

## What is ULF?

Underspecified (Episodic) Logical Forms

*"She wants to eat the cake"*

```
(she.pro ((pres want.v)
  (to (eat.v (the.d cake.n))))))
```

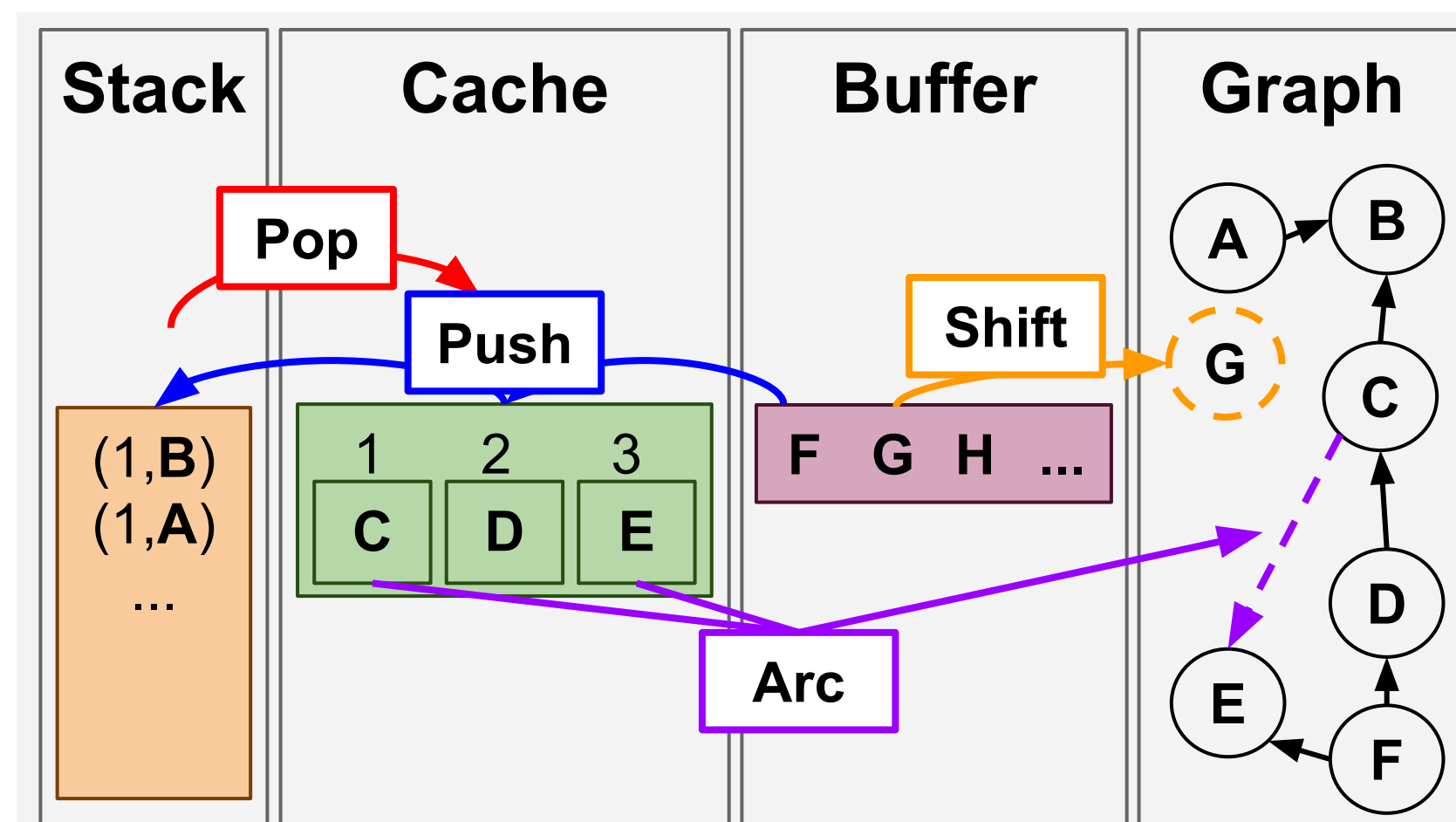
- Closely matches surface form of language
- Grounded in Episodic Logic Types
  - she.pro - entity  $\mathcal{D}$
  - cake.n - noun predicate  $(\mathcal{D} \rightarrow 2)_N$
  - to - action reifier  $((\mathcal{D} \rightarrow 2)_V \rightarrow \mathcal{D})$

- Resolves predicate-argument structure  
**Brackets!**

- Unresolved
  1. Anaphora - she.pro, (the.d cake.n)
  2. Operator scope - the.d, pres
  3. Word sense - want.v, eat.v, cake.n

## What is a Cache Transition Parser?

A transition system for parsing graphs using a fixed-sized cache. It's been used to parse abstract meaning representations.



