Яндекс

Yandex Translate

Statistical Machine Translation

David Talbot

Statistical Machine Translation

- Noisy channel model

Statistical Machine Translation

- Noisy channel model
- Word alignments

Statistical Machine Translation

- Noisy channel model
- Word alignments
- Phrasal models

Statistical Machine Translation

- Noisy channel model
- Word alignments
- Phrasal models
- Log linear model

Statistical Machine Translation

- Noisy channel model
- Word alignments
- Phrasal models
- Log linear model
- Decoding

Noisy Channel Model of Sentence Translation

$$e^* = argmax_e \Pr(e) \Pr(f|e)$$

- Why not model $\Pr(e|f)$ directly?
- Why are approximations in $\Pr(f|e)$ less important than approximations in $\Pr(e|f)$?

Noisy Channel Model of Sentence Translation

$$e^* = argmax_e \Pr(e) \Pr(f|e)$$

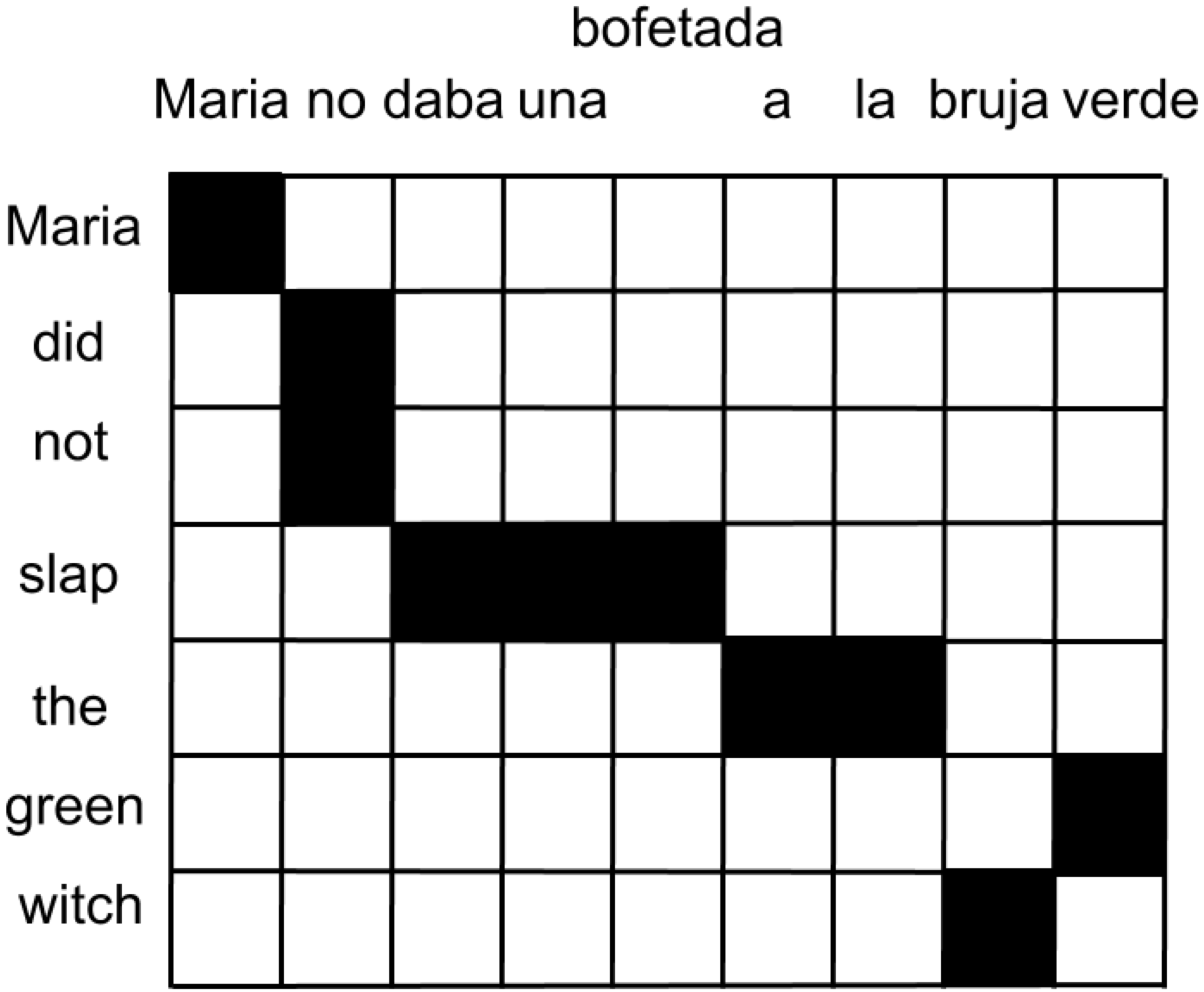
- How can we factorize $\Pr(f|e)$?
- How about $\Pr(f|e)$?

