First part recap

Imitation Learning

- Meta-learning framework, over an existing classifier(s).
- In practice, generates more (and better) training data, to improve the existing classifier(s).

For applied Imitation Learning we need to define:

- Transition system
- Task loss
- Expert policy

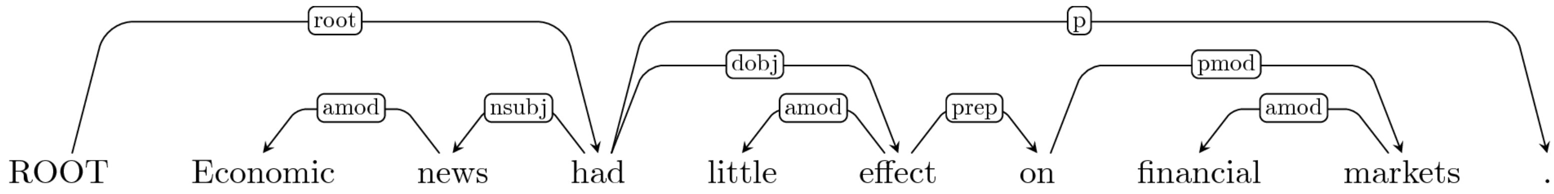
Part 2: NLP Applications and practical advice

- Applications:
  - Dependency parsing
  - Semantic parsing
  - Natural language generation
- Practical advice
  - Expert policy definition
  - Accelerating cost estimation
  - Trouble-shooting

Applying Imitation Learning on Dependenc y Parsing

Dependency parsing

([Goldberg and Nivre 2012 (http://www.aclweb.org/anthology/C12-1059)](http://www.aclweb.org/anthology/C12-1059), [Goldberg and Nivre 2013 (https://www.aclweb.org/anthology/Q/Q13/Q13-1033.pdf)](https://www.aclweb.org/anthology/Q/Q13/Q13-1033.pdf))



To represent the syntax of a sentence as directed labeled edges between words.

- Where labels represent dependencies between words.

Error propagation

Due to greedy decoding, where the parser builds the parse while maintaining only the best hypothesis at each step.

- The first error encountered will confuse the classifier , since it moves the sequence to space not explored by the gold sequence of actions.
- More errors will likely follow, as the transition increasingly moves into more foreign states.

How can Imitation Learning help with that?

Imitation Learning addresses error propagation, by considering the interaction among the transition being considered and transitions to be predicted later in the sentence.

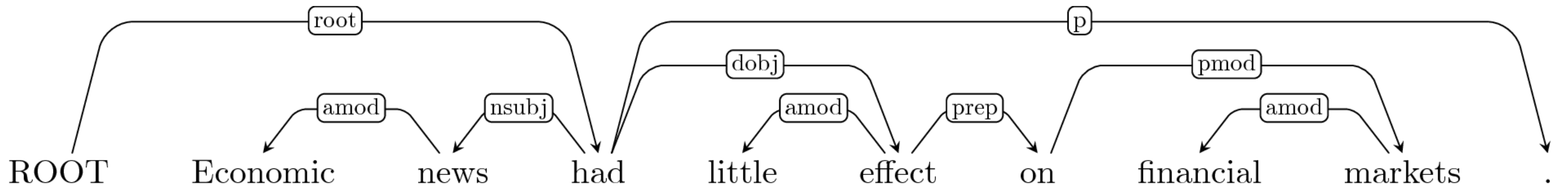
- Explores the search space, but avoids enumerating all possible outputs.
- Also learns how to recover from errors.

Transition system?

We can assume any transition-based system (Arc-Eager, Arc-Standard, Easy-First, etc.).

- Each action, transforms the current state/graph until a terminal state/graph is reached.
- In essence, which arc and label should we add ne xt?

Transition-based dependency parsing in action!



Task loss?

Hamming loss: the number of incorrectly predicted labeled or unlabeled dependency arcs.

- Directly related to the attachment score metrics used to e valuate dependency parsers.

