Neural Distant Supervision for Relation Extraction

Deepanshu Jindal

Elements and Images borrowed from
Happy Mittal, Luke Zettlemoyer

Outline

- What is Relation Extraction (RE)?
- (Very) Brief overview of extraction methods
- Distant Supervision (DS) for RE
- Distant Supervision for RE using Neural Models
- Distant Supervision for RE using Neural Models

Outline

- ~~What is Relation Extraction (RE)?~~
- ~~(Very) Brief overview of extraction methods~~
- Distant Supervision (DS) for RE
- Distant Supervision for RE using Neural Models
- Distant Supervision for RE using Neural Models

Relation Extraction

- Predicting relation between two named entities
  - Subtask of Information Extraction

Edwin Hubble was born in *Marshfield*, Missouri.

Relation Extraction →

BornIn(Edwin Hubble, Marshfield)

Relation Extraction Methods

1. Hand-built patterns
2. Boot Strapping methods
3. Supervised Methods
4. Unsupervised Methods
5. Distant Supervision

Relation Extraction Methods

1. Hand-built patterns
    - Lexico-Syntactic Patterns
    - Hard to maintain, Non scalable
    - Poor Recall

2. Boot Strapping methods
3. Supervised Methods
4. Unsupervised Methods
5. Distant Supervision

Relation Extraction Methods

1. Hand-built patterns
2. Boot Strapping methods
    - Give initial seed patterns and facts
    - Generate more facts and patterns
    - Suffers from semantic drift
3. Supervised Methods
4. Unsupervised Methods
5. Distant Supervision

Relation Extraction Methods

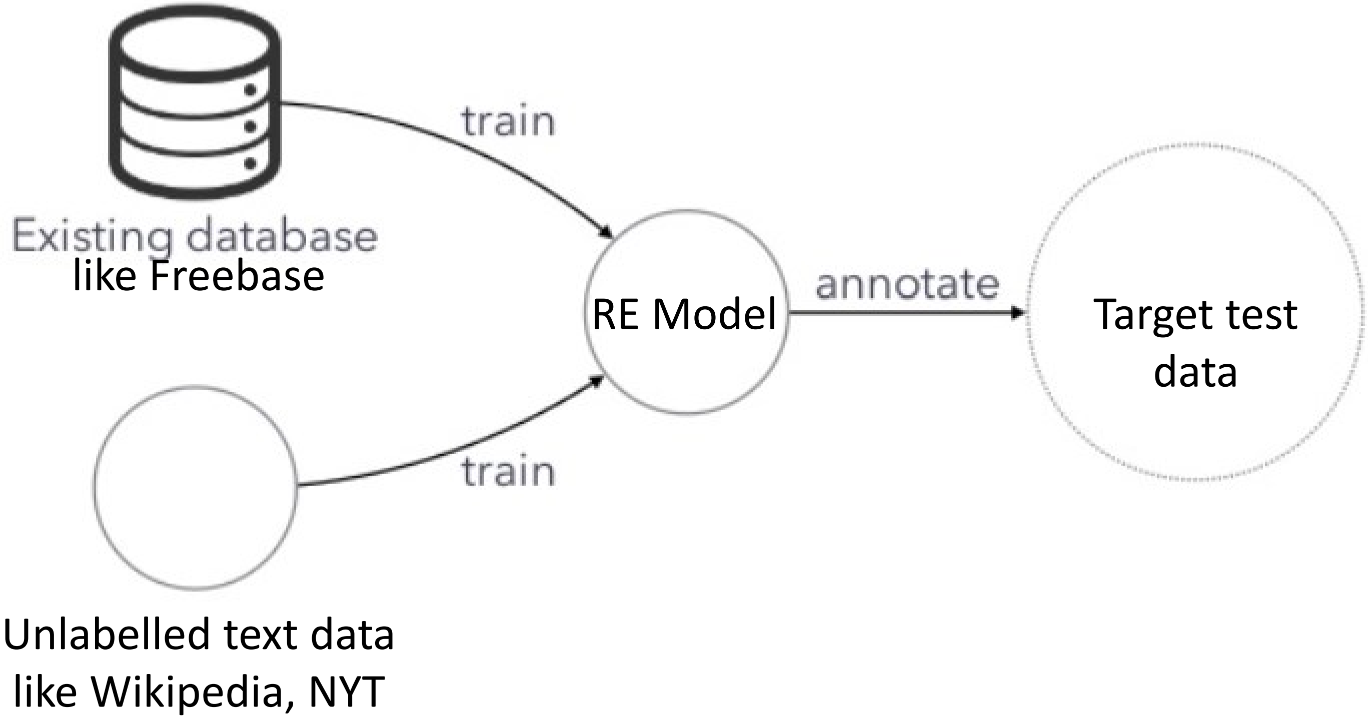
1. Hand-built patterns
2. Boot Strapping methods
3. Supervised Methods
    - Labeled corpora of sentences over which classifier is trained
    - Suffers from small dataset, domain bias.