Word Alignments

- Latent variables not observed in training data
- Assume words are generated independently given alignments

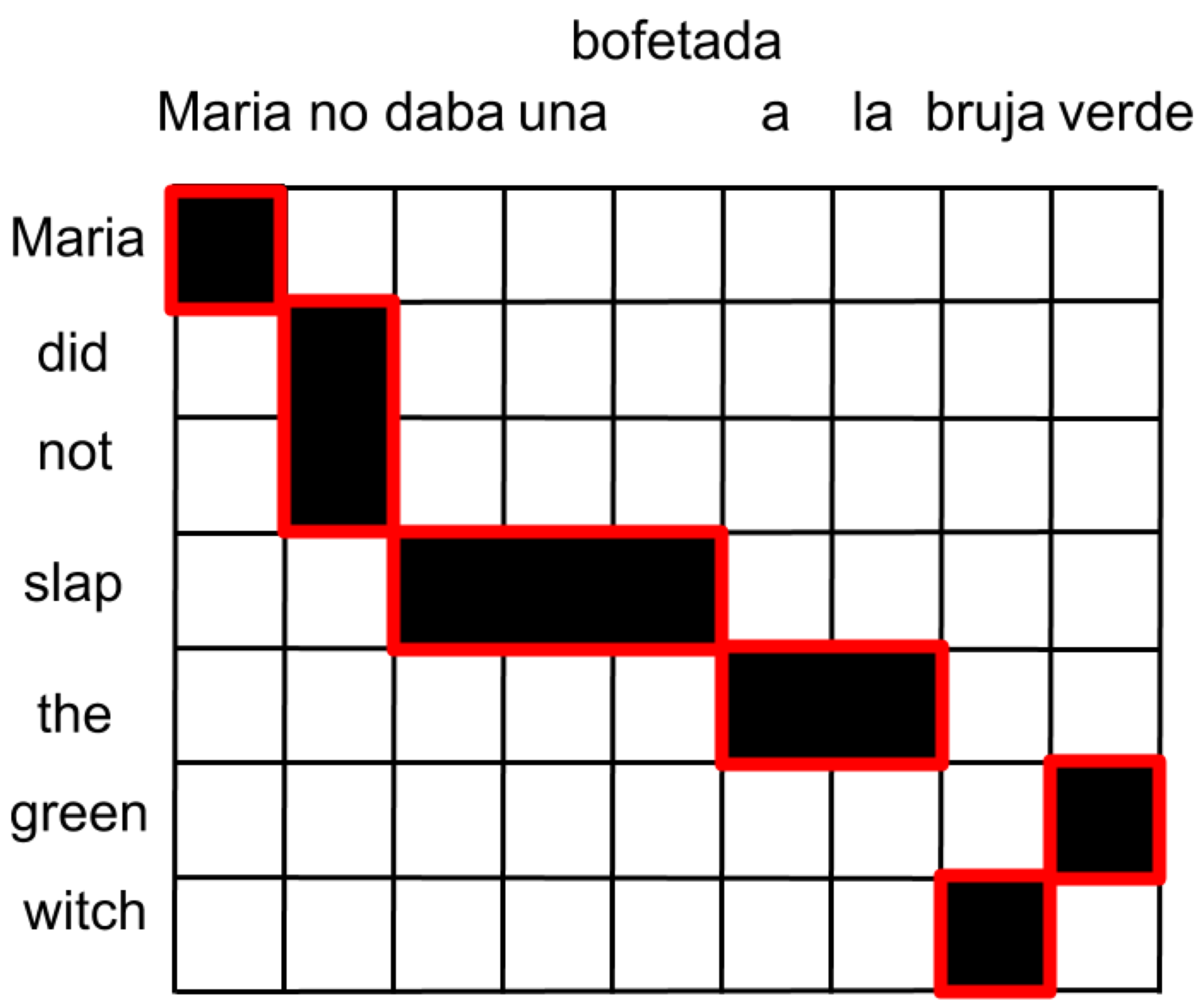
$$\Pr(f|e,a) \approx \prod_{j=1}^{J} \Pr\left(f_j \middle| e_{a_j}\right)$$