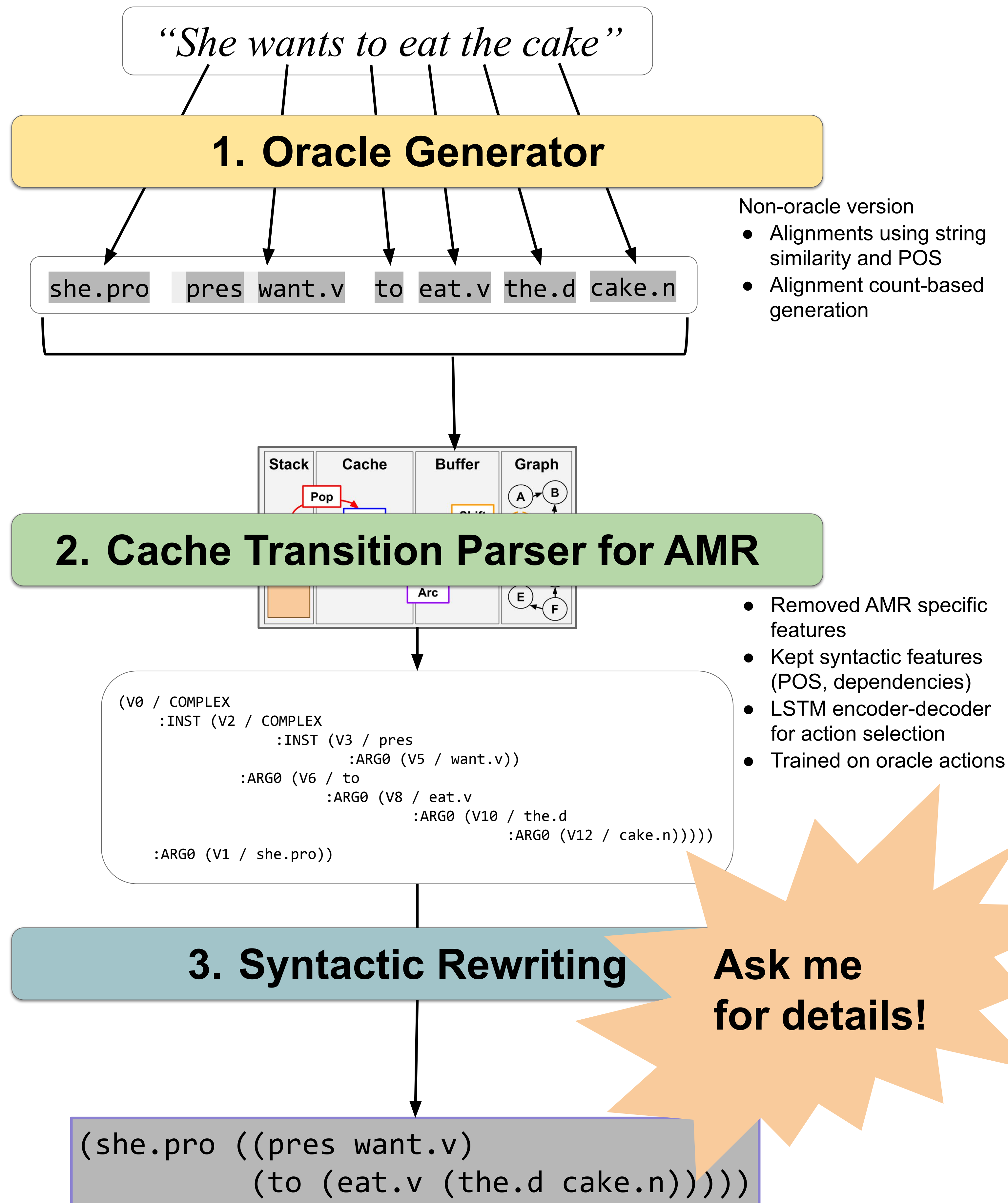
**Pop:** pops the top element from stack to its indexed position in cache

**Shift:** moves the front of the buffer by one and adds a vertex to the graph for the front element

**Push:** moves the front of the buffer to the cache and pushes the old cache value to the stack

**Arc:** forms an arc between a given index of the cache and the rightmost element of the cache

## System Outline



## So how do we do?

**Best model: 0.738 EL-smatch score**  
cache size 2

## Experimental Details

- Gold generation (step 1 is known)
- 725/90/90 sentence train/dev/test split
- Tested on cache sizes 2, 4, and 6

Cache Size	Dev Action Acc.	Dev EL-smatch	Test EL-smatch
2	0.92	0.73	0.738
4	0.90	0.74	0.736
6	0.92	0.72	0.716

Cache Size	Oracle Success Rate
2	0.01
4	0.73
6	0.82

**Weird!** Cache size 2 performs best even though the oracle fails!

## What Next? (Lots!)

1. Evaluating performance with the generator (step 1)
2. Fixing ULF Atom Ordering  
The curious oracle failure is likely due to poor atom ordering
3. Integrating the ULF Type System  
The ULF type system restricts the available interpretations and combinations
  - New action for introducing type shifters
  - Type-based composition constraints
4. Categorization  
Foreign languages, dates, currency, etc. are likely better handled procedurally.

## Acknowledgements

The work was supported by a Sproull Graduate Fellowship from the University of Rochester, DARPA CwC subcontract W911NF-15-1-0542, NSF EAGER grant IIS-1543758, and informed by the Fall 2018 Semantic Parsing Seminar by Dan Gildea.