1. Unsupervised Methods
2. Distant Supervision

Relation Extraction Methods

1. Hand-built patterns
2. Boot Strapping methods
3. Supervised Methods
4. Unsupervised Methods
    - Cluster patterns to identify relations
    - Large corpora available
    - Can’t give name to relations identified.
5. Distant Supervision

Distant Supervision for Relation Extraction



Training

- Find a sentence in unlabelled corpus with two entities

  Steve Jobs is the CEO of *Apple*.

- Find the entities in the KB and determine their relation

Relation	ARG1	ARG2
EmployedBy	Steve Jobs	Apple

- Train the model to extract relation found in KB from the given sentence

Problems

Heuristic based training data

- Very Noisy
- High false positive rate

Distant Supervision assumption is too strong.

Mention of two entities doesn’t imply same relation.

FounderOf(Steve Jobs, Apple)

Steve Jobs was co-founder of Apple and formerly Pixar.

Steve Jobs passed away a day before Apple unveiled Iphone 4S.

Problems

Feature Design and Extraction

- Hand coded features
  - Non Scalable
  - Poor Recall
- Ad Hoc features based on NLP tools (POS, NER Taggers, Parsers)
  - Accumulation of errors during feature extraction

Distant Supervision for Relation Extraction using Neural Networks

Two variations of Neural Network application:

- Neural model for relation extraction
- Neural RL model for distant supervision

Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks

Daojian Zeng, Kang Liu, Yubo Chen and Jun Zhao
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
{djzeng,kliu,yubo.chen,jzhao}@nlpr.ia.ac.cn

Addressing the problems

- Handling Noisy Training Data - Multi Instance Learning

- Neural models for feature extraction and representation

Multi Instance Learning

- Bag of instances
- Labels of the bags are known - labels of the instances unknown
- Objective function at the bag level

Multi Instance Learning

- Bag of instances
- Labels of the bags are known - labels of the instances unknown
- Objective function at the bag level

$T\ bags\ \{M_1, M_2, ...M_T\}\ where\ i^{th}\ bag\ contains\ q_i\ instances\ M_i = \{m_i^1, m_i^2, ..m_i^{q_i}\}$

Multi Instance Learning

- Bag of instances
- Labels of the bags are known - labels of the instances unknown
- Objective function at the bag level

$$T \; bags \; \{M_1, M_2, ...M_T\} \; where \; i^{th} \; bag \; contains \; q_i \; instances \; M_i = \{m_i^1, m_i^2, ..m_i^{q_i}\}$$

$$J(\theta) = \sum_{i=1}^{T} \log p(y_i | m_i^j; \theta)$$

Multi Instance Learning

- Bag of instances
- Labels of the bags are known - labels of the instances unknown
- Objective function at the bag level

T $bags$ $\{M_1, M_2, ...M_T\}$ $where$ i^{th} bag $contains$ q_i $instances$ $M_i = \{m_i^1, m_i^2, ..m_i^{q_i}\}$

