Efficient Semantic Deduction and Approximate Matching over Compact Parse Forests

Roy Bar-Haim[1], Jonathan Berant[2], Ido Dagan[1], Iddo Greental[2], Shachar Mirkin[1], Eyal Shnarch[1], Idan Szpektor[1]

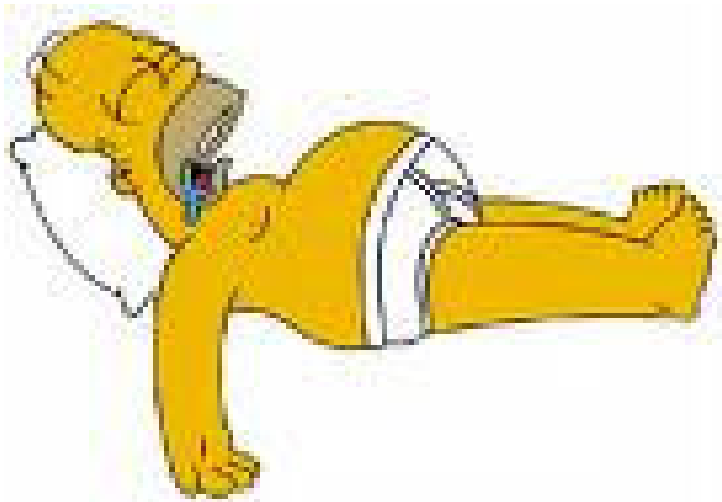
[1]Bar-Ilan University
[2]Tel-Aviv University

Our RTE-3 Recipe

1. Represent *t* and *h* as parse trees
2. Try to *prove h* from *t based on available knowledge*
3. Measure the "distance" from *h* to the generated consequents of *t.*
4. Determine entailment based on distance
5. Cross your fingers …

I forgot one thing...

6. Wait a few hours...



This year: hours → minutes

Textual Entailment – Where do we want to get to?

- Long term goal:
  robust semantic inference engine
  - To be used as a generic component in text understanding applications
  - Encapsulating all required inferences
- Based on compact, well defined formalism:
  - Knowledge representation
  - Inference mechanisms

Our Inference Formalism
Bar-Haim et al., AAAI-07 & RTE3

- A ***proof system*** over parse trees:
  - Represents diverse kinds of semantic knowledge uniformly as entailment (inference) rules
  - Allows unified inference mechanism
- Analogous to logic proof systems:

Propositions	Parse Trees
Inference Rules	Tree transformations
Proof	A sequence of trees, generated by rule application

- Given *Text (T)* and *Hypothesis (H)*, try to *generate H* from *T*
- Formalizes common transformation-based approaches

A Sample Proof

Text: McCain congratulated the elected president, Barack Obama.

Consequent	Entailment Rule
⇒Barack Obama is the elected president.	*Apposition* (syntactic)
⇒Barack Obama won the elections	*X is the elected president* *→ X won the elections* (lexical-syntactic)
⇒The Democratic nominee won the elections.	*Barack Obama* *→ the Democratic nominee* (lexical)

Hypothesis: The Democratic nominee won the elections.

Explicit Generation of Consequents

Text: Children like candies.

Rules: *children→kids ; like→enjoy ; candies→sweets*

Consequents:

Kids like candies.

Kids enjoy candies.

Children like sweets.

…

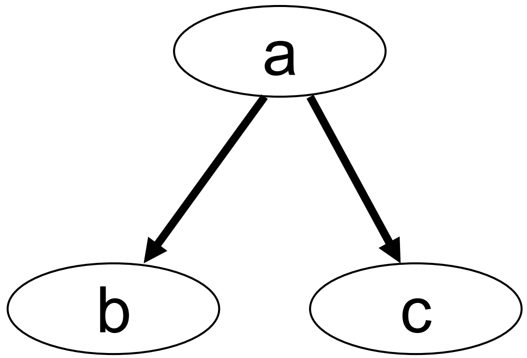
2^3 alternatives!

Exponential explosion!

We need compact representation of consequents!

Intuitive Solution: Add only the Entailed Part

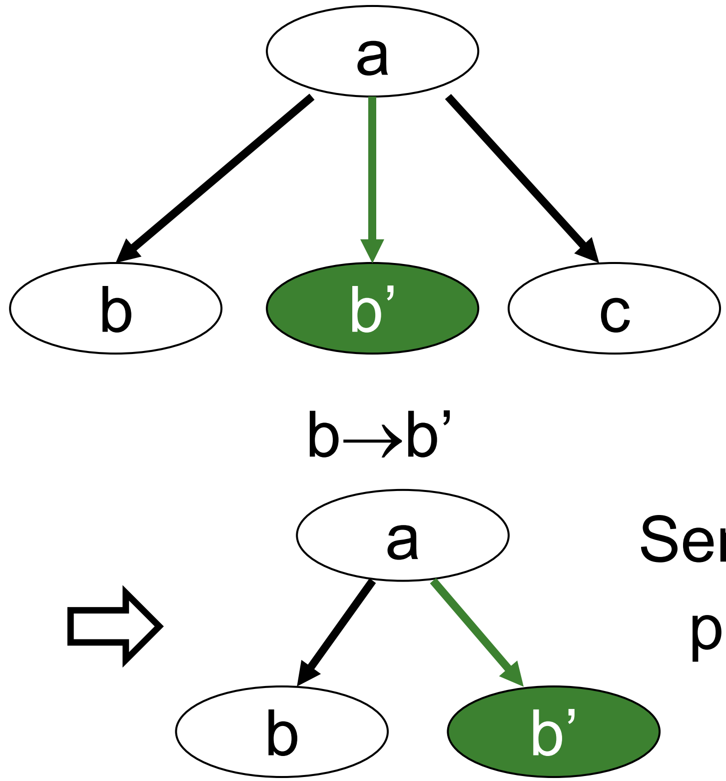
- Resulting structure is a union of sentences (trees)
- Each of its sub-structures is assumed to be entailed



b→b’

Intuitive Direction: Add only the Entailed Part

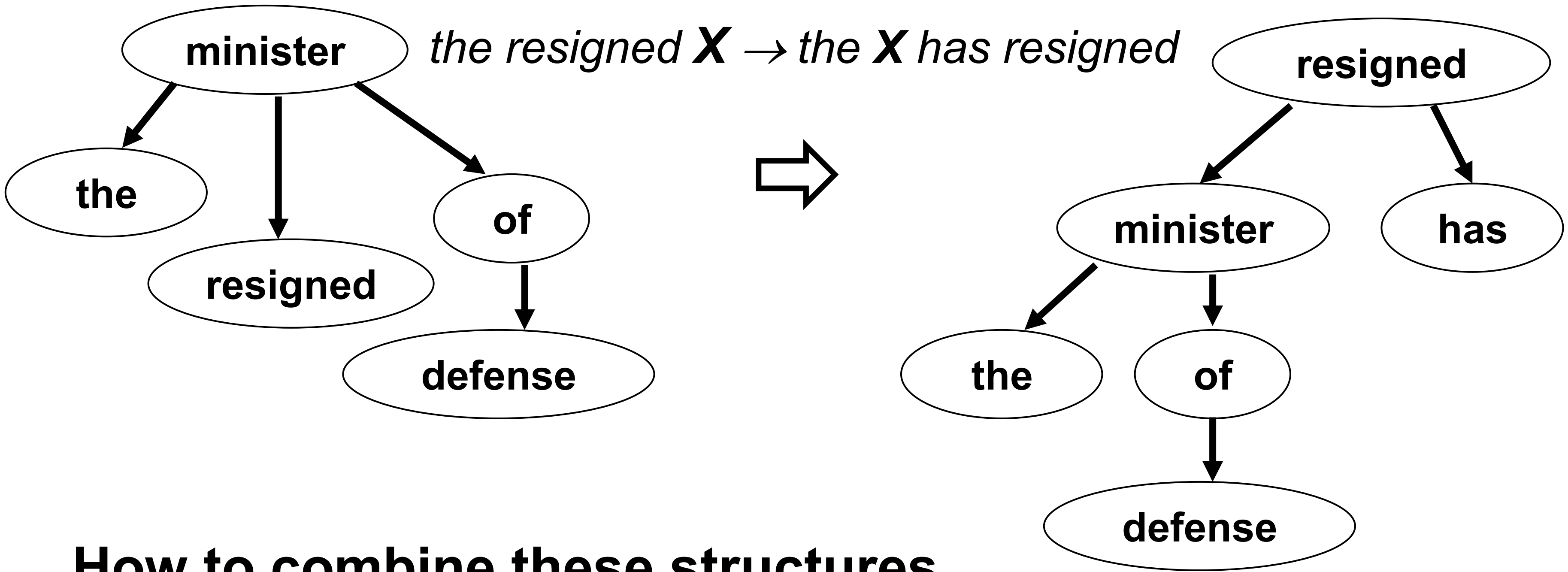
- Resulting structure is a union of sentences (trees)
- Each of its sub-structures is assumed to be entailed



Semantics not preserved!

