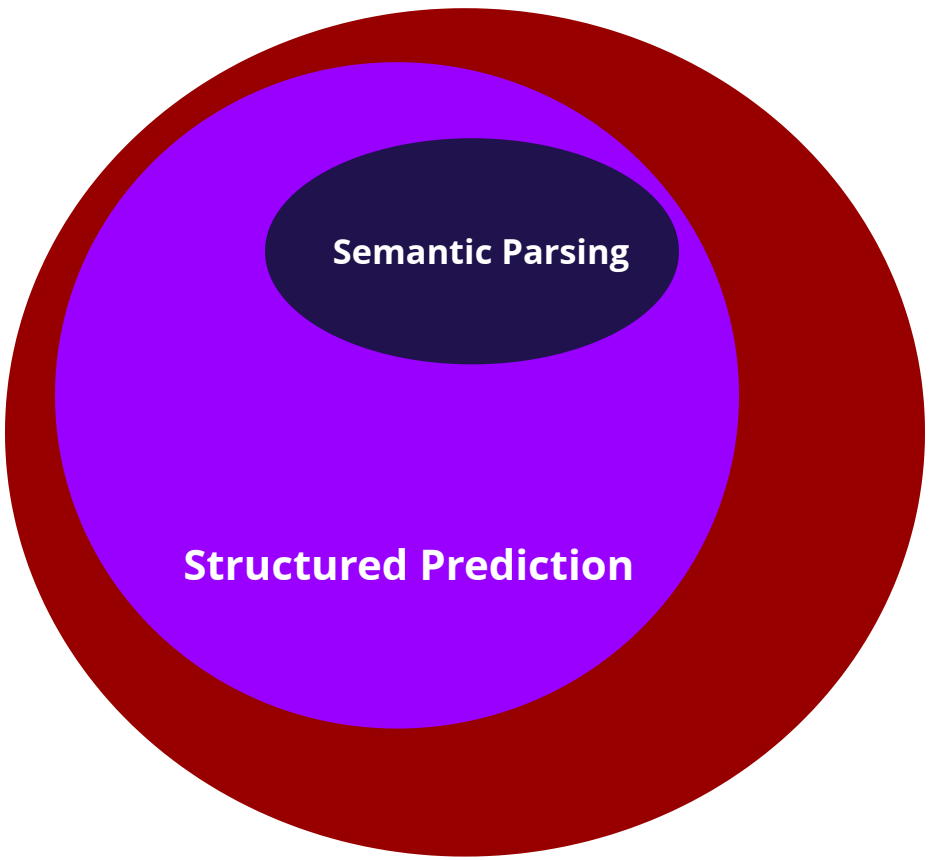
50.040
Natural Language Processing

Lu, Wei

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Tasks in NLP



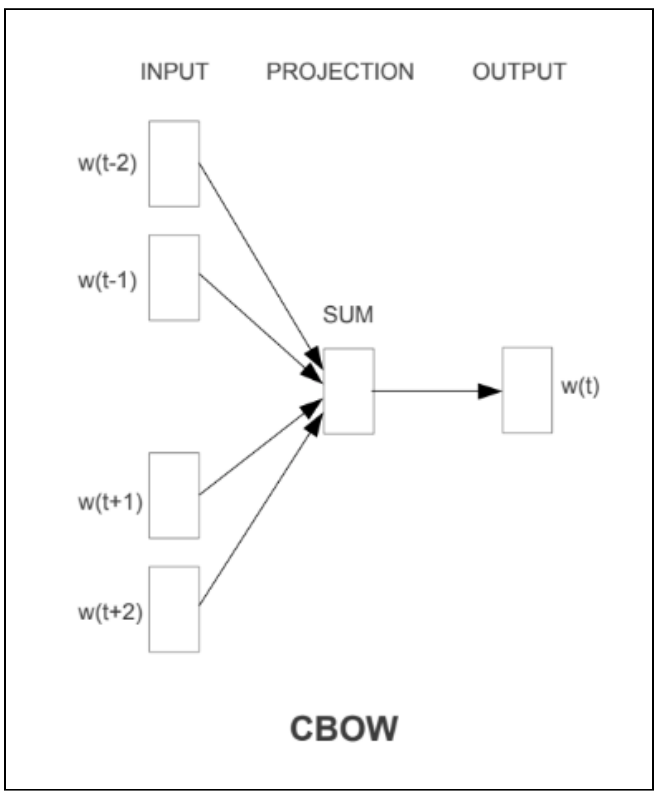
Semantic Parsing

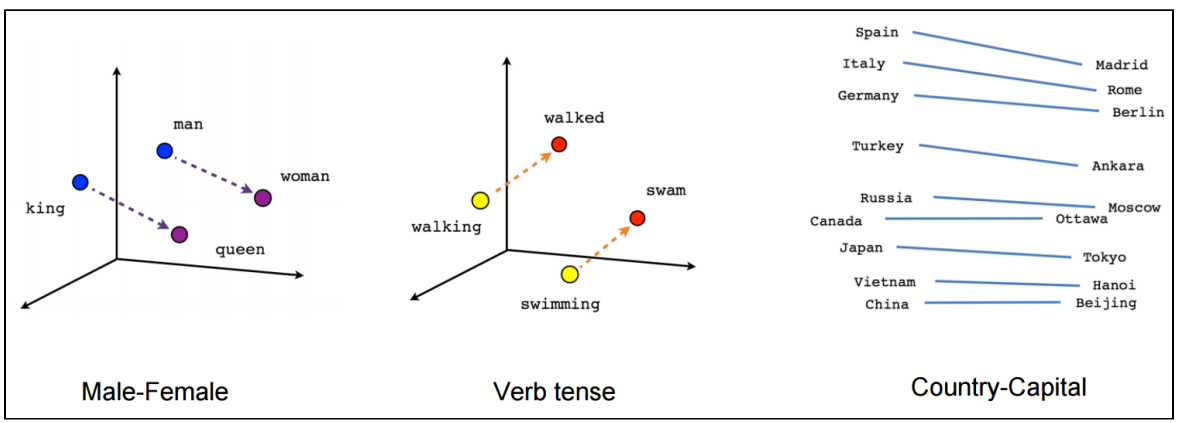
Parsing a sentence into its semantics

But… what is semantics?

Semantics?

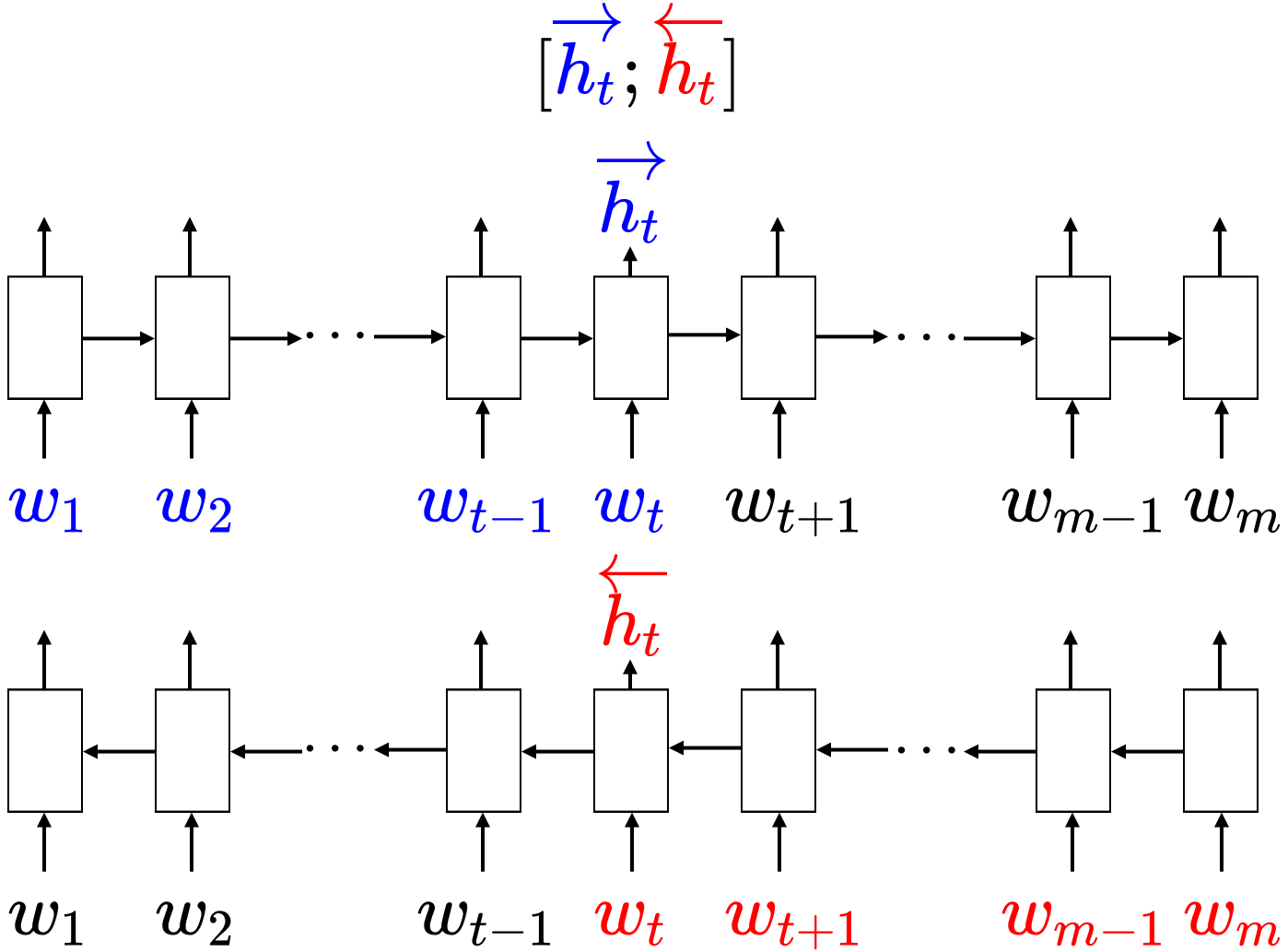
Word Embeddings





Semantics?

Contextual/Sentence Embeddings



Semantics?

“You can’t cram the meaning of a single $&!#* sentence into a single $!#&* vector!”

- Raymond Mooney

Semantics

Distributed representations for words, phrases, or sentences are typically learned in an unsupervised manner.

Thus, they may be used to capture some level of *relative semantics*, or *semantic proximity* (e.g., how semantically *similar* two words or two sentences are), but not *absolute semantics*.

Semantics

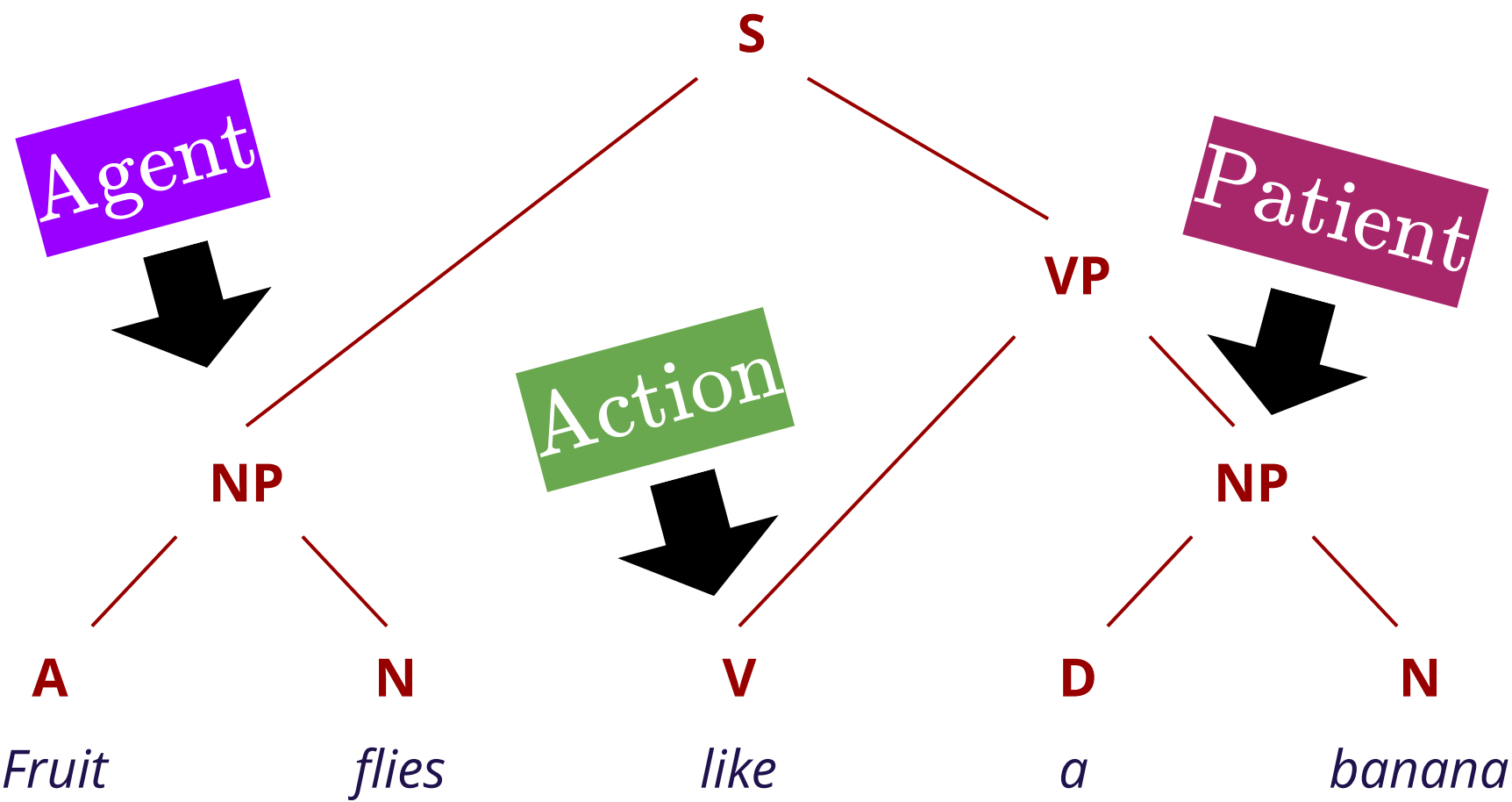
Distributed representations for words, phrases, or sentences are typically learned in an unsupervised manner.

Semantic representations must be defined by human. It is human that is giving or defining the *absolute semantics*!

Supervised Task!

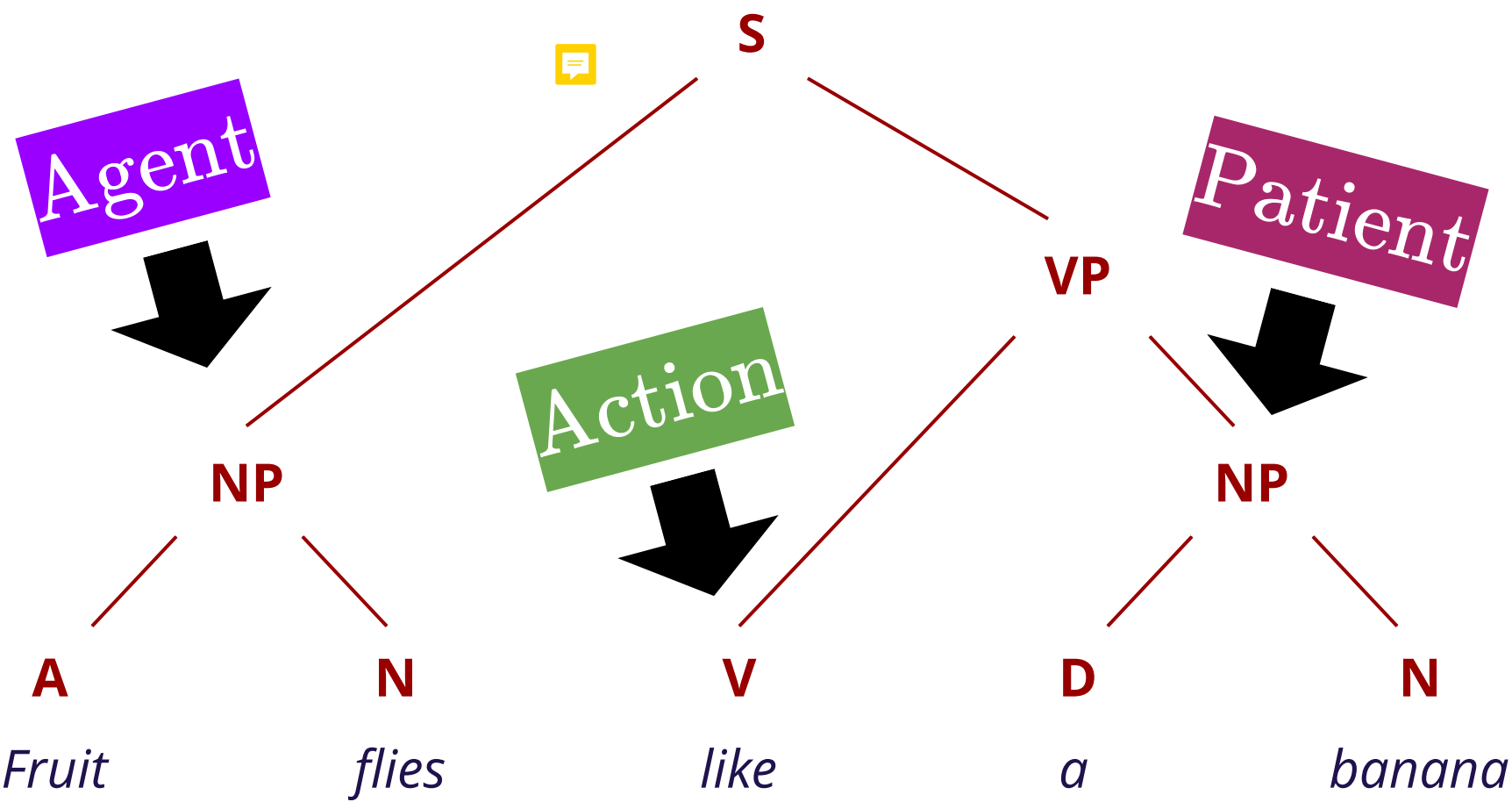
Semantic Role Labeling
(Shallow Semantic Parsing)

The process of assigning "semantic roles" into words and phrases in a sentence.