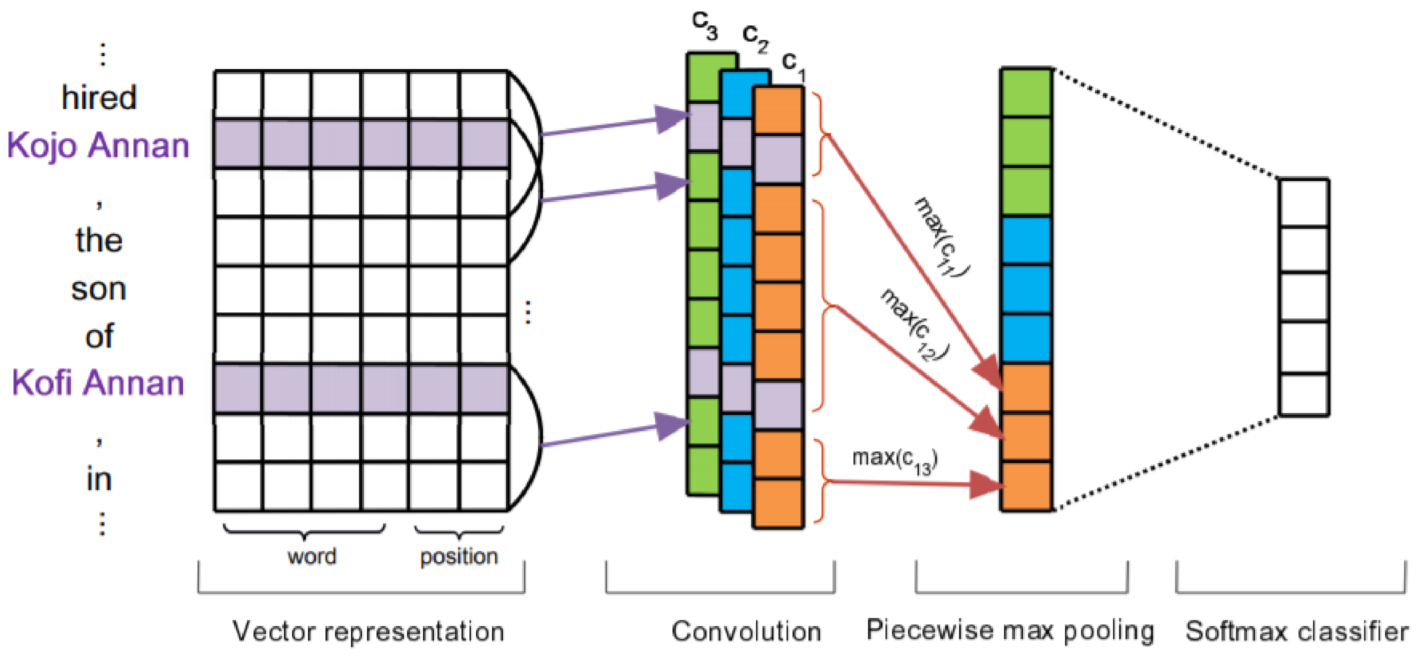
$$J(\theta) = \sum_{i=1}^{T} \log p(y_i | m_i^j; \theta)$$ where $$j^* = \arg\max_j p(y_i | m_i^j; \theta) \;\; 1 \leq j \leq q_i$$

