Transition-based Dependency Tree to AMR Parser

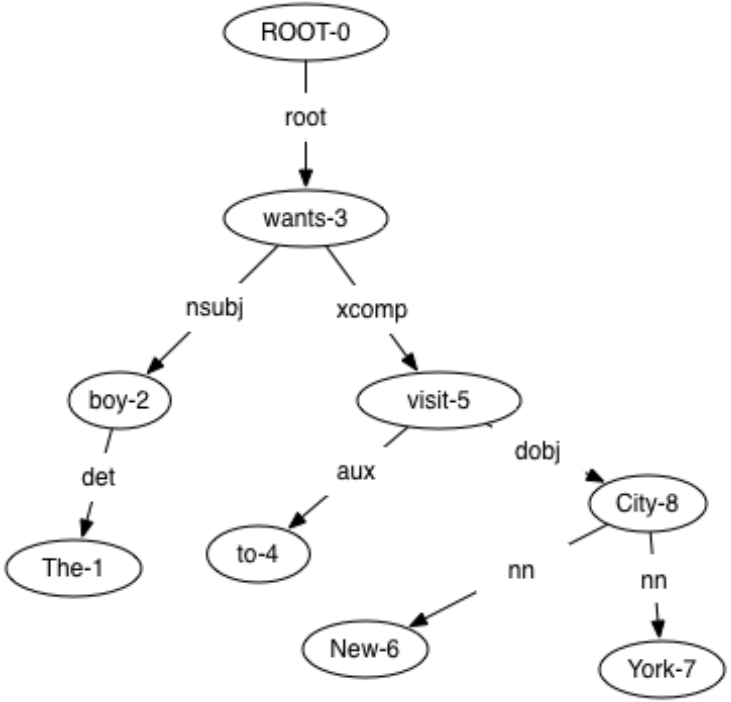
Chuan Wang, Yuchen Zhang, Nianwen Xue
Brandeis University
August.01.2014

Outline

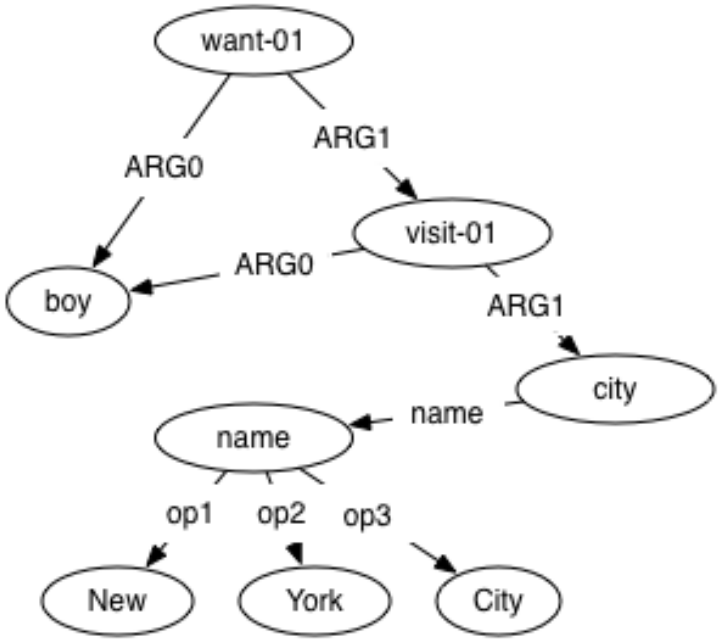
- Motivation
- System pipeline
- Preprocessing
  - Alignment
  - Gold graph generation
- Dependency tree to Graph transition system
- Preliminary Results

Motivation

The boy wants to visit New York City .