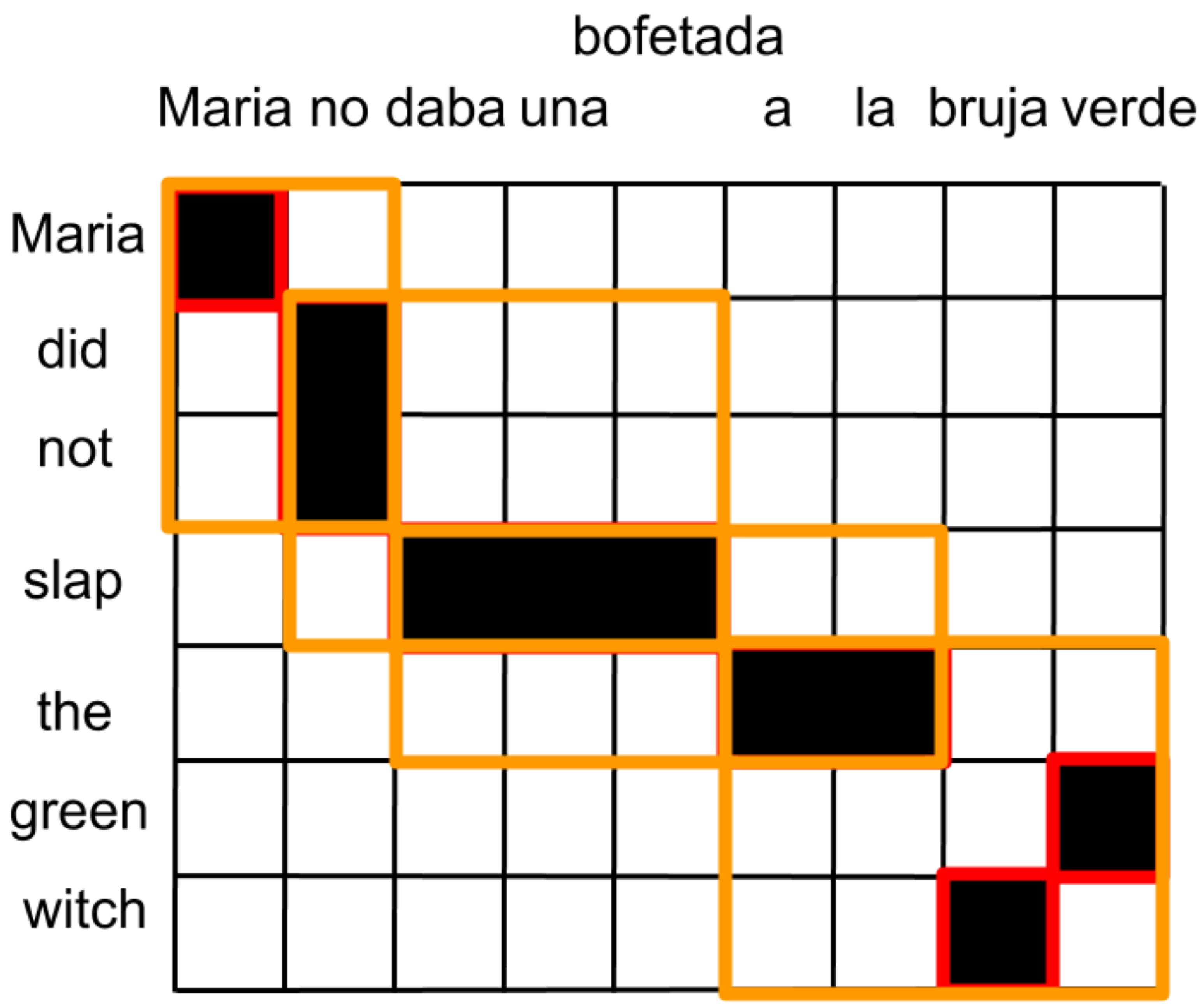
Word Alignments



From Words to Phrases

- Estimate word alignments using EM
- Use word alignments as constraints to align phrases
- Build phrasal model of $\Pr(f|e)$





