NERO: A Neural Rule Grounding Framework for Label-Efficient Relation Extraction

Wenxuan Zhou, Hongtao Lin, Bill Yuchen Lin, Ziqi Wang, Junyi Du, Leonardo Neves, and Xiang Ren

Presented By: Bernal Jimenez Gutierrez

CSE 5539 Fall 2020

Outline

- Introduction
  - Relation Extraction
  - Neural Model Baseline
  - Semi Supervised Methods
- Neural Rule Grounding (NERO)
  - Relation Classifier Module
  - SRM Module
  - Joint Learning
- Experiments
  - Datasets
  - Baselines
  - Results
- Conclusion

Introduction

Sentence Level Relation Extraction

Microsoft was founded by **Bill Gates**.

Relation: founded_by

Mike was born March 26, 1965, in **US**.

Relation: origin

What is the **semantic relationship** between the given entities?

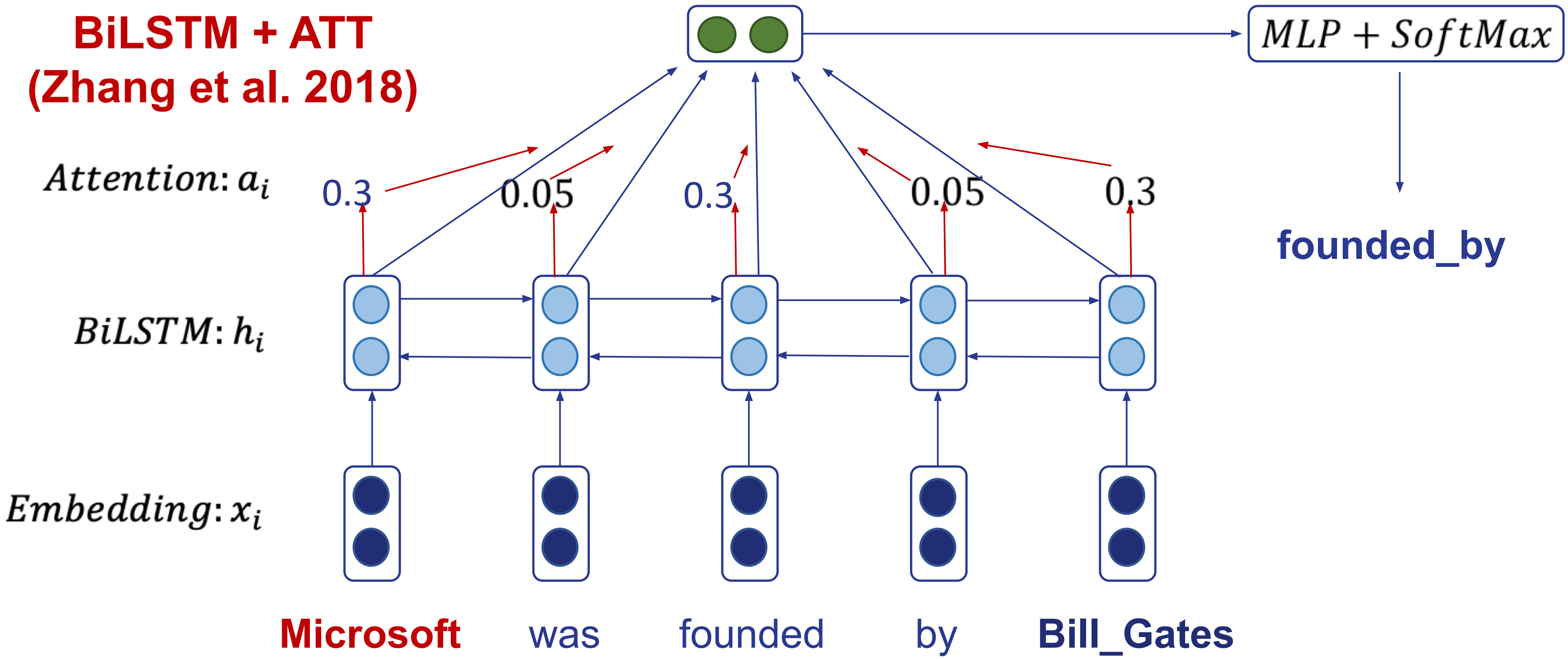
Sentence Level Relation Extraction

$$\mathcal{S} = \{(e^i_{\text{subj}}, e^i_{\text{obj}}; s^i)\}_{i=1}^{N}$$

$$f : \mathcal{S} \rightarrow \mathcal{R} \cup \{\textsc{None}\}$$

What is the **semantic relationship** between the given entities?

