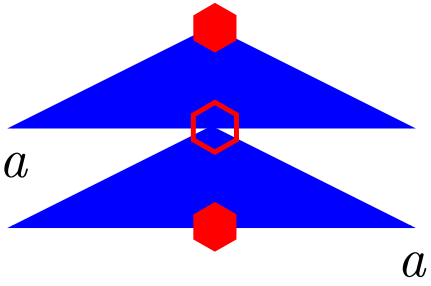
CFGs generate ww^R dependencies (4)

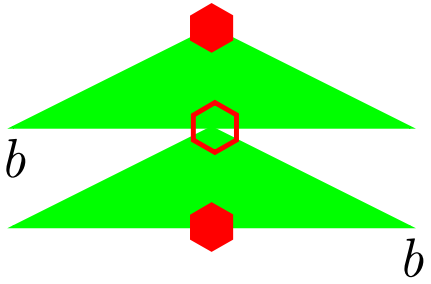


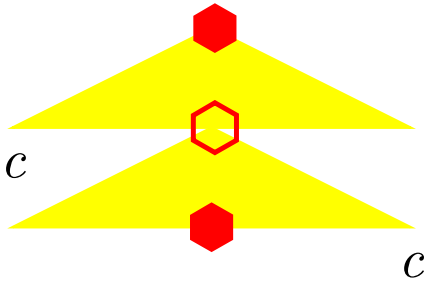
- CFGs generate *nested dependencies* between a string w and its reverse w^R

TAGs generate ww dependencies (1)

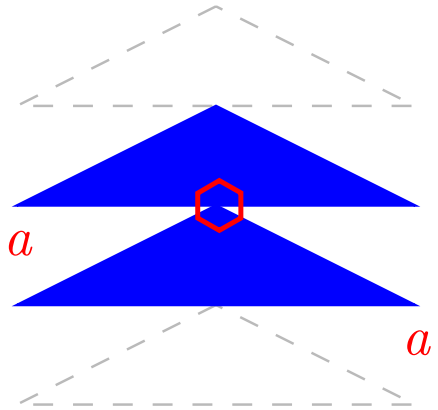


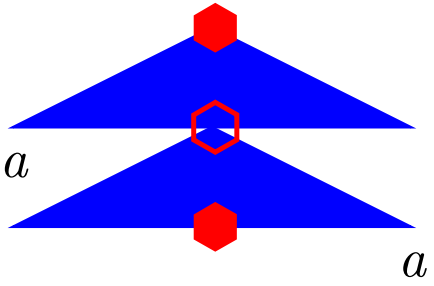


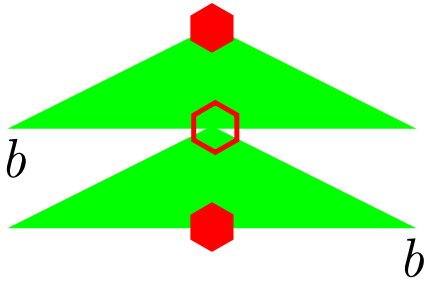


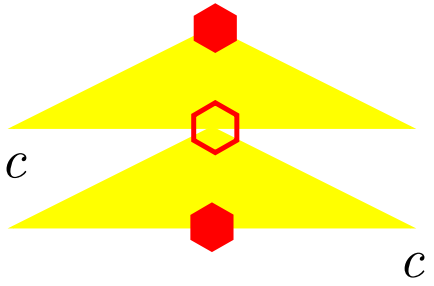


TAGs generate ww dependencies (2)