Piecewise Convolution Network

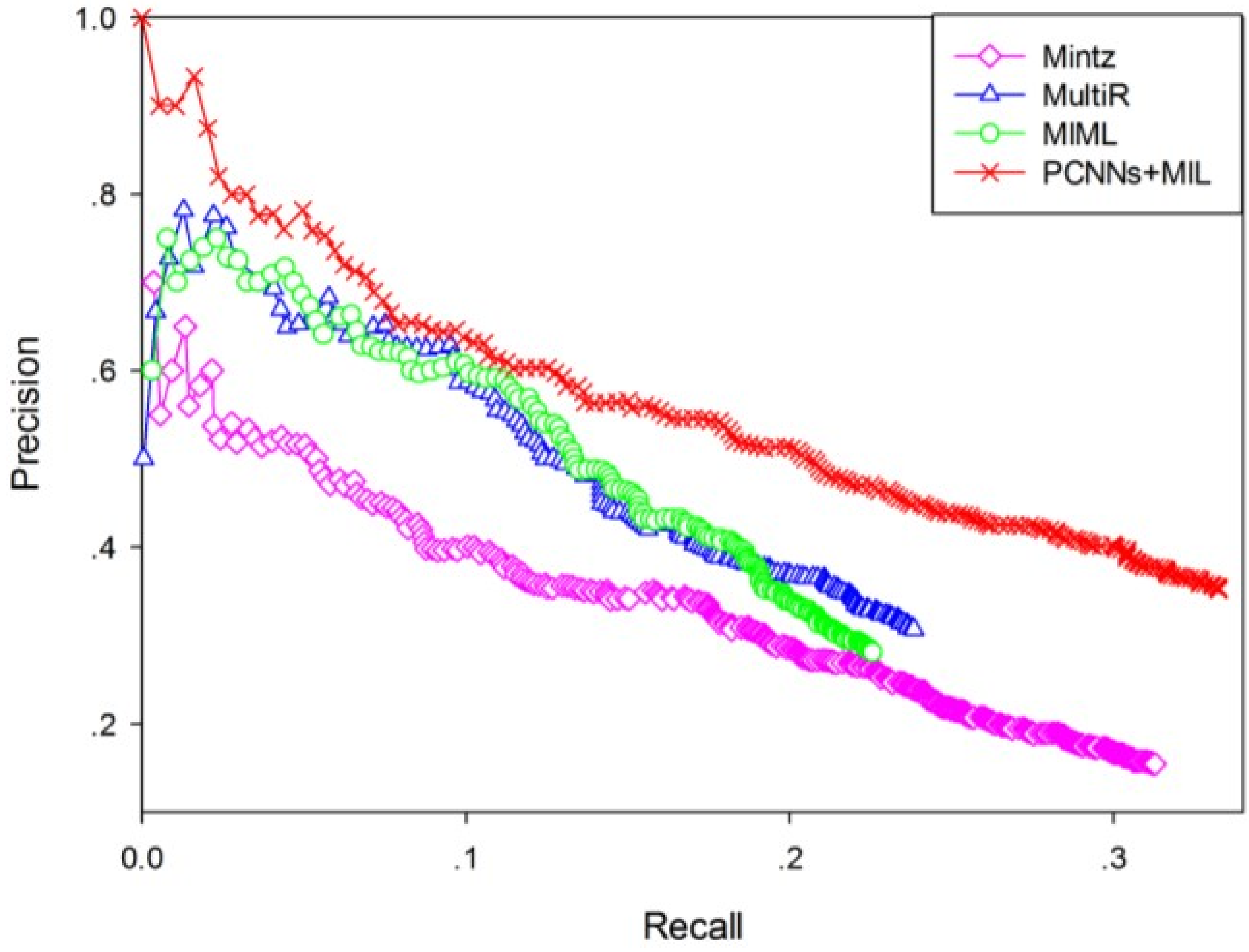
- Doing MaxPool over the entire sentence is too restrictive
- Do separate pooling for left context, inner context and right context

Piecewise Convolution Network

- Doing MaxPool over the entire sentence is too restrictive
- Do separate pooling for left context, inner context and right context



Results



Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning

Pengda Qin[♯], Weiran Xu[♯], William Yang Wang[♭]
[♯]Beijing University of Posts and Telecommunications, China
[♭]University of California, Santa Barbara, USA
{qinpengda, xuweiran}@bupt.edu.cn
{william}@cs.ucsb.edu

Addressing the problem

False Positives – Bottleneck for performance

- Previous approaches
  - Don’t explicitly remove noisy instances
    Hope model would be able to suppress noise [Hoffman ’11, Surdeanu ‘12]
  - Choose one best sentence and ignore rest [Zeng ‘14, ‘15]
  - Attention mechanism to upweight relevant instances [Lin ‘17]