How to share common variables?

the resigned ***X*** → *the* ***X*** *has resigned*



How to combine these structures efficiently (and correctly)?

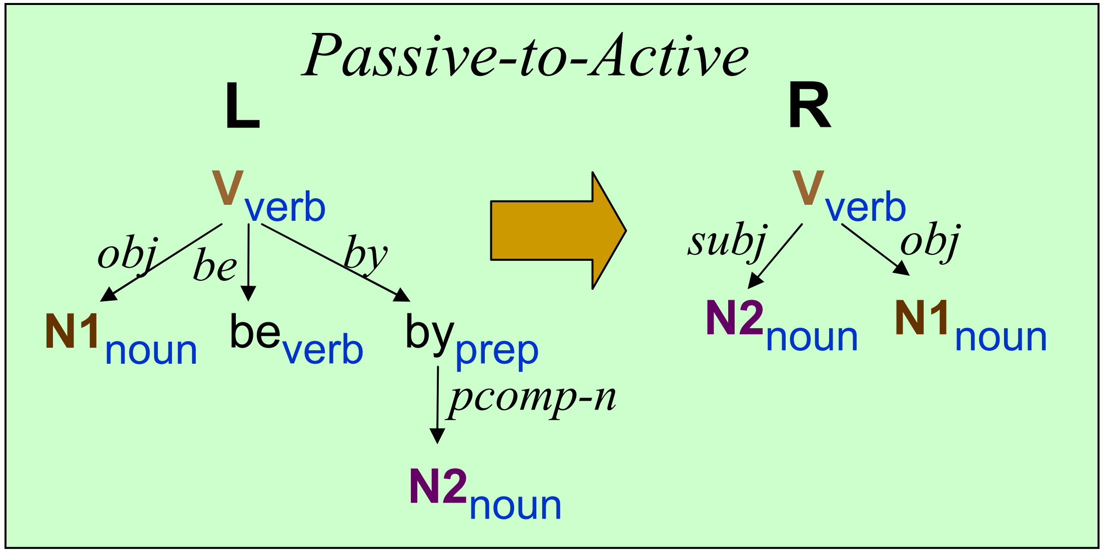
Talk Outline

- A novel data structure, ***compact forest,*** for efficient inference
  - Preserves semantics
- New sources of entailment rules
  - Scale made feasible by compact forest
- Approximate matching
- Results