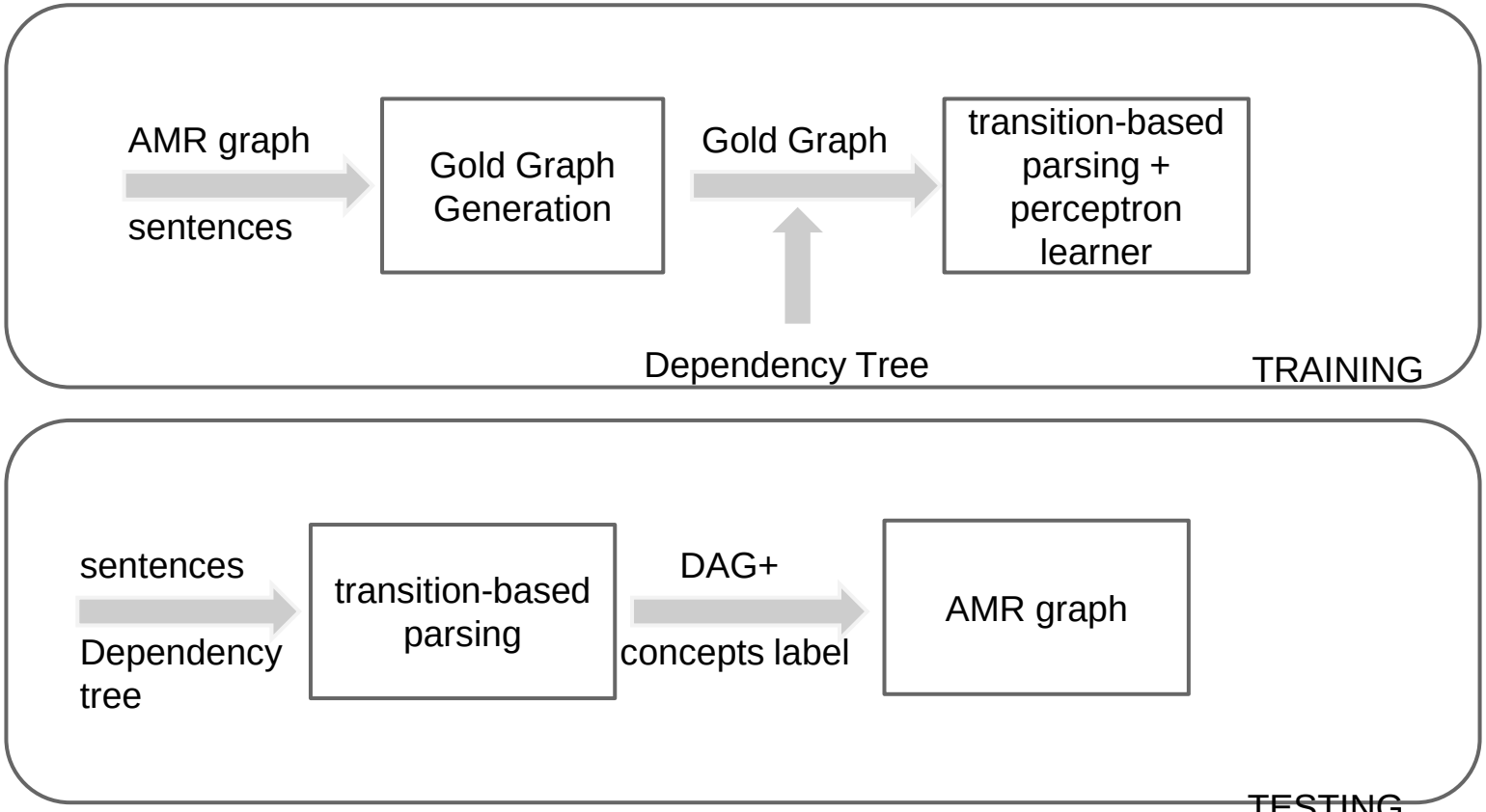


Dependency Tree

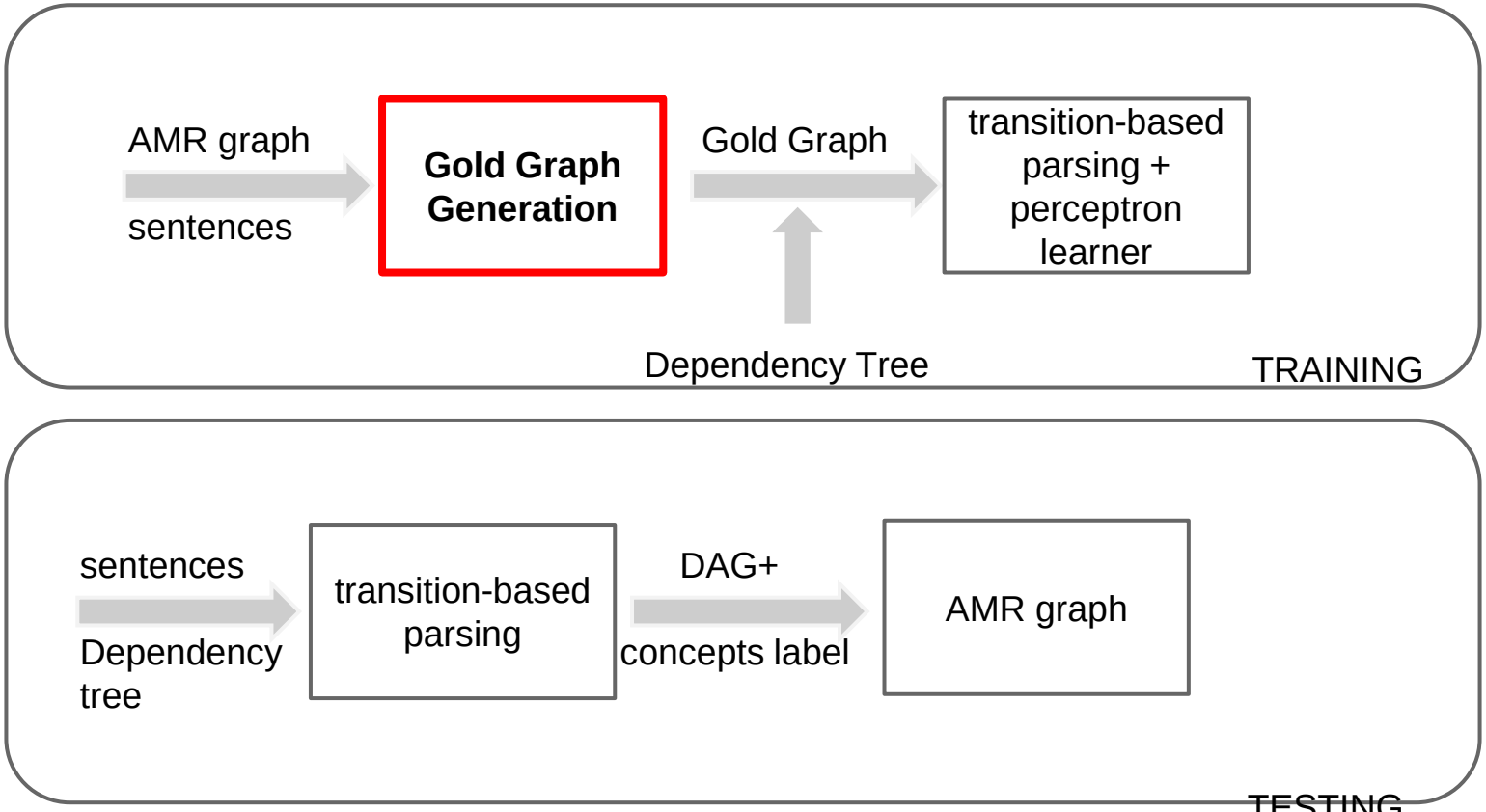


AMR Graph

Pipeline


AMR graph
sentences
Gold Graph Generation
Gold Graph
transition-based parsing + perceptron learner
Dependency Tree
TRAINING
sentences
Dependency tree
transition-based parsing
DAG+
concepts label
AMR graph
TESTING