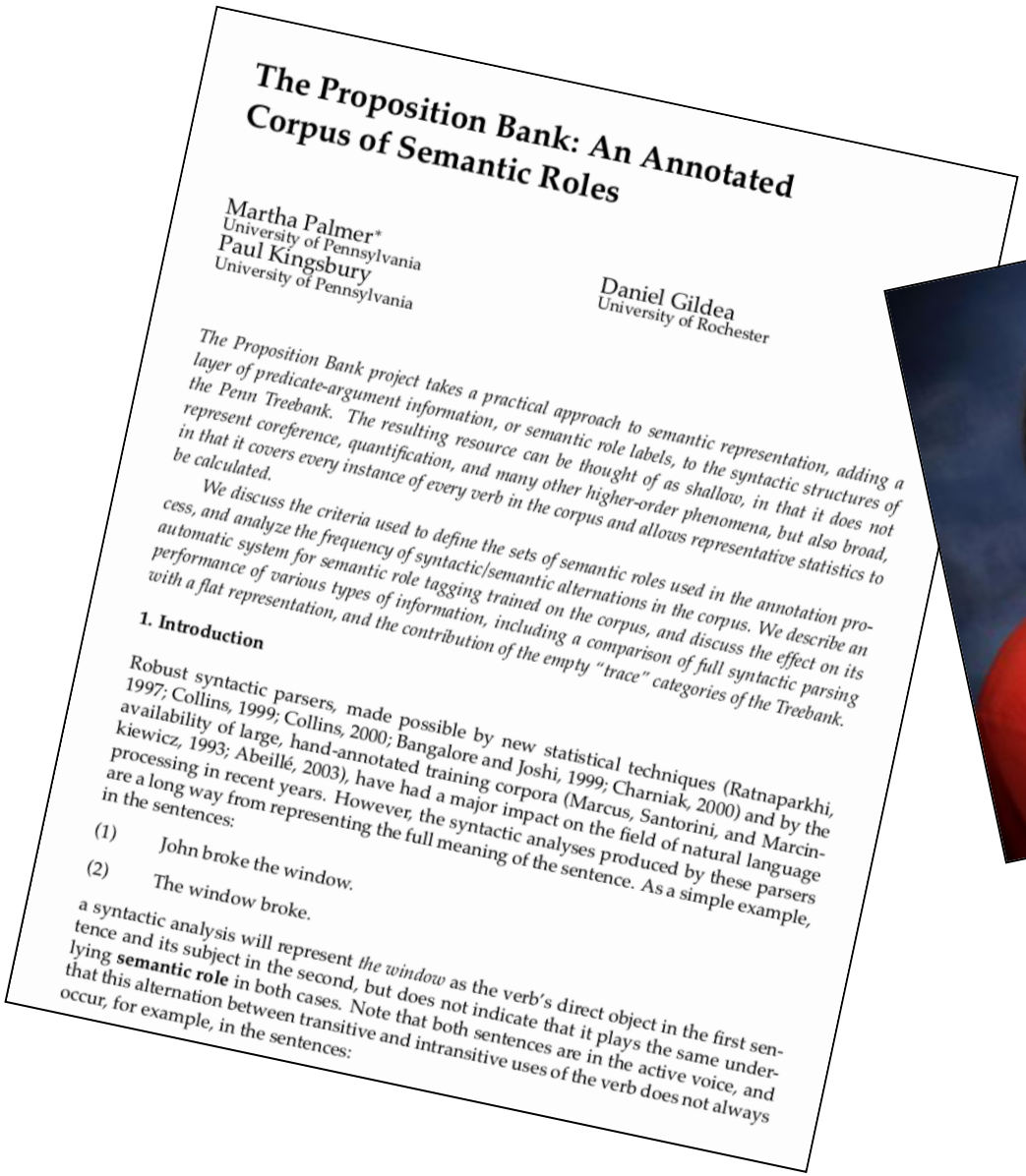


Semantic Role Labeling

Proposition $\text{like}(\text{“Fruit Flies”}, \text{“a banana”})$



Proposition Bank



The Proposition Bank: An Annotated Corpus of Semantic Roles

Martha Palmer*
University of Pennsylvania
Paul Kingsbury
University of Pennsylvania

Daniel Gildea
University of Rochester

The Proposition Bank project takes a practical approach to semantic representation, adding a layer of predicate-argument information, or semantic role labels, to the syntactic structures of the Penn Treebank. The resulting resource can be thought of as shallow, in that it does not represent coreference, quantification, and many other higher-order phenomena, but also broad, in that it covers every instance of every verb in the corpus and allows representative statistics to be calculated.

We discuss the criteria used to define the sets of semantic roles used in the annotation process, and analyze the frequency of syntactic/semantic alternations in the corpus. We describe an automatic system for semantic role tagging trained on the corpus, and discuss the effect on its performance of various types of information, including a comparison of full syntactic parsing with a flat representation, and the contribution of the empty "trace" categories of the Treebank.

1. Introduction

Robust syntactic parsers, made possible by new statistical techniques (Ratnaparkhi, 1997; Collins, 1999; Collins, 2000; Bangalore and Joshi, 1999; Charniak, 2000) and by the availability of large, hand-annotated training corpora (Marcus, Santorini, and Marcinkiewicz, 1993; Abeillé, 2003), have had a major impact on the field of natural language processing in recent years. However, the syntactic analyses produced by these parsers are a long way from representing the full meaning of the sentence. As a simple example, in the sentences:

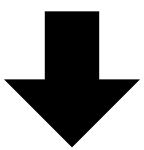
(1) John broke the window.

(2) The window broke.

a syntactic analysis will represent *the window* as the verb's direct object in the first sentence and its subject in the second, but does not indicate that it plays the same underlying **semantic role** in both cases. Note that both sentences are in the active voice, and that this alternation between transitive and intransitive uses of the verb does not always occur, for example, in the sentences:

Proposition Bank

```
(S (NP-SBJ Analysts)
 (VP have
  (VP been
   (VP expecting
    (NP (NP a GM-Jaguar pact)
        (SBAR (WHNP-1 that)
           (S (NP-SBJ *T*-1)
                (VP would
              (VP give
                      (NP the U.S. car maker)
               (NP (NP an eventual (ADJP 30 %) stake)
                (PP-LOC in (NP the British company)))))))))))
```



```
(S Arg0 (NP-SBJ Analysts)
 (VP have
  (VP been
   (VP expecting
    Arg1 (NP (NP a GM-Jaguar pact)
        (SBAR (WHNP-1 that)
           (S Arg0 (NP-SBJ *T*-1)
                (VP would
              (VP give
               Arg2 (NP the U.S. car maker)
                 Arg1 (NP (NP an eventual (ADJP 30 %) stake)
                (PP-LOC in (NP the British company)))))))))))
```

Proposition Bank

```
(S Arg0 (NP-SBJ Analysts)
 (VP have
  (VP been
   (VP expecting
    Arg1 (NP (NP a GM-Jaguar pact)
         (SBAR (WHNP-1 that)
           (S Arg0 (NP-SBJ *T*-1)
                (VP would
              (VP give
               Arg2 (NP the U.S. car maker)
                 Arg1 (NP (NP an eventual (ADJP 30 %) stake)
                  (PP-LOC in (NP the British company)))))))))))
```

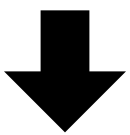
Propositions

expect(Analysts, GM-J pact)
give(GM-J pact, US car maker, 30% stake)

Semantic Role Labeling

Argument Identification

1. Extract features from sentence, syntactic parse, and other sources for each candidate constituent
2. Train classifier to identify arguments

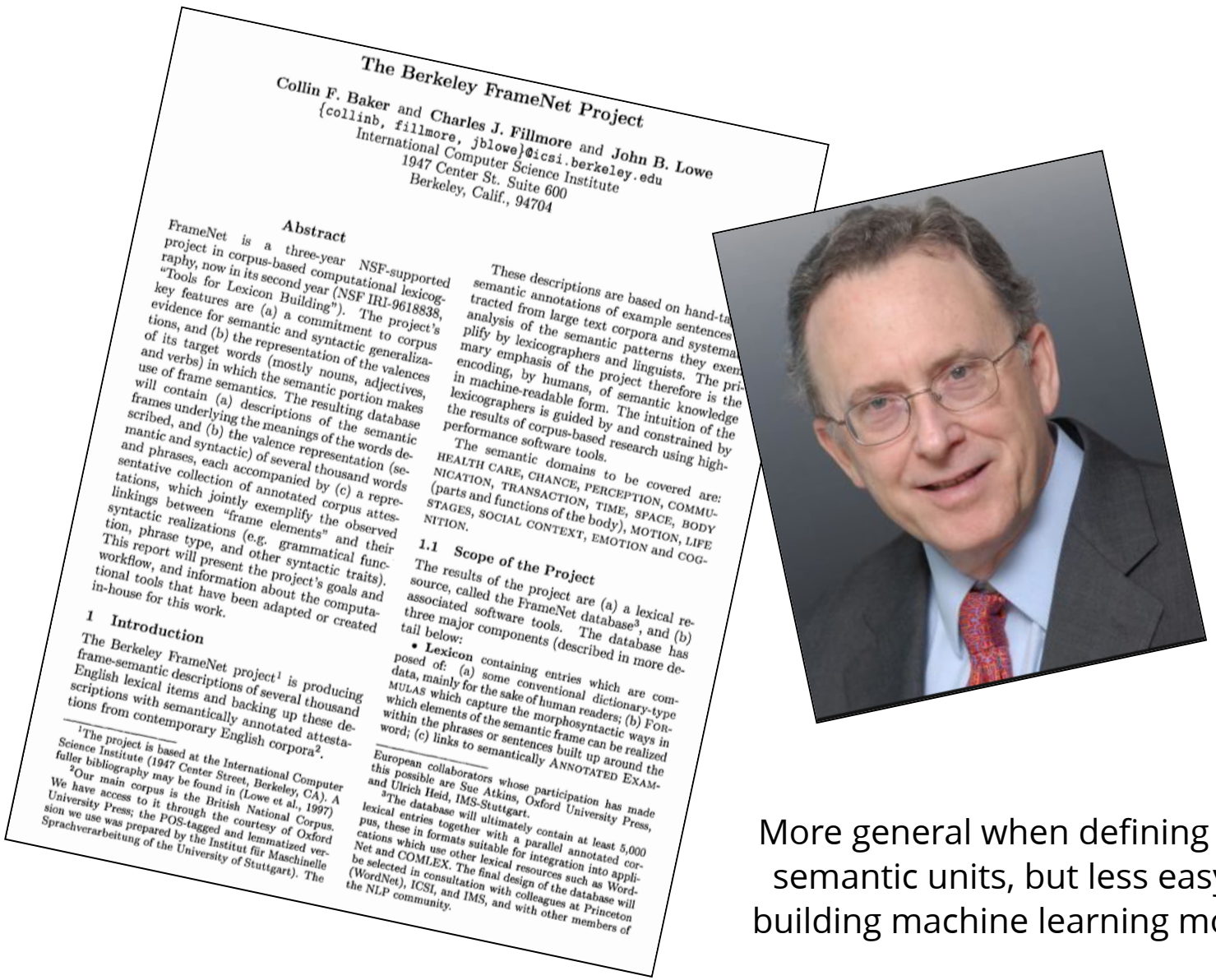


Argument Labeling

1. Extract features same as or similar to used in argument identification
2. Train classifier to select label for arguments

Optional

FrameNet



The Berkeley FrameNet Project

Collin F. Baker and Charles J. Fillmore and John B. Lowe
{collinb, fillmore, jblowe}@icsi.berkeley.edu
International Computer Science Institute
1947 Center St. Suite 600
Berkeley, Calif., 94704

Abstract

FrameNet is a three-year NSF-supported project in corpus-based computational lexicography, now in its second year (NSF IRI-9618838, "Tools for Lexicon Building"). The project's key features are (a) a commitment to corpus evidence for semantic and syntactic generalizations, and (b) the representation of the valences of its target words (mostly nouns, adjectives, and verbs) in which the semantic portion makes use of frame semantics. The resulting database will contain (a) descriptions of the semantic frames underlying the meanings of the words described, and (b) the valence representation (semantic and syntactic) of several thousand words and phrases, each accompanied by (c) a representative collection of annotated corpus attestations, which jointly exemplify the observed linkings between "frame elements" and their syntactic realizations (e.g. grammatical function, phrase type, and other syntactic traits). This report will present the project's goals and workflow, and information about the computational tools that have been adapted or created in-house for this work.

1 Introduction

The Berkeley FrameNet project[1] is producing frame-semantic descriptions of several thousand English lexical items and backing up these descriptions with semantically annotated attestations from contemporary English corpora[2].

These descriptions are based on hand-ta semantic annotations of example sentences tracted from large text corpora and systema analysis of the semantic patterns they exem plify by lexicographers and linguists. The pri mary emphasis of the project therefore is the encoding, by humans, of semantic knowledge in machine-readable form. The intuition of the lexicographers is guided by and constrained by the results of corpus-based research using high-performance software tools.

The semantic domains to be covered are: HEALTH CARE, CHANCE, PERCEPTION, COMMUNICATION, TRANSACTION, TIME, SPACE, BODY (parts and functions of the body), MOTION, LIFE STAGES, SOCIAL CONTEXT, EMOTION and COGNITION.

1.1 Scope of the Project

The results of the project are (a) a lexical resource, called the FrameNet database[3], and (b) associated software tools. The database has three major components (described in more detail below:

- **Lexicon** containing entries which are composed of: (a) some conventional dictionary-type data, mainly for the sake of human readers; (b) FORMULAS which capture the morphosyntactic ways in which elements of the semantic frame can be realized within the phrases or sentences built up around the word; (c) links to semantically ANNOTATED EXAM-

[1]The project is based at the International Computer Science Institute (1947 Center Street, Berkeley, CA). A fuller bibliography may be found in (Lowe et al., 1997)

[2]Our main corpus is the British National Corpus. We have access to it through the courtesy of Oxford University Press; the POS-tagged and lemmatized version we use was prepared by the Institut für Maschinelle Sprachverarbeitung of the University of Stuttgart). The

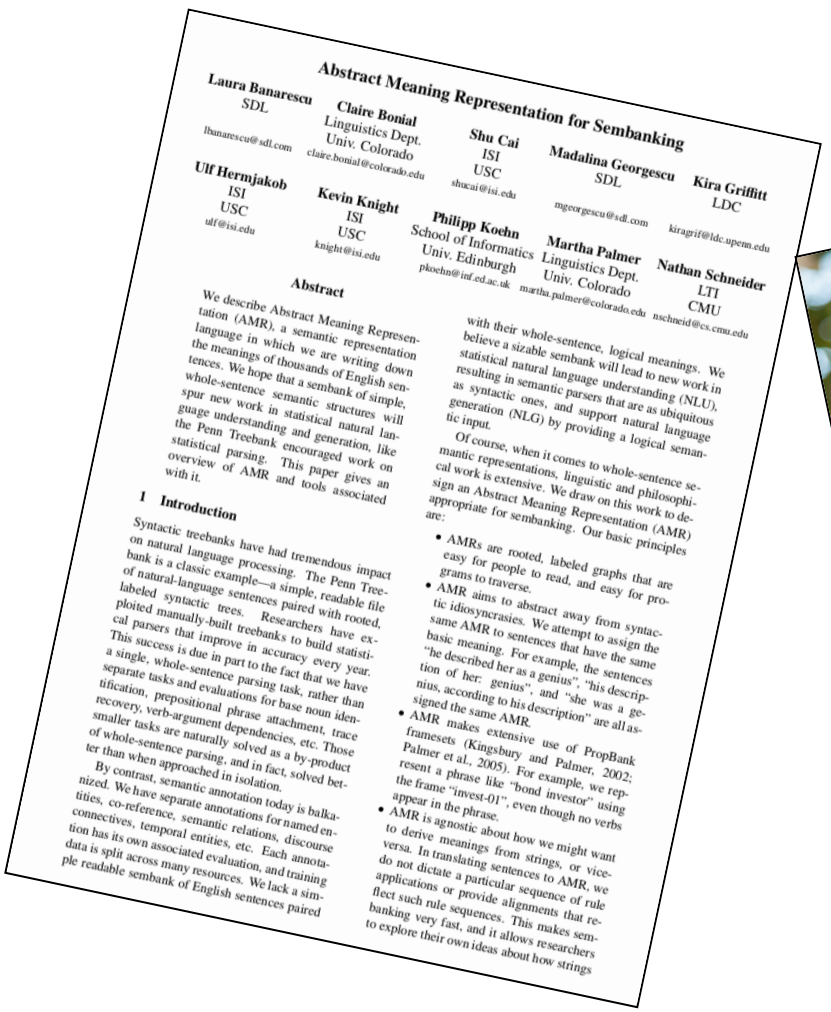
European collaborators whose participation has made this possible are Sue Atkins, Oxford University Press, and Ulrich Heid, IMS-Stuttgart.

[3]The database will ultimately contain at least 5,000 lexical entries together with a parallel annotated corpus, these in formats suitable for integration into applications which use other lexical resources such as WordNet and COMLEX. The final design of the database will be selected in consultation with colleagues at Princeton (WordNet), ICSI, and IMS, and with other members of the NLP community.

More general when defining basic semantic units, but less easy for building machine learning models.

Optional

Abstract Meaning Representation



Abstract Meaning Representation for Sembanking

Laura Banarescu
SDL
lbanarescu@sdl.com

Claire Bonial
Linguistics Dept.
Univ. Colorado
claire.bonial@colorado.edu

Shu Cai
ISI
USC
shucai@isi.edu

Madalina Georgescu
SDL
mgeorgescu@sdl.com

Kira Griffitt
LDC
kiragrif@ldc.upenn.edu

Ulf Hermjakob
ISI
USC
ulf@isi.edu

Kevin Knight
ISI
USC
knight@isi.edu

Philipp Koehn
School of Informatics
Univ. Edinburgh
pkoehn@inf.ed.ac.uk

Martha Palmer
Linguistics Dept.
Univ. Colorado
martha.palmer@colorado.edu

Nathan Schneider
LTI
CMU
nschneid@cs.cmu.edu

Abstract

We describe Abstract Meaning Representation (AMR), a semantic representation language in which we are writing down the meanings of thousands of English sentences. We hope that a sembank of simple, whole-sentence semantic structures will spur new work in statistical natural language understanding and generation, like the Penn Treebank encouraged work on statistical parsing. This paper gives an overview of AMR and tools associated with it.

1 Introduction

Syntactic treebanks have had tremendous impact on natural language processing. The Penn Treebank is a classic example—a simple, readable file of natural-language sentences paired with rooted, labeled syntactic trees. Researchers have exploited manually-built treebanks to build statistical parsers that improve in accuracy every year. This success is due in part to the fact that we have a single, whole-sentence parsing task, rather than separate tasks and evaluations for base noun identification, prepositional phrase attachment, trace recovery, verb-argument dependencies, etc. Those smaller tasks are naturally solved as a by-product of whole-sentence parsing, and in fact, solved better than when approached in isolation.

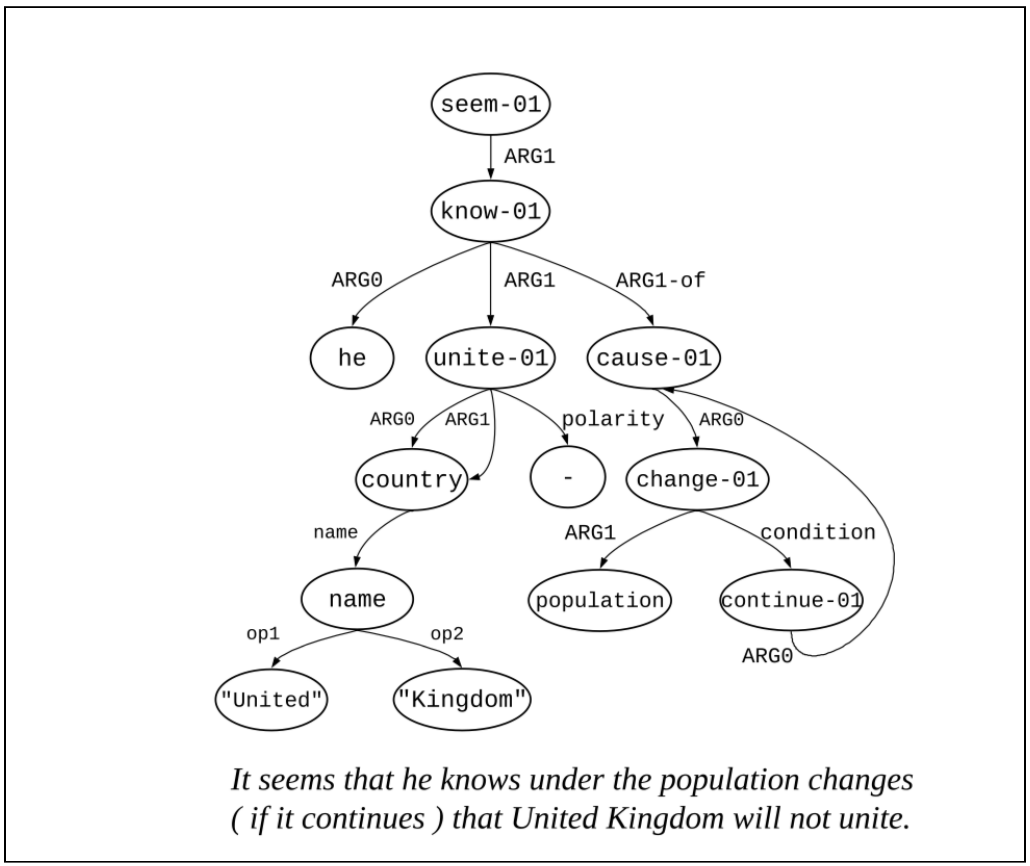
By contrast, semantic annotation today is balkanized. We have separate annotations for named entities, co-reference, semantic relations, discourse connectives, temporal entities, etc. Each annotation has its own associated evaluation, and training data is split across many resources. We lack a simple readable sembank of English sentences paired with their whole-sentence, logical meanings. We believe a sizable sembank will lead to new work in statistical natural language understanding (NLU), resulting in semantic parsers that are as ubiquitous as syntactic ones, and support natural language generation (NLG) by providing a logical semantic input.