Neural Model for Relation Extraction



Need a lot of human-annotated labels!

How do we get them?

Standard Pipeline for Labeling Data

Corpus

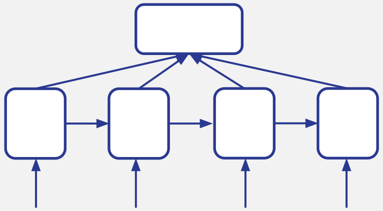
Microsoft was founded by **Bill Gates** in 1975.
Apple was founded by **Steven Jobs** in 1976.
Amazon was founded by **Jeff Bezos** in 1994.
Microsoft was established by **Bill Gates** in 1975.

Labels

ORG: FOUNDED_BY
ORG: FOUNDED_BY
ORG: FOUNDED_BY
ORG: FOUNDED_BY

Annotator

Neural Classifier

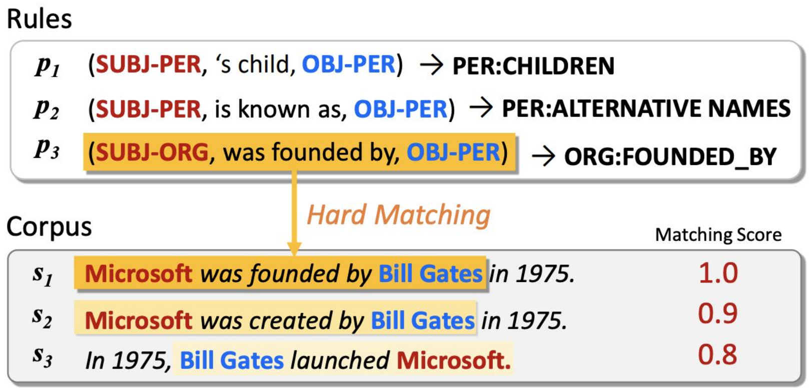


Standard Data Annotation

Slow, redundant annotation efforts on similar instances!

Faster Annotation Methods

- Distant Supervision over a Knowledge Base
  - Uses (subject, relation, object) tuple in curated KB
  - Sentences with subject and object entities in KB tuple are labelled with their specific relation
  - Labels are assigned without inspecting context
  - According to the TACRED paper, up to 31% of distant supervision samples are wrong
- Labeling Rules
  - String pattern based rules are most commonly used
  - Very high precision but low recall problem
  - Most methods which use labelling rules ignore data that was not matched by patterns



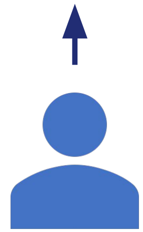
Alternative Labeling Scheme: **Labeling Rules**

Corpus

Microsoft was founded by **Bill Gates** in 1975.
Apple was founded by **Steven Jobs** in 1976.
Amazon was founded by **Jeff Bezos** in 1994.

Labels

ORG: FOUNDED_BY
ORG: FOUNDED_BY
ORG: FOUNDED_BY



Annotator

Labeling Rules

Annotate contextually similar instances
via much fewer rules

(Hearst, 1992)

Challenge: Language Variations

Corpus

Microsoft was founded by **Bill Gates** in 1975.
Apple was founded by **Steven Jobs** in 1976.
Microsoft was established by **Bill Gates** in 1975.
In 1975, **Bill Gates** launched **Microsoft**.

Labels

ORG: FOUNDED_BY
ORG: FOUNDED_BY
No Matched!
No Matched!