Phrasal Translation Model

- Score phrase pairs based on counts of aligned phrase pairs
- Add word level scores to smooth these
- Add arbitrary features to phrase, e.g. $\Pr(e|f)$ in addition to $\Pr(f|e)$

Phrase-based Translation Model

He	$\longrightarrow$	Он
stood	$\longrightarrow$	стоял, стояла, поставил, …
bank	$\longrightarrow$	берега, берегу, банк, банка …
He stood	$\longrightarrow$	Он стоял
by the bank	$\longrightarrow$	на берегу, рядом с банком …

Log Linear Translation Model

$$e^* = argmax_e \Sigma_k \lambda_k \phi(e, f)$$

Log Linear Translation Model

- Arbitrary features: phrase-table, language model, length penalty, reordering costs, word-sense disambiguation, etc.

$$e^* = argmax_e \Sigma_k \lambda_k \phi(e, f)$$

Log Linear Translation Model

- Arbitrary features: phrase-table, language model, length penalty, reordering costs, word-sense disambiguation, etc.
- Move from generative model to discriminative model with generative models as features

$$e^* = argmax_e \Sigma_k \lambda_k \phi(e, f)$$

Log Linear Translation Model

- Arbitrary features: phrase-table, language model, length penalty, reordering costs, word-sense disambiguation, etc.
- Move from generative model to discriminative model with generative models as features
- Optimize evaluation metric (BLEU) directly with beam search on dev (MERT)

$$e^* = argmax_e \Sigma_k \lambda_k \phi(e, f)$$

Phrase-based Translation

English

He stood on the bank

Phrase-based Translation

Phrase Table

- English $\longrightarrow$ Pieces of Russian

He stood on the bank

{Он | стояла | на | берегу ... }

Он | стоял | у | банка, ... }

Он | поставил | у берега, ... }

Phrase-based Translation

Phrase Table Language Model

English ⟶ Pieces of Russian ⟶ Russian

He stood on the bank Он стоял на

{Он | стояла | на | берегу … }

Он | стоял | у | банка, … }

Он | поставил | у берега, …. }

Phrase-based Translation

	Phrase Table		Language Model	
English	$\longrightarrow$	Pieces of Russian	$\longrightarrow$	Russian

He stood on the bank

Он стоял на ...

{Он | стояла | на | берегу ... }

Он | стоял | у | банка, ... }

Он | поставил | у берега, ... }

Pr(Он стояла|He stood)	≅	Pr(Он|he) Pr(стояла|stood) ...	YES
Pr(Он стояла)	≅	Pr(Он) Pr(стояла|Он) ...	NO

Phrase Based Decoder

Problem: Find the highest scoring translation that translated all the input

Solution: Stack based decoding

- Start with an empty hypothesis
- Extend hypotheses by translating some (still) untranslated source words
- Backtrack from highest scoring hypothesis that translates all words

Phrase Based Decoder

Problem: Naïve search is exponential

Solution (1): Recombination

- Recombine hypotheses that are the same or equivalent under the model

1. Consist of the same words, e.g. ‘ab’ --> ‘AB’ vs ‘a’ --> ‘A’+ ’b’ --> ‘B’
2. Would be indistinguishable from this point (e.g. end with the same n-1 words)

Phrase Based Decoder

Problem: Naïve search is exponential

Solution (2): Beam search

- Store hypotheses on a stack
- Prune stack when its size goes beyond some threshold

Phrase Based Decoder

Problem: How to organize stacks

Solution (3): ?

(A) In a single stack

(B) By the number of words translated so far

(C) By the exact words translated so far

Phrase Based Decoder

Problem: How to organize stacks

Solution (3): ?

~~(A) In a single stack~~

(B) By the number of words translated so far

~~(C) By the exact words translated so far~~

Phrase Based Decoder

Problem: How to ‘fairly’ compare hypotheses that translated different words

Solution (4): ?