Of course, when it comes to whole-sentence semantic representations, linguistic and philosophical work is extensive. We draw on this work to design an Abstract Meaning Representation (AMR) appropriate for sembanking. Our basic principles are:

- AMRs are rooted, labeled graphs that are easy for people to read, and easy for programs to traverse.
- AMR aims to abstract away from syntactic idiosyncrasies. We attempt to assign the same AMR to sentences that have the same basic meaning. For example, the sentences "he described her as a genius", "his description of her: genius", and "she was a genius, according to his description" are all assigned the same AMR.
- AMR makes extensive use of PropBank framesets (Kingsbury and Palmer, 2002; Palmer et al., 2005). For example, we represent a phrase like "bond investor" using the frame "invest-01", even though no verbs appear in the phrase.
- AMR is agnostic about how we might want to derive meanings from strings, or vice-versa. In translating sentences to AMR, we do not dictate a particular sequence of rule applications or provide alignments that reflect such rule sequences. This makes sembanking very fast, and it allows researchers to explore their own ideas about how strings

Optional

Abstract Meaning Representation



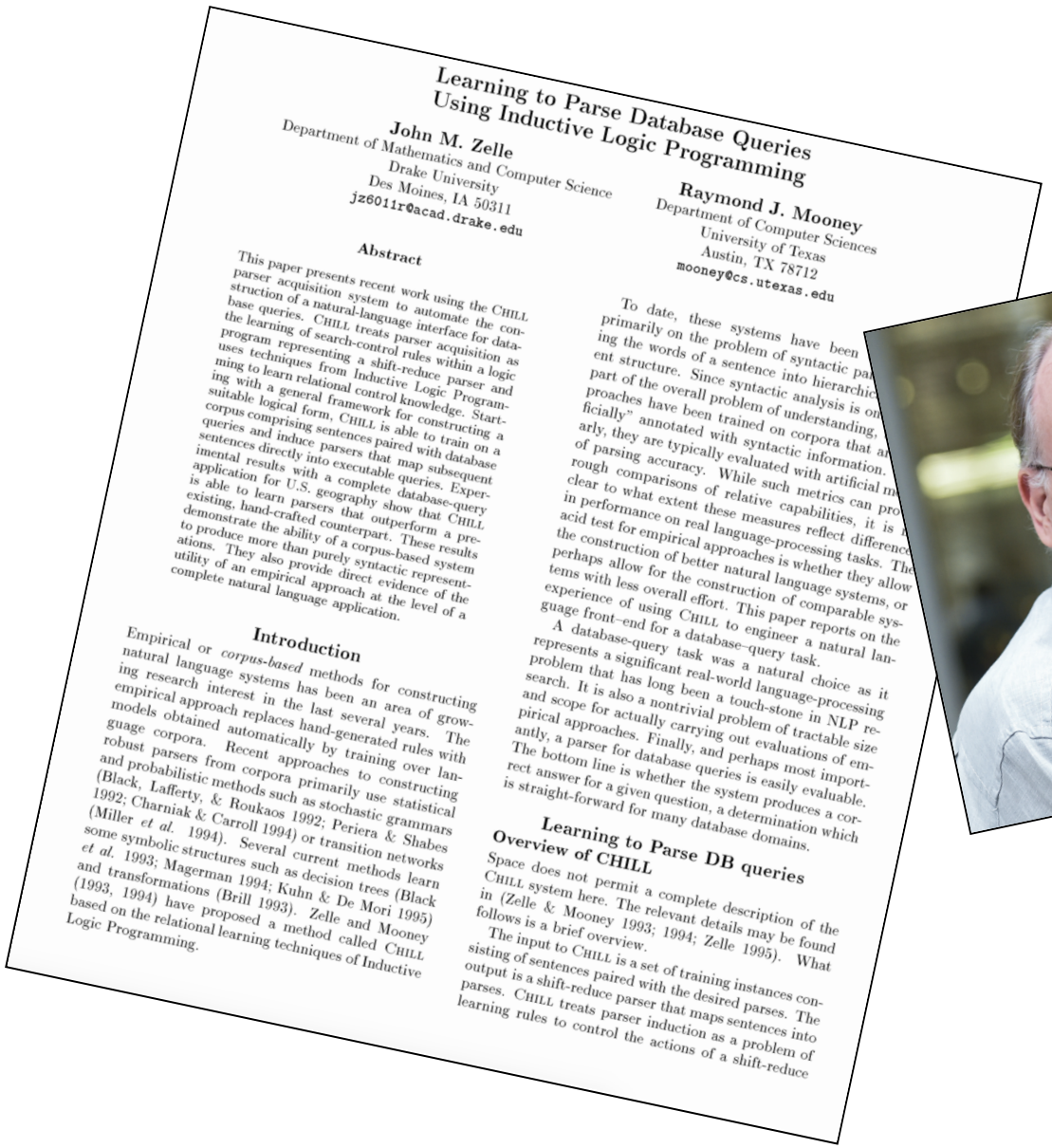
Built based on the development of Propbank

Question

What is non-shallow semantic parsing?

Semantics should be "useful"!

Semantic Parsing



Learning to Parse Database Queries
Using Inductive Logic Programming

John M. Zelle
Department of Mathematics and Computer Science
Drake University
Des Moines, IA 50311
jz6011r@acad.drake.edu

Raymond J. Mooney
Department of Computer Sciences
University of Texas
Austin, TX 78712
mooney@cs.utexas.edu

Abstract

This paper presents recent work using the CHILL parser acquisition system to automate the construction of a natural-language interface for database queries. CHILL treats parser acquisition as the learning of search-control rules within a logic program representing a shift-reduce parser and uses techniques from Inductive Logic Programming to learn relational control knowledge. Starting with a general framework for constructing a suitable logical form, CHILL is able to train on a corpus comprising sentences paired with database queries and induce parsers that map subsequent sentences directly into executable queries. Experimental results with a complete database-query application for U.S. geography show that CHILL is able to learn parsers that outperform a pre-existing, hand-crafted counterpart. These results demonstrate the ability of a corpus-based system to produce more than purely syntactic representations. They also provide direct evidence of the utility of an empirical approach at the level of a complete natural language application.

Introduction

Empirical or *corpus-based* methods for constructing natural language systems has been an area of growing research interest in the last several years. The empirical approach replaces hand-generated rules with models obtained automatically by training over language corpora. Recent approaches to constructing robust parsers from corpora primarily use statistical and probabilistic methods such as stochastic grammars (Black, Lafferty, & Roukaos 1992; Periera & Shabes 1992; Charniak & Carroll 1994) or transition networks (Miller *et al.* 1994). Several current methods learn some symbolic structures such as decision trees (Black *et al.* 1993; Magerman 1994; Kuhn & De Mori 1995) and transformations (Brill 1993). Zelle and Mooney (1993, 1994) have proposed a method called CHILL based on the relational learning techniques of Inductive Logic Programming.

To date, these systems have been primarily on the problem of syntactic pa ing the words of a sentence into hierarchic ent structure. Since syntactic analysis is ou part of the overall problem of understanding, proaches have been trained on corpora that a ficially" annotated with syntactic information. arly, they are typically evaluated with artificial m of parsing accuracy. While such metrics can pro rough comparisons of relative capabilities, it is clear to what extent these measures reflect differences in performance on real language-processing tasks. The acid test for empirical approaches is whether they allow the construction of better natural language systems, or perhaps allow for the construction of comparable systems with less overall effort. This paper reports on the experience of using CHILL to engineer a natural language front–end for a database–query task.

A database-query task was a natural choice as it represents a significant real-world language-processing problem that has long been a touch-stone in NLP research. It is also a nontrivial problem of tractable size and scope for actually carrying out evaluations of empirical approaches. Finally, and perhaps most importantly, a parser for database queries is easily evaluable. The bottom line is whether the system produces a correct answer for a given question, a determination which is straight-forward for many database domains.

Learning to Parse DB queries

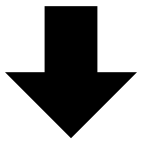
Overview of CHILL

Space does not permit a complete description of the CHILL system here. The relevant details may be found in (Zelle & Mooney 1993; 1994; Zelle 1995). What follows is a brief overview.

The input to CHILL is a set of training instances consisting of sentences paired with the desired parses. The output is a shift-reduce parser that maps sentences into parses. CHILL treats parser induction as a problem of learning rules to control the actions of a shift-reduce

Semantic Parsing

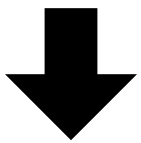
Which rivers run through the states bordering Texas?



$$answer(tranverse(next\_to(stateid(\text{`texas}'))))$$

Semantic Parsing

Which rivers run through the states bordering Texas?



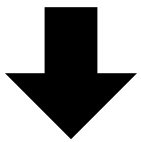
$answer(tranverse(next\_to(stateid(\text{'texas'}))))$

Querying databases!

Arkansas, Mississippi, . . .

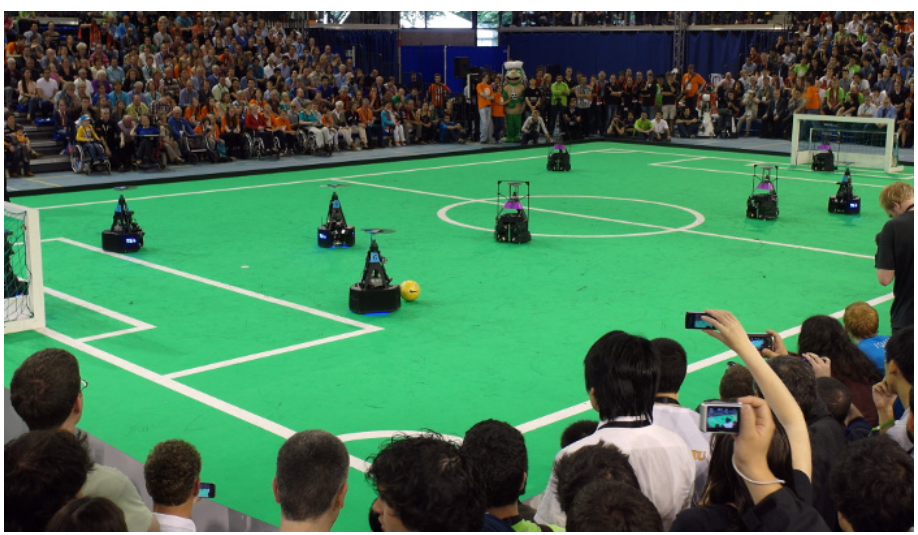
Semantic Parsing

If player 3 has the ball, then position player 5 in the midfield.



$((bowser(player\ our\ 3))\ (do\ (player\ our\ 5)\ (pos\ (midfield))))$

Controlling robots!



Semantic Parsing

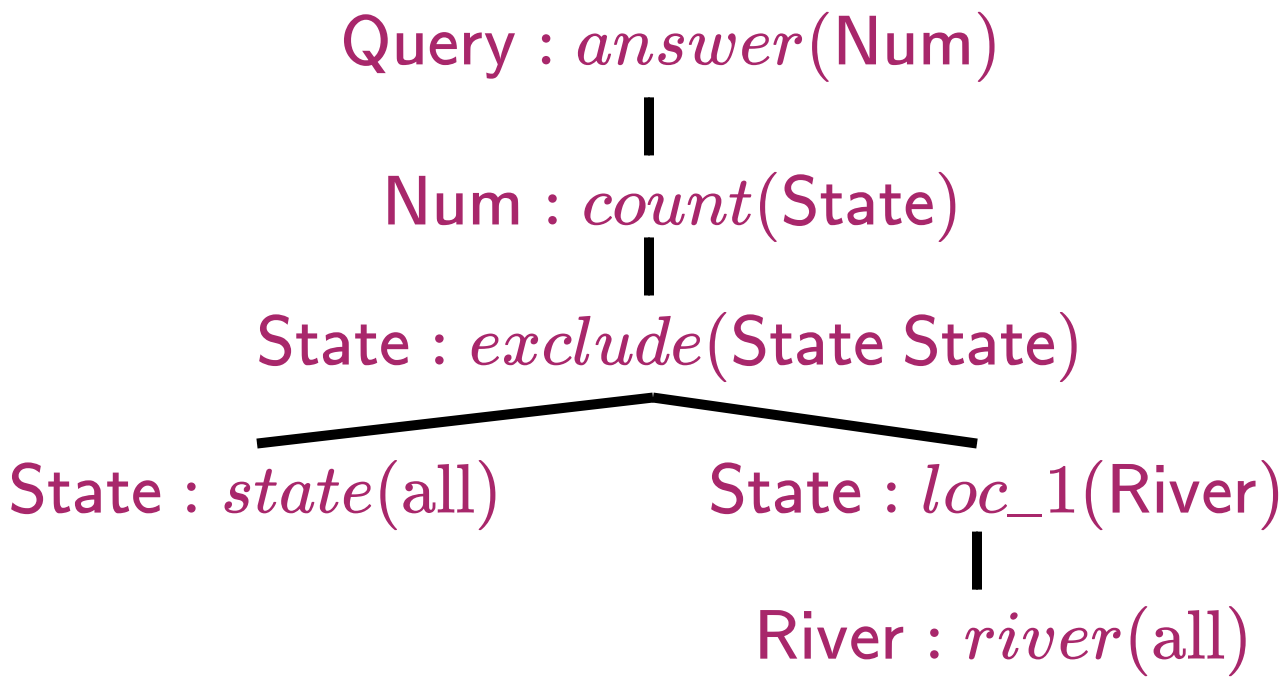
How many states do not have rivers ?

$$answer(count(exclude(state(\text{all}), loc\_1(river(\text{all})))))$$

Semantic Parsing

How many states do not have rivers ?

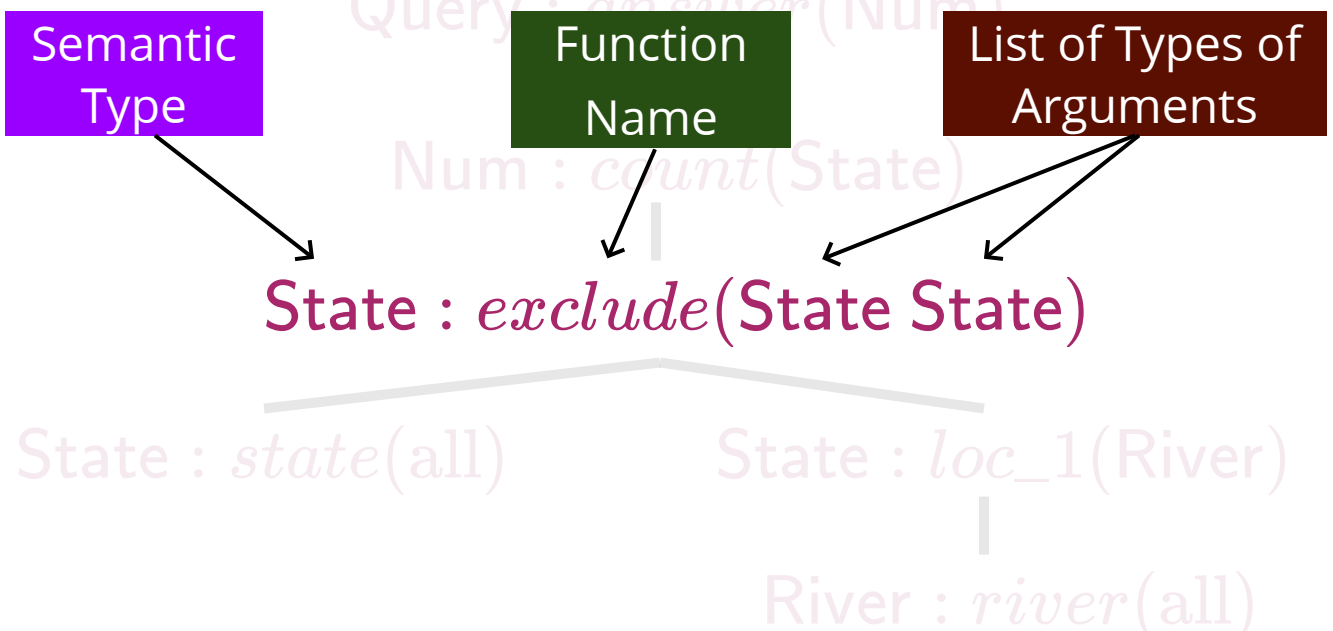
$answer(count(exclude(state(\text{all}), loc\_1(river(\text{all})))))$



Semantic Parsing

How many states do not have rivers ?

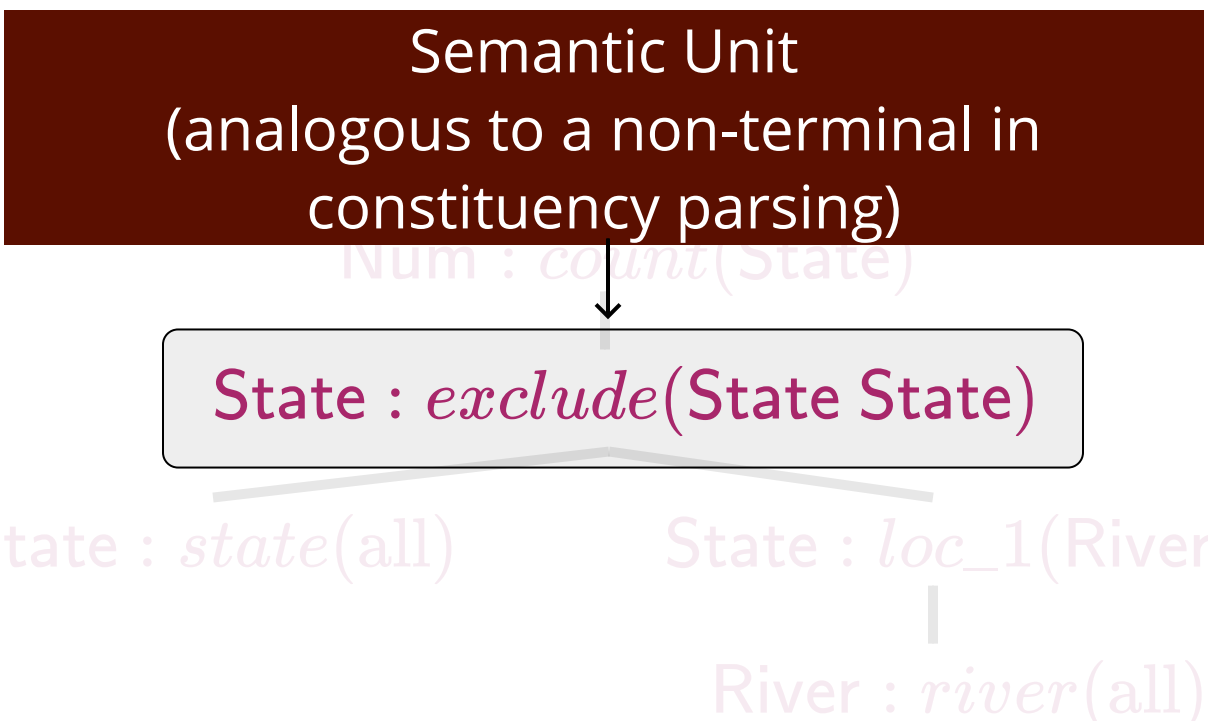
$answer(count(exclude(state(\text{all}), loc\_1(river(\text{all})))))$



Semantic Parsing

How many states do not have rivers ?