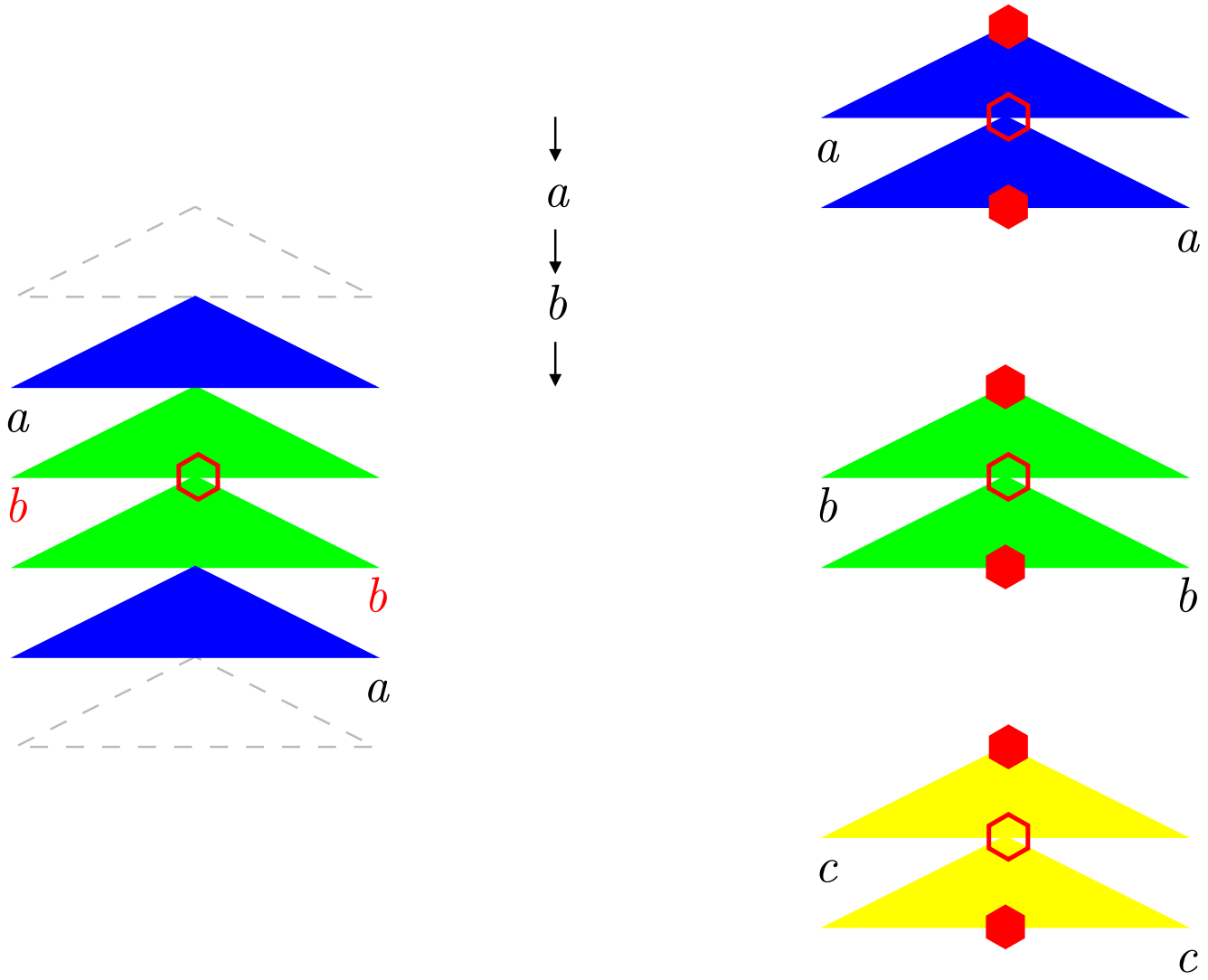




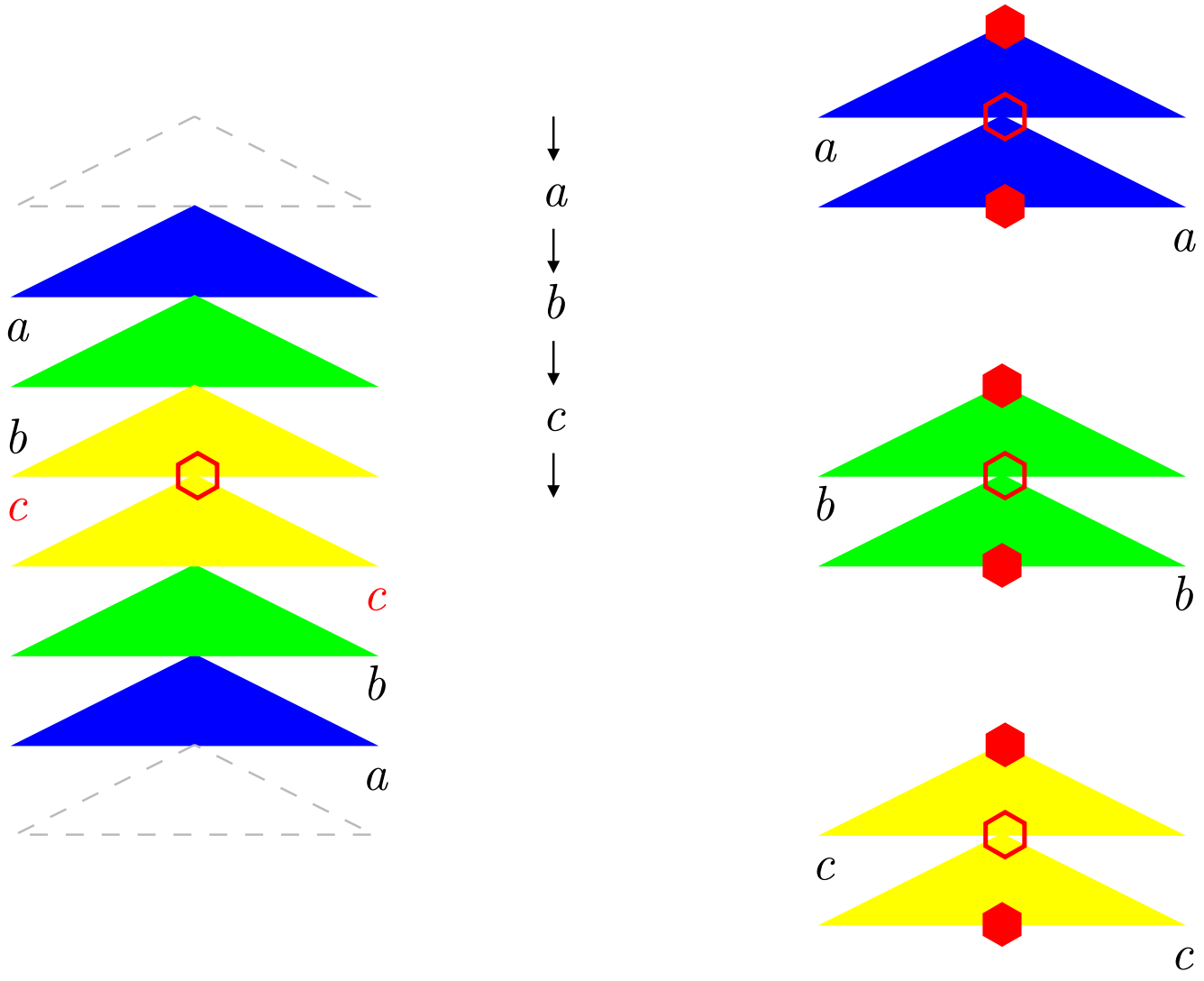




TAGs generate ww dependencies (3)



TAGs generate ww dependencies (4)



Derivation of *a flight* . . . (1)

- *a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday*

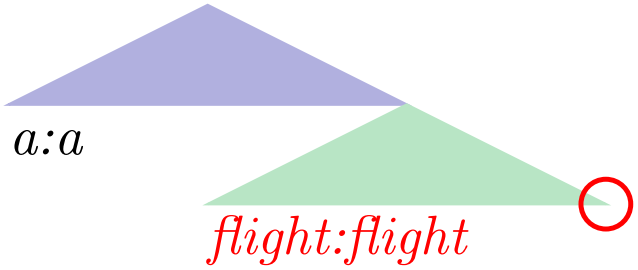
Derivation of *a flight . . .* (2)

a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday

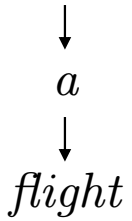
↓

a

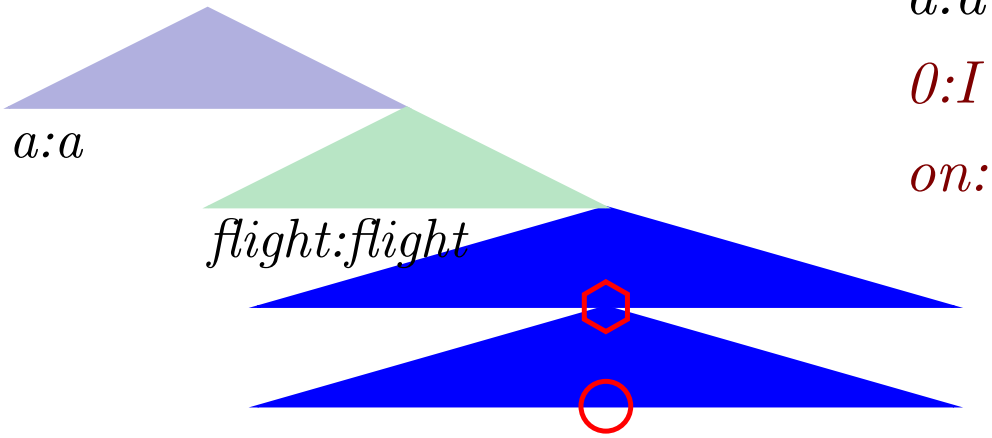
Derivation of *a flight . . .* (3)



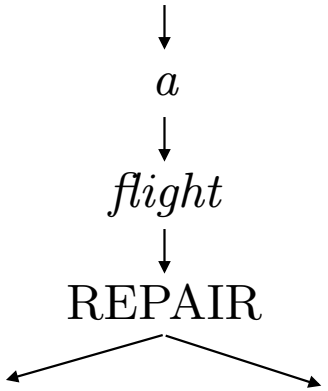
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



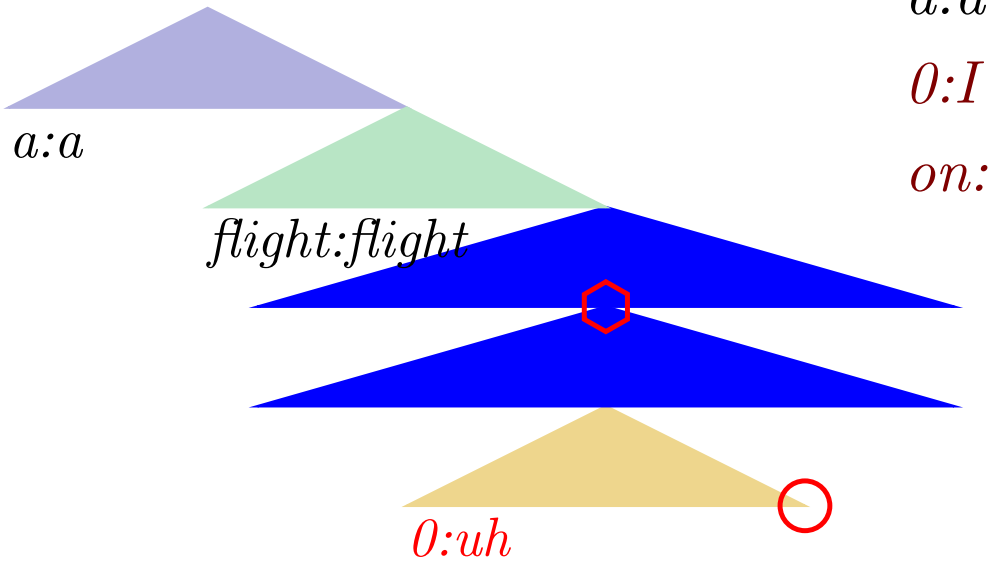
Derivation of *a flight . . .* (4)