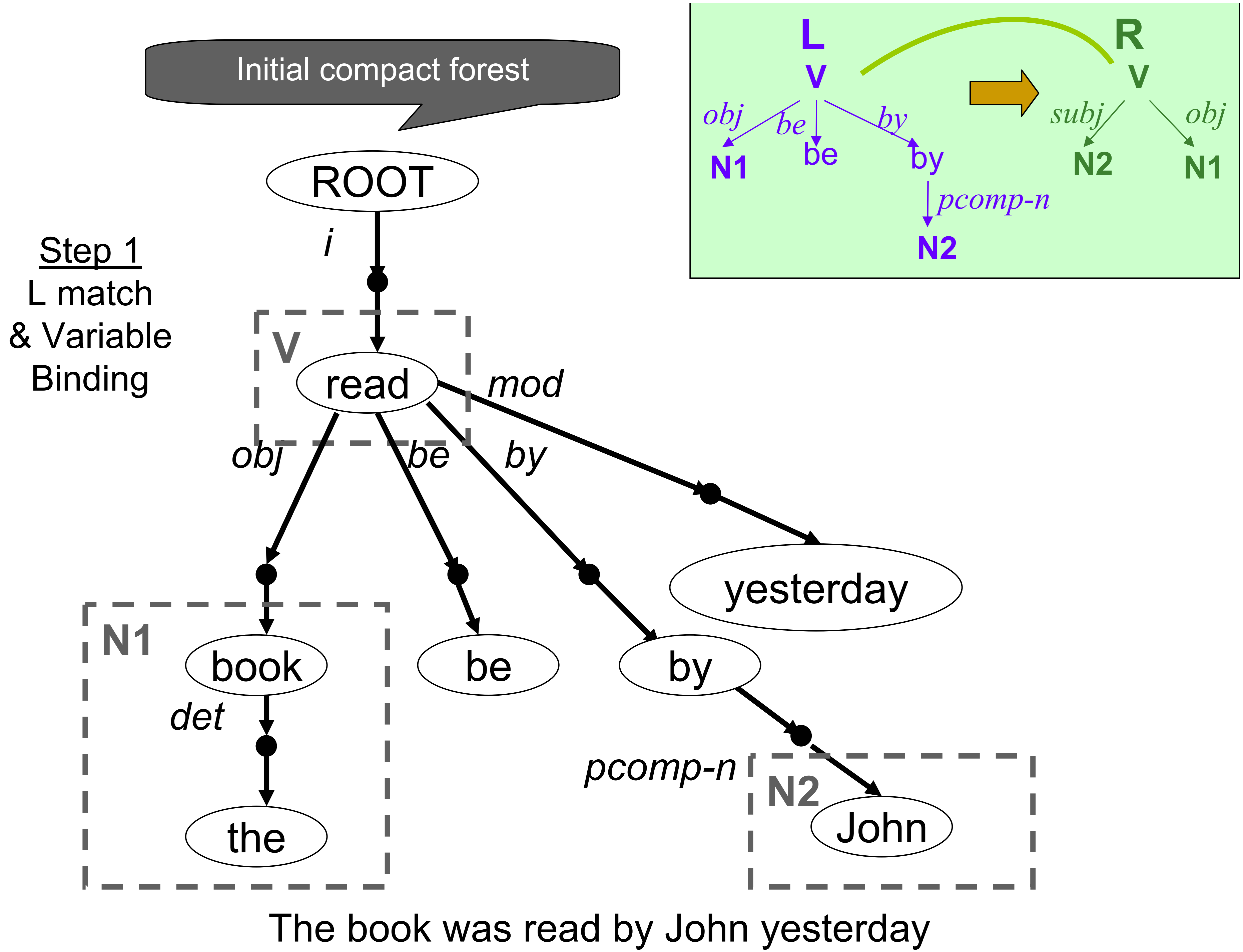
The Compact Forest

The Compact Forest – a Walk-Through Example

- A sample rule: passive-to active transformation

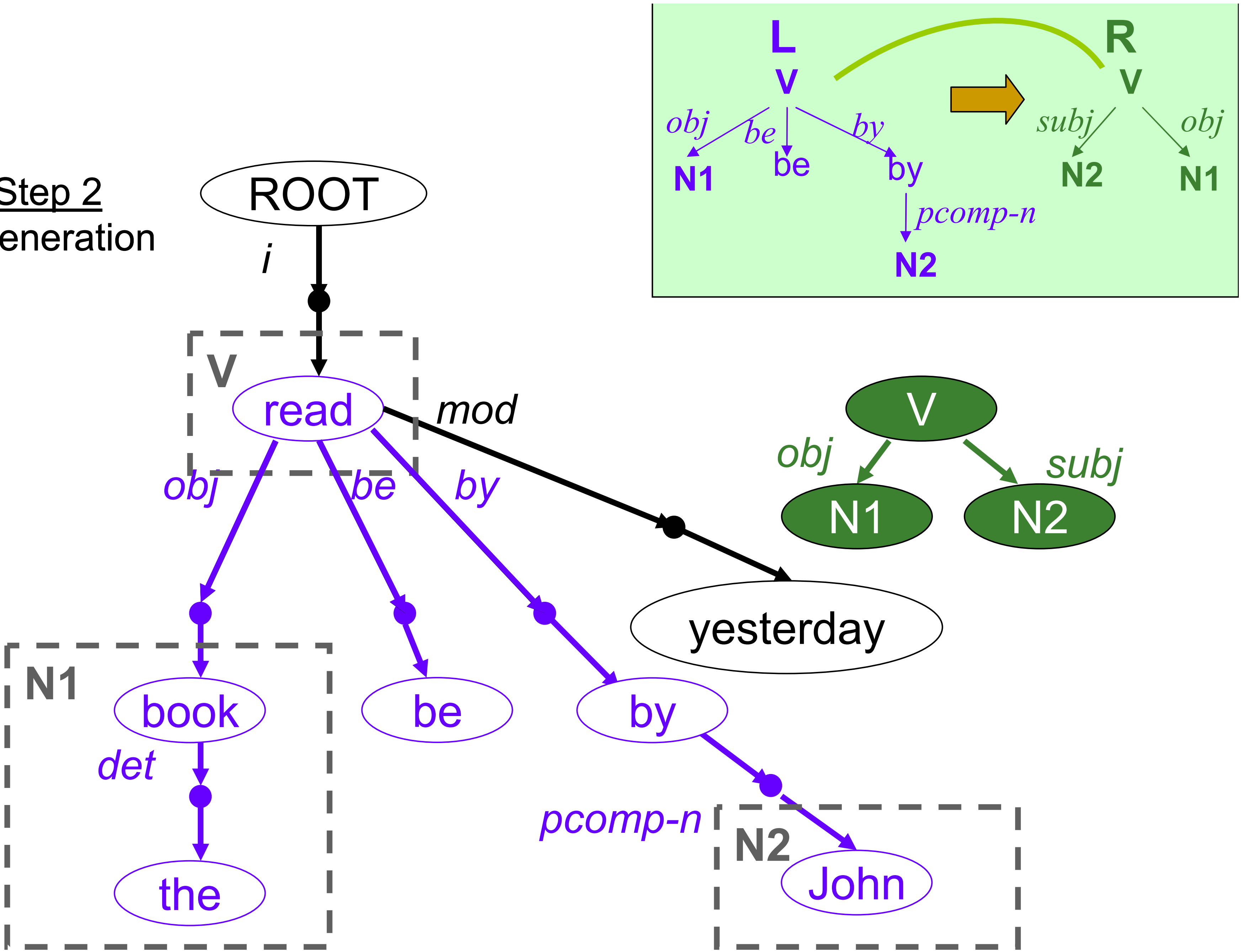


The book was read by John yesterday → John read the book yesterday



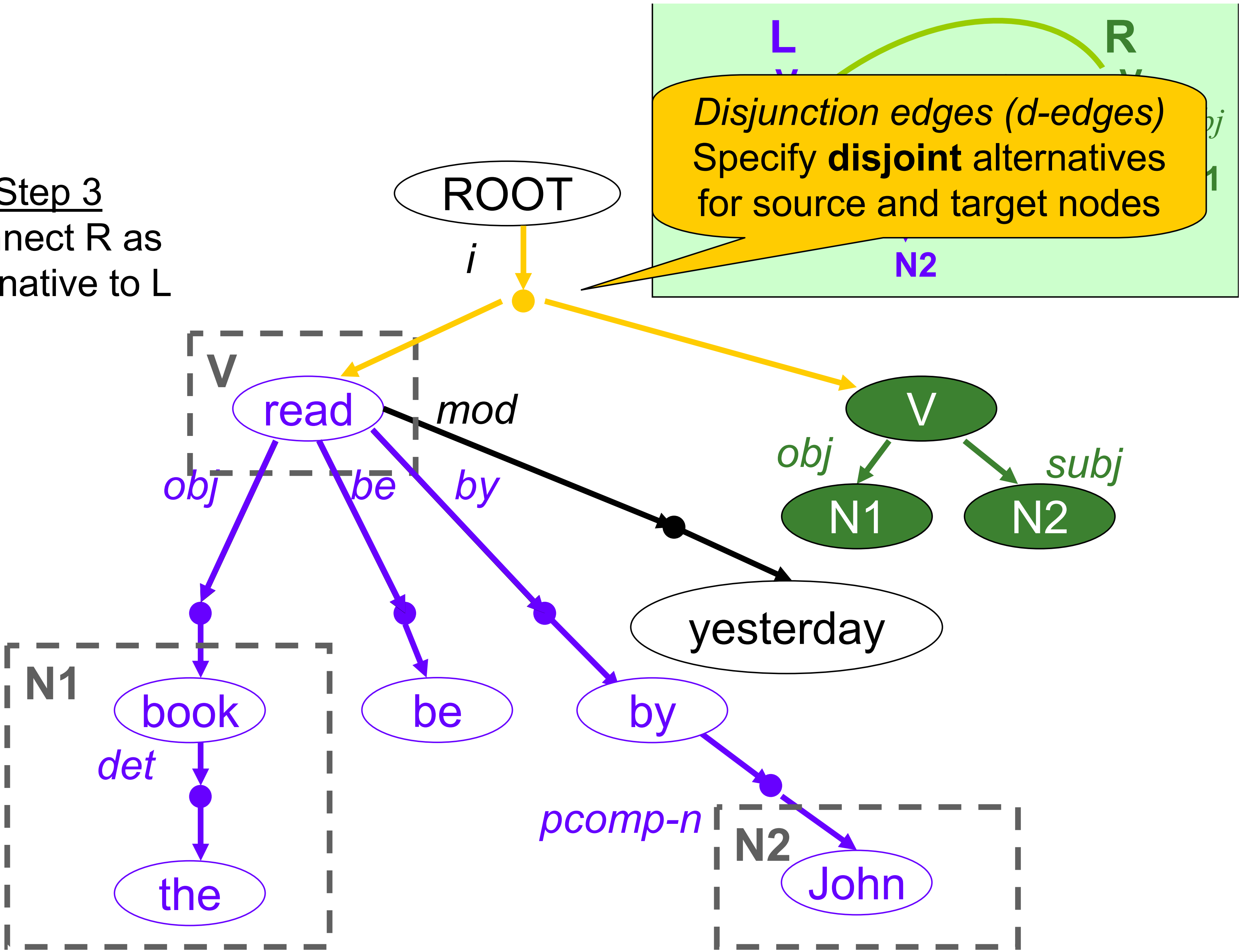
Initial compact forest
Step 1
L match
& Variable
Binding
ROOT
i
V
read
mod
obj
be
by
book
det
the
N1
be
by
yesterday
pcomp-n
N2
John
L
V
obj
be
by
N1
be
by
pcomp-n
N2
R
V
subj
obj
N2
N1
The book was read by John yesterday