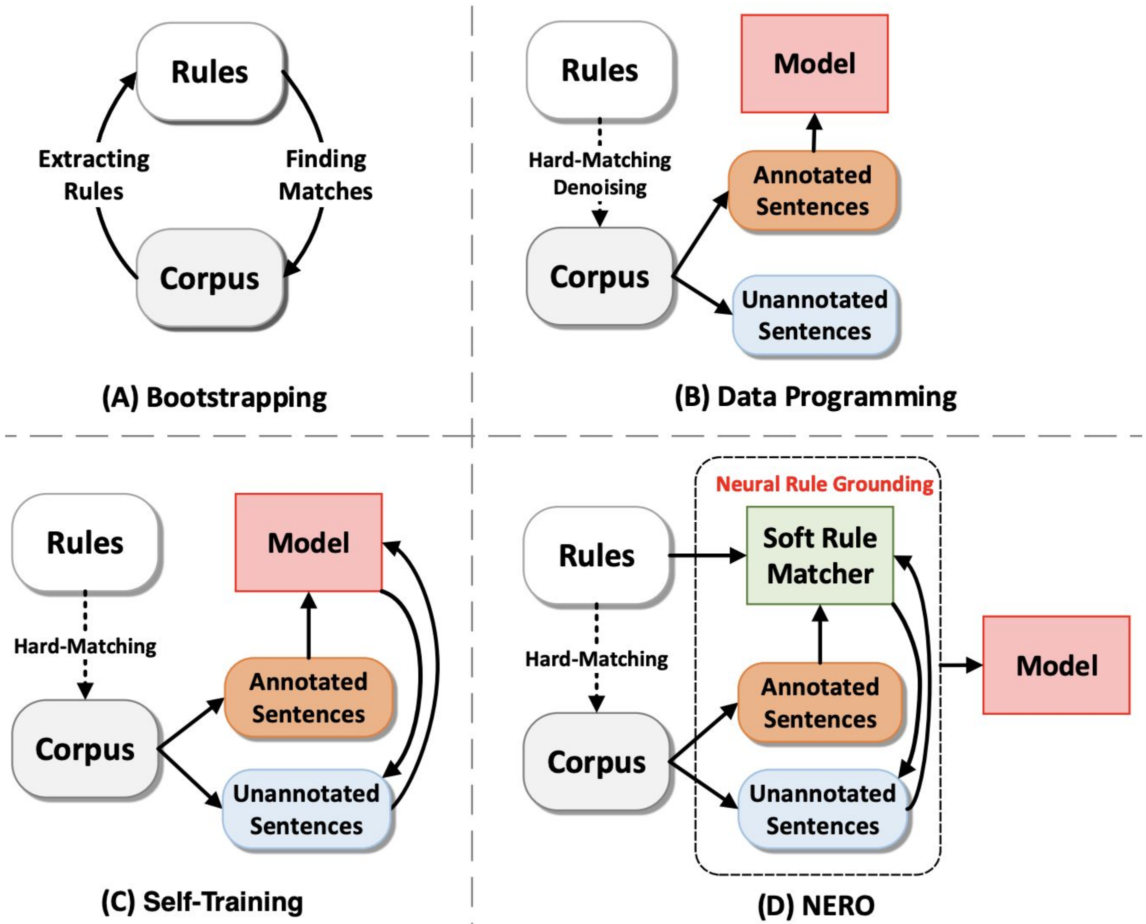
SUBJ-ORG was founded by **OBJ-PER** → **ORG: FOUNDED_BY**