Proposal

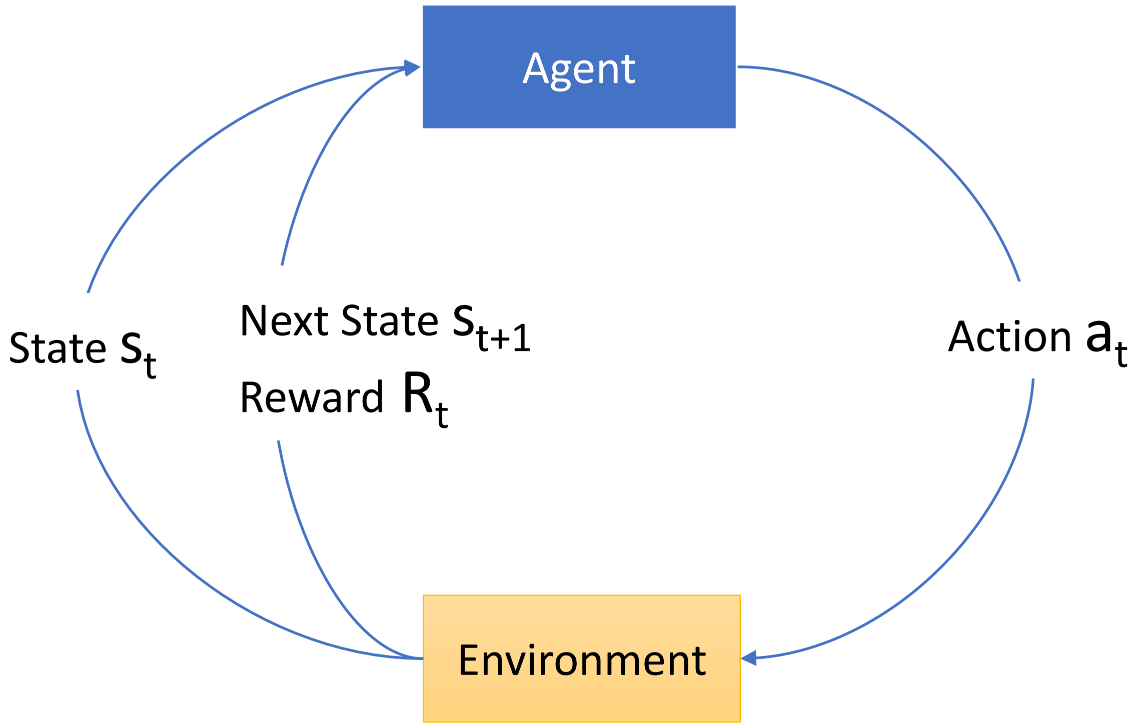
- Agent to determine where to retain or remove instance
- Put removed instances as negative examples

Proposal

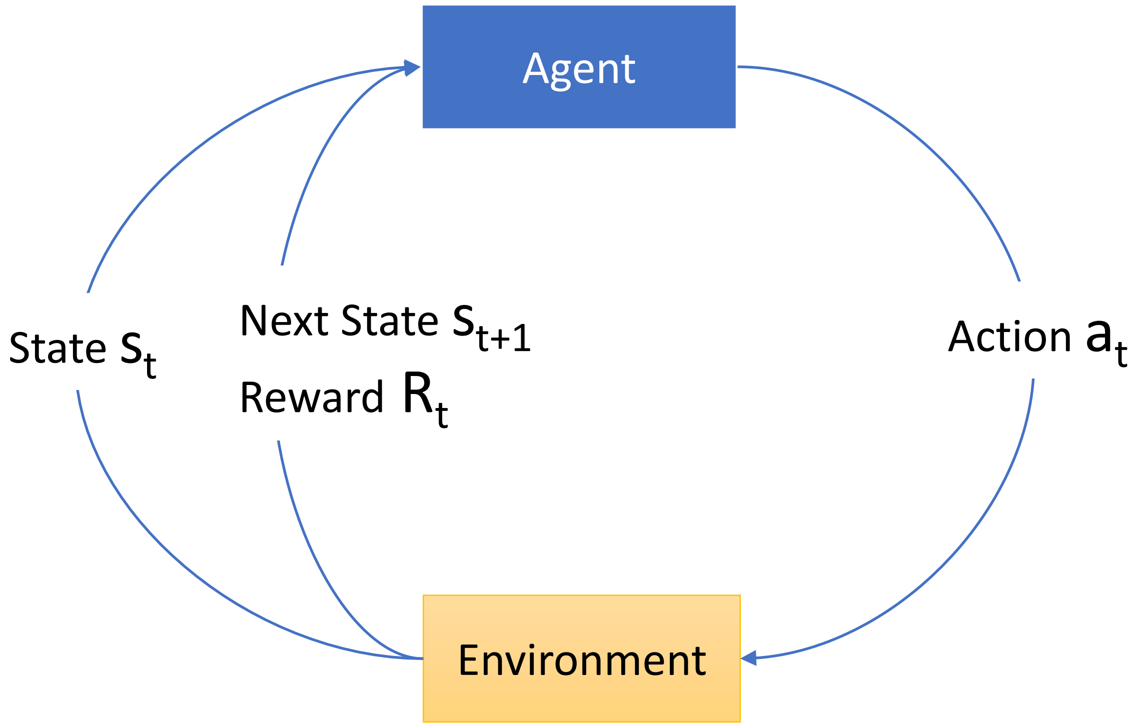
- Agent to determine where to retain or remove instance
- Put removed instances as negative examples