Step 2
R generation



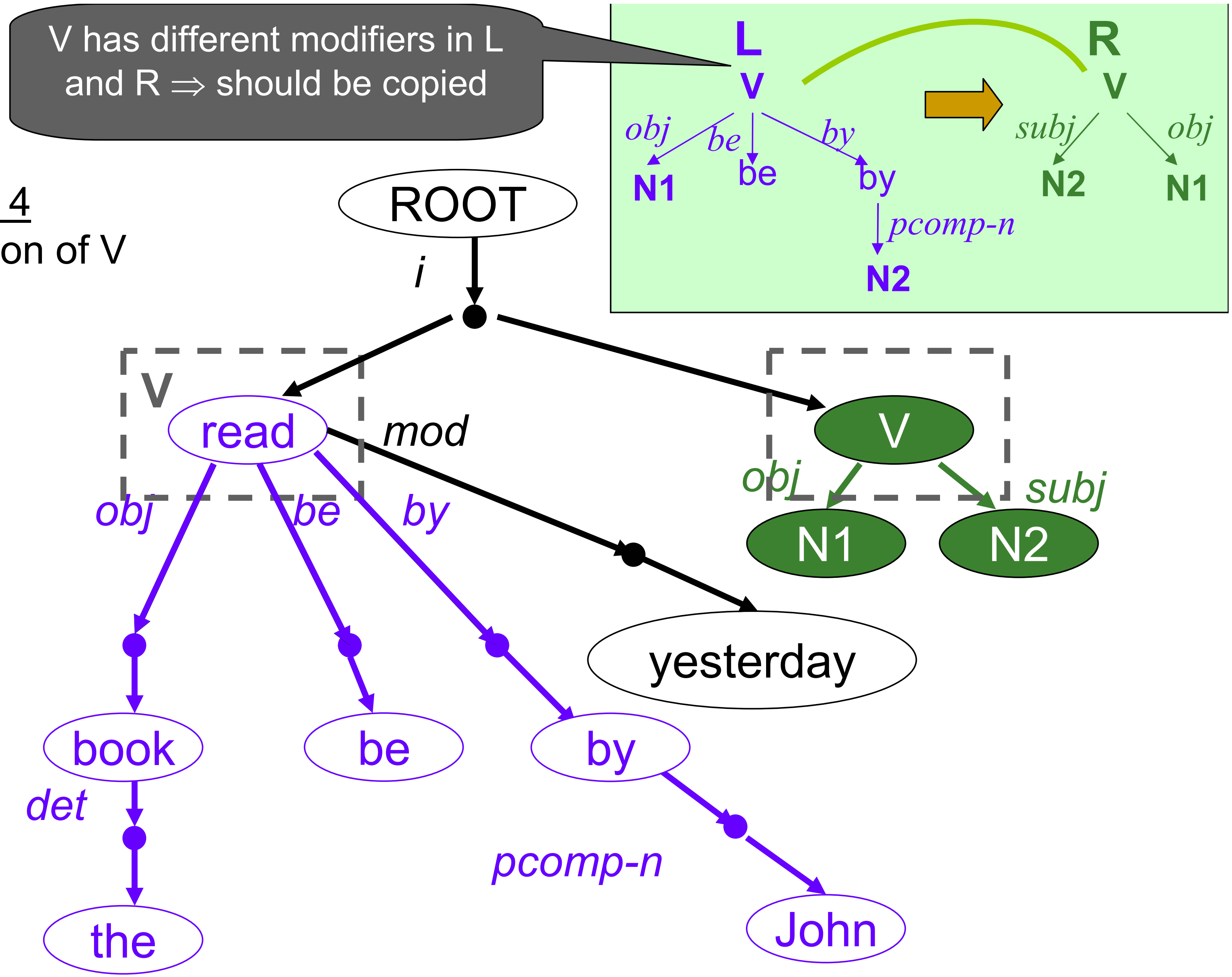
The book was read by John yesterday

Step 3
Connect R as alternative to L



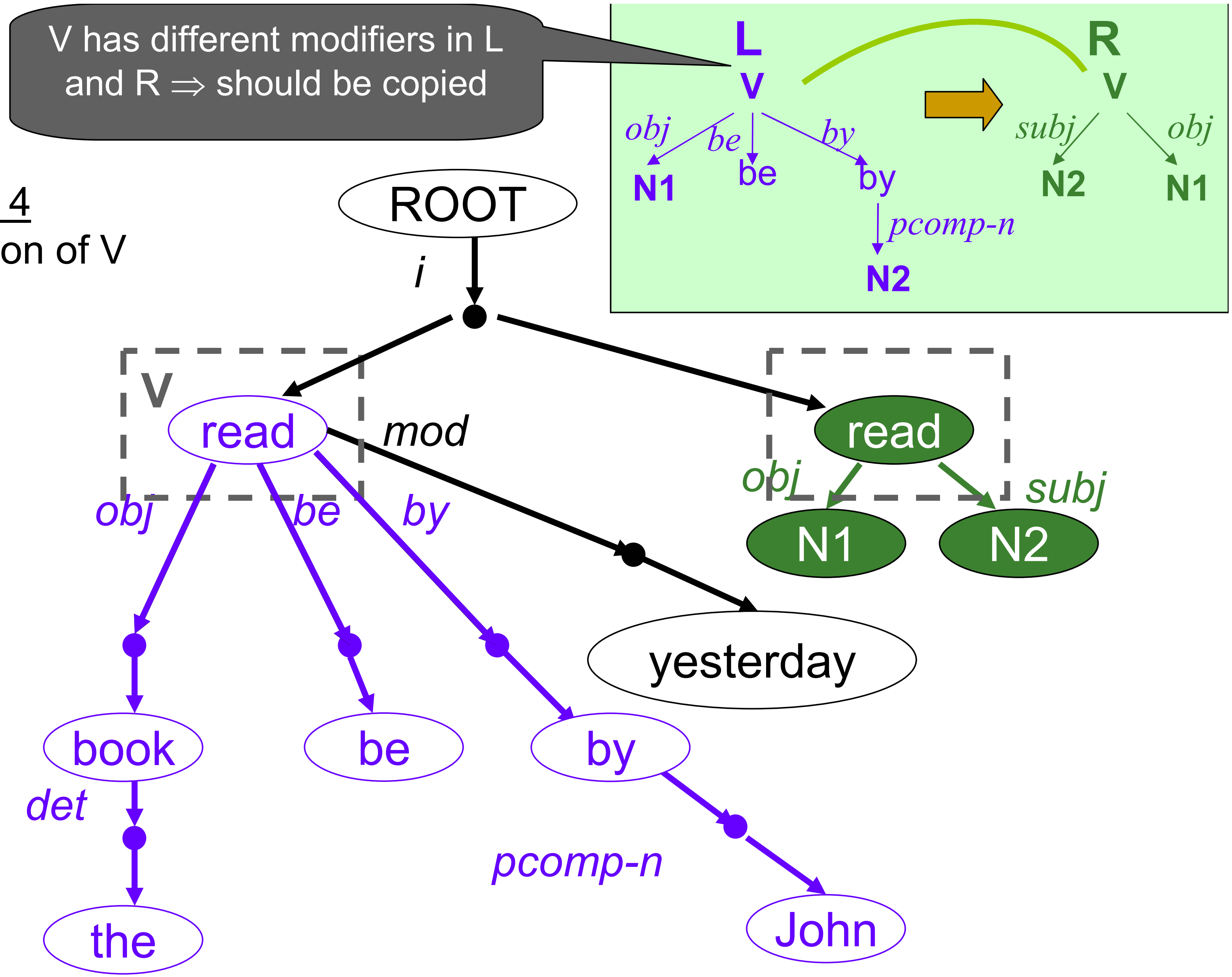
The book was read by John yesterday ⇒ N1 V N2