Annotator

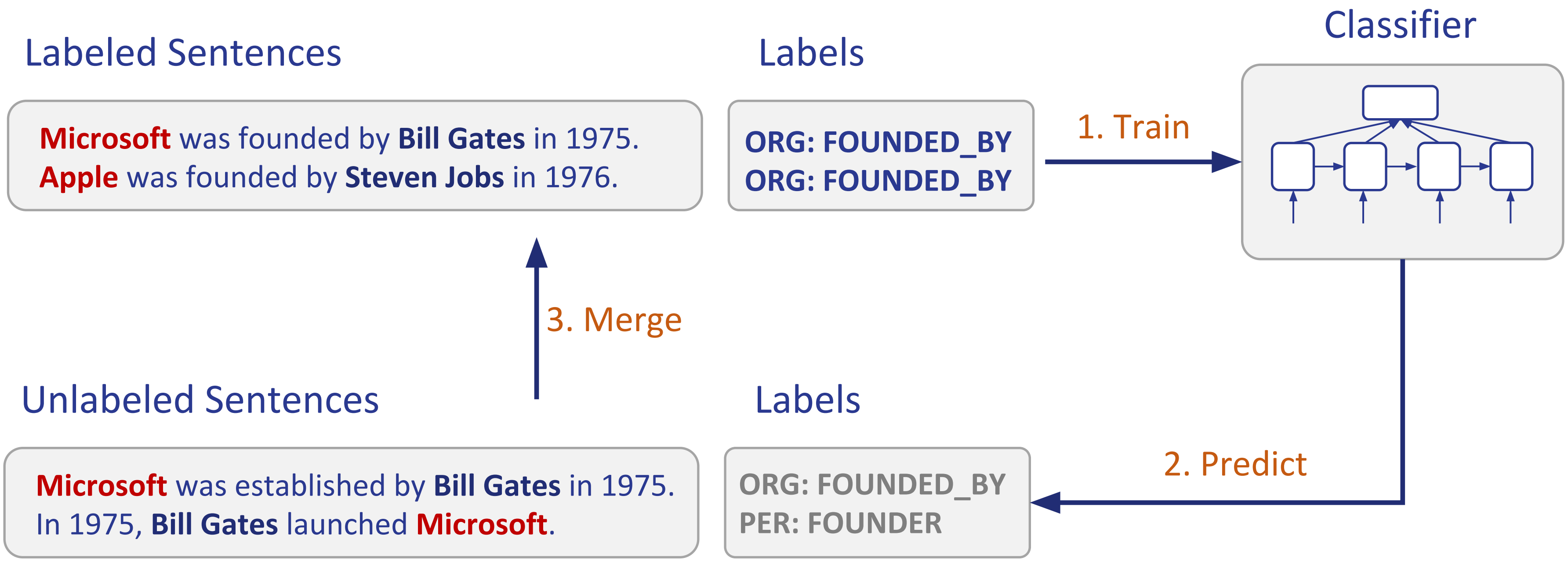
A lot of similar sentences cannot be matched
⇒ Not enough training data
⇒ Poor performance

Do we have to add more labeling rules?

Previous Semi Supervised Methods



Self-Training



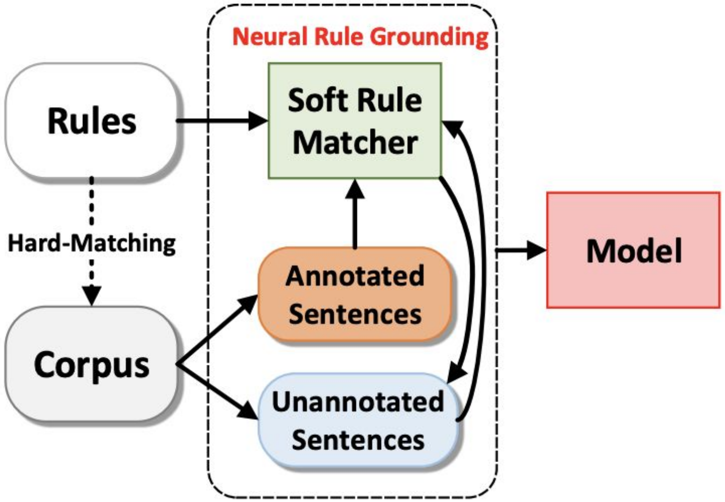
Can create pseudo-labeled data, but will suffer from cascading error propagation

(Rosenberg et al., 2005)

NEural Rule GrOunding Framework (NERO)