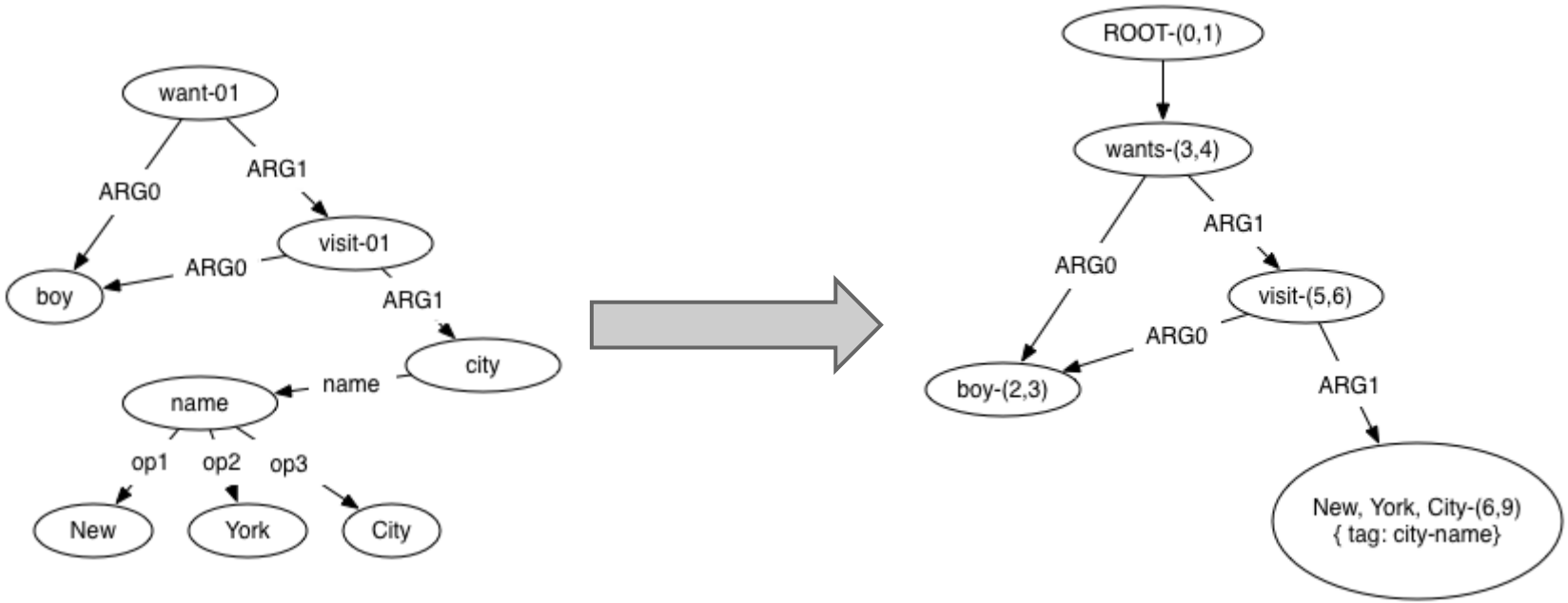
Pipeline


AMR graph
sentences
Gold Graph Generation
Gold Graph
transition-based parsing + perceptron learner
Dependency Tree
TRAINING
sentences
Dependency tree
transition-based parsing
DAG+
concepts label
AMR graph
TESTING

Gold graph generation

Alignment + AMR graph = Gold graph “Our learning target”