Reinforcement Learning agent to optimize Relation Classifier

Reinforcement Learning



Reinforcement Learning



State space	S
Action space	A

Environment

- Reward Model — R
- Transition Model — T

Agent

- Policy Model — π

Problem Formulation

Agent for each relation type

- State
  - Current instance + Instances removed until now
  - Concat(Current Sentence Vector, Avg. Vector of Sentence removed)
- Action
  - Remove/Retain current instance

Problem Formulation

- Reward
  - Change in classifier performance(F1) between consecutive epochs

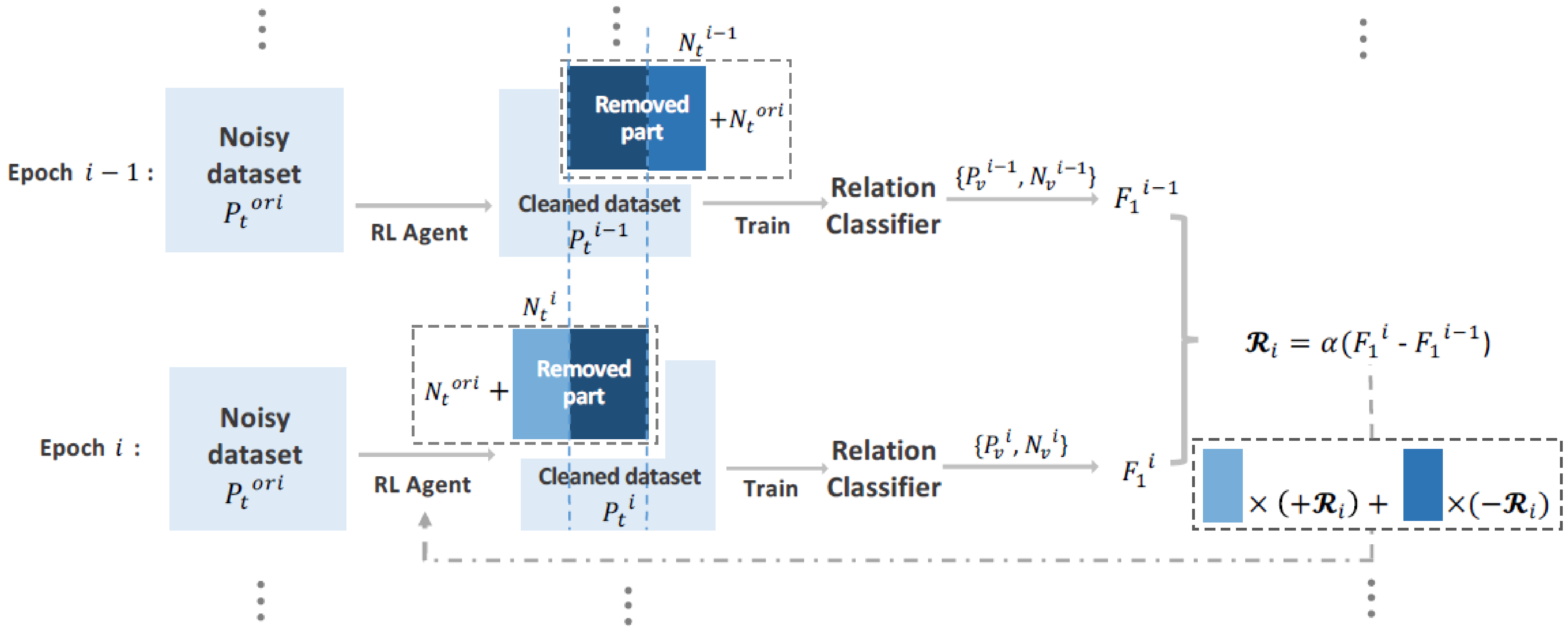
$$R_i = \alpha(F_1^i - F_1^{i-1})$$

- Policy Network
  - Simple CNN (???)