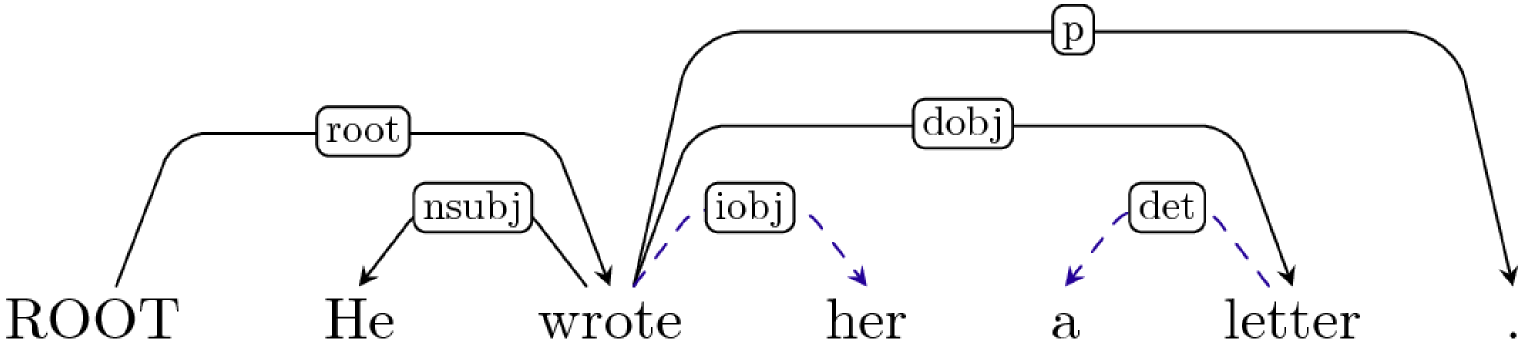
Expert policy?

A single static canonical sequence of actions from the initial to the terminal state.

- Derived from the reference graph.

Single static policies worked well for the Part-of-Speech tagging task.

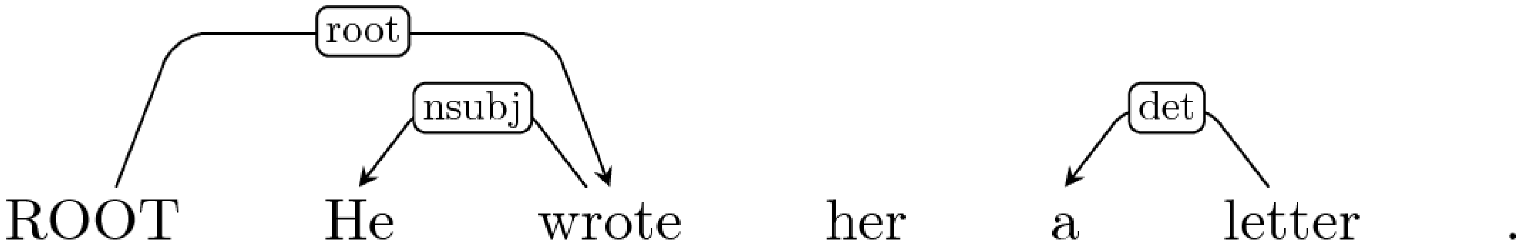
But what if there are multiple correct transitions?



A static policy could arbitarilly chose a tr ansition (e.g. prioritize shifts over other actions).

- But this indirectly labels the alternativ e transition as false!

And what if a mistake happens?

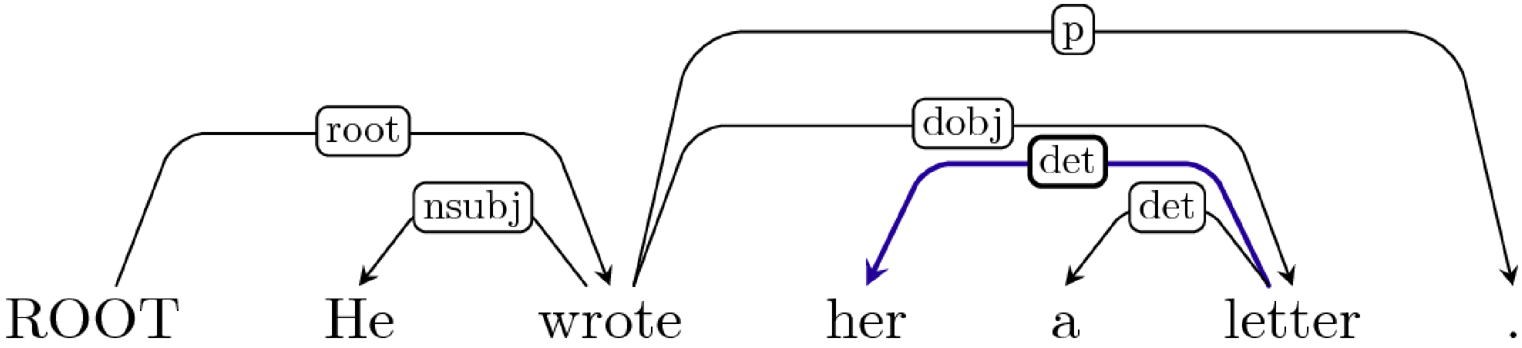


A static policy is not well defined in states that are not part of the gold tr ansition.