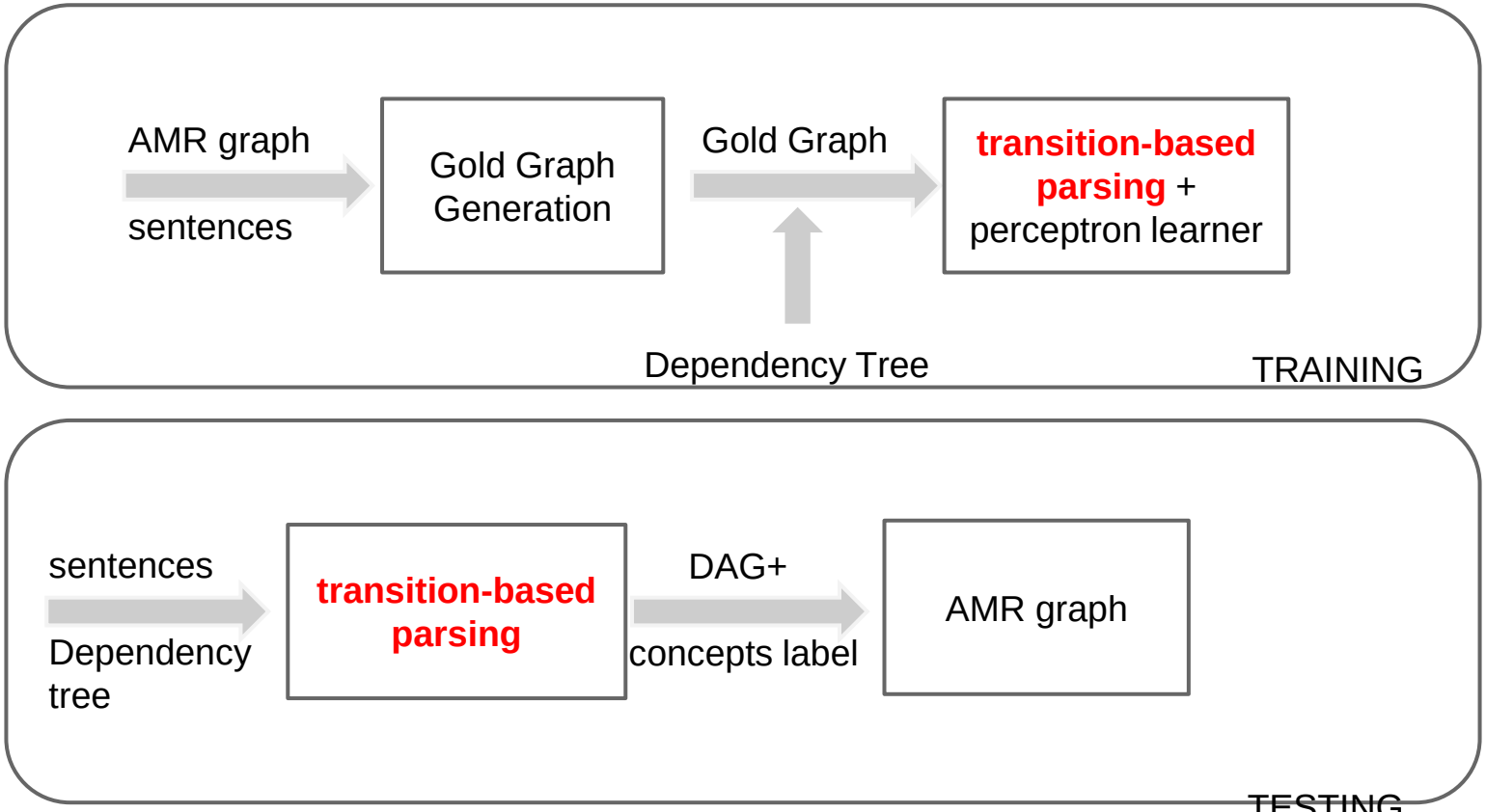
1. each node in gold graph represents one of the spans in the sentence.
2. each node also contains its entity tag in the AMR graph.



AMR graph

Gold graph

Pipeline

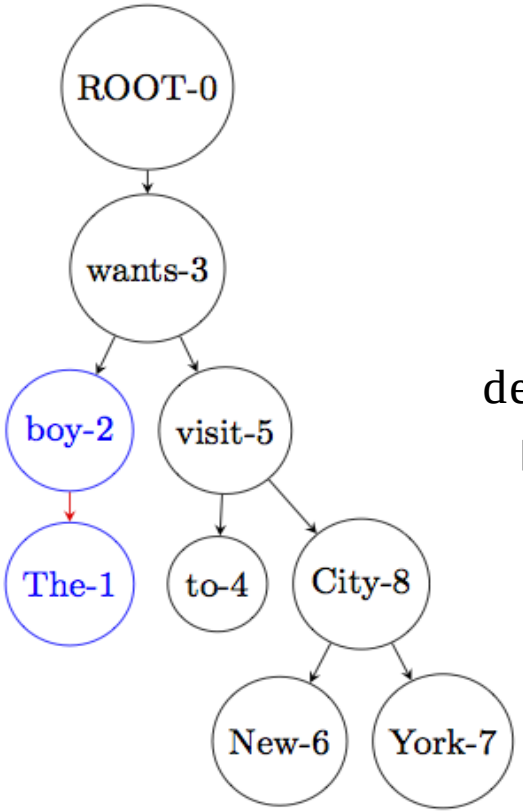

AMR graph
sentences
Gold Graph Generation
Gold Graph
transition-based parsing + perceptron learner
Dependency Tree
TRAINING
sentences
Dependency tree
transition-based parsing
DAG+
concepts label
AMR graph
TESTING