Step 4
instantiation of V



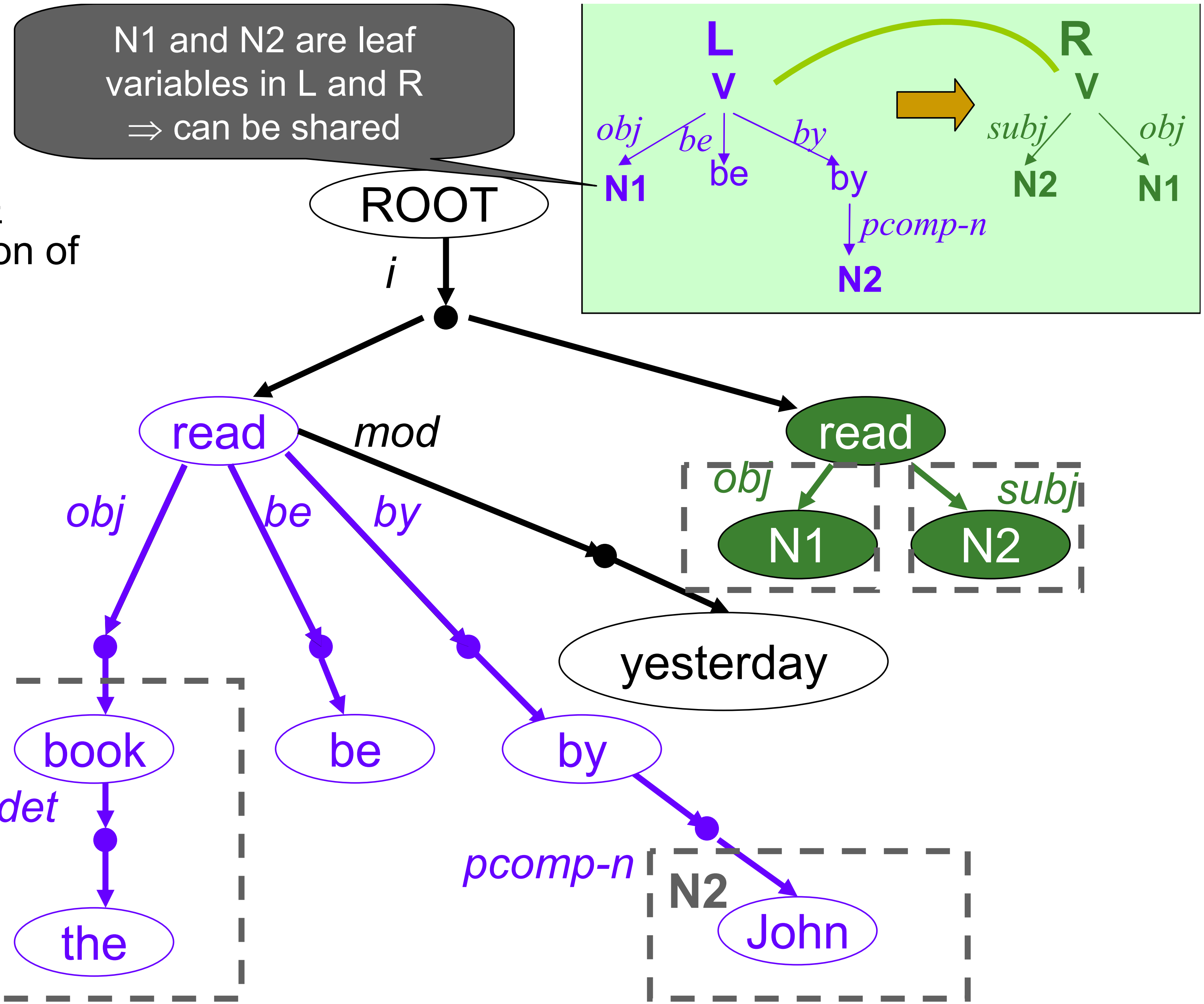
The book was read by John yesterday ⇒ N1 V N2

Step 4
instantiation of V



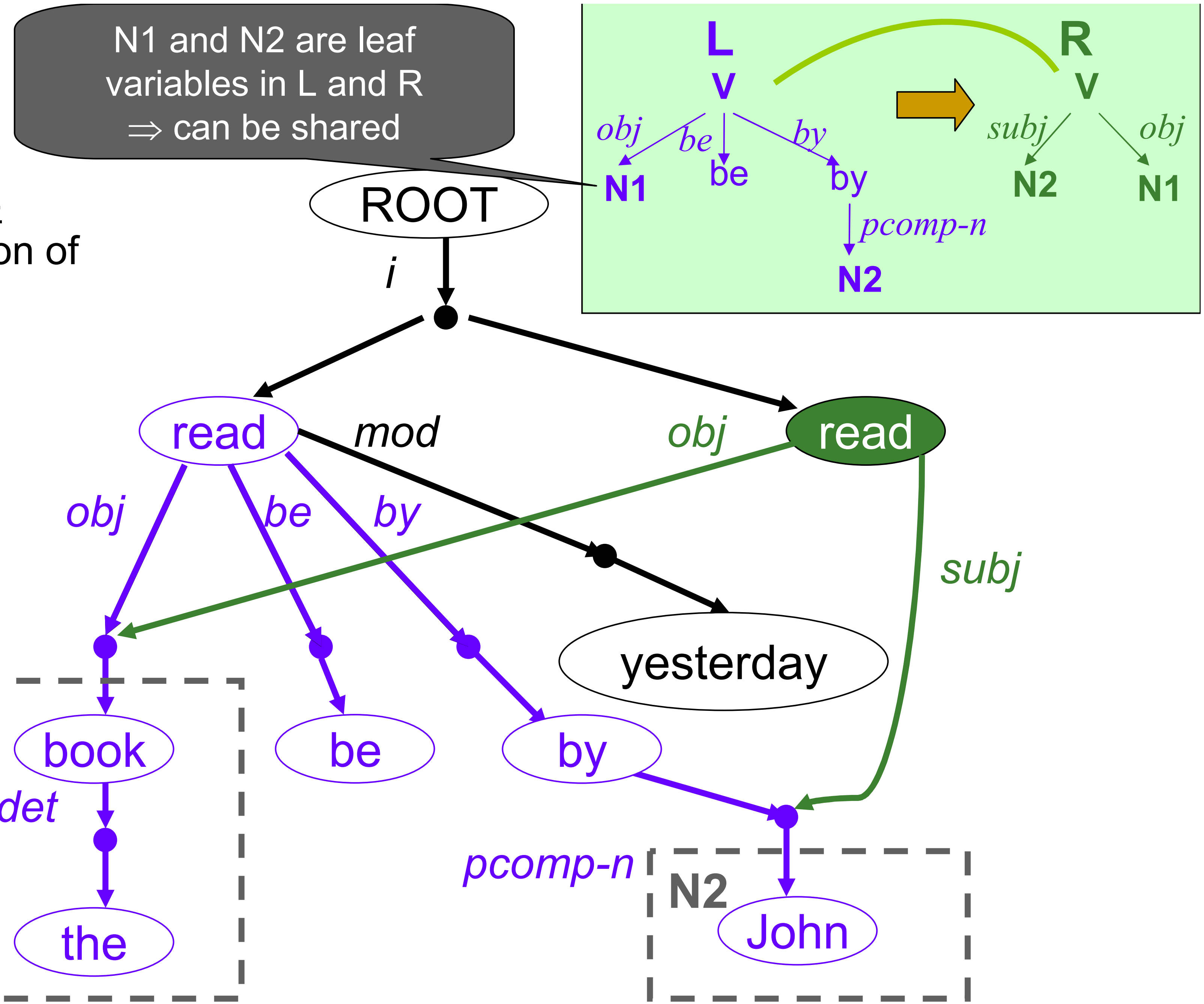
The book was read by John yesterday ⇒ N1 read N2