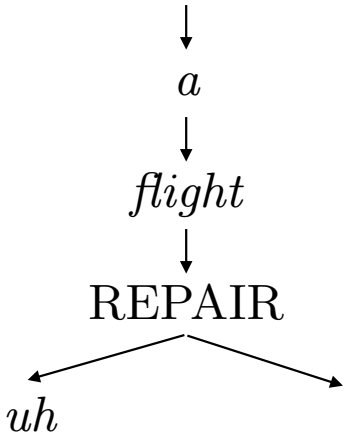
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



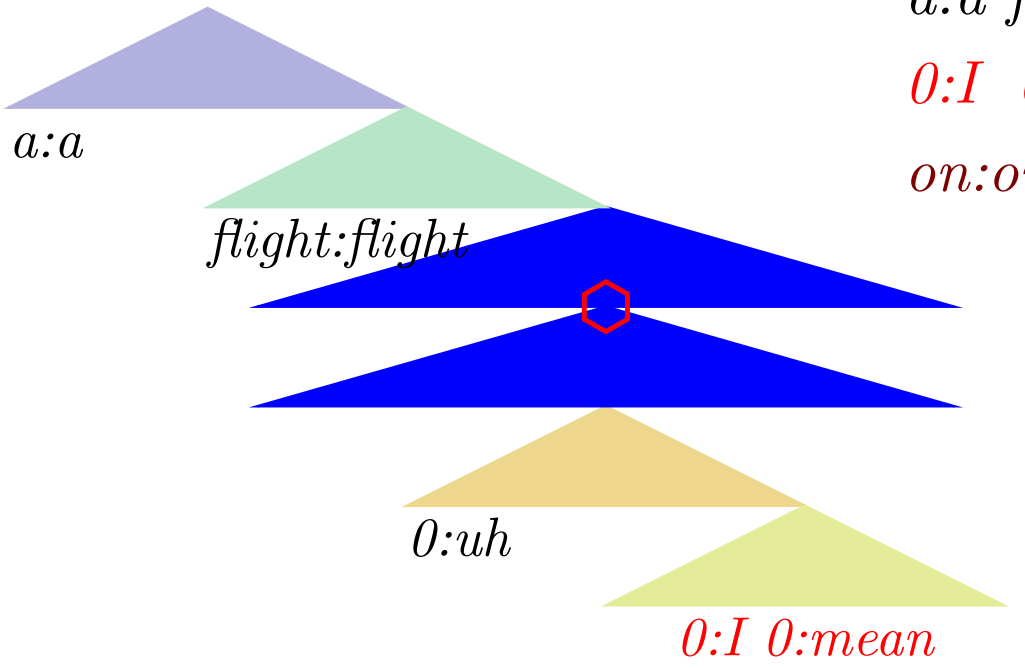
Derivation of *a flight . . .* (5)



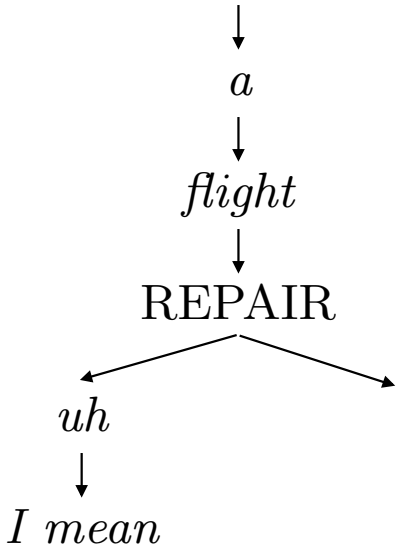
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



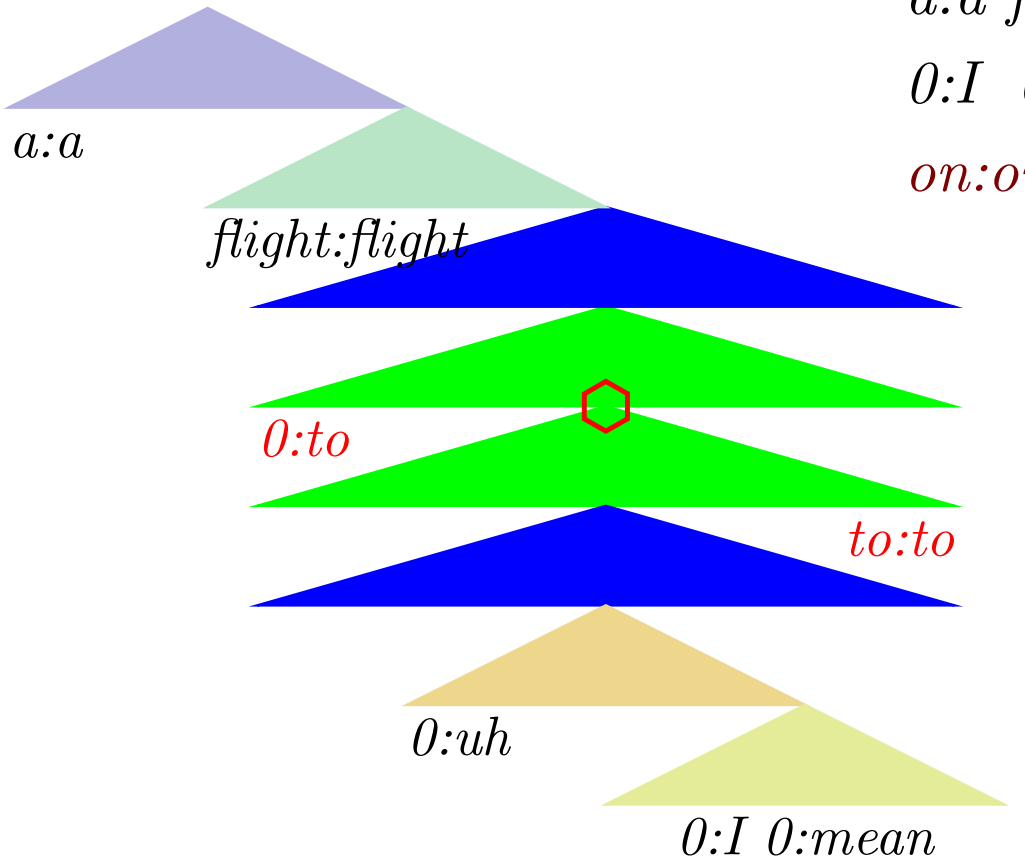
Derivation of *a flight . . .* (6)



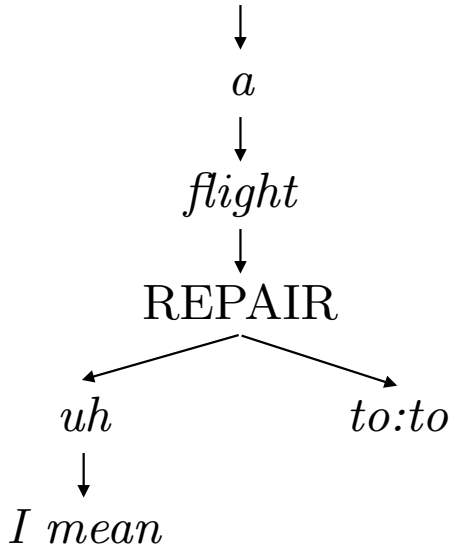
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