Dependency Tree to Graph Transition Parsing

Parsing actions:

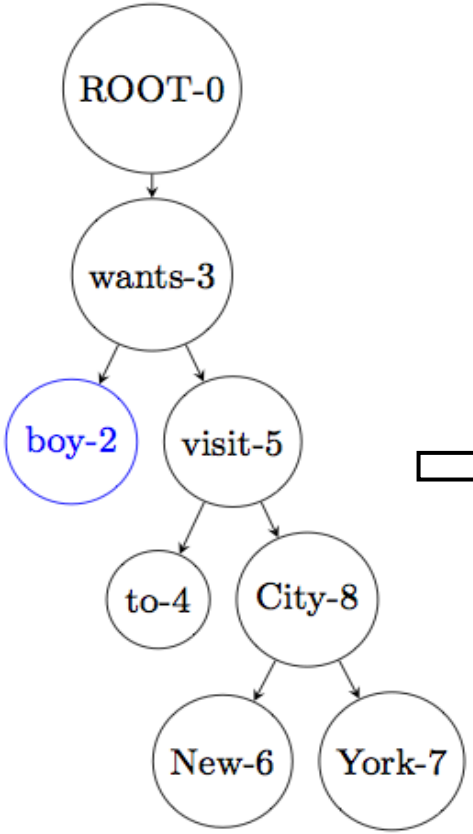
- if β is not empty:
  - delete edge
  - swap
  - replace head
  - merge
  - next
- if β is empty:
  - add child k
  - finish
  - delete node