Step 4
instantiation of
N1,N2



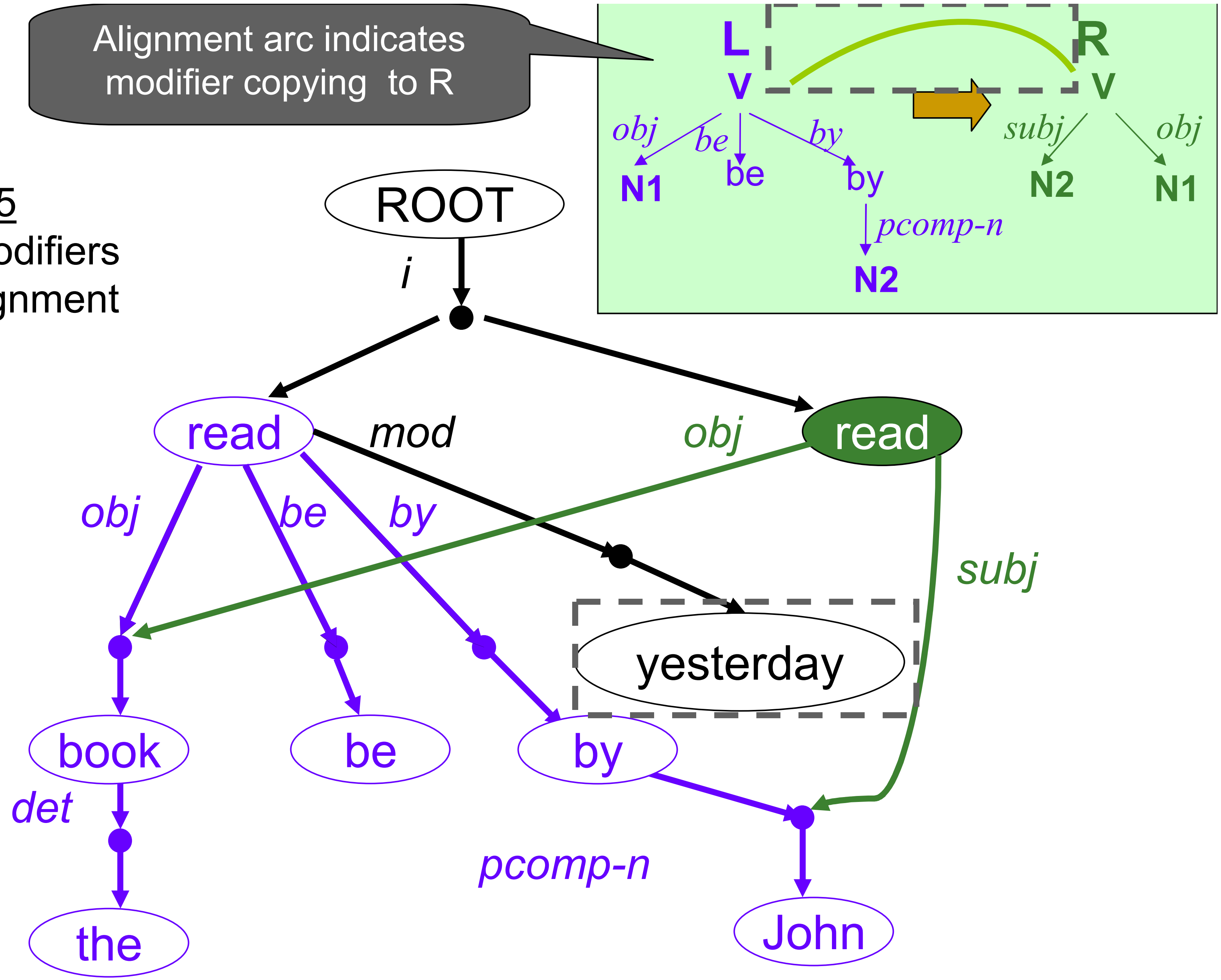
The book was read by John yesterday ⇒ N1 read N2

Step 4
instantiation of N1,N2



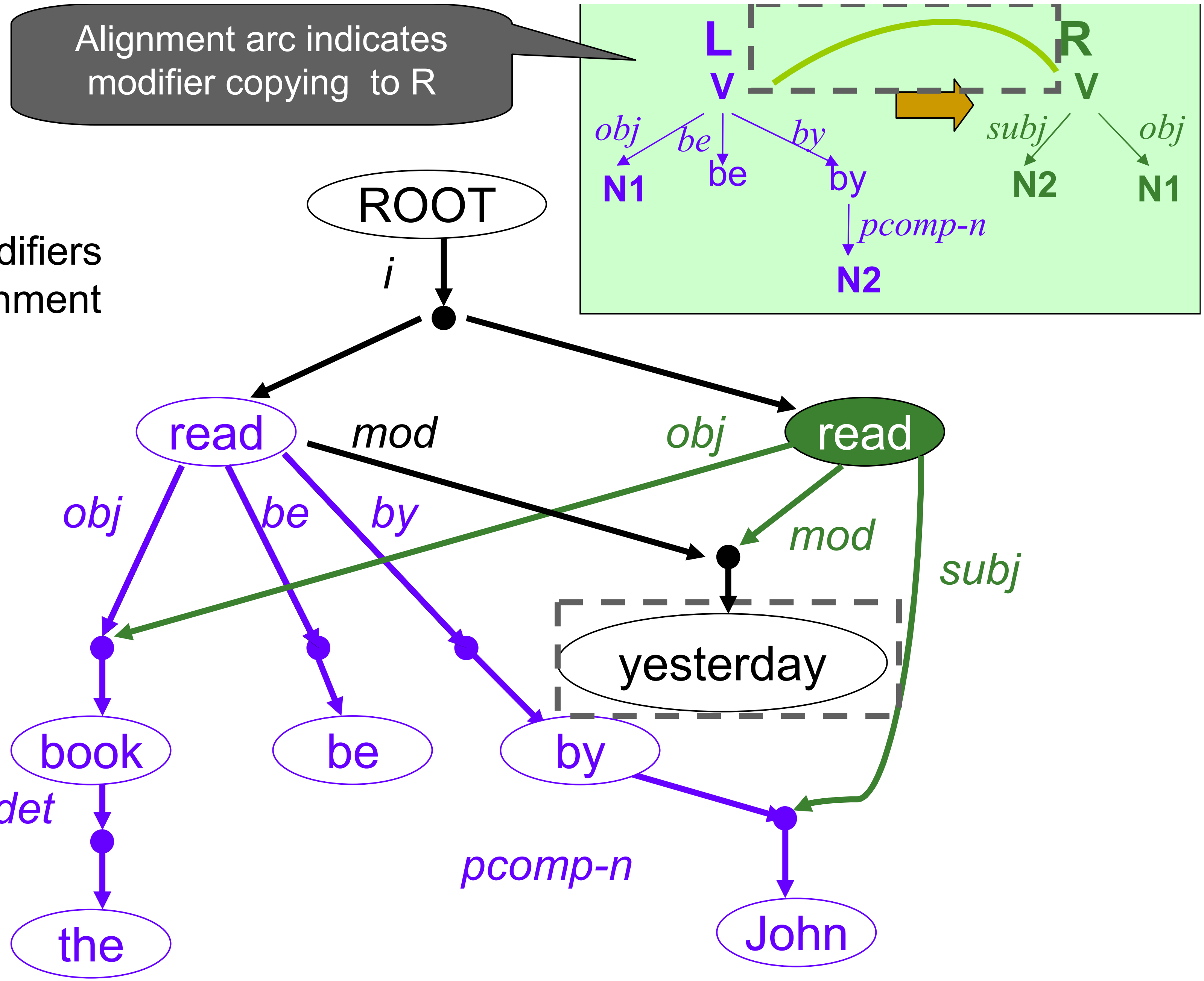
The book was read by John yesterday ⇒ John read the book