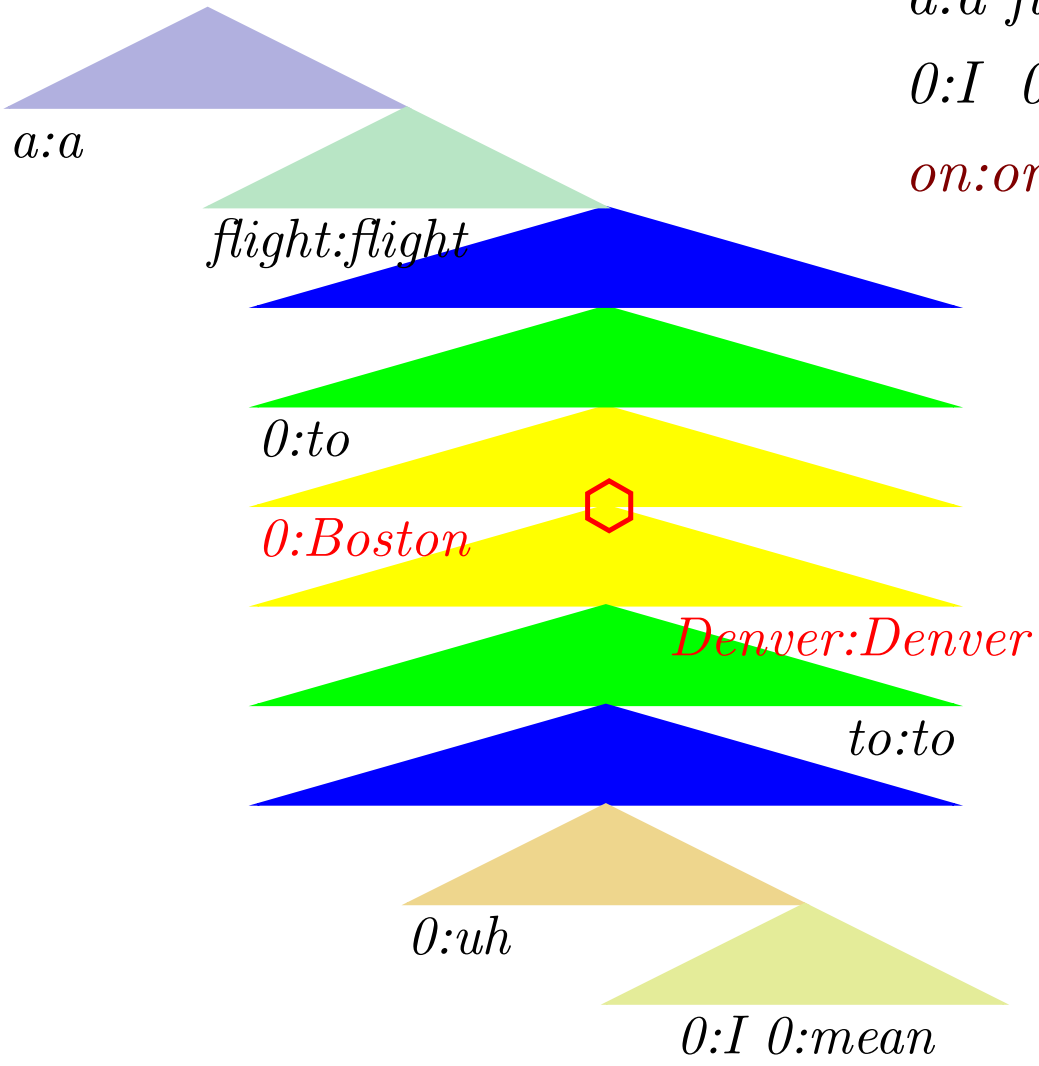
Derivation of *a flight . . .* (7)



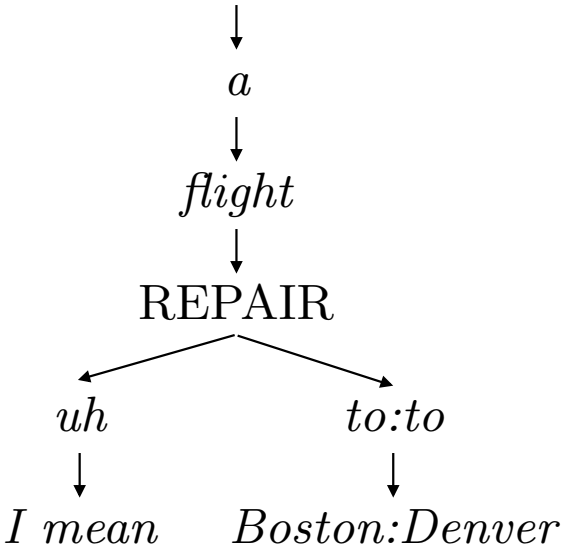
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



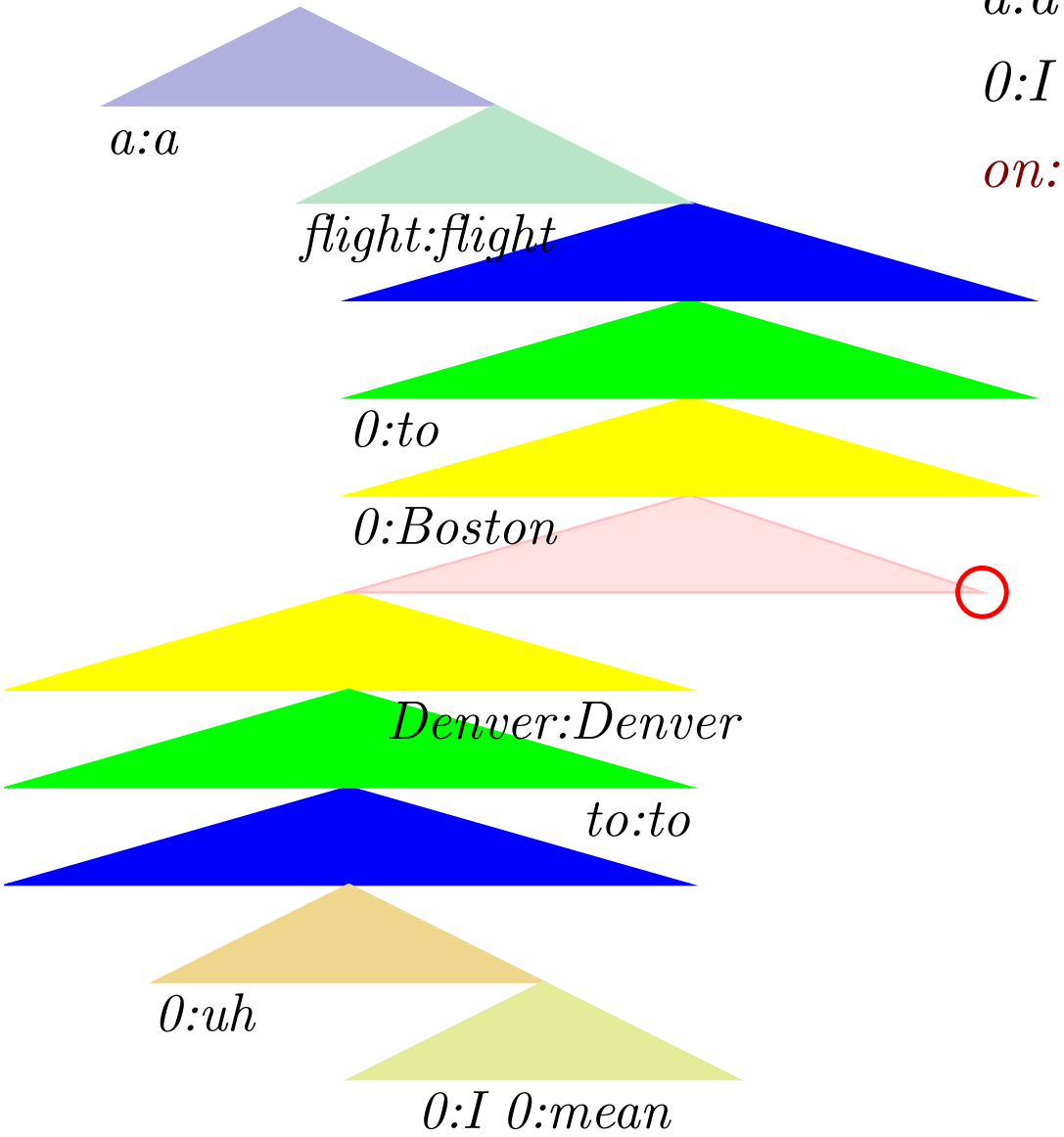
Derivation of *a flight . . .* (8)