State 1

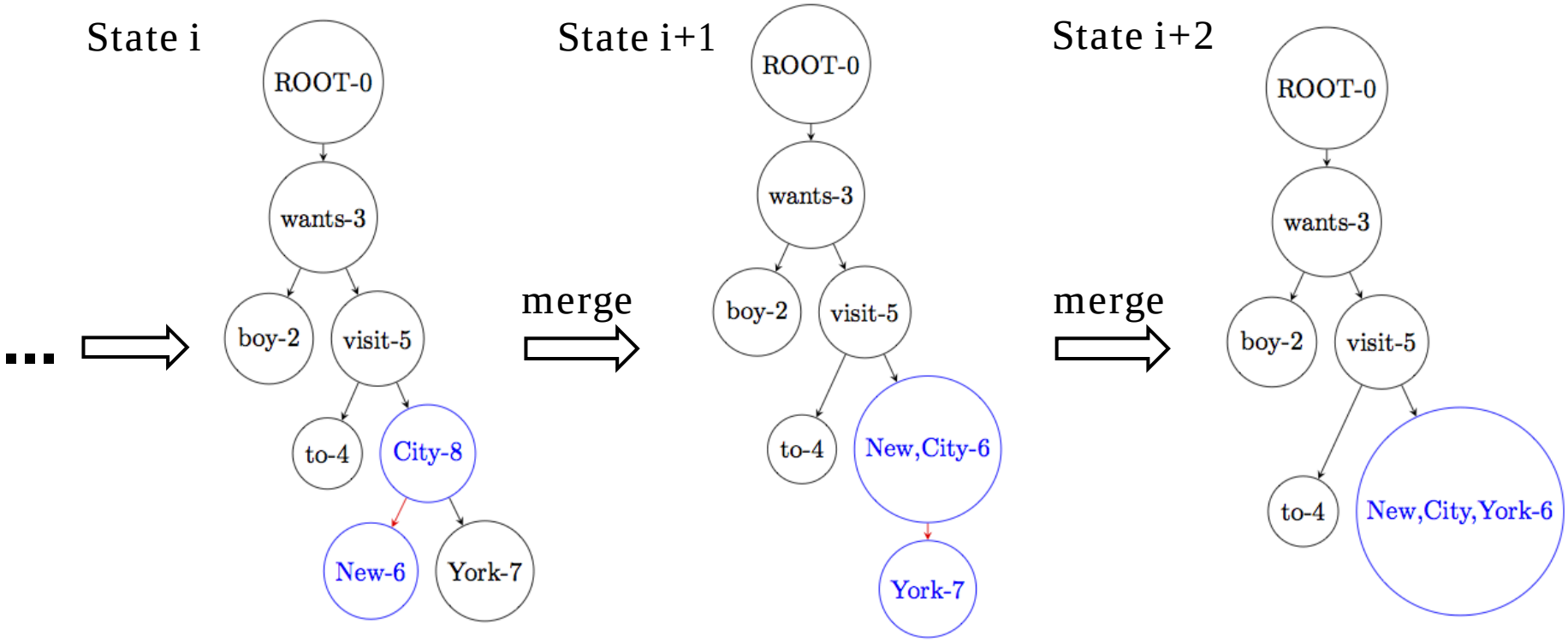


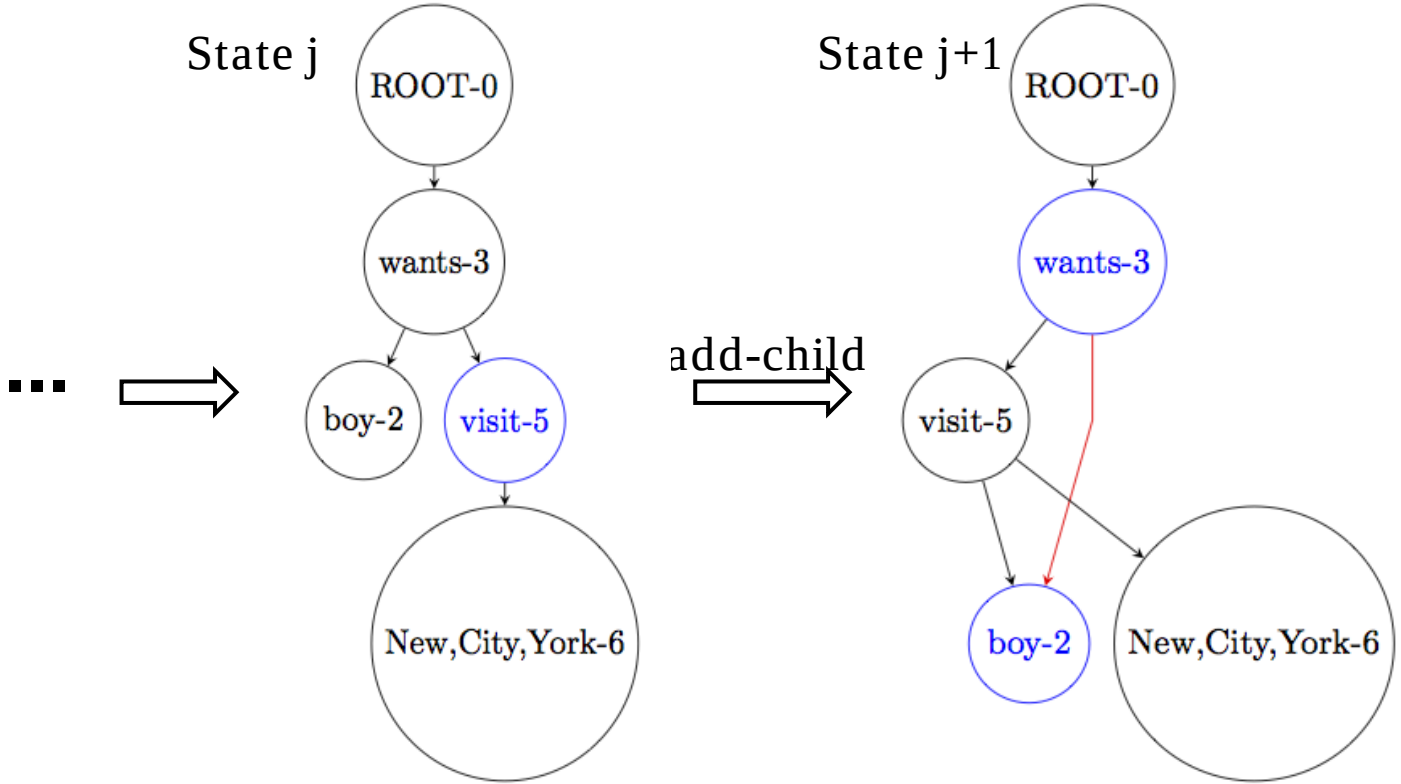
delete-node

State 2

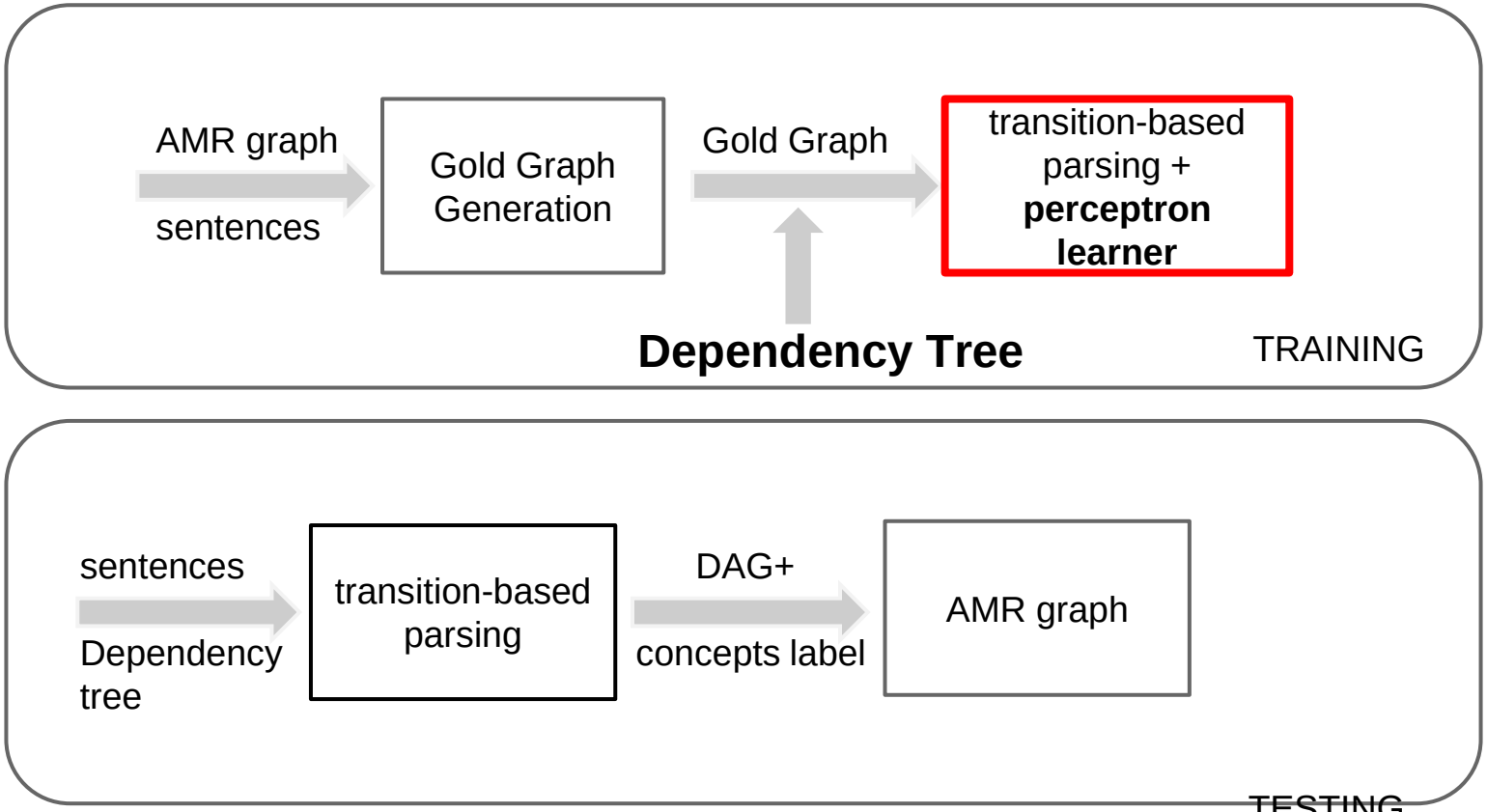


...


State i
ROOT-0
wants-3
boy-2
visit-5
to-4
City-8
New-6
York-7
merge
State i+1
ROOT-0
wants-3
boy-2