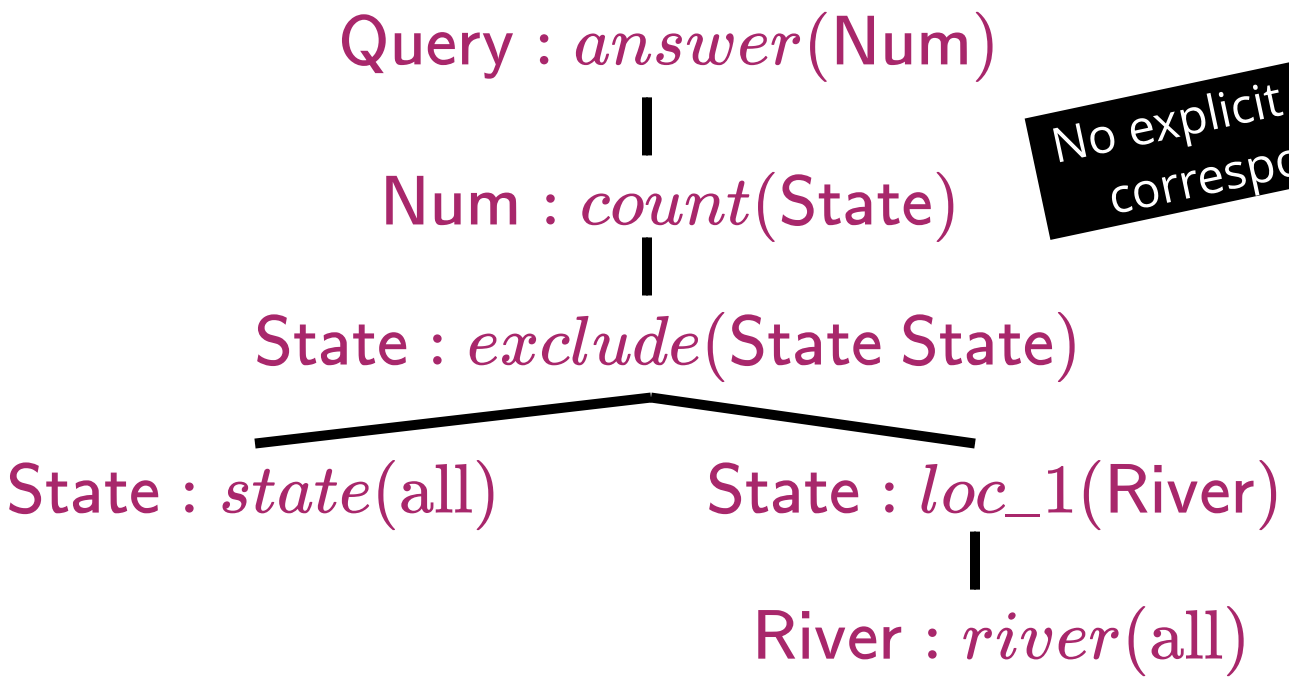
$answer(count(exclude(state(\text{all}), loc\_1(river(\text{all})))))$



Semantic Parsing

How is this problem different from constituency parsing?

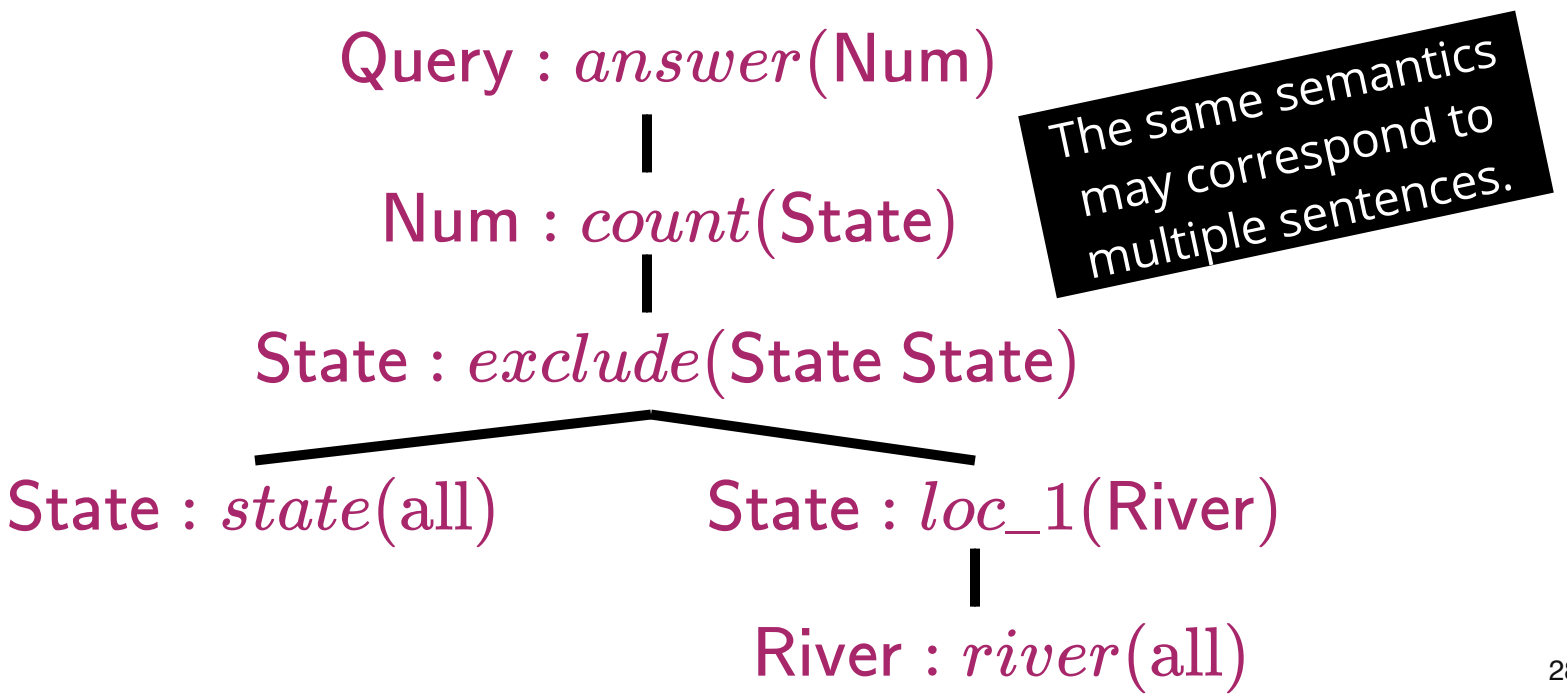
How many states do not have rivers ?



Semantic Parsing

Why is this problem more challenging than constituency parsing?

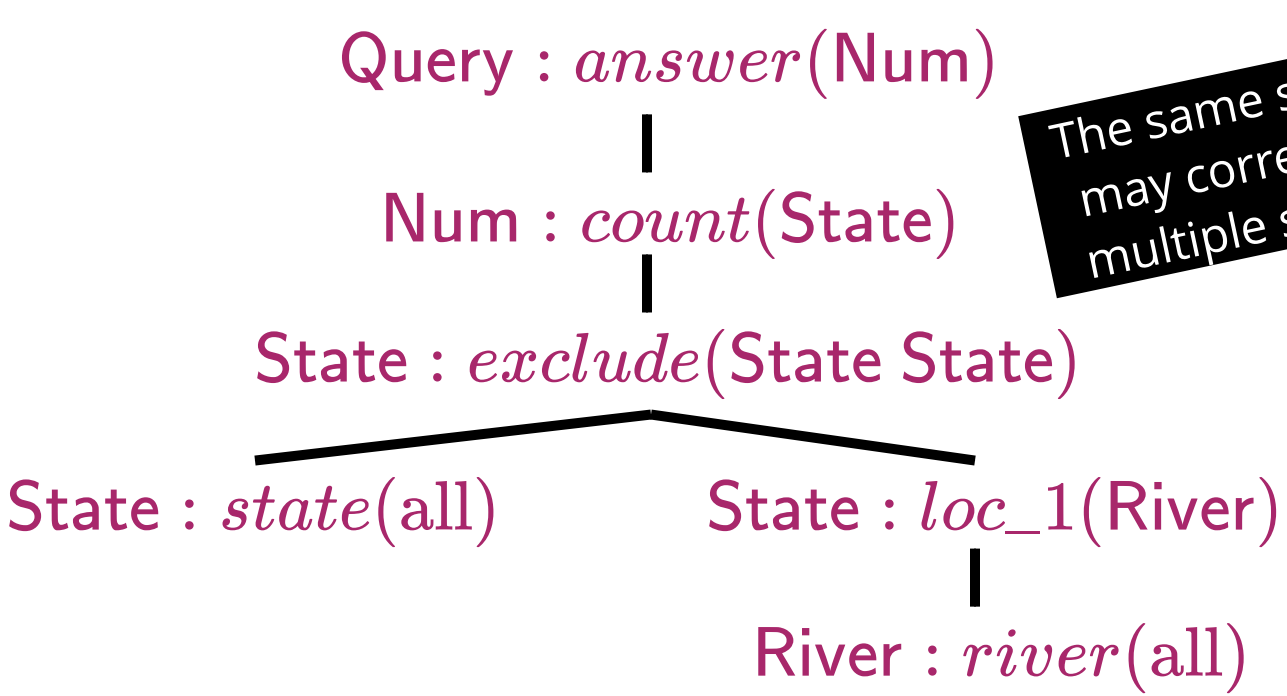
What is the number of states that do not have rivers ?



Semantic Parsing

Why is this problem more challenging than constituency parsing?

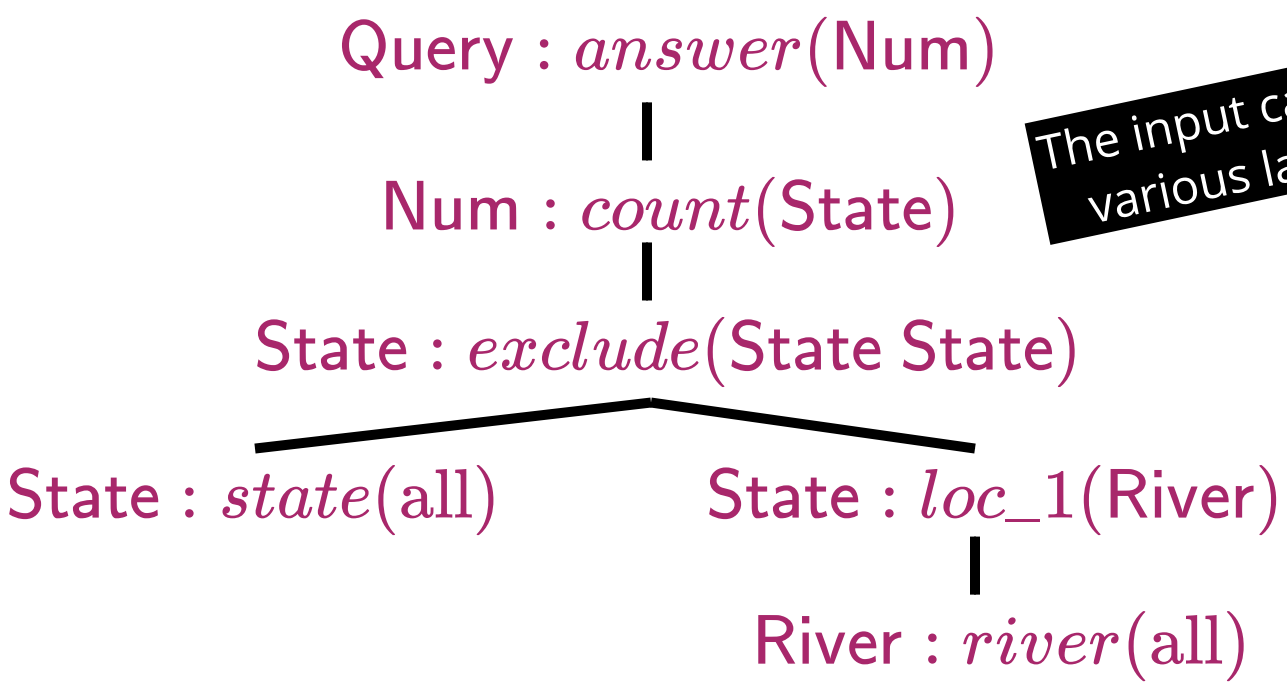
Count the number of riverless states !



Semantic Parsing

Why is this problem more challenging than constituency parsing?

有 多少 州 没有 河流 ?



The input can be from various languages.

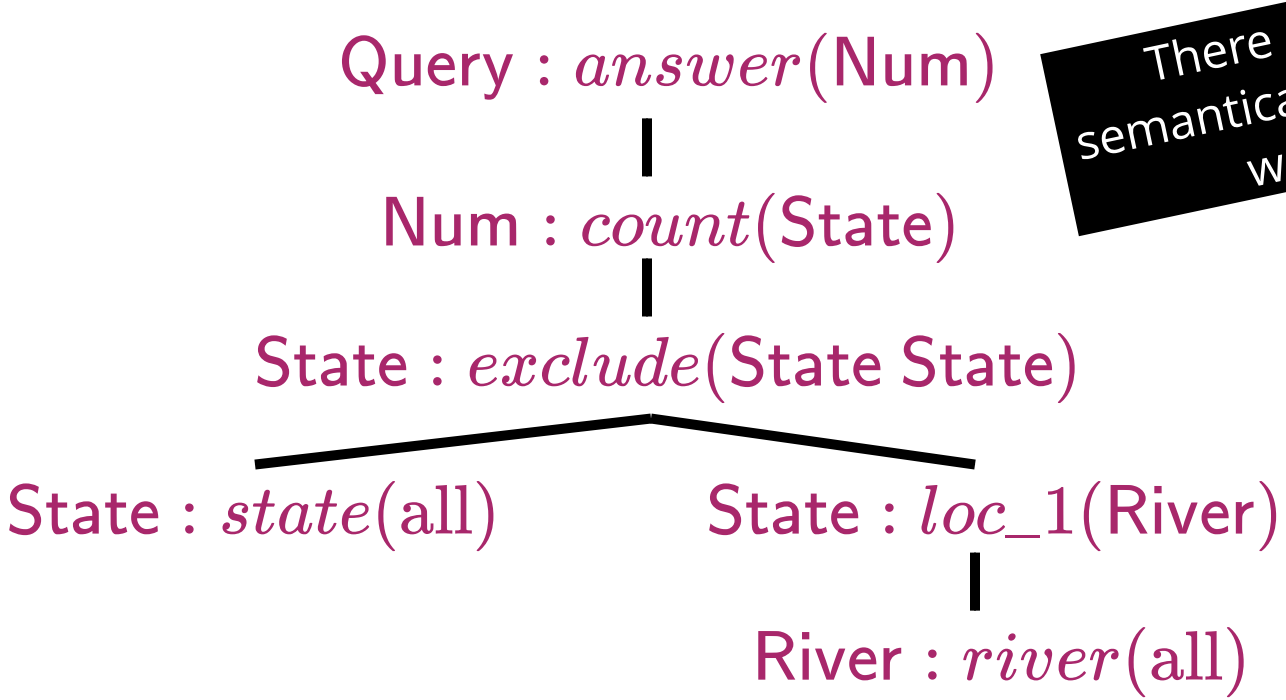
Semantic Parsing

Why is this problem more challenging than constituency parsing?

Semantic Parsing

Why is this problem more challenging than constituency parsing?

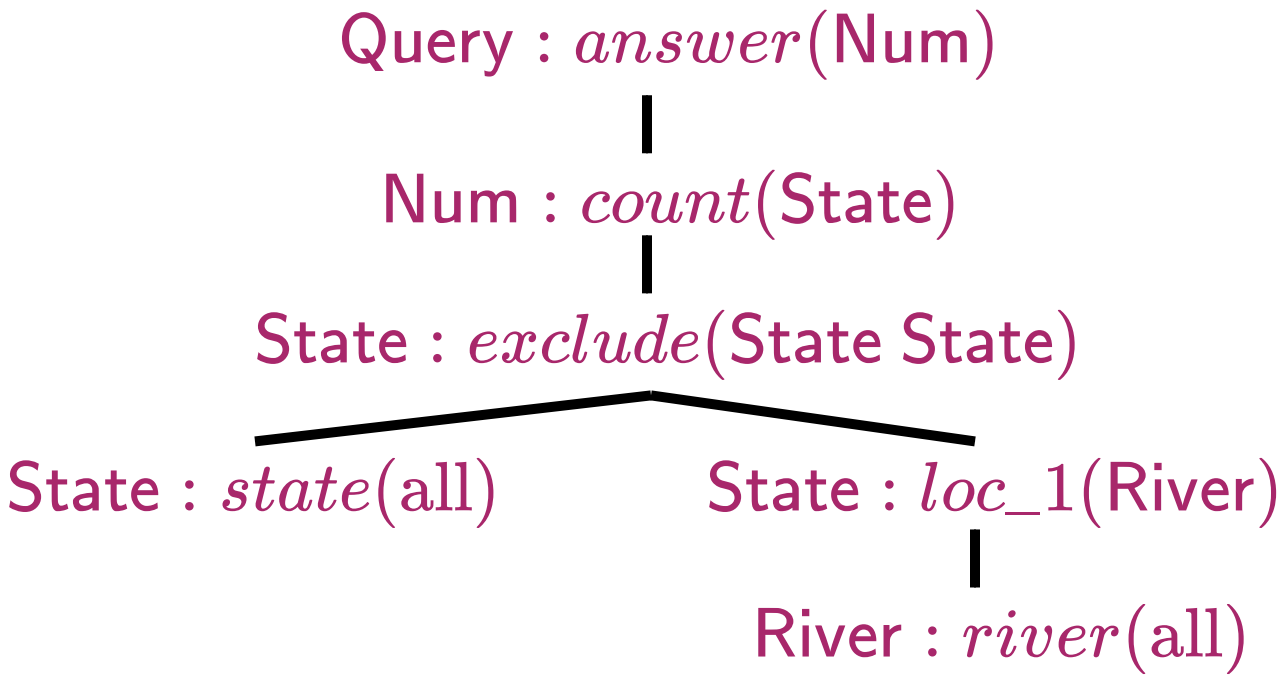
河流 不 经过 的 有 几个 州 呢 ?



There may be semantically vacuous words!