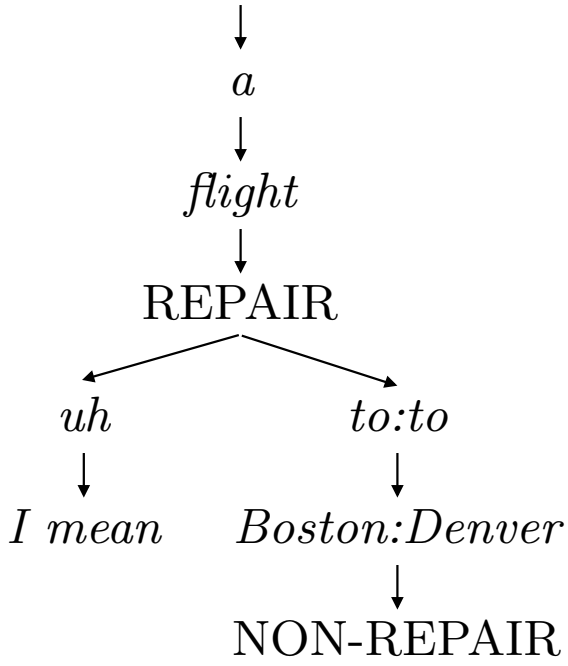
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



Derivation of *a flight . . .* (9)

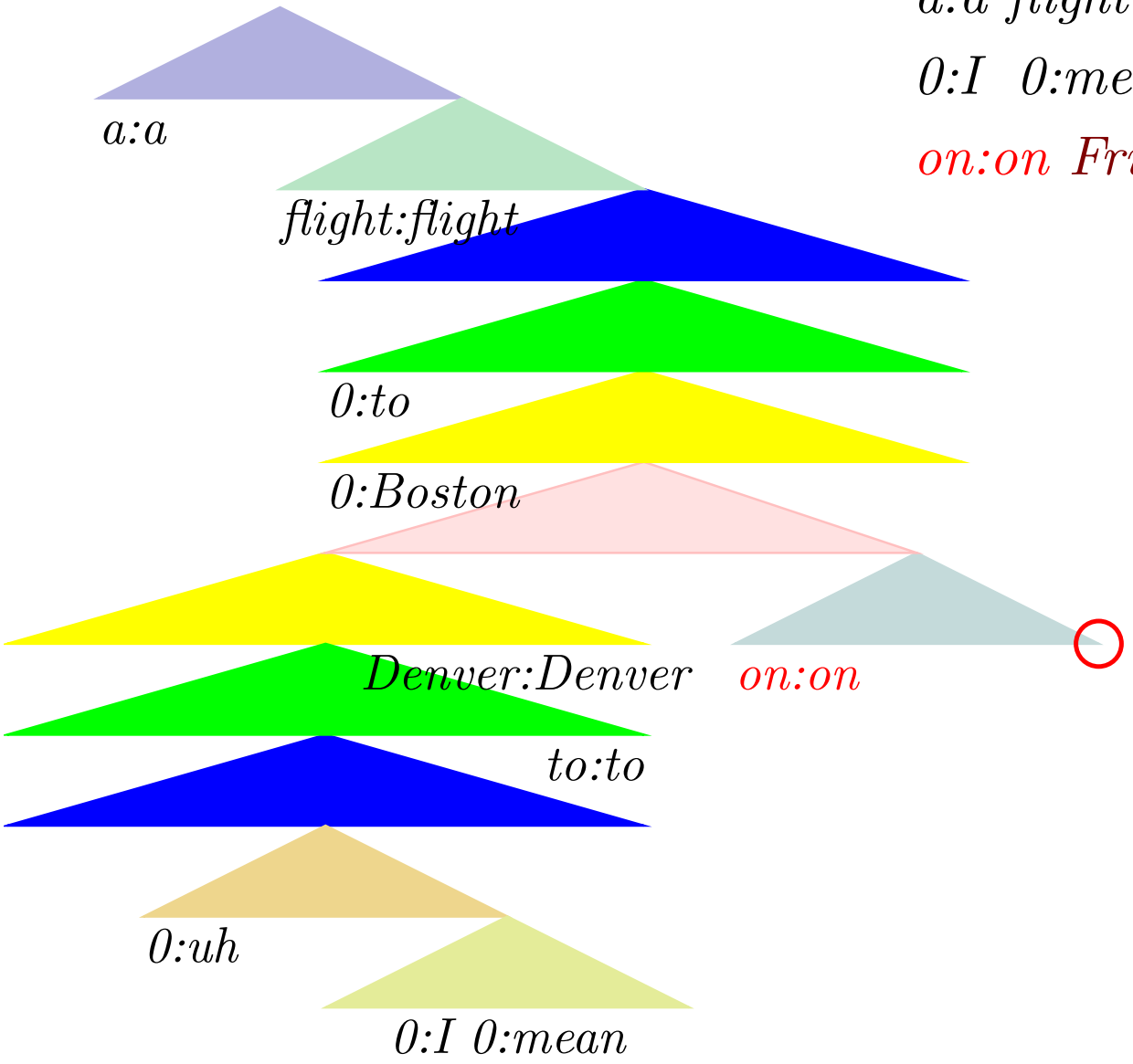


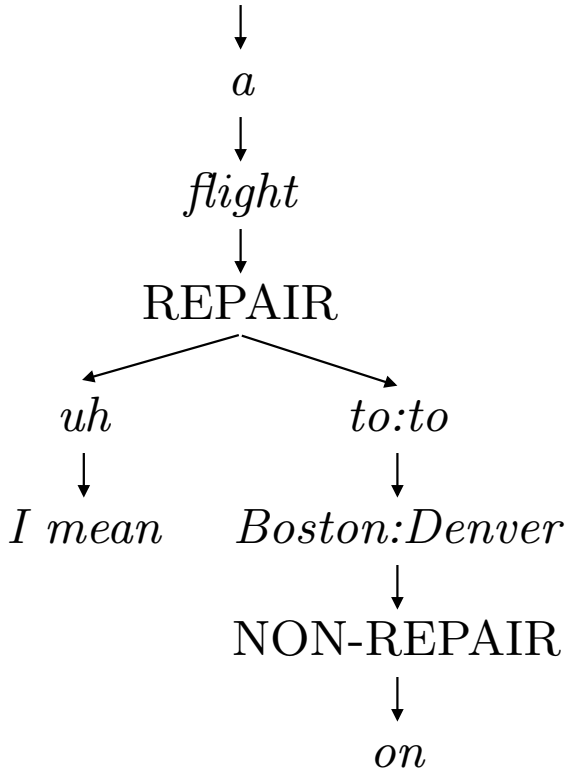
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



Derivation of *a flight . . .* (10)