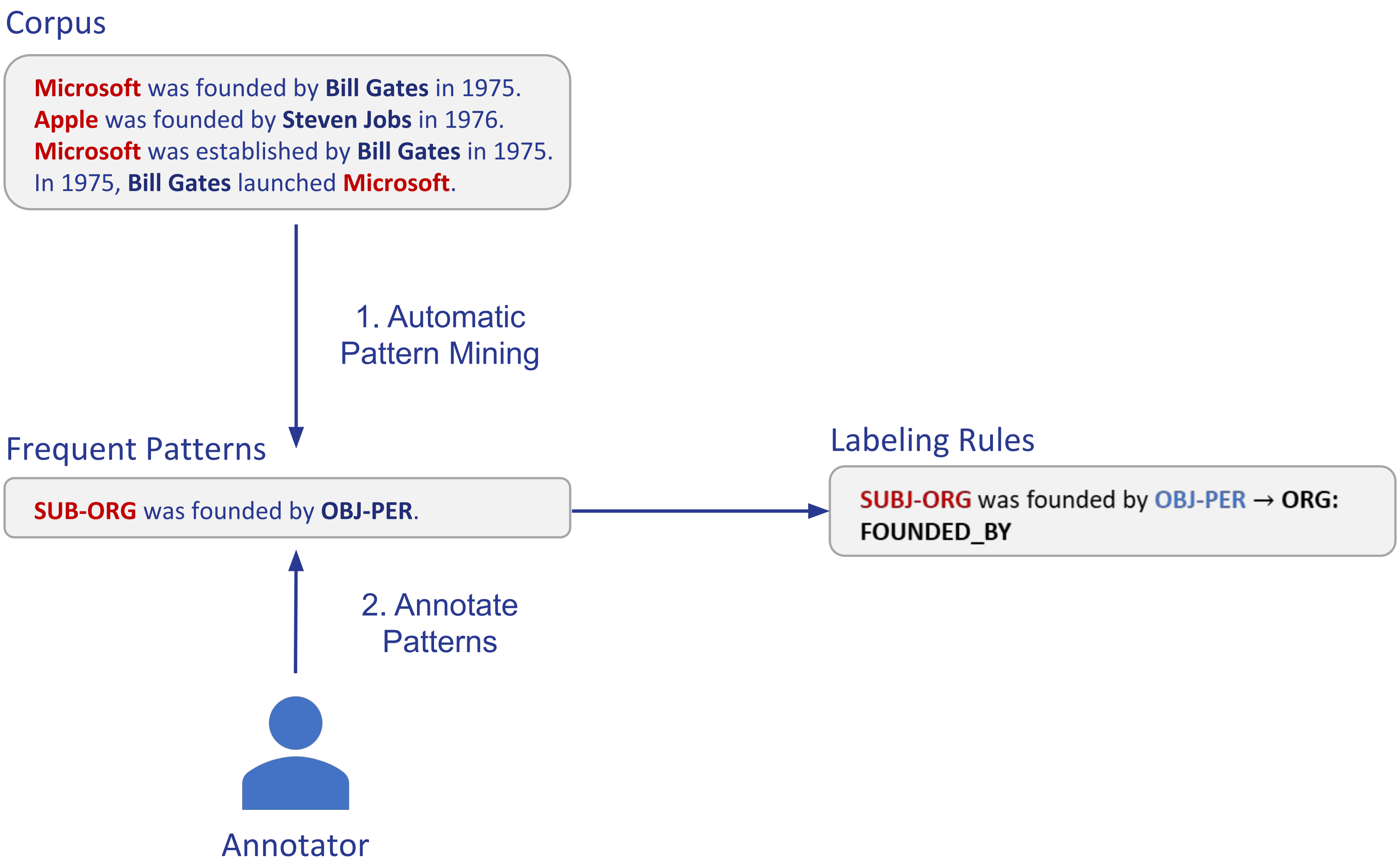
Framework Overview

- NERO splits off the RE model from the self-supervision loop and “grounds” the pseudo labels directly to the rules
- Main aspects
  - Generating Labelling Rules
  - Soft Matcher Module
  - Relation Classifier
  - Joint Training Framework
  - Extra Loss Functions
    - Clustering Loss
    - Rule Loss