Training RL Agent

- Positive and Negative examples from Distance Supervision $\{P^{ori}, N^{ori}\}$
- Create P_t^{ori}, P_v^{ori} from P^{ori} and N_t^{ori}, N_v^{ori} from N^{ori}
- Sample false positive instances ψ from P_t^{ori} based on agent’s policy
- $P_t = P_t^{ori} - \psi$ $\qquad\qquad$ $N_t = N_t^{ori} + \psi$
- Reward = performance difference on validation set between two epochs

Training RL agent



Pretraining

Pretrain policy networks using Distance Supervision data

Stop this training process when the accuracy reaches 85% ~ 90%

- Difficult to correct biases later
- Better exploration

Training Heuristics

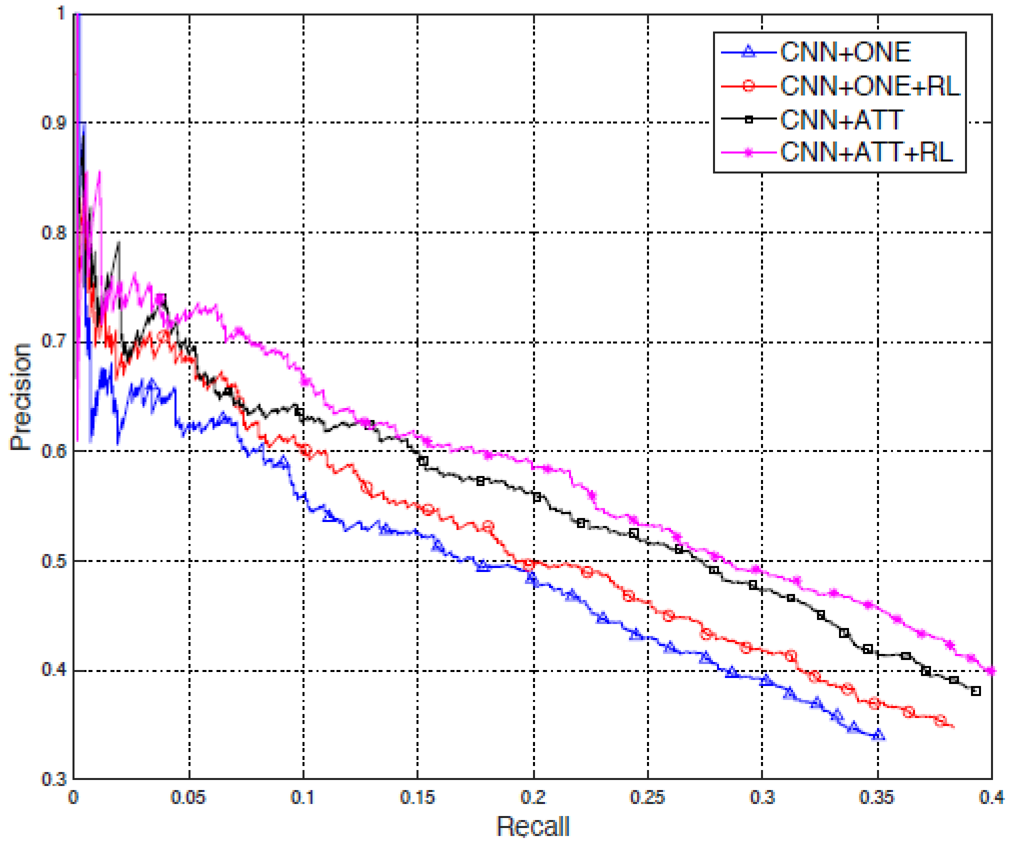
- Hard upper limit on size of ψ
- Loss computation only for non-obvious false positives