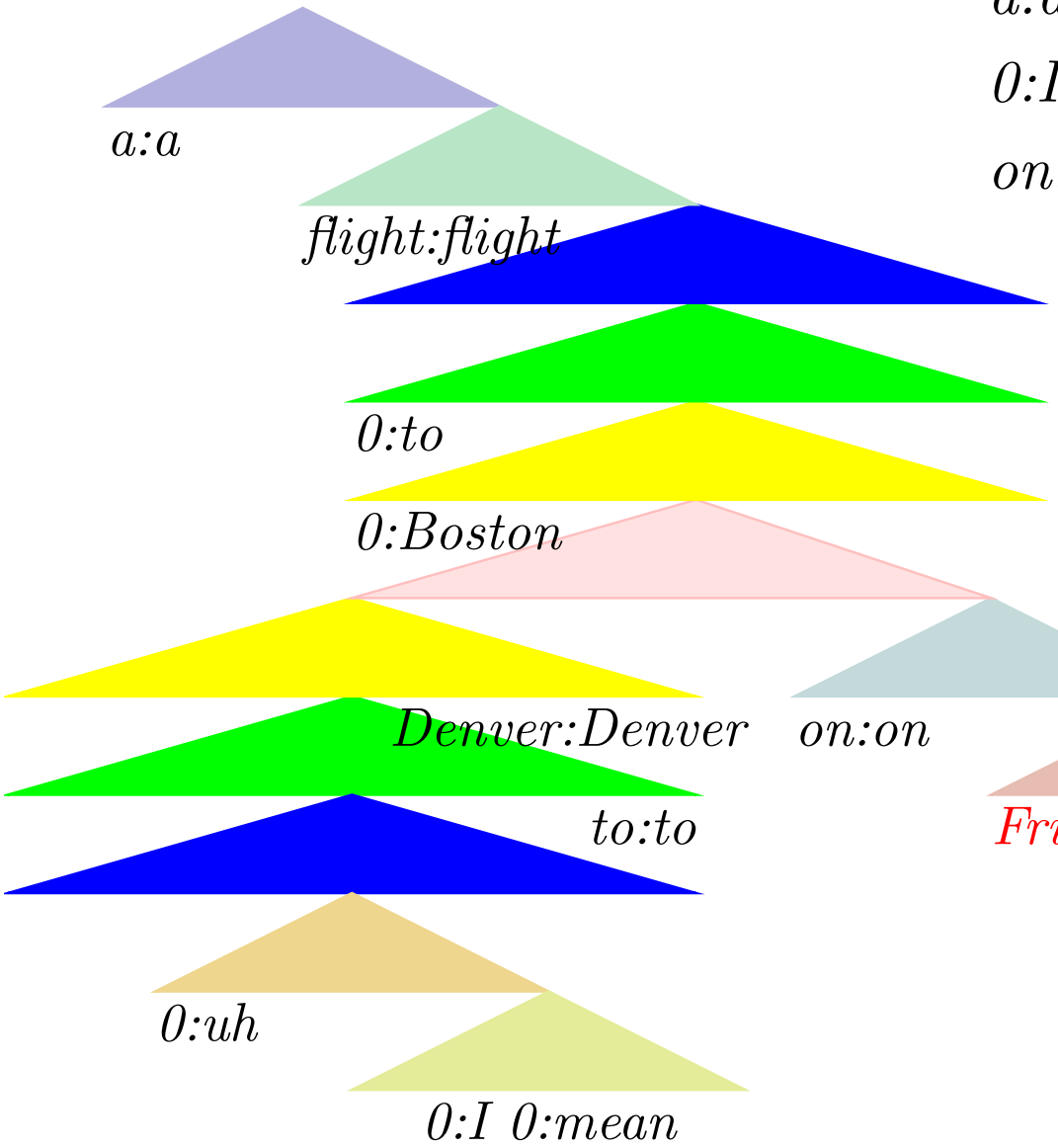
a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday

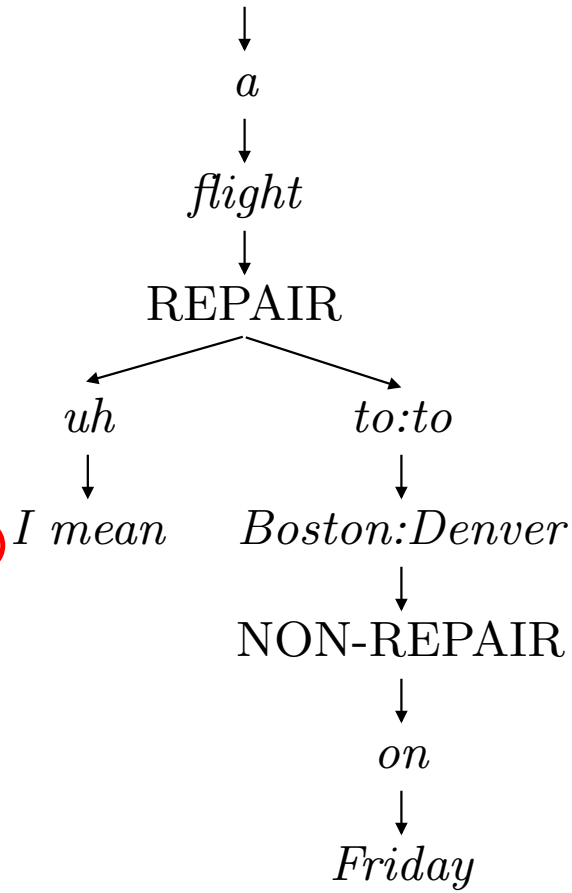




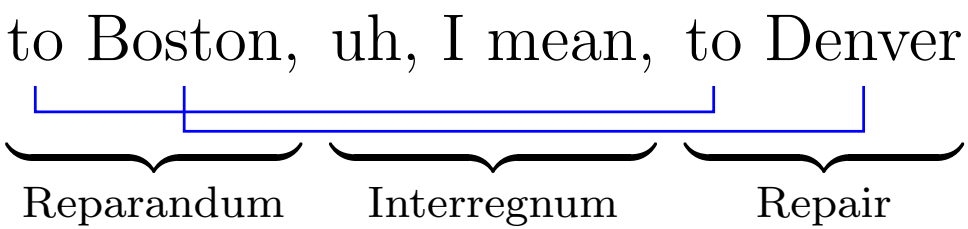
Derivation of *a flight . . .* (11)

a:a flight:flight 0:to 0:Boston 0:uh 0:I 0:mean to:to Denver:Denver on:on Friday:Friday



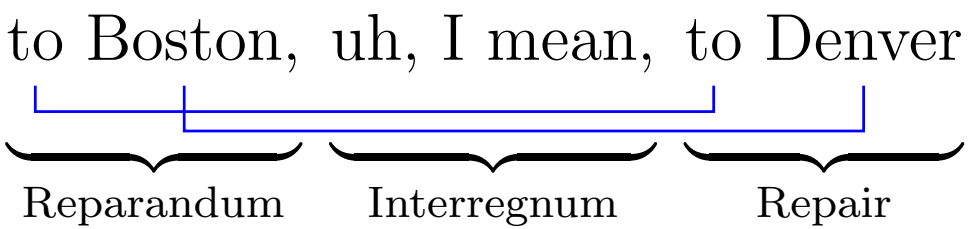


Training data (1)