Semantic Parsing

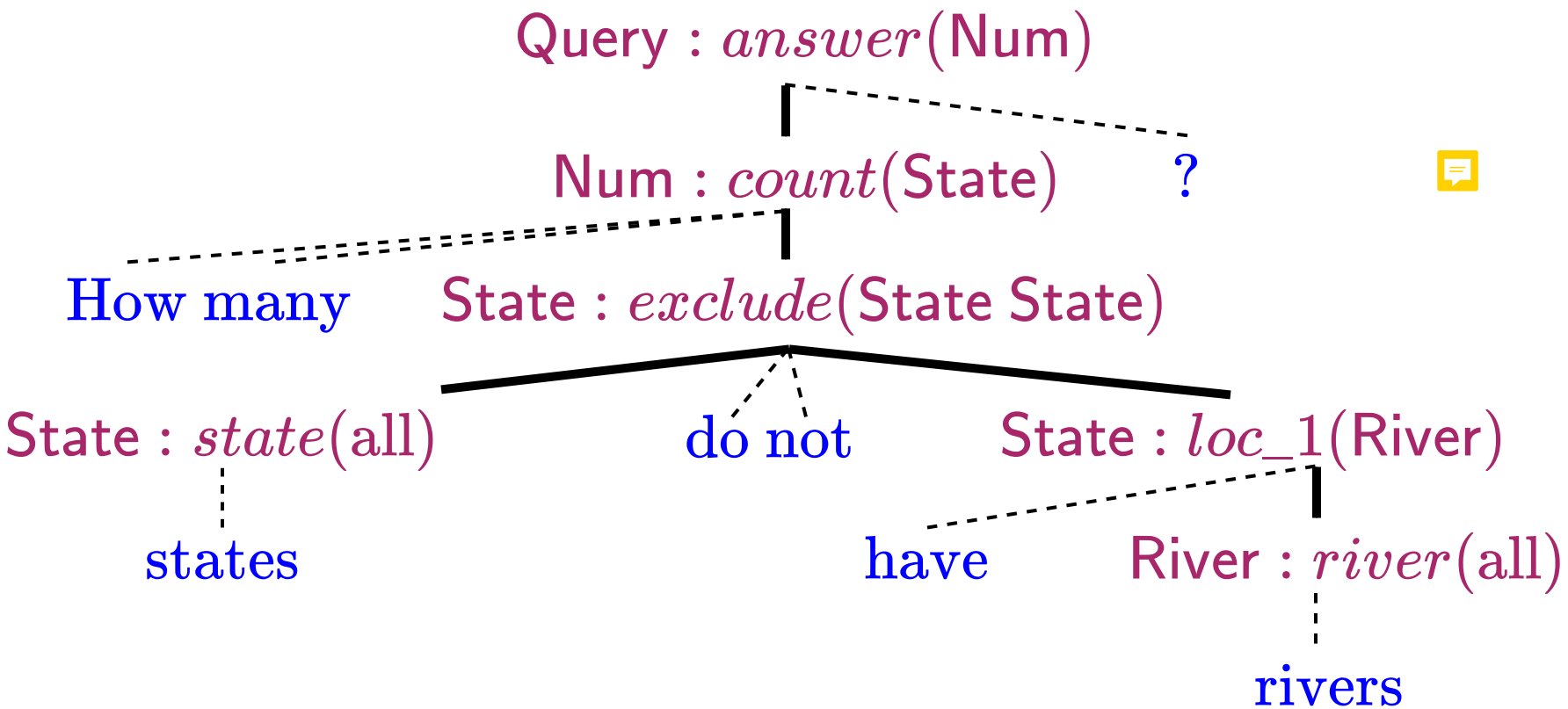
How many states do not have rivers ?



What if I would like to align the words with semantics?

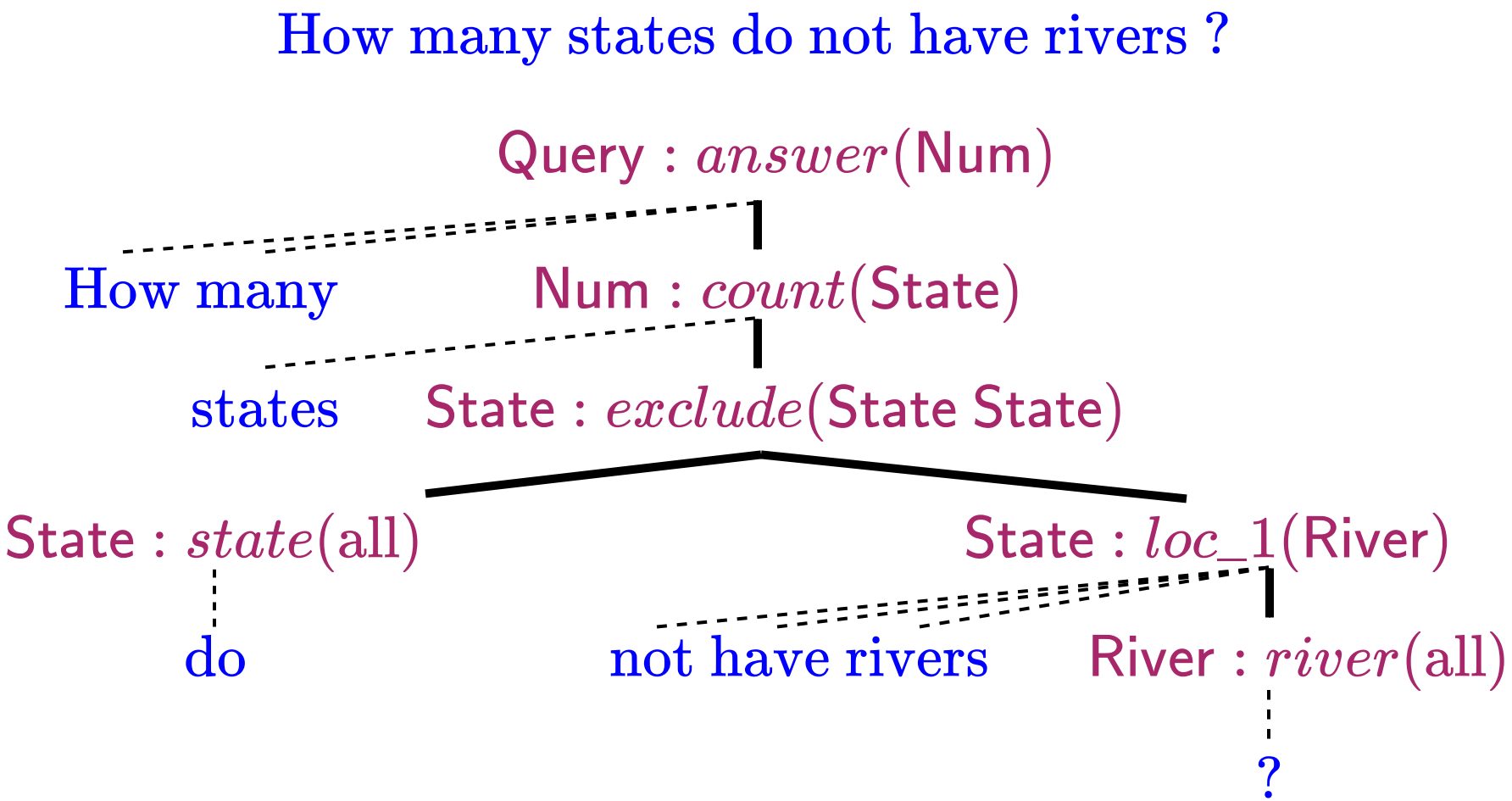
Hybrid Tree
(Lu et al. 2008)

How many states do not have rivers ?



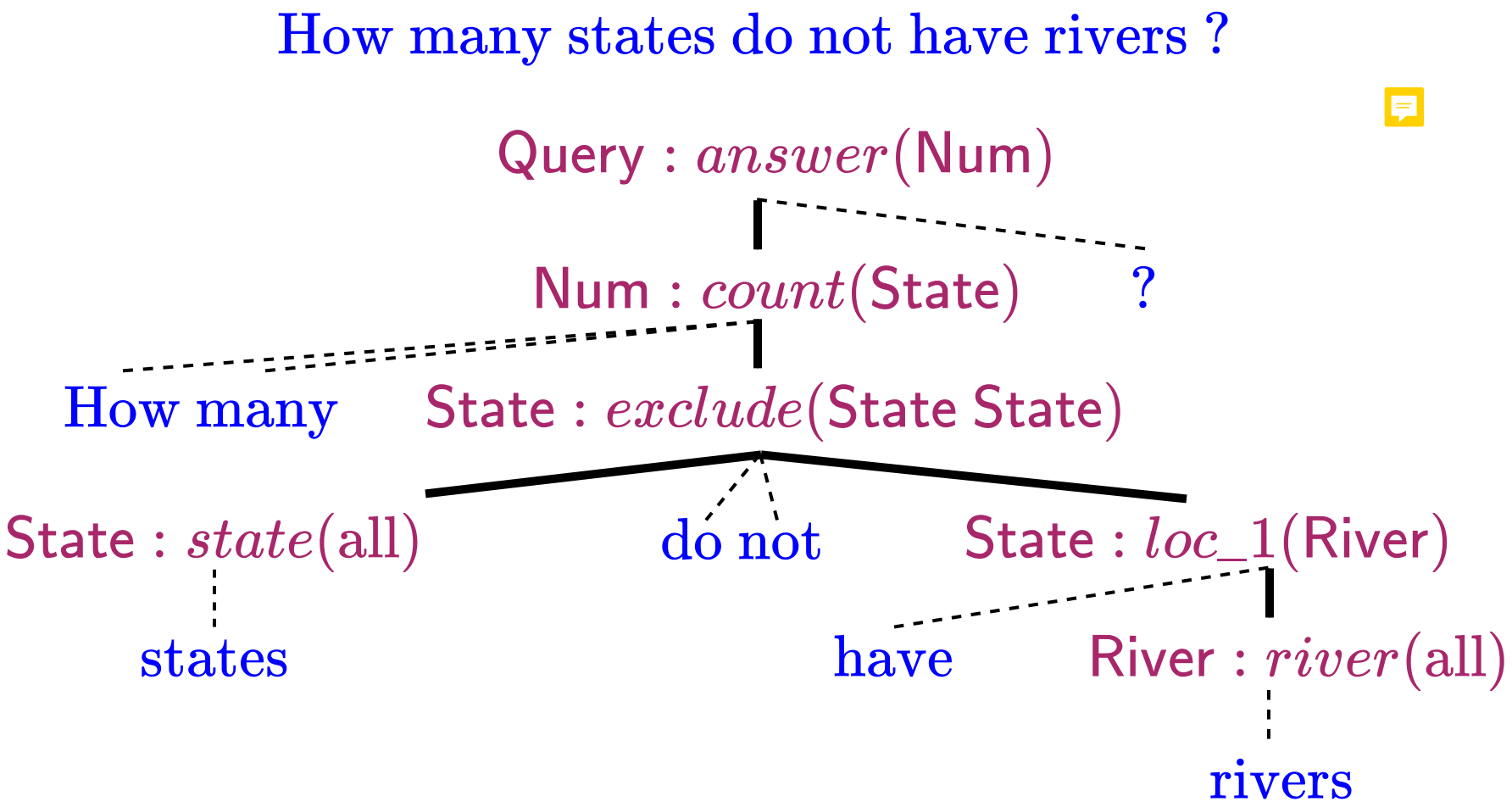
One possible way of alignment.

Hybrid Tree



Another way of aligning the tree and the sentence.

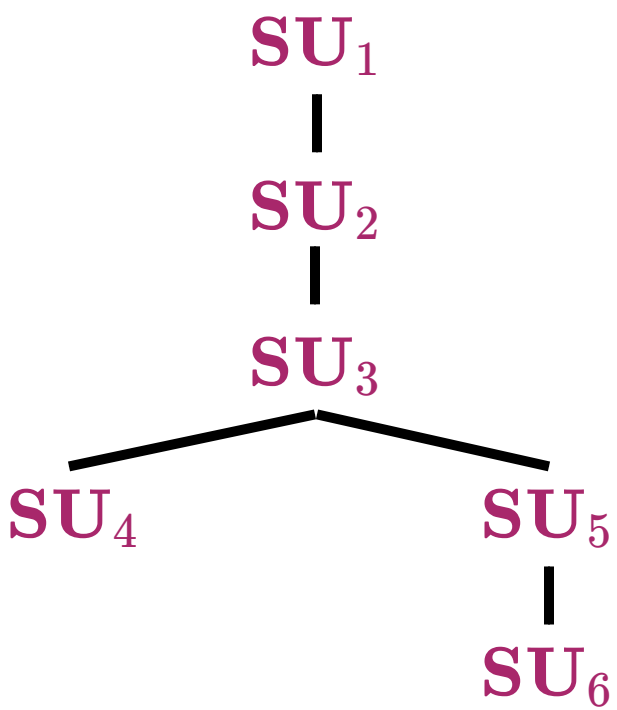
Hybrid Tree



Let's take a closer look at the tree.

Hybrid Tree

How many states do not have rivers ?



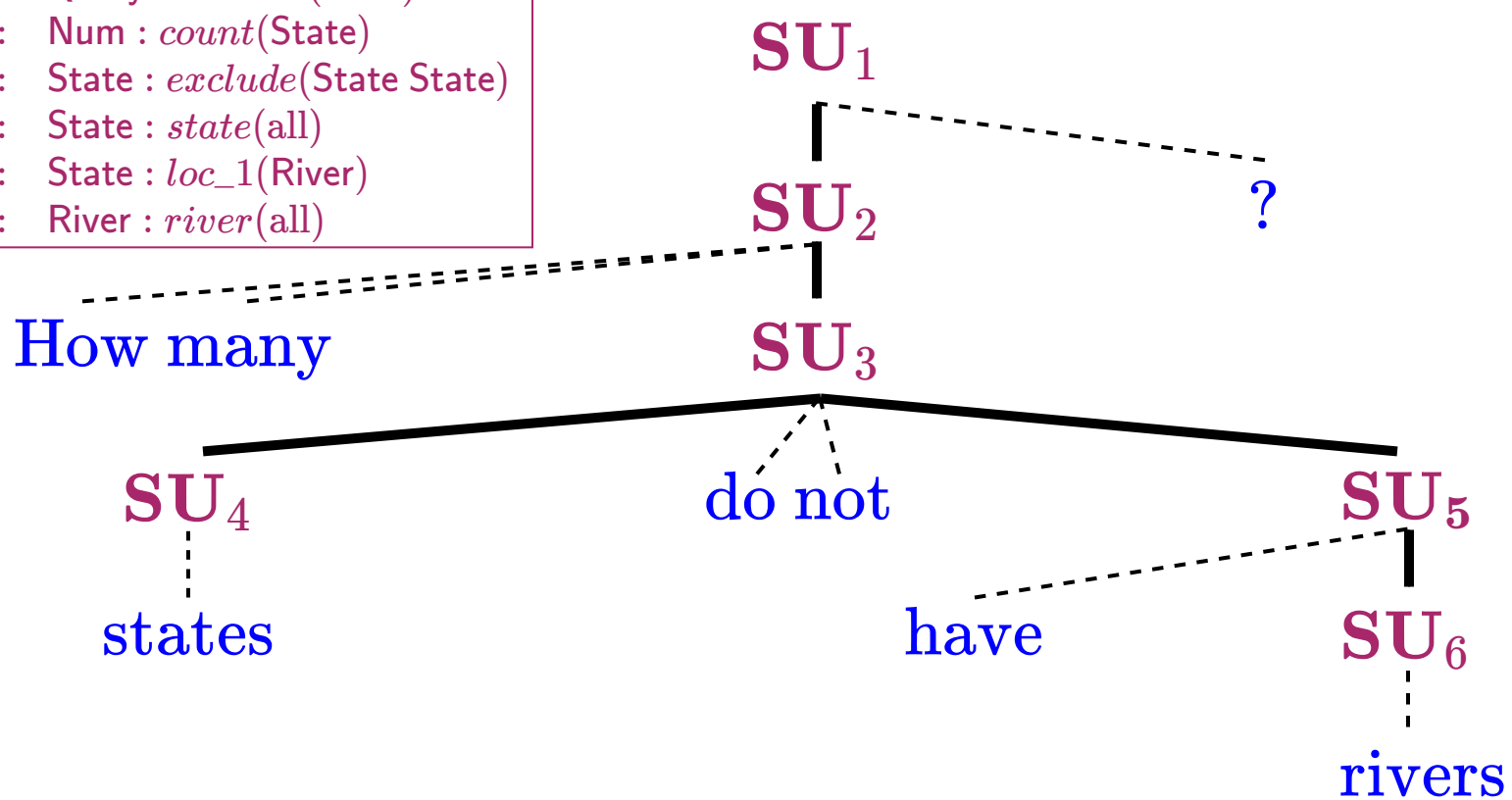
$\mathbf{SU}_1$:	Query : $answer$(Num)
$\mathbf{SU}_2$:	Num : $count$(State)
$\mathbf{SU}_3$:	State : $exclude$(State State)
$\mathbf{SU}_4$:	State : $state$(all)
$\mathbf{SU}_5$:	State : loc_1(River)
$\mathbf{SU}_6$:	River : $river$(all)

Let us try to save some space first...

Hybrid Tree

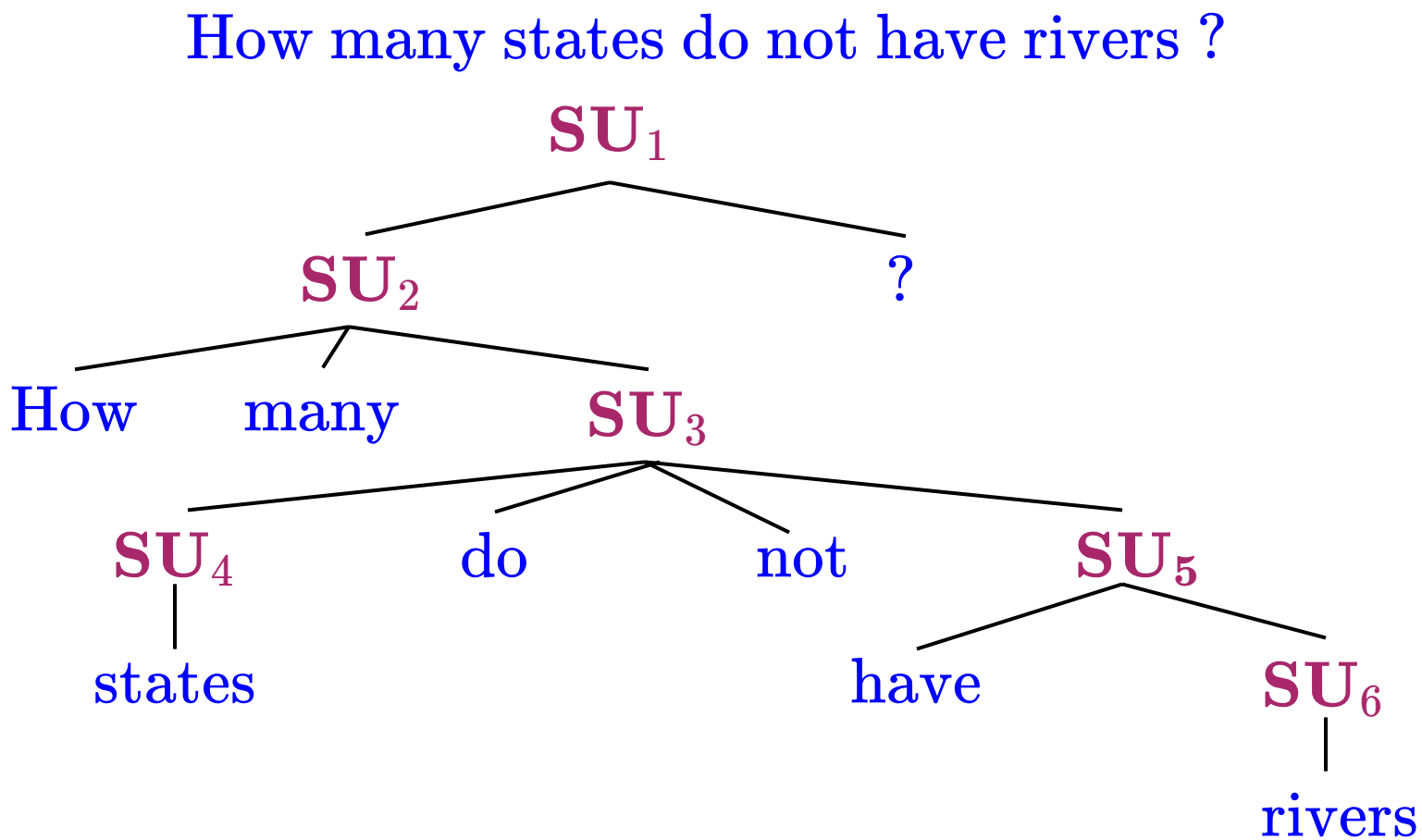
How many states do not have rivers ?

