

# Better Glue for Pipelines

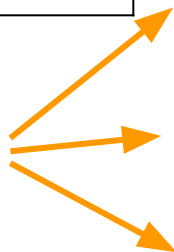
CSE504 Project Proposal

Luheng He

# Motivation: Pipelined Software for NLP/ML Tasks

(Mostly) task-independent, off-the-shelf tools
Task-dependent code

**Glue  
Code**



	Typical subtasks for NLP	Typical subtasks for ML
1	Input Reader	Input Reader
2	Segmentation/tokenization	Pre-processing/Data filtering
3	Pos-tagging/Parsing/Named-entity Recognition	
4	Feature Extraction for the target task	
5	Parameter Fitting (Learning)	
6	Evaluation/Cross validation	
7	Model Ensemble	
8	Output/Analysis/Visualization	

# Can we automatically generate glue code?

## What's wrong with glue code:

- Takes time to write, slows down research progress
- Boring and repetitive
- Error-prone
- ...

## Automatically generate glue code:

- Focus on NLP/ML pipelines for now
- Focus on the case where we need to **transform** the output data from an upstream software A to the input of a downstream task B

## Code (Data structure, API):

```
class ParsedSentence {  
    int[] tokenIds;  
    int[] depParents;  
    ....  
}
```

## Specification/Comments:

```
/* output format =  
word_id \t word \t parent_id \t label  
*/  
/* input format =  
parent_id,child_id,label_id */
```

## Sample input/output:

1	the	2	DT ...
2	cat	3	NN ...
3	sits	0	VB ...

## Formal representation and invariants for the data:

tokenIds: List[Int], parseTreeArcs: List[(Int, Int)] ...  
 $\forall t \in \text{tokenIds}: 0 \leq t \leq \text{numWords}, \forall (x,y) \in \text{parseTreeArcs}: 0 \leq x, y \leq |\text{tokenIds}| \dots$

## Glue code

that transform output data from software A to the input data of software B

## Tests

based on the invariants

## Specifications

that explains the input/output format