Phrase Based Decoder

Problem: How to ‘fairly’ compare hypotheses that translated different words

Solution (4): Assigning an estimate of the future cost

Phrase Based Decoder

Problem: How to ‘fairly’ compare hypotheses that translated different words

Solution (4): Assigning an estimate of the future cost

- Translation costs known (usually independent)
- Language model costs approximated (without context)
- Reordering costs ignored

Phrase Based Machine Translation

- Developed mostly in 2000s
- Resulted in a huge advance in MT quality
- Allowed launch of online MT services

What do you think its problems are?

Phrase Based Machine Translation

- Adequacy was okay
- Fluency was often horrible
- Reordering was a huge problem

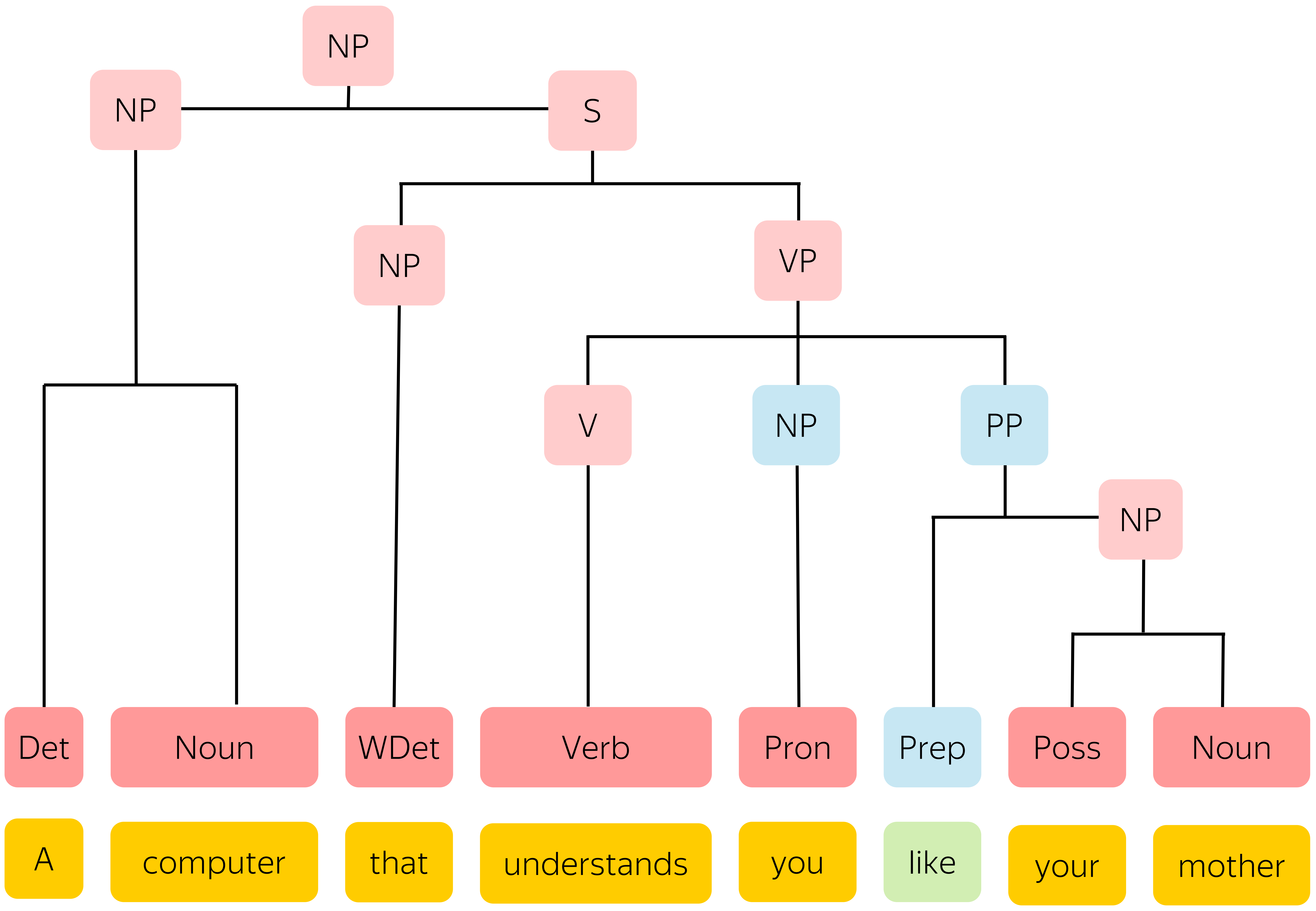
Phrase Based Machine Translation

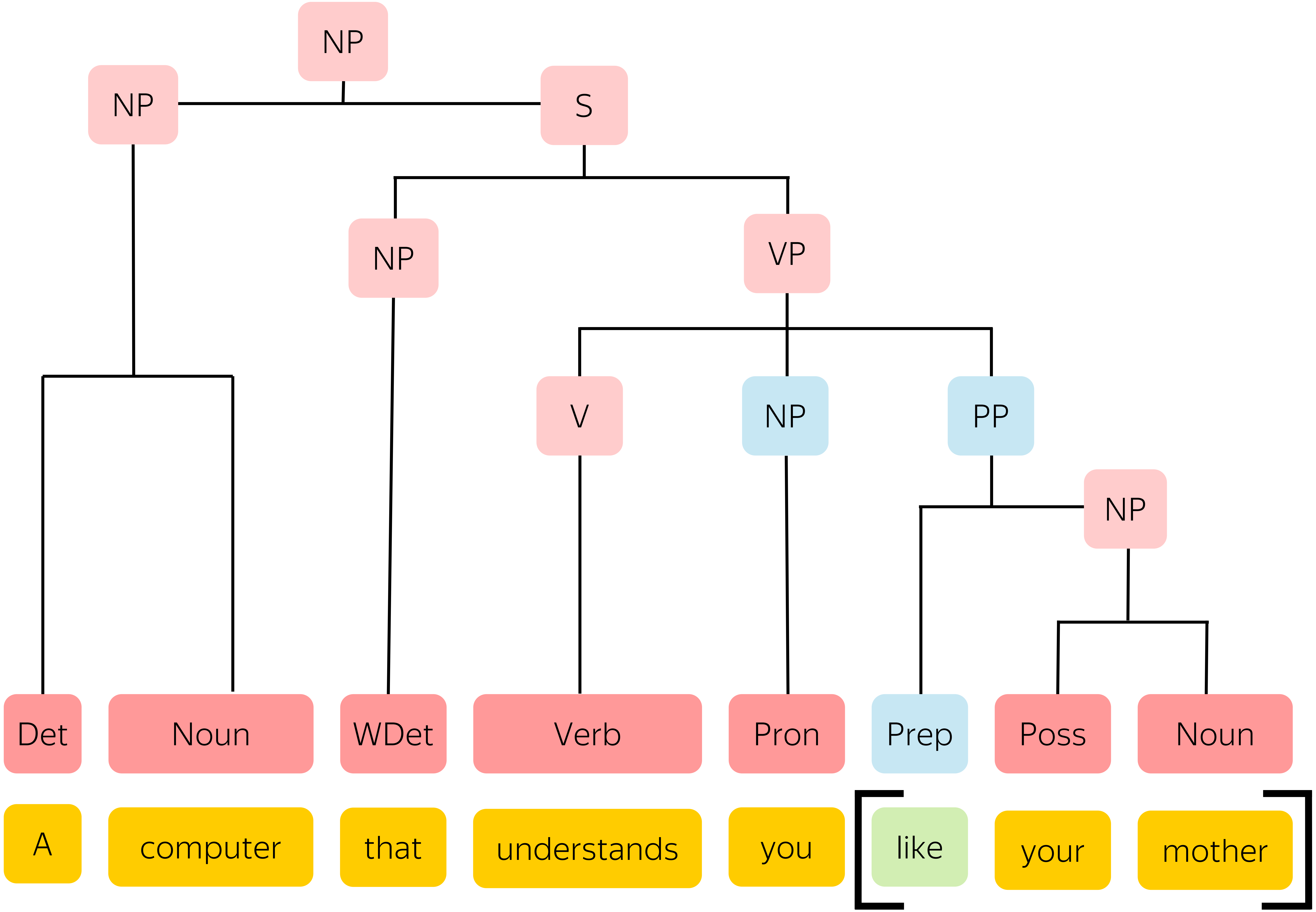
- Worked relatively well for close language pairs
- Worked relatively well if the target language is not rich in morphology
- Worked with surprisingly little data (compared to Neural MT)