$\mathbf{SU}_1$:	Query : $answer$(Num)
$\mathbf{SU}_2$:	Num : $count$(State)
$\mathbf{SU}_3$:	State : $exclude$(State State)
$\mathbf{SU}_4$:	State : $state$(all)
$\mathbf{SU}_5$:	State : loc_1(River)
$\mathbf{SU}_6$:	River : $river$(all)



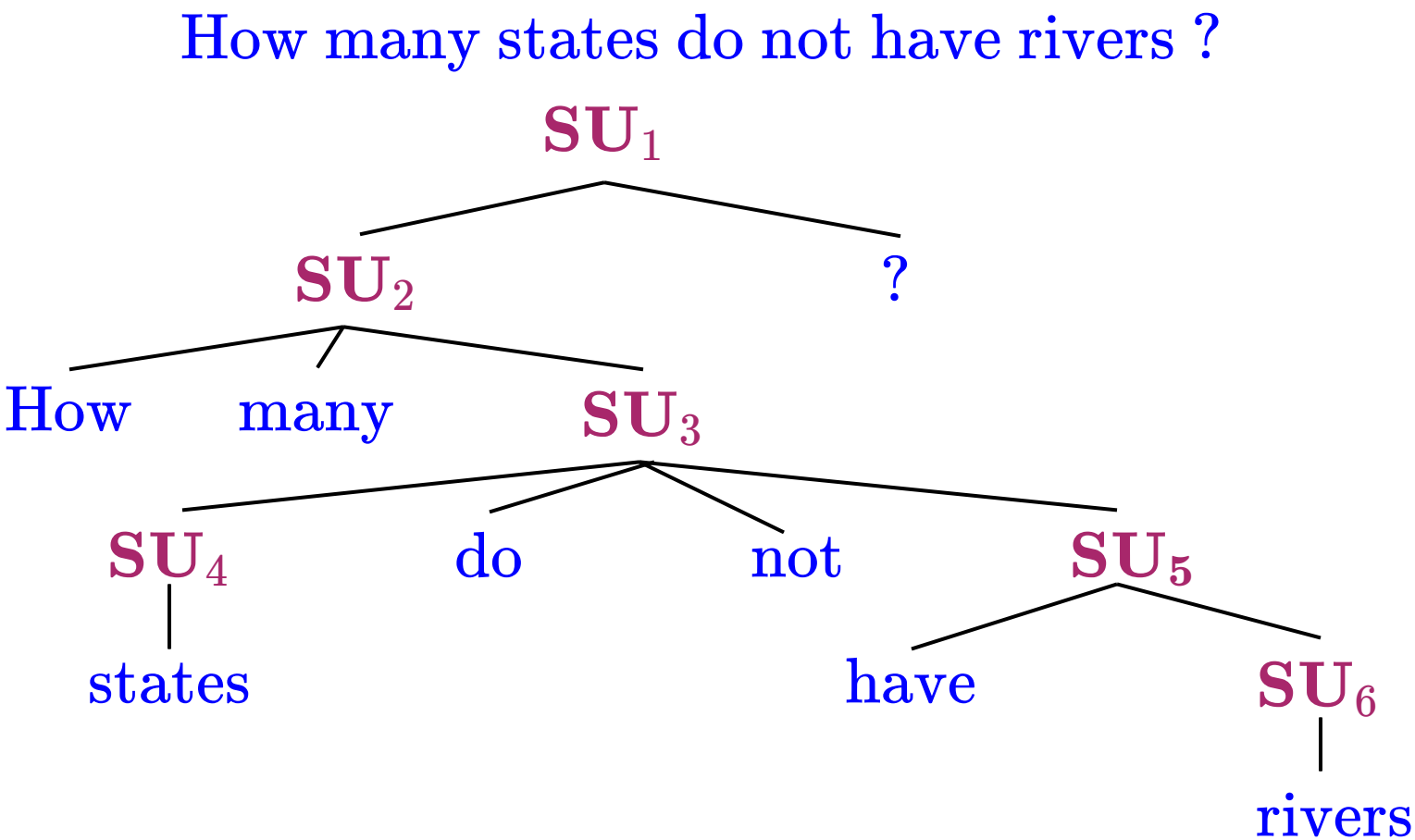
Now, let's take a closer look at the tree.

Hybrid Tree



This is a hybrid tree, where the leaf nodes are words, and the internal nodes are semantic units.

Hybrid Tree

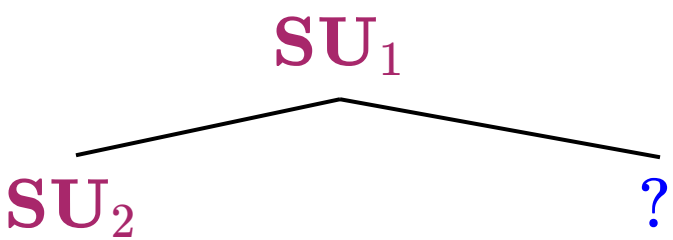


We assume there is an underlying generative process that is producing such a tree.

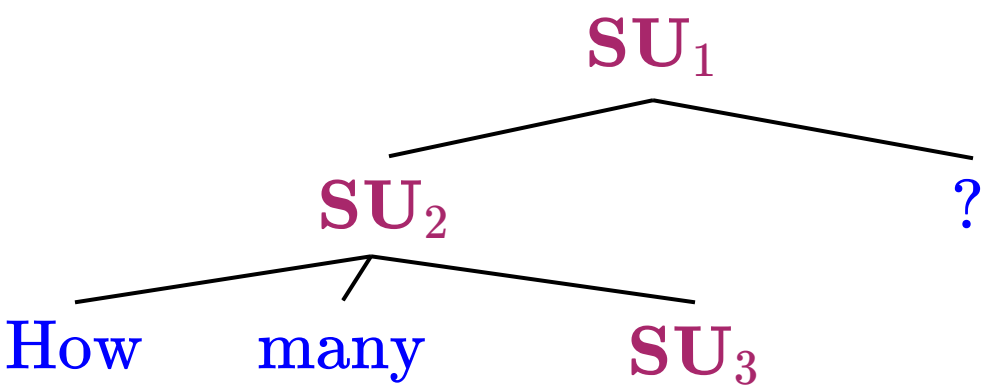
Hybrid Tree

$\mathbf{SU}_1$

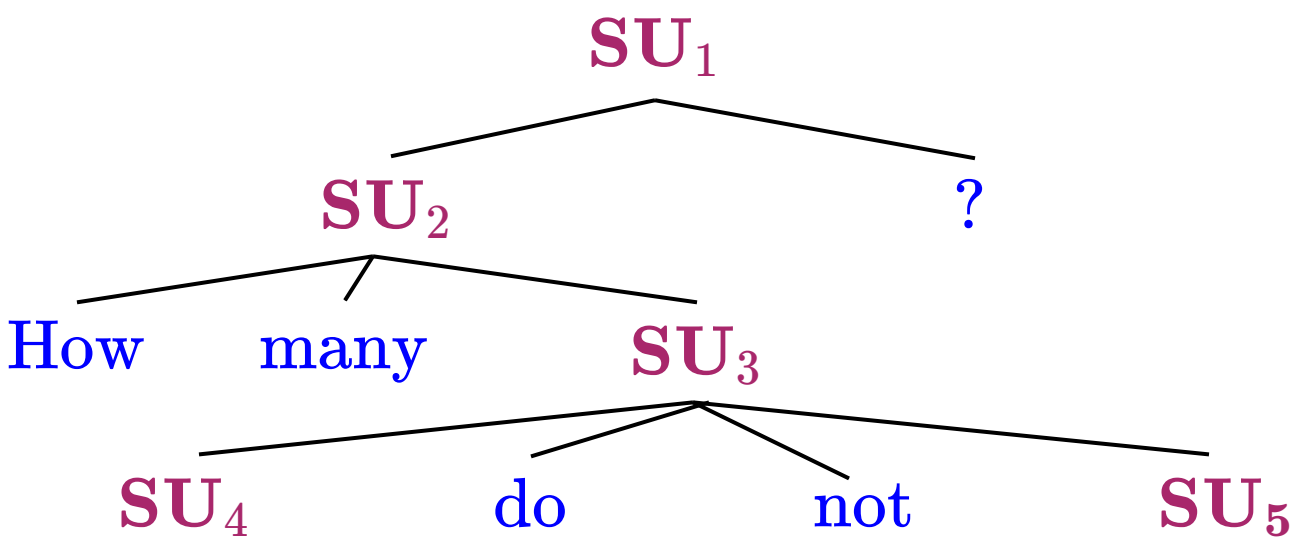
Hybrid Tree



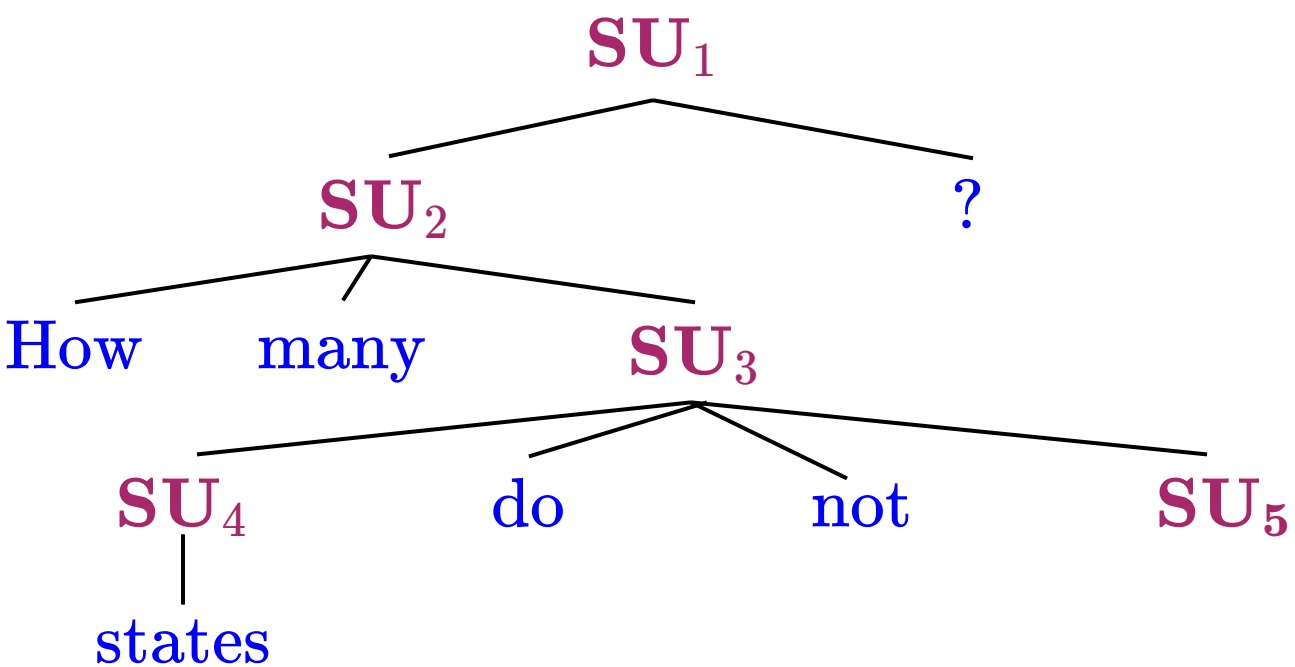
Hybrid Tree



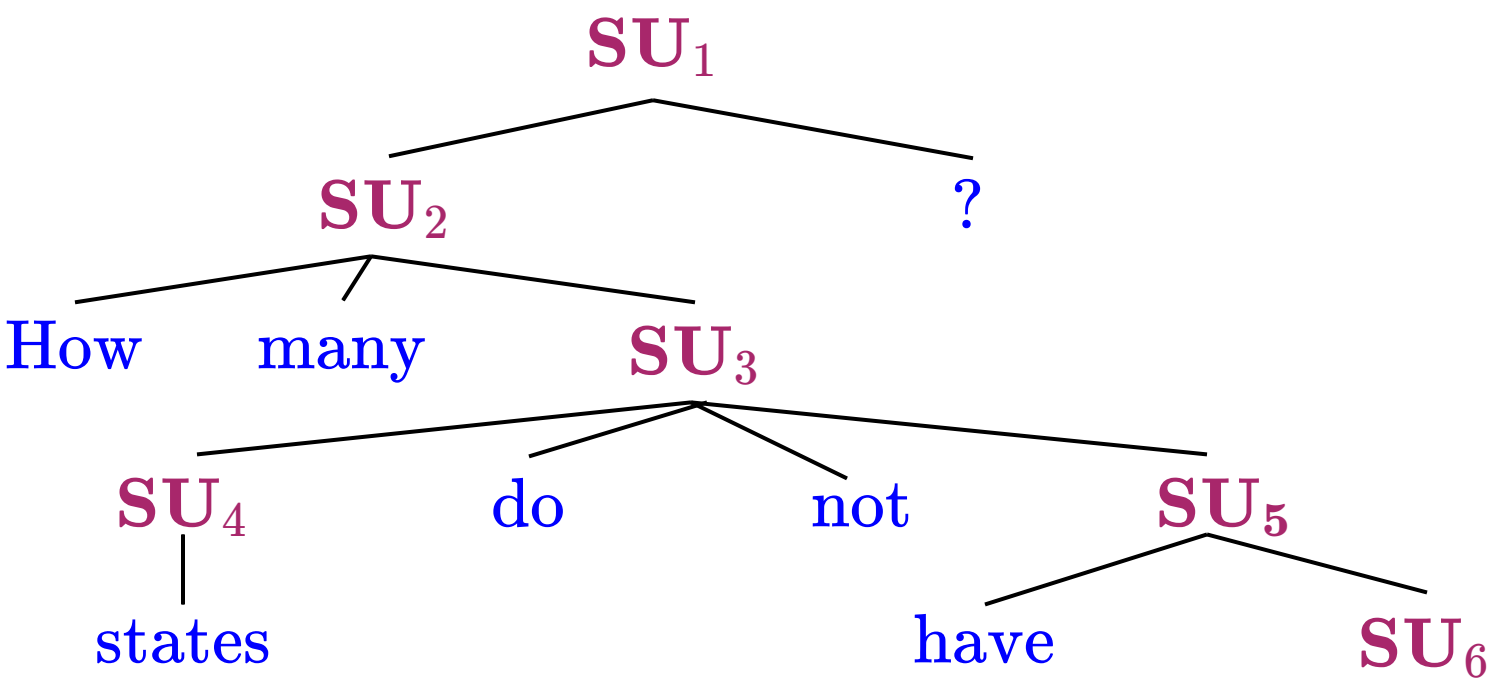
Hybrid Tree



Hybrid Tree

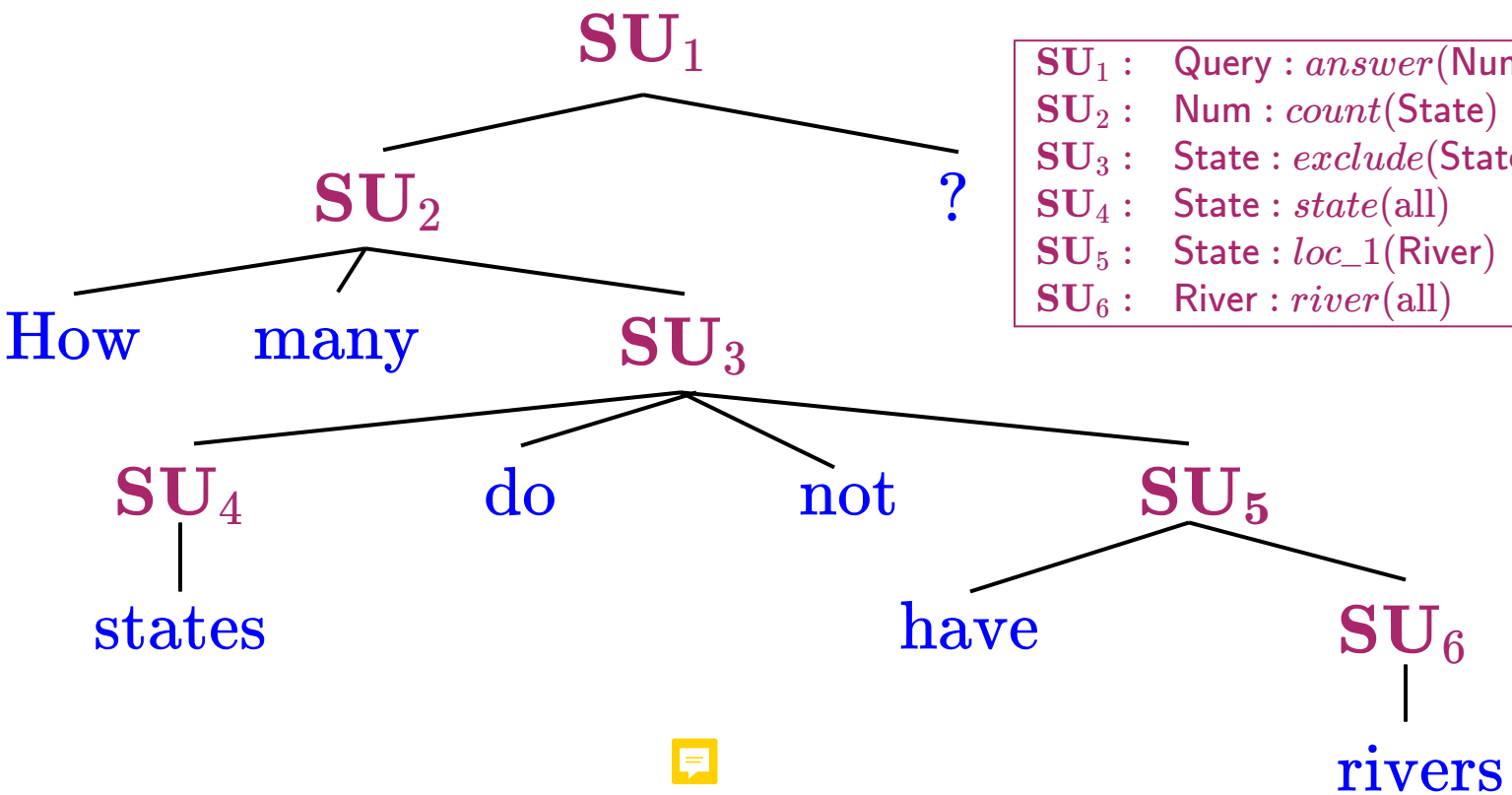


Hybrid Tree



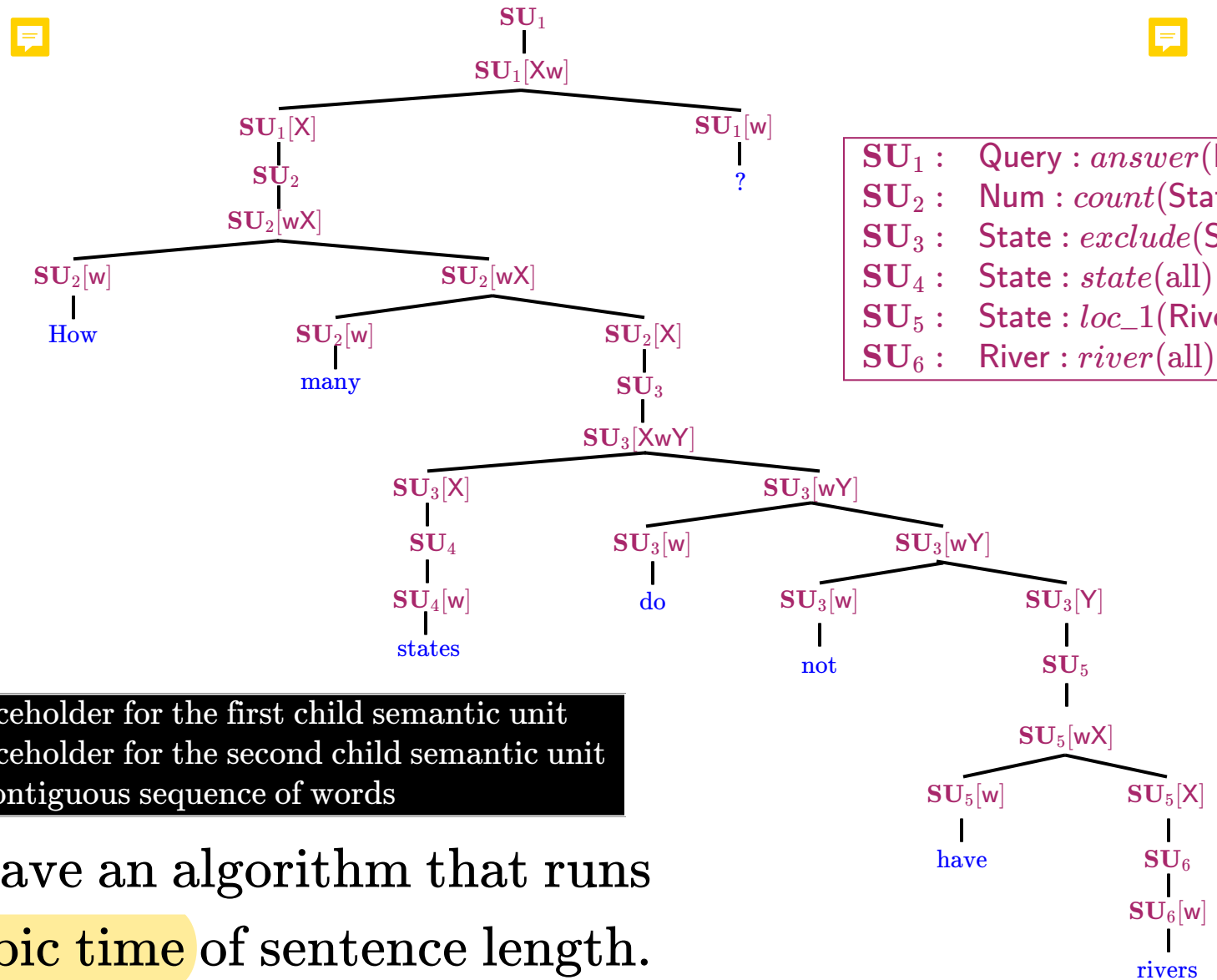
Hybrid Tree

How many states do not have rivers ?