Step 5
Sharing modifiers through alignment



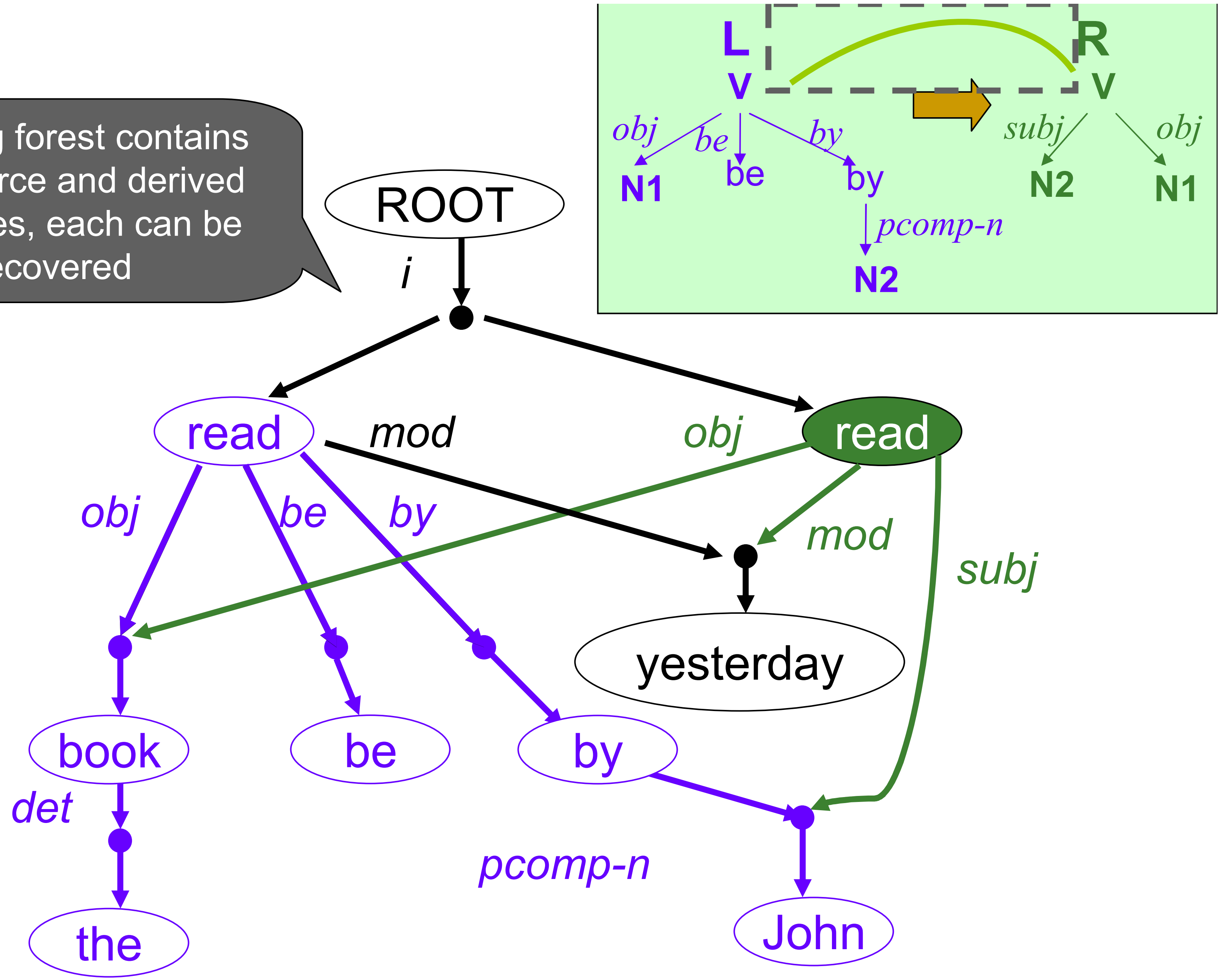
The book was read by John yesterday ⇒ John read the book

Step 5
Sharing modifiers through alignment



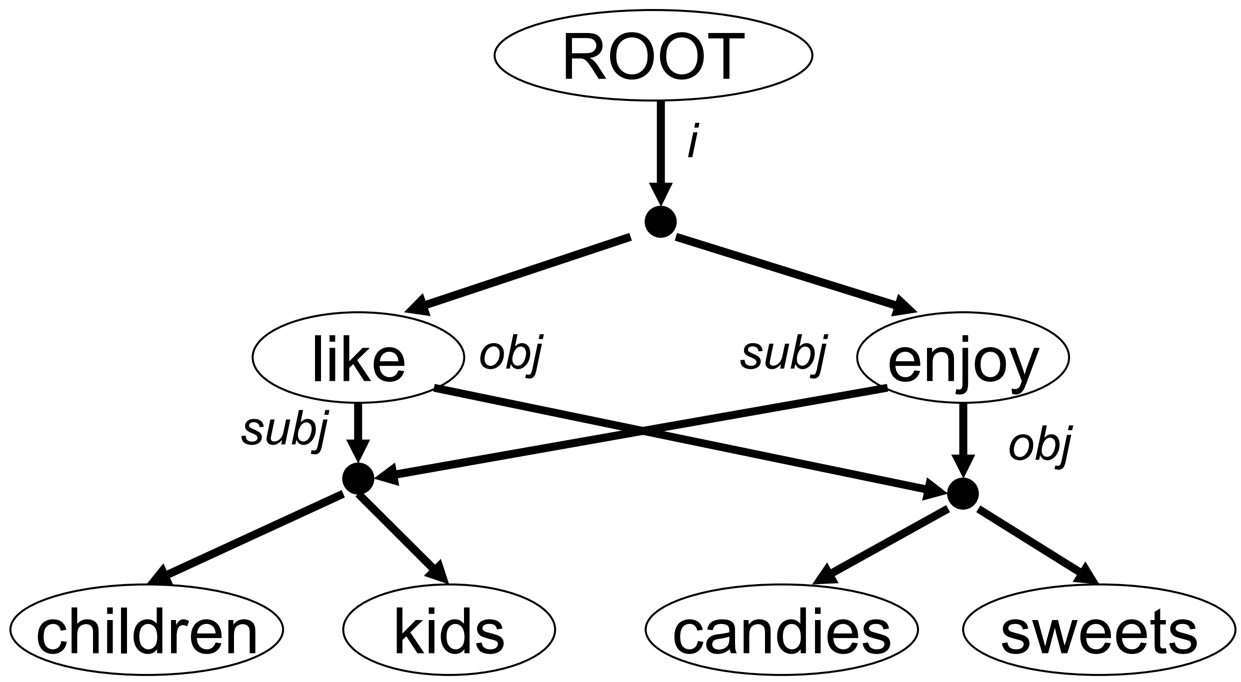
The book was read by John yesterday ⇒ John read the book yesterday

Resulting forest contains both source and derived sentences, each can be recovered



⇒An efficient data structure, preserving inference semantics

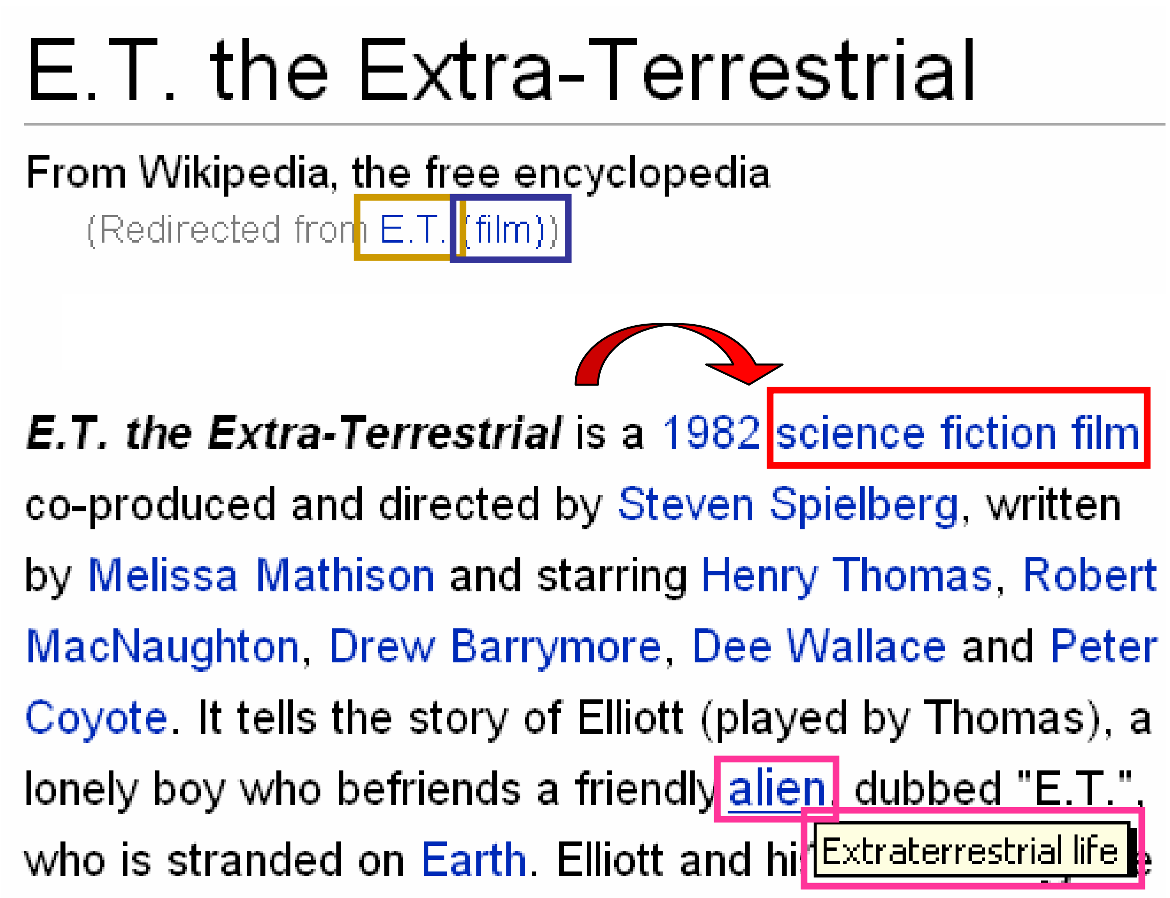
Children and Sweets – the Compact Version…



Complexity reduction from exponential to linear!