Phrase Based Machine Translation++

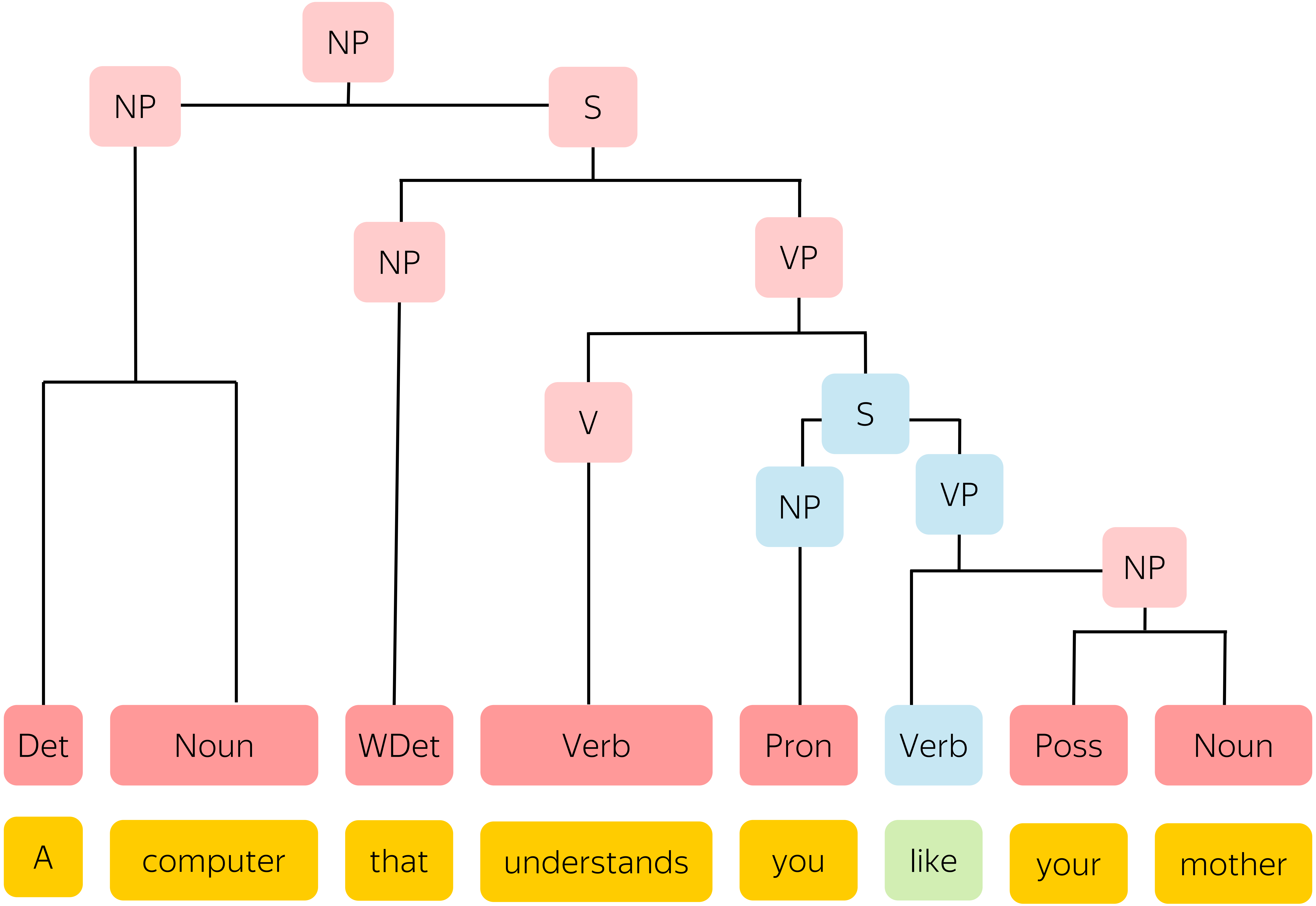
- Significant improvements from introducing syntax
- Reordering based on syntactic parse trees
- Disambiguation based on syntactic analysis