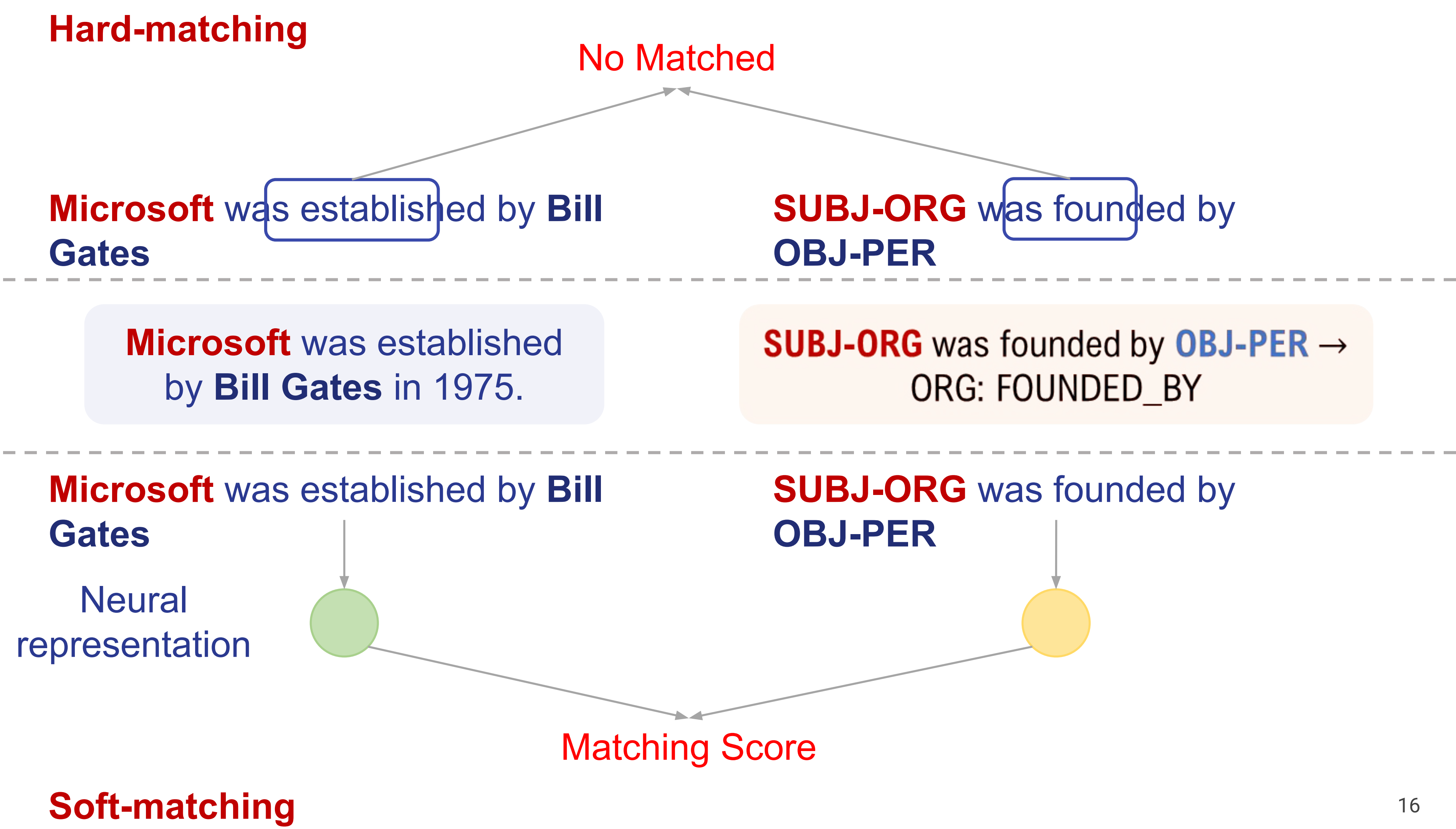


(D) NERO

Generating Labeling Rules



Soft Matcher Module



Soft Rule Matcher: Architecture

Unmatched Sentences

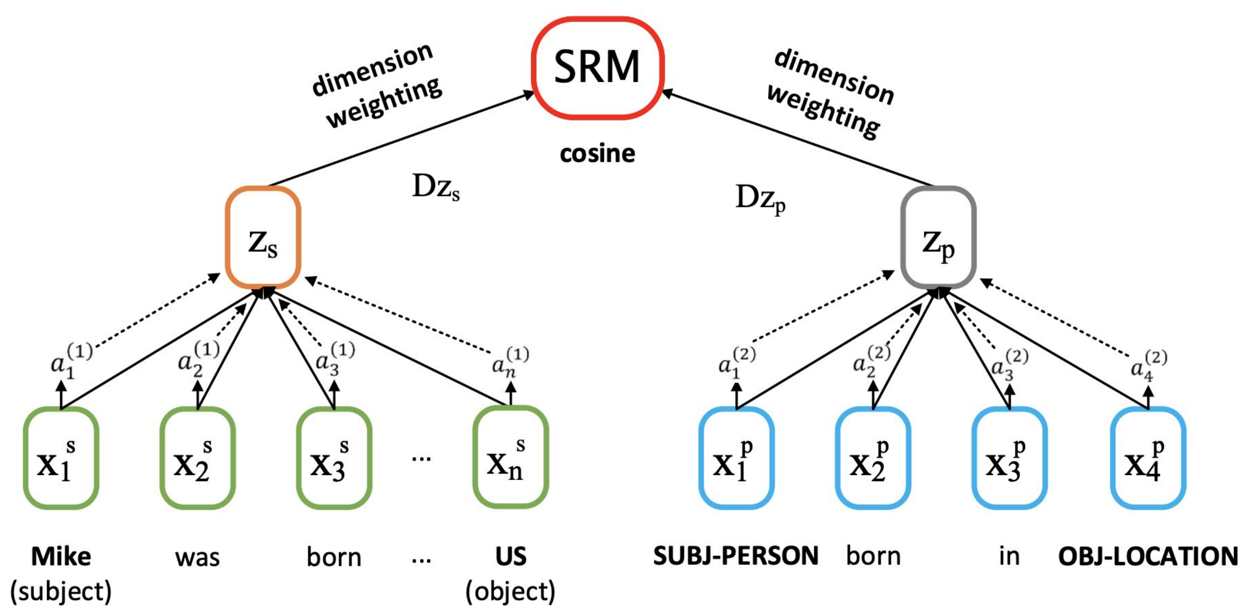
Microsoft was established by **Bill Gates** in 1975.
In 1975, **Bill Gates** launched **Microsoft**.