$\mathbf{SU}_1$:	Query : $answer$(Num)
$\mathbf{SU}_2$:	Num : $count$(State)
$\mathbf{SU}_3$:	State : $exclude$(State State)
$\mathbf{SU}_4$:	State : $state$(all)
$\mathbf{SU}_5$:	State : loc_1(River)
$\mathbf{SU}_6$:	River : $river$(all)

Hybrid Tree

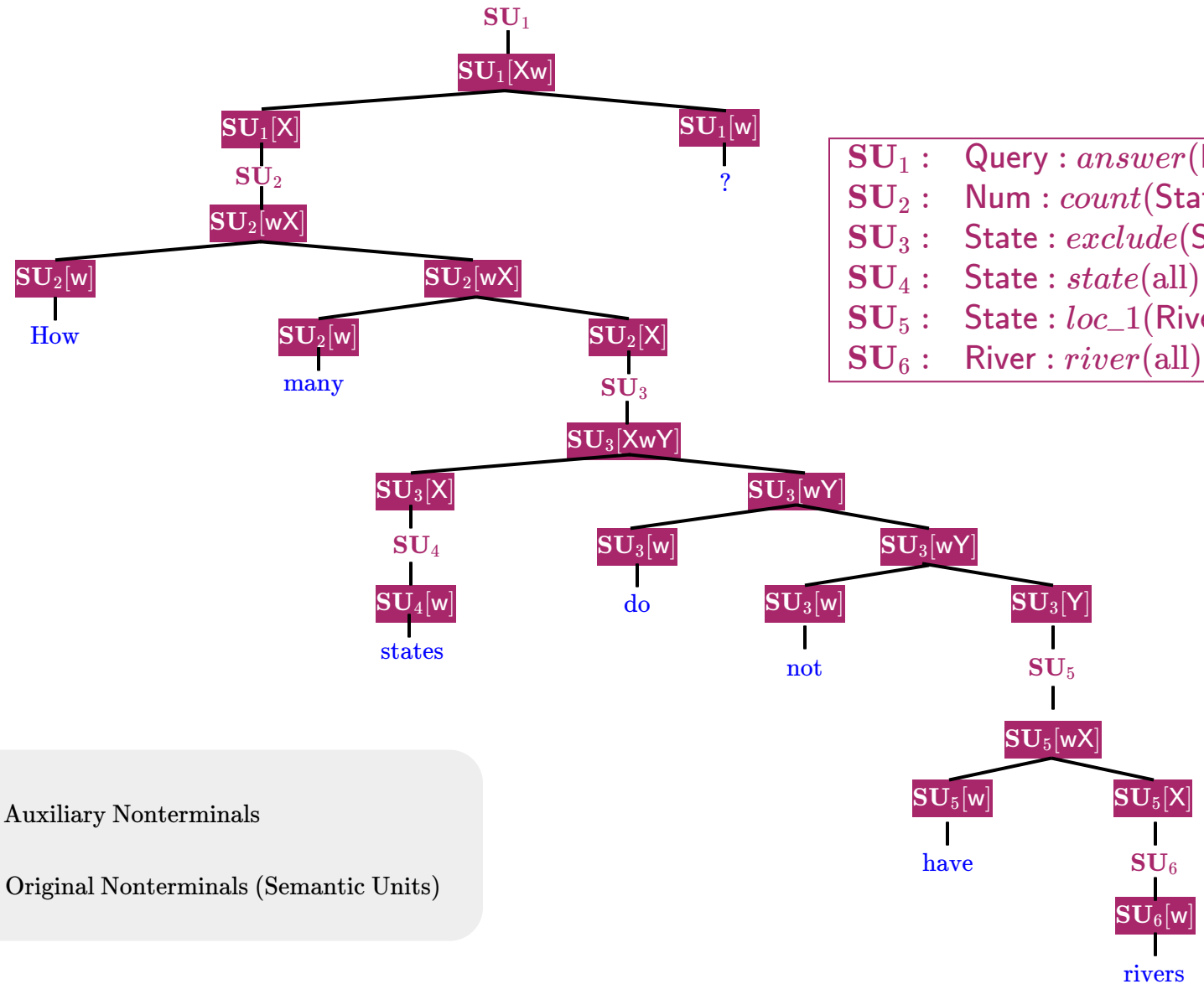


SU_1 :	Query : $answer$(Num)
SU_2 :	Num : $count$(State)
SU_3 :	State : $exclude$(State State)
SU_4 :	State : $state$(all)
SU_5 :	State : loc_1(River)
SU_6 :	River : $river$(all)

X :	placeholder for the first child semantic unit
Y :	placeholder for the second child semantic unit
w :	a contiguous sequence of words

We have an algorithm that runs in cubic time of sentence length.

Hybrid Tree

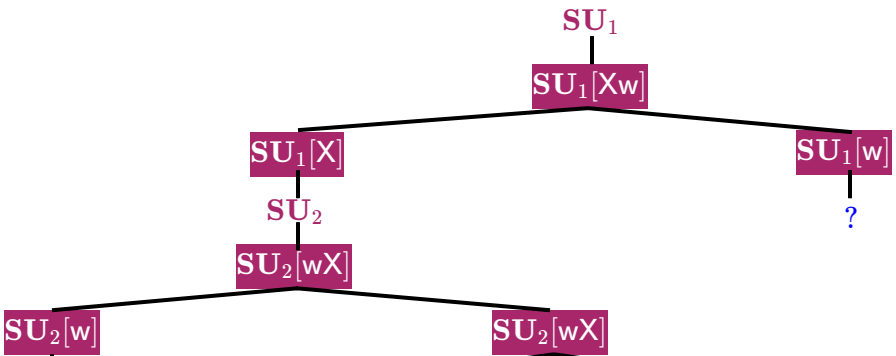


SU_1 :	Query : $answer$(Num)
SU_2 :	Num : $count$(State)
SU_3 :	State : $exclude$(State State)
SU_4 :	State : $state$(all)
SU_5 :	State : loc_1(River)
SU_6 :	River : $river$(all)

$SU_2[w]$ Auxiliary Nonterminals

SU_1 Original Nonterminals (Semantic Units)

Hybrid Tree



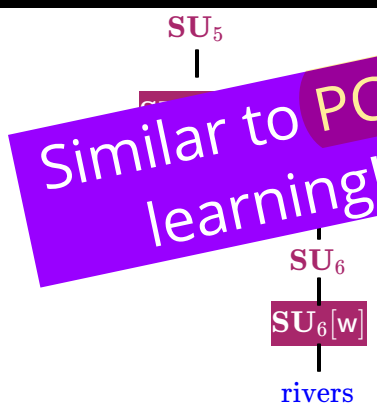
SU_1 : Query : *answer*(Num)
SU_2 : Num : *count*(State)
SU_3 : State : *exclude*(State State)
SU_4 : State : *state*(all)

Assume the training set comes with such hybrid trees.

How to learn under a generative model?

states

not

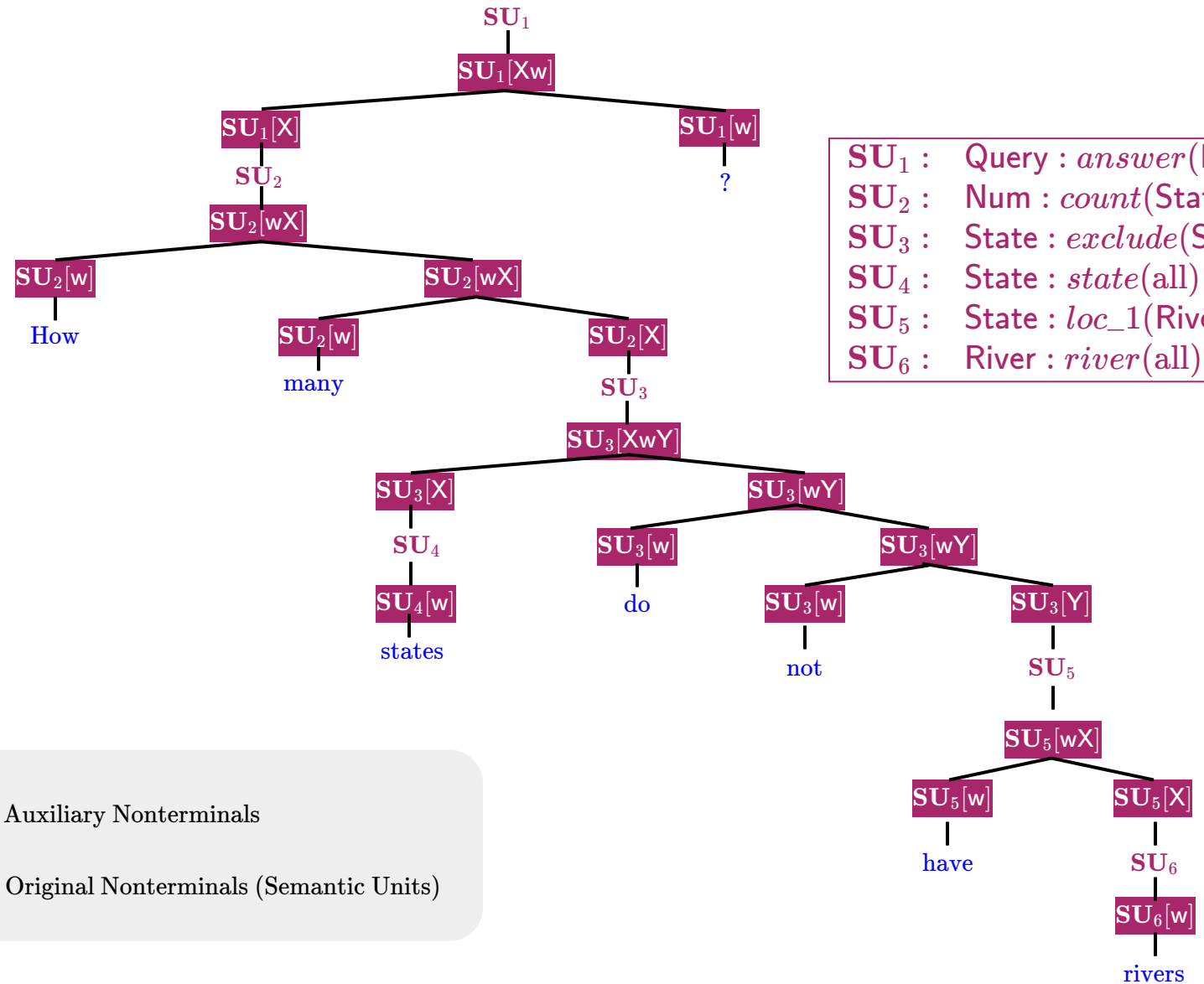


Similar to PCFG learning!

$SU_2[w]$ Auxiliary Nonterminals

SU_1 Original Nonterminals (Semantic Units)

Hybrid Tree

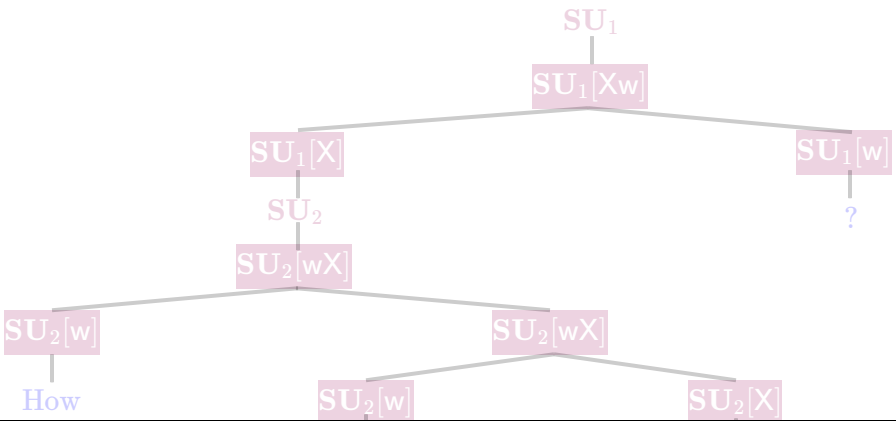


SU_1 :	Query : $answer$(Num)
SU_2 :	Num : $count$(State)
SU_3 :	State : $exclude$(State State)
SU_4 :	State : $state$(all)
SU_5 :	State : loc_1(River)
SU_6 :	River : $river$(all)

$SU_2[w]$ Auxiliary Nonterminals

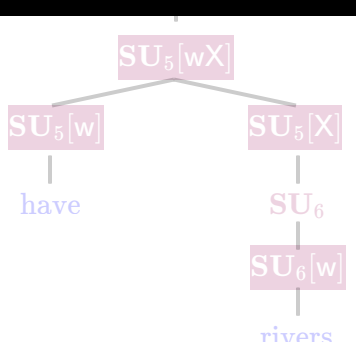
SU_1 Original Nonterminals (Semantic Units)

Hybrid Tree



SU_1 : Query : $answer$(Num)
SU_2 : Num : $count$(State)
SU_3 : State : $exclude$(State State)
SU_4 : State : $state$(all)
SU_5 : State : loc_1(River)

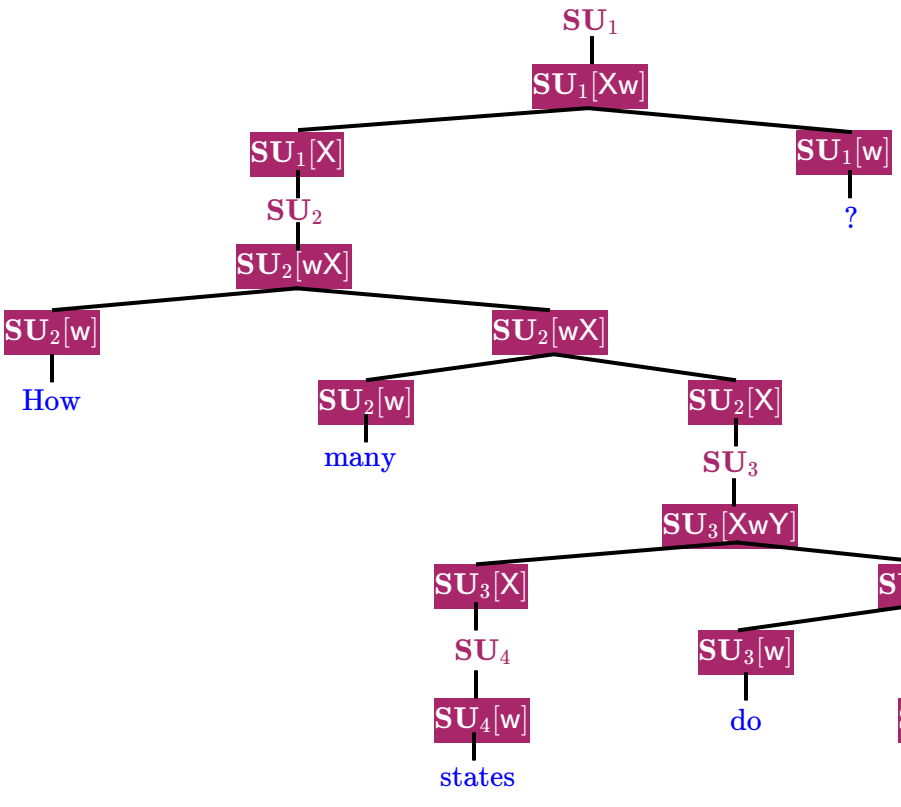
Unfortunately…
We don't know which alignment is correct!
What shall we do?



$SU_2[w]$ Auxiliary Nonterminals

SU_1 Original Nonterminals (Semantic Units)

Hybrid Tree



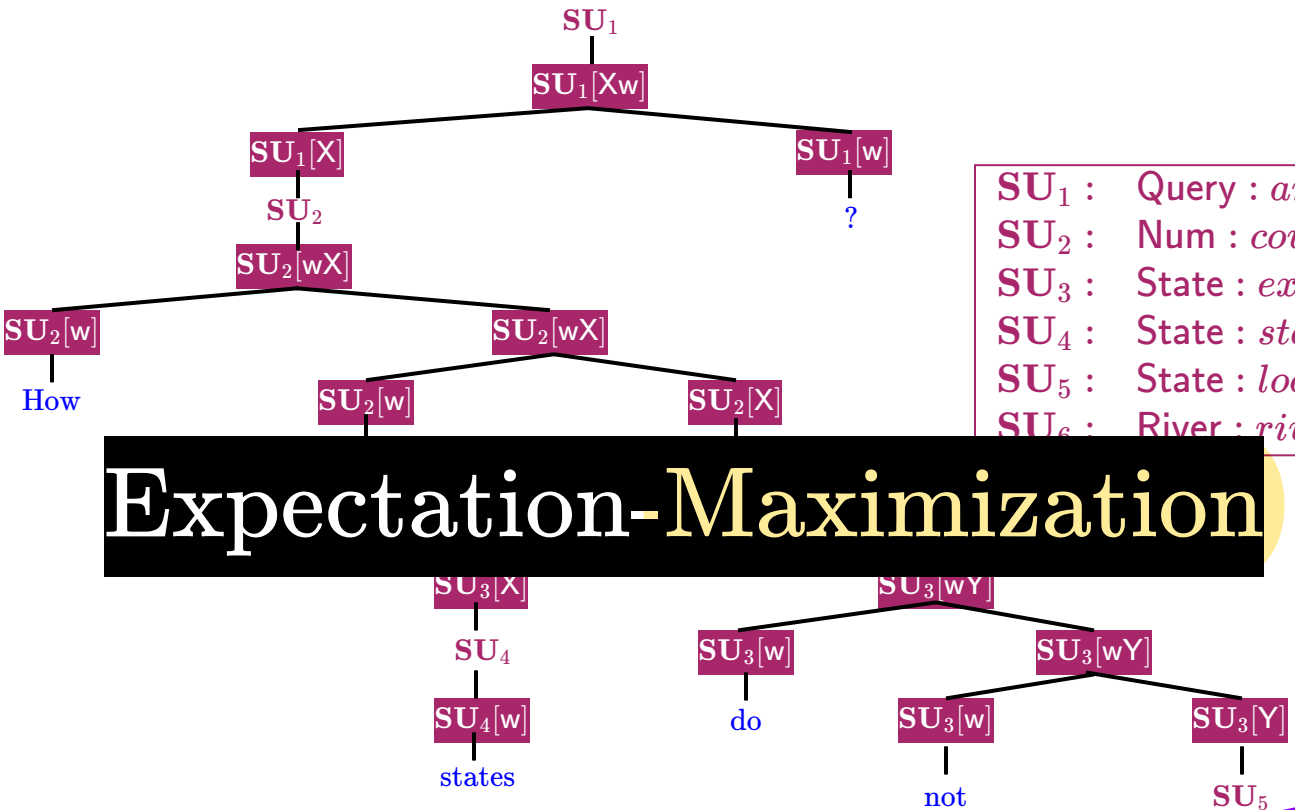
SU_1 :	Query : $answer$(Num)
SU_2 :	Num : $count$(State)
SU_3 :	State : $exclude$(State State)
SU_4 :	State : $state$(all)
SU_5 :	State : loc_1(River)
SU_6 :	River : $river$(all)

$SU_2[w]$ Auxiliary Nonterminals

SU_1 Original Nonterminals (Semantic Units)

Such hybrid trees are not given. We shall model them as hidden variables!