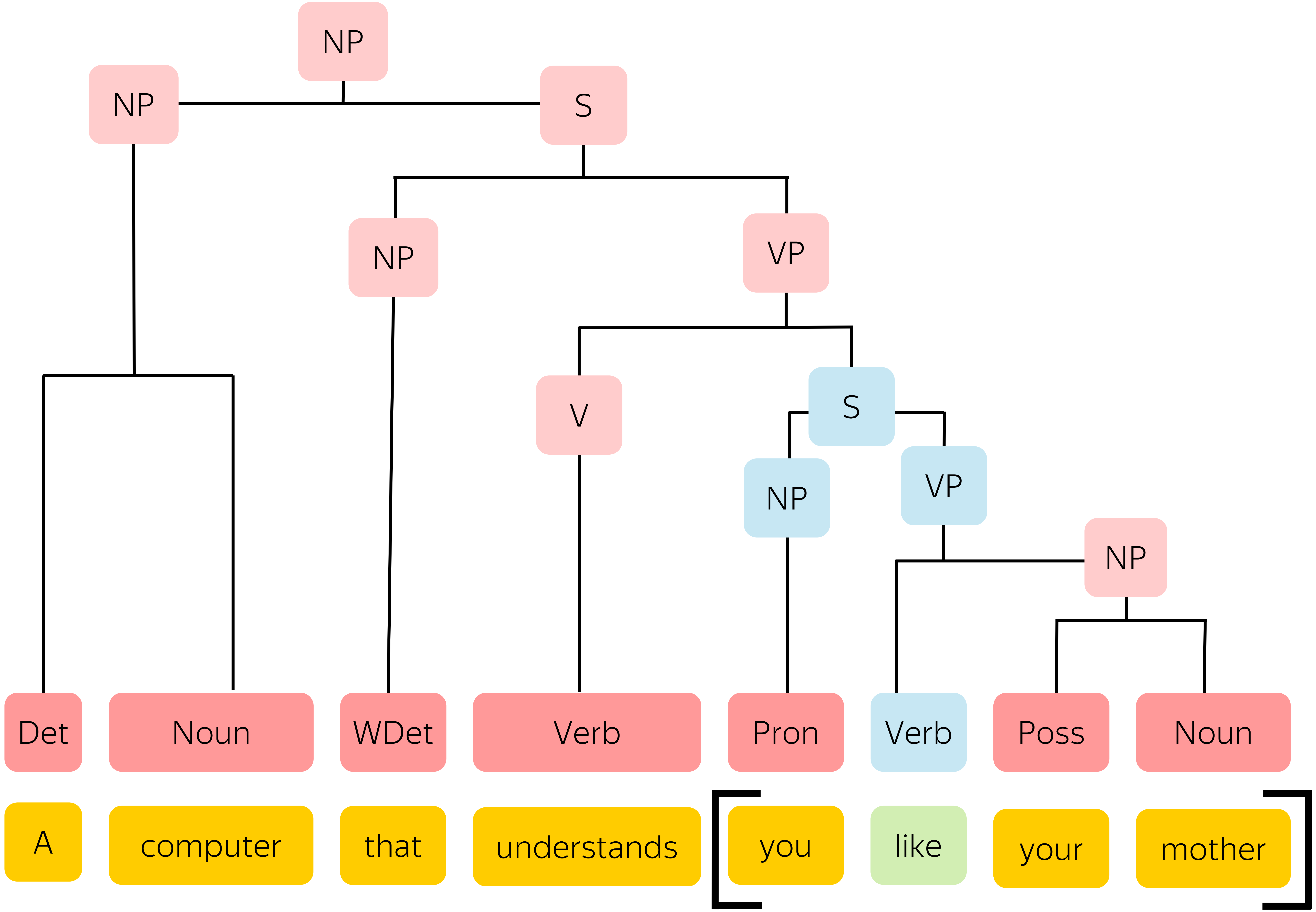
But generally required quite language specific annotations


NP
NP
S
NP
VP
V
NP
PP
NP
Det
Noun
WDet
Verb
Pron
Prep
Poss
Noun
A
computer
that
understands
you
like
your
mother



Компьютер, который понимает вас так же, **как ваша мама**.


NP
NP
S
NP
VP
V
S
NP
VP
NP
Det
Noun
WDet
Verb
Pron
Verb
Poss
Noun
A
computer
that
understands
you
like
your
mother



Компьютер, который понимает, что **вам нравится ваша мама.**

NLP components in Phrase-based MT++

- Word alignment
- Syntactic parser
- Reordering module
- Morphological analyzer/predictor

But impossible to optimize end-to-end