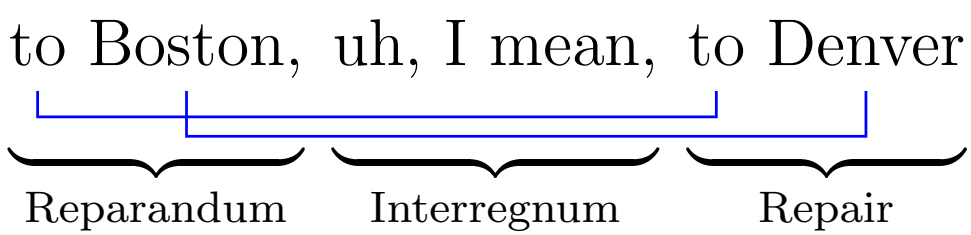
- Switchboard corpus annotates *reparandum, interregnum and repair*
- Trained on Switchboard files sw[23]*.dps (1.3M words)
- Punctuation and partial words ignored
- 5.4% of words are in a reparandum
- 31K repairs, average repair length 1.6 words
- Number of training words: reparandum 50K (3.8%), interregnum 10K (0.8%), repair 53K (4%), too complicated 24K (1.8%)

Training data (2)



- Reparandum and repair word-aligned by *minimum edit distance*
  - Prefers identity, POS identity, similar POS alignments
- Of the 57K alignments in the training data:
  - 35K (62%) are *identities*
  - 7K (12%) are *insertions*
  - 9K (16%) are *deletions*
  - 5.6K (10%) are *substitutions*
    - 2.9K (5%) are substitutions with same POS
    - 148 of 352 substitutions (42%) in heldout are not in training

Estimating the channel model



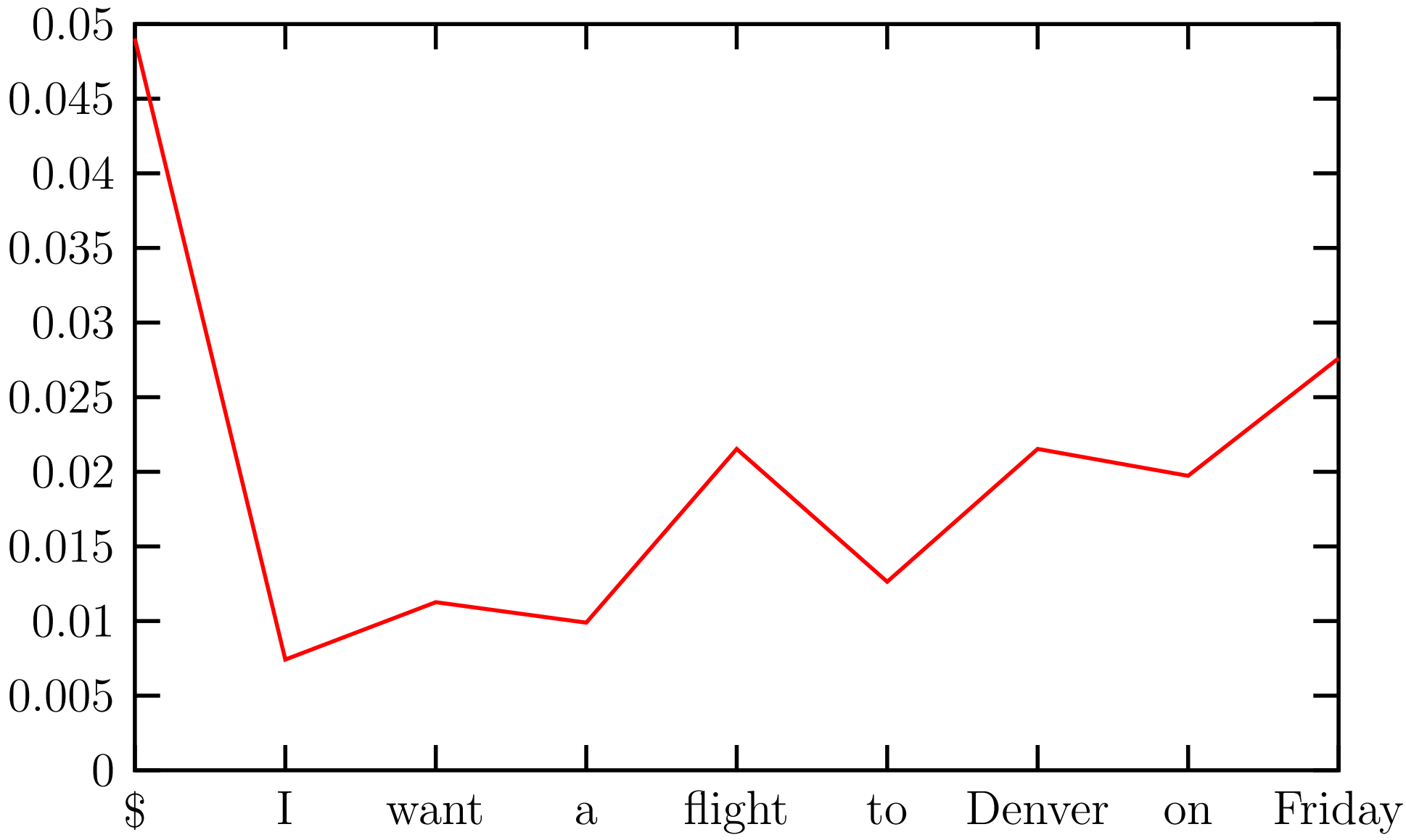
- Channel model is defined in terms of several simpler distributions:

  $\mathrm{P}_r(\mathsf{repair}|\mathit{flight})$: Probability of a repair starting after *flight*

  $\mathrm{P}_t(m|\mathit{Boston}, \mathit{Denver})$, where $m \in \{\mathsf{copy}, \mathsf{substitute}, \mathsf{insert}, \mathsf{delete}, \mathsf{end}\}$
  Probability of m after reparandum *Boston* and repair *Denver*

  $\mathrm{P}_m(\mathit{tomorrow}|\mathit{Boston}, \mathit{Denver})$: Probability that next reparandum word is *tomorrow*