Hybrid Tree



SU_1 : Query : $answer$(Num)
SU_2 : Num : $count$(State)
SU_3 : State : $exclude$(State State)
SU_4 : State : $state$(all)
SU_5 : State : loc_1(River)
SU_6 : River : $river$(all)

Expectation-Maximization

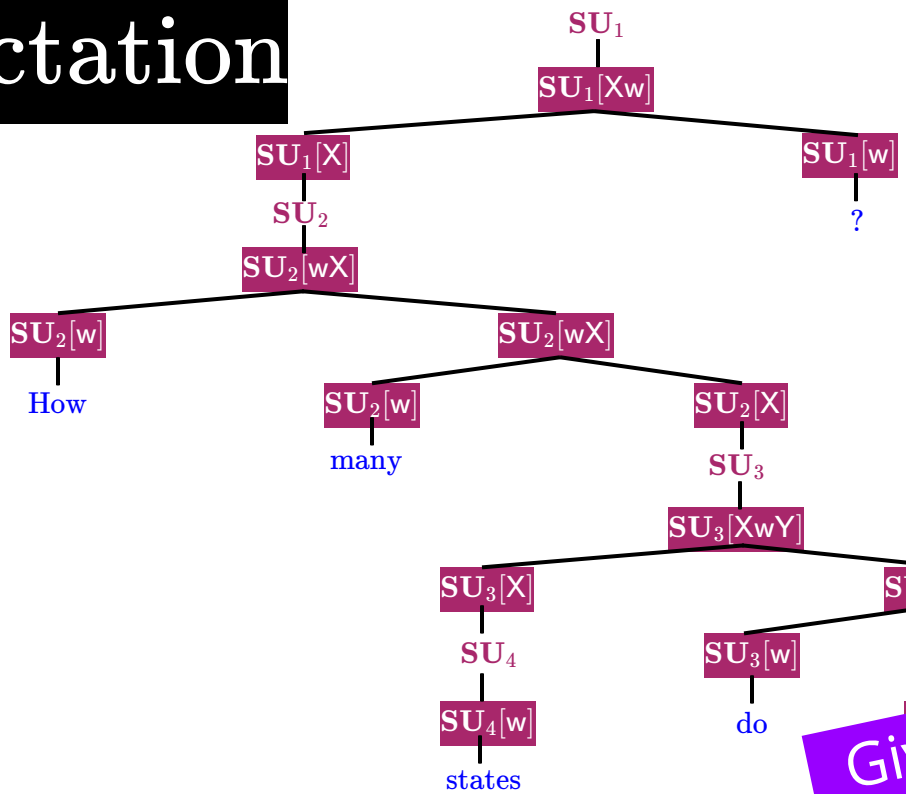
$SU_2[w]$ Auxiliary Nonterminals

SU_1 Original Nonterminals (Semantic Units)

Such hybrid trees are not given. We shall model them as hidden variables!

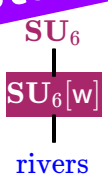
Hybrid Tree

Expectation



SU_1 :	Query : $answer$(Num)
SU_2 :	Num : $count$(State)
SU_3 :	State : $exclude$(State State)
SU_4 :	State : $state$(all)
SU_5 :	State : loc_1(River)
SU_6 :	River : $river$(all)

Given the current model parameters (probabilities of the PCFG rules), find the most probable hybrid tree for each instance!

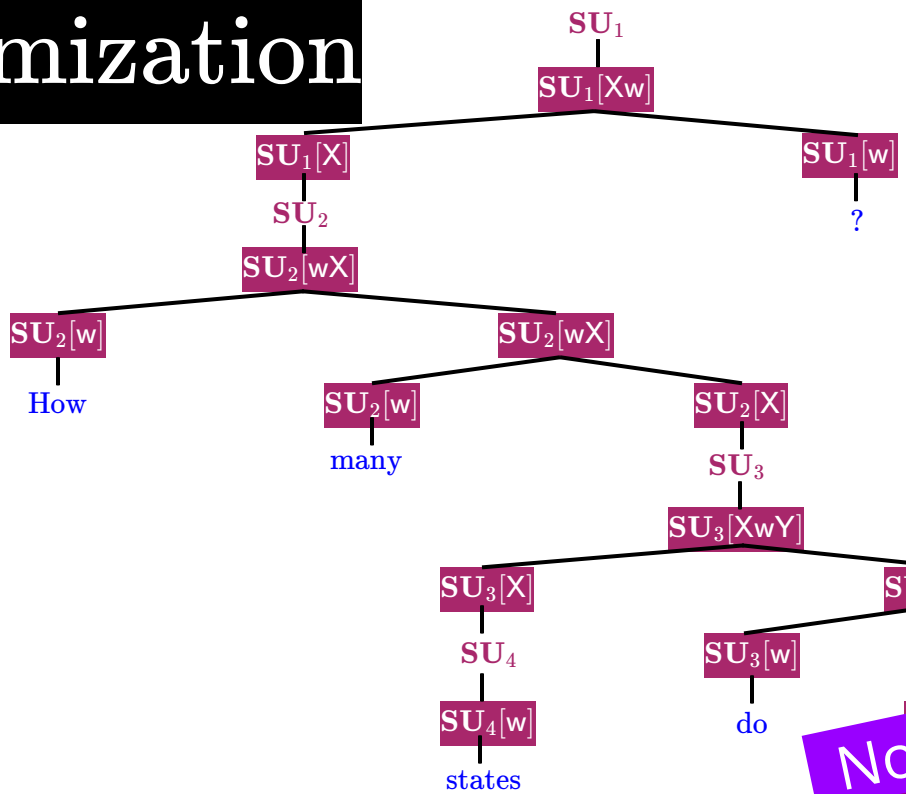


$SU_2[w]$ Auxiliary Nonterminals

SU_1 Original Nonterminals (Semantic Units)

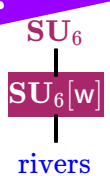
Hybrid Tree

Maximization



SU_1 :	Query : $answer$(Num)
SU_2 :	Num : $count$(State)
SU_3 :	State : $exclude$(State State)
SU_4 :	State : $state$(all)
SU_5 :	State : loc_1(River)
SU_6 :	River : $river$(all)

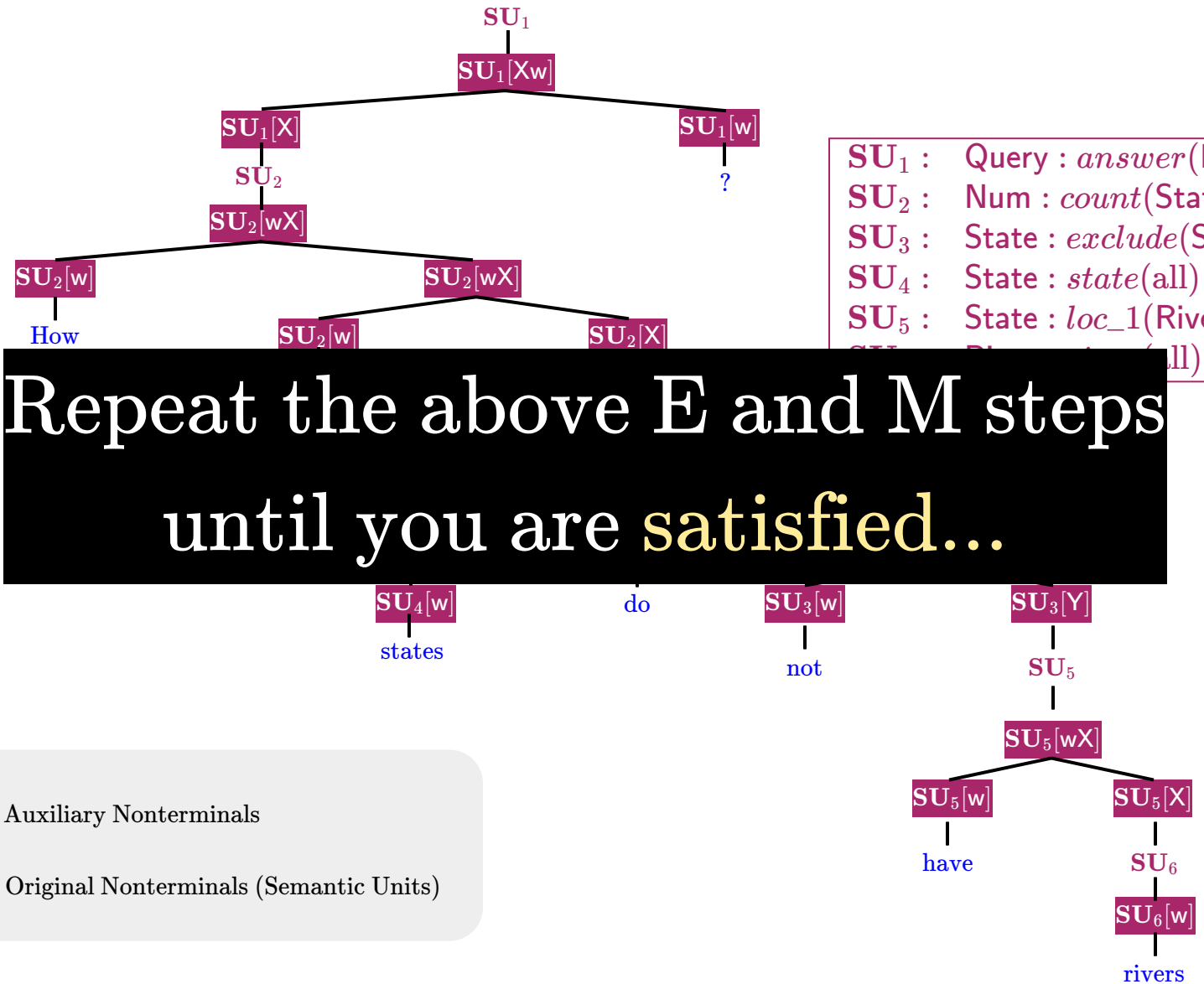
Now, based on the newly found hybrid trees for all instances, re-estimate the probabilities for the PCFG rules!



$SU_2[w]$ Auxiliary Nonterminals

SU_1 Original Nonterminals (Semantic Units)

Hybrid Tree



SU_1 : Query : *answer*(Num)
SU_2 : Num : *count*(State)
SU_3 : State : *exclude*(State State)
SU_4 : State : *state*(all)
SU_5 : State : *loc_1*(River)
ll)

Repeat the above E and M steps until you are satisfied...

$SU_2[w]$ Auxiliary Nonterminals

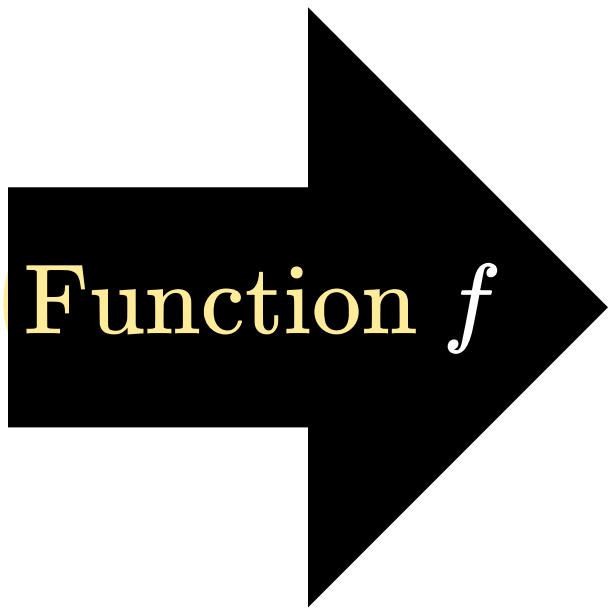
SU_1 Original Nonterminals (Semantic Units)

Question

How do we do soft EM for hybrid trees?

Semantic Parsing

Natural Language Sentence

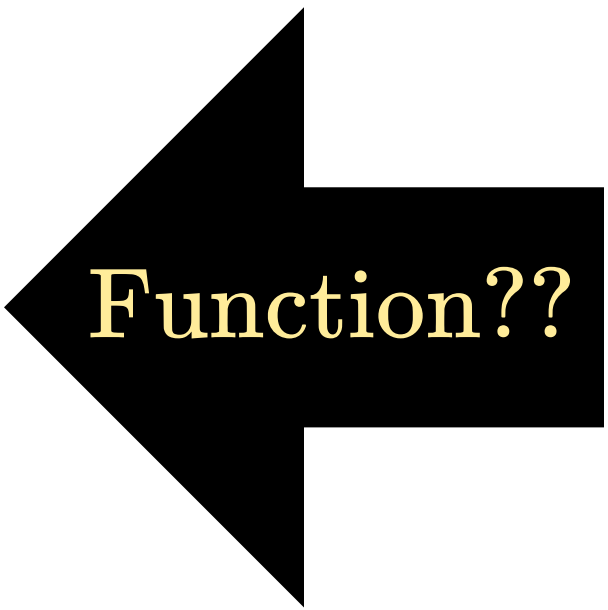


Semantic Representation

What if we work on the other direction?

Language Generation

Natural Language Sentence



Semantic Representation

What would be the major challenges?

Language Generation

We shall work out a way to perform proper evaluations!

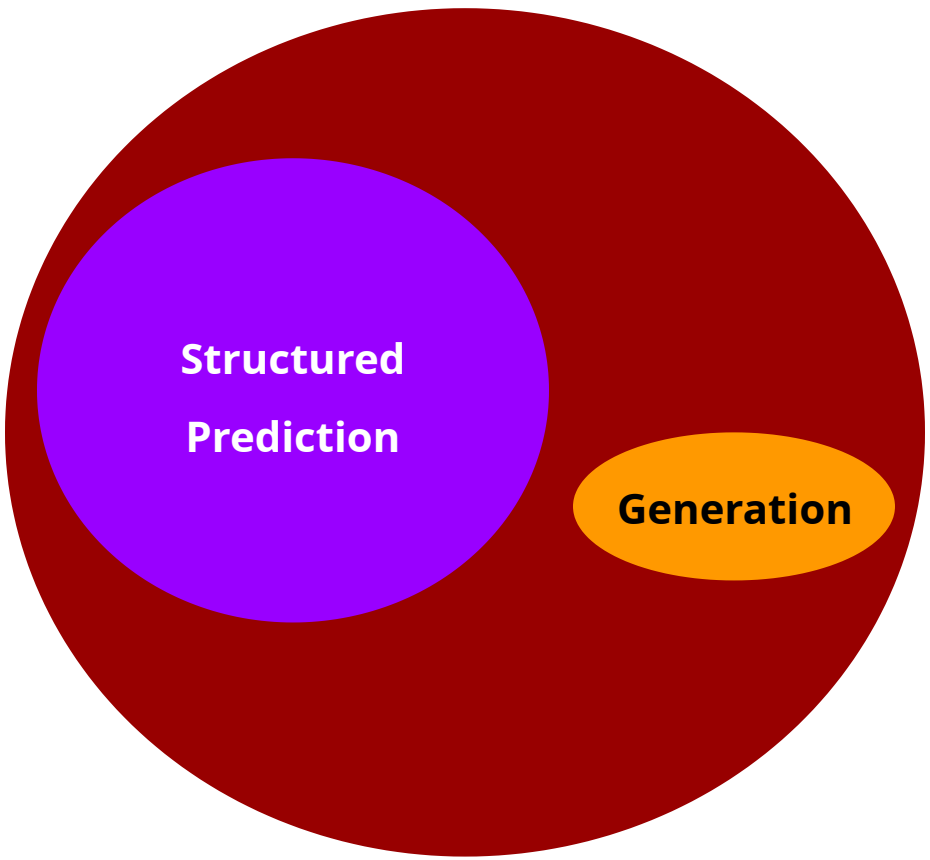
Natural Language Sentence

This should not be a function any more!

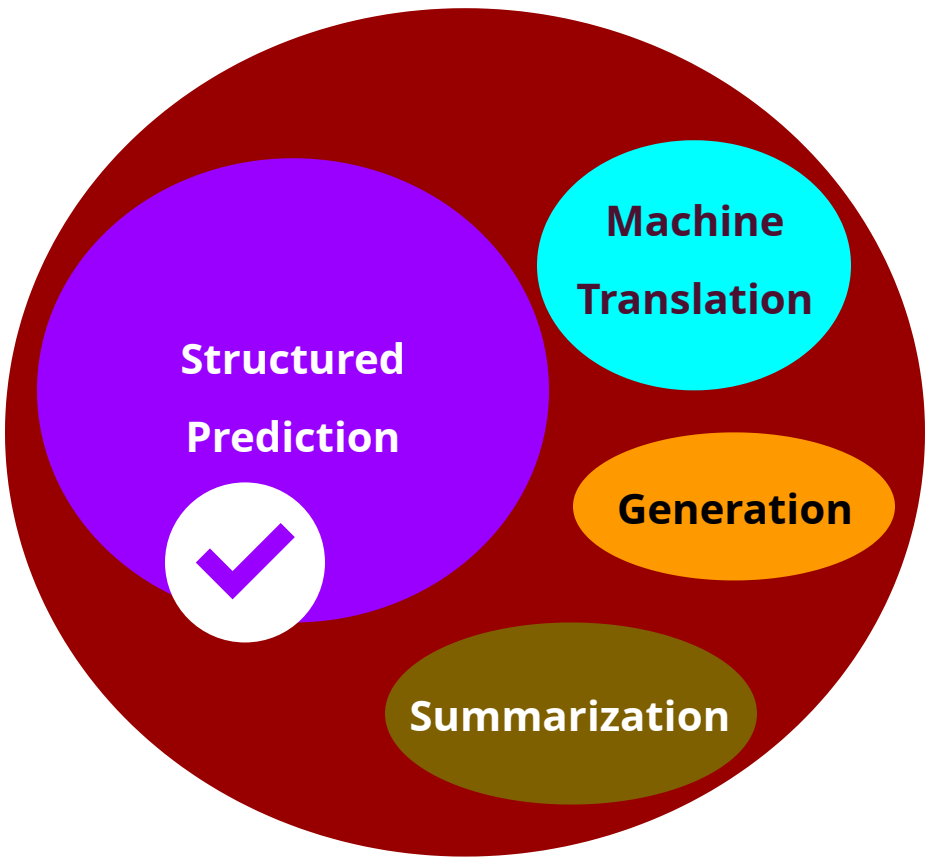
Semantic Representation

What would be the major challenges?

Tasks in NLP



Tasks in NLP



Supervised