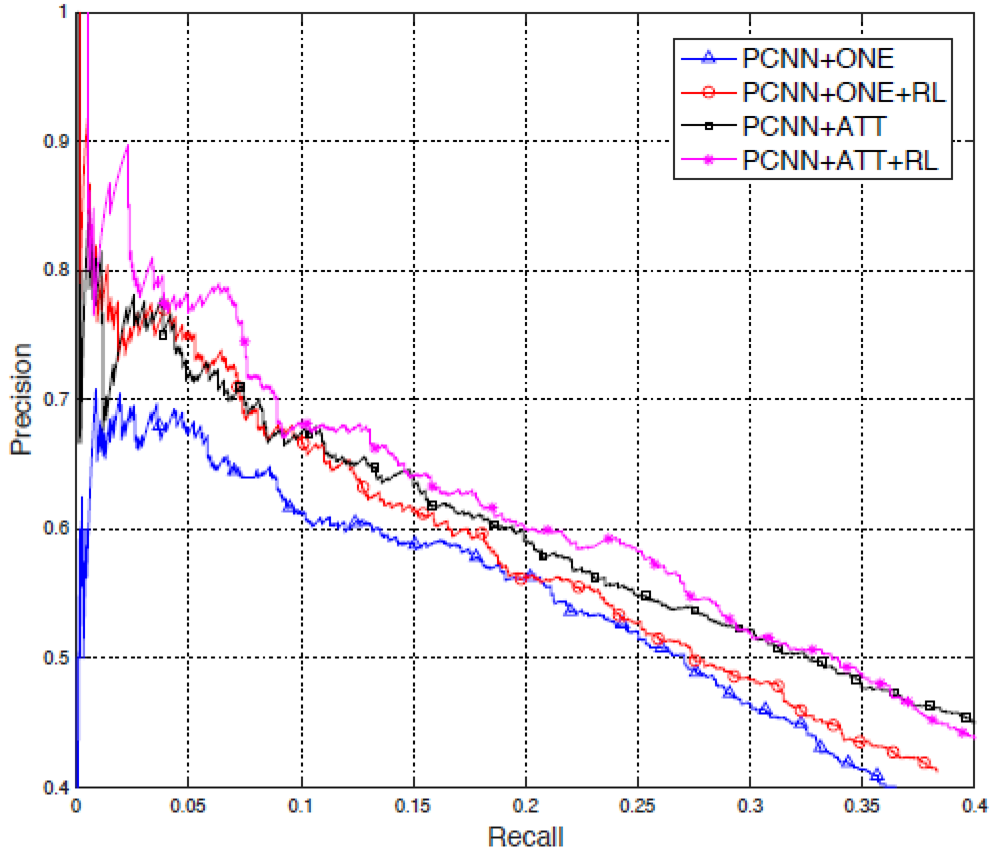
$$\Omega_{i-1} = \Psi_{i-1} - (\Psi_i \cap \Psi_{i-1})$$
$$\Omega_i = \Psi_i - (\Psi_i \cap \Psi_{i-1})$$

$$J(\theta) = \sum^{\Omega_i} \log \pi(a|s;\theta)R + \sum^{\Omega_{i-1}} \log \pi(a|s;\theta)(-R)$$

- Entity pair which has no positive examples left is shifted entirely to negative example set

Results





Results reported are only for the top 10 frequent relation classes in dataset.

Positives

- Applicability to different classifiers
- Pretraining Strategy
- Getting RL to work for NLP task
- Use of simple CNN instead of complex model
    - more sensitive to training data
    - Works with low training data
- It works! Improves performance
- Pseudo Code helps

Negatives

- Evaluation only on top 10 frequent relations
- Non Scalable
  - Retraining relation extraction classifiers from scratch at each epoch
  - Different classifiers for each relation
- Ill defined reward function/MDP
  - Reward function dependent on agent’s choice of val set?
  - Poor intuition of state space definition

Some extensions

- Scope for joint training instead of individual FP classifiers for each relation
- Incremental training instead of training from scratch
- What is the need for RL? Why not just use relation classifier?
  - Maybe RL agent directly optimizes the metric in question?
- Human labelled validation set