New Knowledge Sources

Extracting Lexical Rules from Wikipedia



E.T. the Extra-Terrestrial

From Wikipedia, the free encyclopedia
(Redirected from E.T. (film))

E.T. the Extra-Terrestrial is a 1982 science fiction film co-produced and directed by Steven Spielberg, written by Melissa Mathison and starring Henry Thomas, Robert MacNaughton, Drew Barrymore, Dee Wallace and Peter Coyote. It tells the story of Elliott (played by Thomas), a lonely boy who befriends a friendly alien, dubbed "E.T.", who is stranded on Earth. Elliott and hi Extraterrestrial life e

- ***Be-complement***
  Nominal complements of ‘be’
- ***Redirect***
  various terms to canonical title
- ***Parenthesis***
  used for disambiguation
- ***Link***
  Maps to a title of another article

Lexical-Syntactic Rules from Lexical Resources

- Entailment rules between predicates + argument mapping
- Combining information from various lexical resources:
  - **WordNet:** semantic relations (synonyms, hyponyms)
  - **VerbNet:** subcategorization frames for verbs
  - **Nomlex-Plus:**
    - Mapping between verbs and nominalizations (acquire ⇔ acquisition)
    - Subcategorization frames for nominalizations
  - Example:

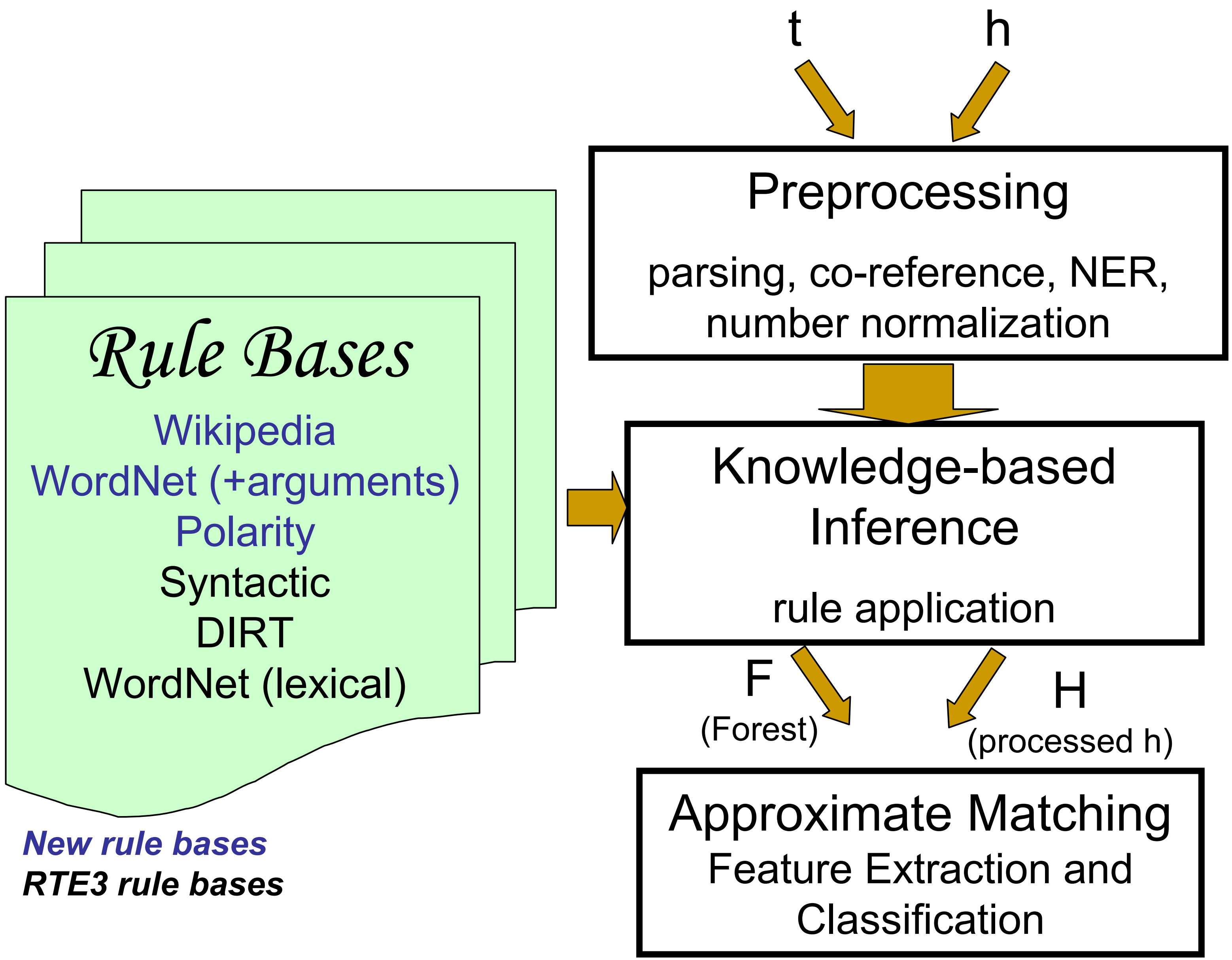
    homicide of X by Y ⇔ killing of X by Y ⇔ Y kill X

Polarity Rules

- Annotate **polarity** for verbs and Nouns
  - **Positive:** *John called*$^{(+)}$ *Mary.*
  - **Negative:** *John forgot to call*$^{(-)}$ *Mary.*
  - **Unknown:** *John wanted to call*$^{(?)}$ *Mary.*
- Expressed by
  - Verbal negation
  - modal verbs and adverbials
  - conditional sentences
  - Verbs inducing **negative** polarity - *forget, failed*
  - Verbs inducing **unknown** polarity - *want, attempt*
- Polarity mismatch is indicative for no-entailment
  - Used as feature

Adding Approximate Matching

RTE4 System Architecture



Features for Approximate Matching

- Coverage of H by F
  - Lexical coverage (words, verbs, numbers, named entities)
  - Local syntactic coverage (edges)
  - Global structural matching
    - Aim to match maximal subtrees of H in F
    - Efficient computation using dynamic progamming
- Polarity mismatch (*forgot to buy* vs. *bought*)
- Argument matching for corresponding predicates

Results

- Using SVM classifier (outperforms decision tree)

Train	Test	Knowledge-Based Inference	Accuracy
RTE3 Dev+Test	RTE4	✓	58.4
RTE3 Dev	RTE4	✓	**60.3**
RTE3 Dev	RTE3 Test	✓	66.9
RTE3 Dev	RTE3 Test		64.6
RTE3 Dev	½RTE4	✓	58.8
RTE3 Dev	½RTE4		55.8

- Competitive accuracy on RTE-3 (~4-6 out of 26)
- Knowledge-based inference improves accuracy (+3%)
- New efficient architecture reduced running time from hours to minutes, allowing application of many more rules

Conclusion

- **Takeouts**
  - A formalized proof system over parse trees can be implemented ***efficiently***
    - While preserving semantics – equivalent to explicit generation of consequents
    - Substantial run time reduction
    - Substantial increase in number of rules applied
  - Knowledge helps!
    - But not much for now...
- Future research
  - Analysis (errors, missing knowledge, impact of resources, search strategies)
  - Additional knowledge sources
  - Improve approximate matching

Thank You!