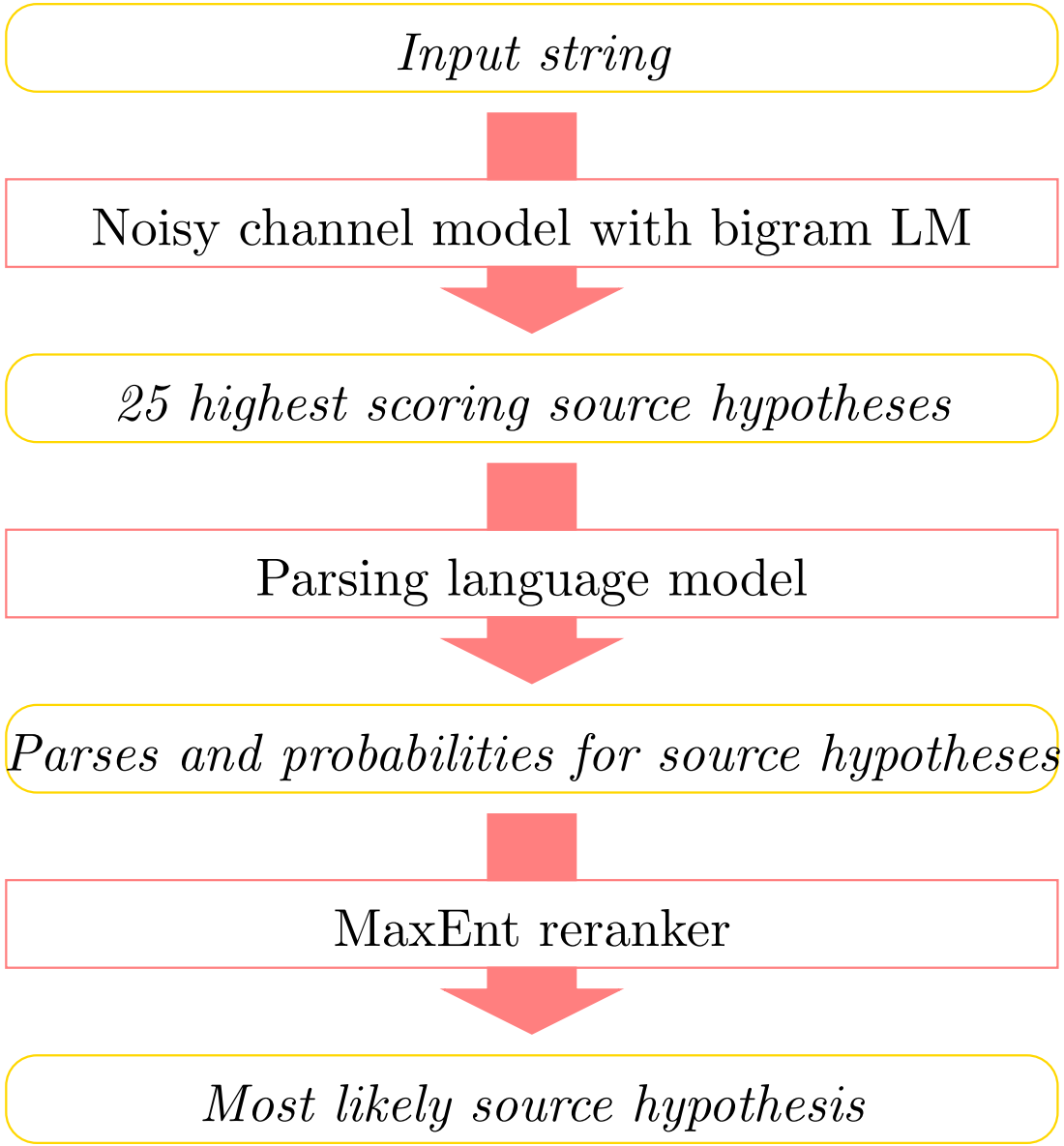
Estimated repair start probabilities



Implementation details (1)

- Don't know how to efficiently *search* for best analysis using Charniak parser LM
  ⇒ find 25-best hypothesized sources for each sentence using simpler *bigram* LM
- Recalculate probability of each hypothesized source using Charniak parsier LM
- Two different ways of combining channel and language model log probabilities
  - Add them (noisy channel model)
  - Use them as *features* in a machine learning algorithm
    ⇒ a *reranking* approach to finding best hypothesis

Implementation details (2)

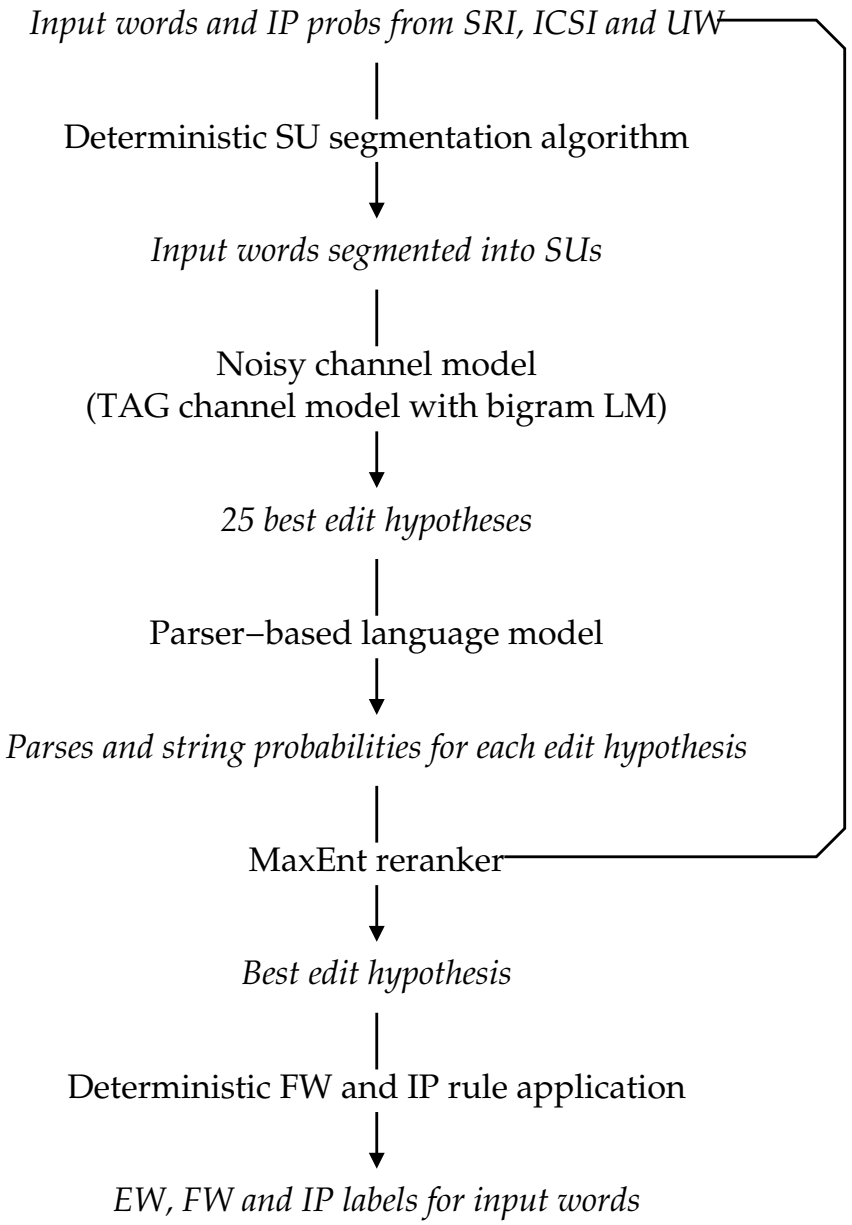


Evaluation of model's performance

	f-score	error rate
NCM + bigram LM	0.75	0.45
NCM + parser LM	0.81	0.35
MaxEnt reranker using NCM + parser LM	0.87	0.25
MaxEnt reranker alone	0.78	0.38

- Evaluated on an unseen portion of Switchboard corpus
- *f-score* is a geometric average of EDITED words precision and recall (bigger is better)
- *error rate* is the number of EDITED word errors made divided by number of true edited words (smaller is better)

RT04F competition



- RT04F evaluated *meta-data extraction*
- Test material was unsegmented speech
- ICSI, SRI and UW supplied us with ASR output, SU boundaries and acoustic IP probabilities

RT04F evaluation results

Task/error rate	Oracle words	ASR words
EDITED word detection	46.1	76.3
Filler word detection	23.7	40.0
Interruption point detection	28.6	55.9

- EDITED word detection used noisy channel reranker
- Filler word detection used *deterministic rules*
- Interruption point detection combined these two models

Evaluation of model's performance

Error rate on dev2 data	Oracle words	ASR words
Full model	0.525	0.773
– parsing model	0.55	0.790
– repair model	0.567	0.805
– prosodic features	0.541	0.772

- DARPA runs a competitive evaluation (RT04) of speech understanding systems
- EDITED word detection was one task in this evaluation
- Our system was not designed to deal with the RT04 data
  - our system assumes input is segmented into sentences

Conclusion and future work

- *Syntactic parsers make good language models*
- Grammars are useful for lots of things besides syntax!
- *Noisy channel model* can *combine very different kinds of models*
  - a lexicalized CFG model of syntactic structure
  - a TAG model of "rough copy" dependencies in speech repairs
- Modern *machine learning techniques* are very useful
  - can exploit *prosodic* and other kinds of information
- *Noisy channel model of robust language comprehension*
- Performs well in practice
- Future work:
  - Repair detection/correction in languages other than English
  - Semi-supervised and unsupervised training