visit-5
to-4
New,City-6
York-7
merge
State i+2
ROOT-0
wants-3
boy-2
visit-5
to-4
New,City,York-6

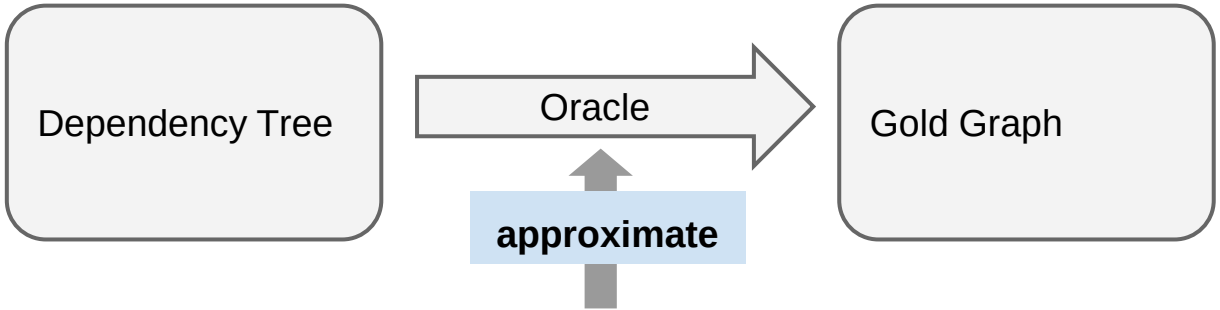

State j
ROOT-0
wants-3
boy-2
visit-5
New,City,York-6
add-child
State j+1
ROOT-0
wants-3
visit-5
boy-2
New,City,York-6