Dynamic policy

Non-deterministic and complete policy

- Allows ambiguous transitions.
- Defined for all states.
- Recovers from errors.

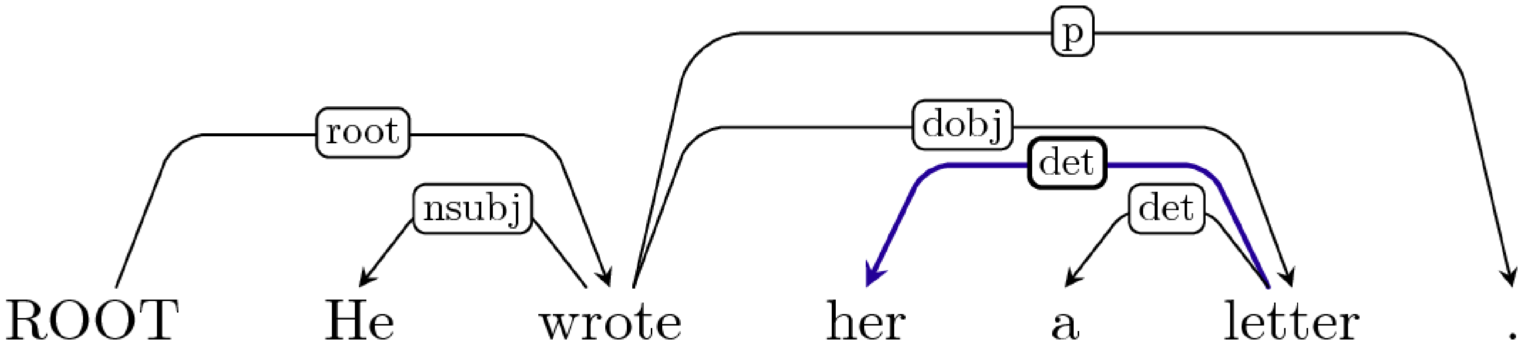


What is the expert policy then?

Given a particular state, where an error may or may not have already occured:

- We need to determine the best reachable terminal state
  - Quite possibly not an optimal terminal state, if we have made an error before.
  - "Best" according to some loss function in relation to the gold terminal state.
- Return the set of transition actions that lead to that state.

Expert policy in action!



Cost of 1, if we use the labeled attachment score as a loss function.

Is that DAgger?

Goldberg and Nivre (2013) (https://www.aclweb.org/anthology/Q/Q13/Q13-1033.pdf) proposed a system that employed dynamic expert policies for dependency parsing.

- As well as an algorithm to learn parameters by exploration.
- Very similar to DAgger.
    - Roll-in is a mix of the learned and expert policies, at the time-step level.
    - There may be multiple correct actions at each time-step.

Results

system / language	hungarian	chinese	greek	czech	basque	catalan	english	turkish	arabic	italian
					UAS					
eager:static	76.42	85.01	79.53	78.70	75.14	91.30	86.10	77.38	81.59	84.40
eager:dynamic	77.48	85.89	80.98	80.25	75.97	92.02	88.69	77.39	83.62	84.30
hybrid:static	76.39	84.96	79.40	79.71	73.18	91.30	86.43	75.91	83.43	83.43
hybrid:dynamic	77.54	85.10	80.49	80.07	73.70	91.06	87.62	76.90	84.04	83.83
easyfirst:static	81.27	87.01	81.28	82.00	75.01	92.50	88.57	78.92	82.73	85.31
easyfirst:dynamic	81.52	87.48	82.25	82.39	75.87	92.85	89.41	79.29	83.70	85.11
					LAS					
eager:static	66.72	81.24	72.44	71.08	65.34	86.02	84.93	66.59	72.10	80.17
eager:dynamic	68.41	82.23	73.81	72.99	66.63	86.93	87.69	67.05	73.92	80.43
hybrid:static	66.54	80.17	70.99	71.88	62.84	85.57	84.96	64.80	73.16	78.78
hybrid:dynamic	68.05	80.59	72.07	72.15	63.52	85.47	86.28	66.12	74.10	79.25