Pipeline

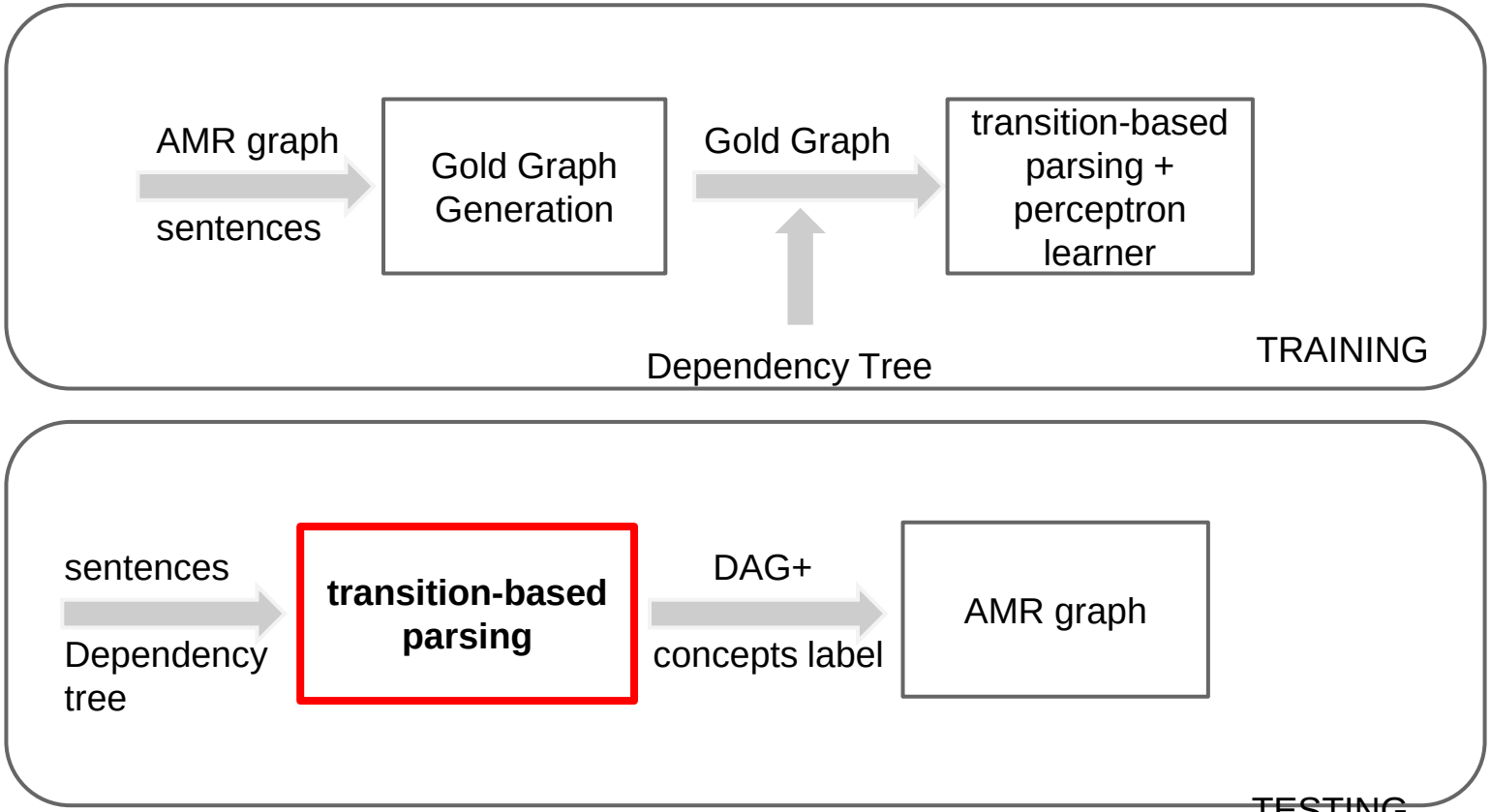

AMR graph
sentences
Gold Graph Generation
Gold Graph
transition-based parsing + **perceptron learner**
Dependency Tree
TRAINING
sentences
Dependency tree
transition-based parsing
DAG+
concepts label
AMR graph
TESTING

Deterministic Parsing

If we assume alignment is correct, which means we always get the gold graph (alignment + AMR graph), we can design an oracle to generate unique sequence of parsing actions for each sentence in corpus and use perceptron learner to approximate oracle.



Pipeline


AMR graph
sentences
Gold Graph Generation
Gold Graph
transition-based parsing + perceptron learner
Dependency Tree
TRAINING
sentences
Dependency tree
transition-based parsing
DAG+
concepts label
AMR graph
TESTING

Experiment results

Currently we don’t have edge label and concept label, we can measure the performance by measuring only the unlabeled score.

$$\text{Precision} = \frac{N_{correct}}{N_{parsed}}$$

$$\text{Recall} = \frac{N_{correct}}{N_{gold}}$$

$N_{correct}$ is the number of correct attachments predicted
N_{parsed} is the number of predicted attachments
N_{gold} is the number of gold attachments

Experiment results

- “The little price”: 1562 sentences with English AMRs in total; 312 held out as development set; the rest 1250 as training set.
- Similarity of attachment between dependency tree and gold graph? -- baseline measured on the development set.
- CAMR: using only word form, POS tag and named entity tag as features -- our system trained 19 iterations on the training set and measured on the development set.

	unlabeled precision	unlabeled recall	unlabeled f1
baseline	0.30	0.48	0.37
CAMR	0.45	0.53	0.48