Labels + Matching Score

ORG: FOUNDED_BY **0.8**
ORG: FOUNDED_BY **0.7**

Labeling Rules

2. Dynamic Soft-matching

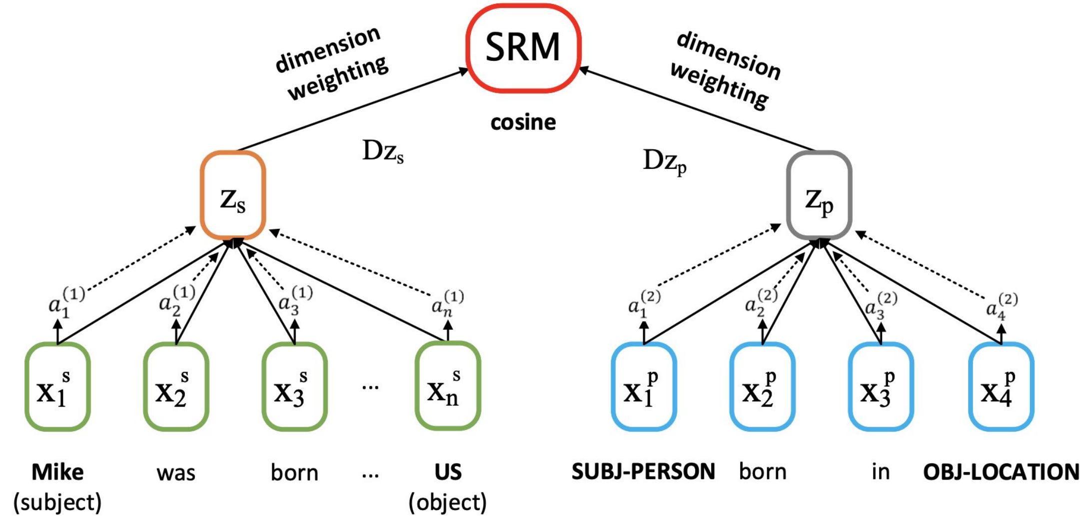


Soft Rule Matcher: Architecture

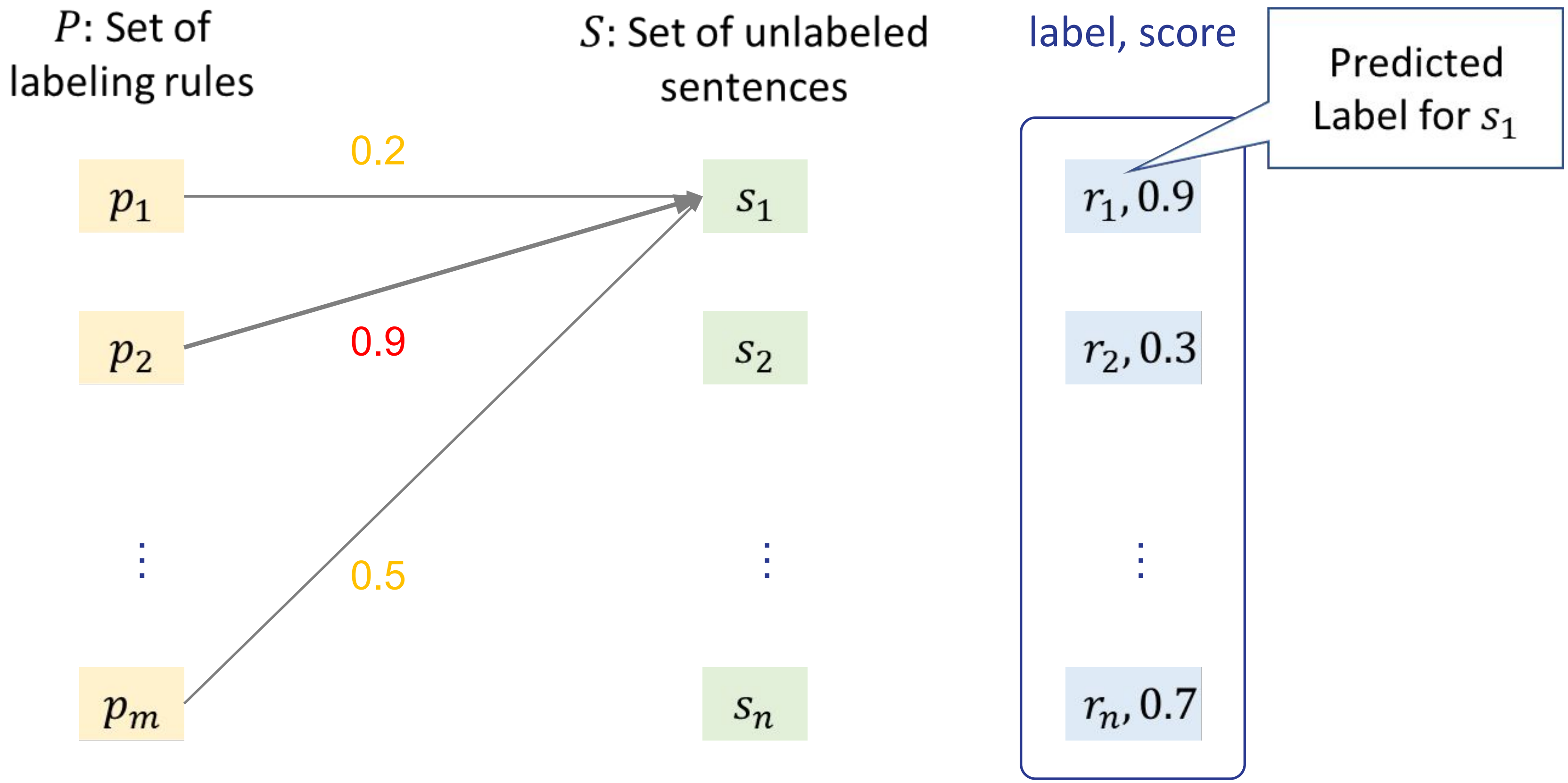
$$\mathbf{z_s} = \sum_{t=1}^{n} \frac{\exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x_t^s}))}{\sum_{t'=1}^{n} \exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x_{t'}^s}))} \mathbf{x_t^s},$$

$$\mathbf{z_p} = \sum_{t=1}^{m} \frac{\exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x_t^p}))}{\sum_{t'=1}^{m} \exp(\mathbf{u}^{\mathrm{T}} \tanh(\mathbf{B}\mathbf{x_{t'}^p}))} \mathbf{x_t^p},$$

$$\mathrm{SRM}(s,p) = \frac{(\mathbf{D}\mathbf{z_s})^T(\mathbf{D}\mathbf{z_p})}{\|\mathbf{D}\mathbf{z_s}\|\|\mathbf{D}\mathbf{z_p}\|},$$



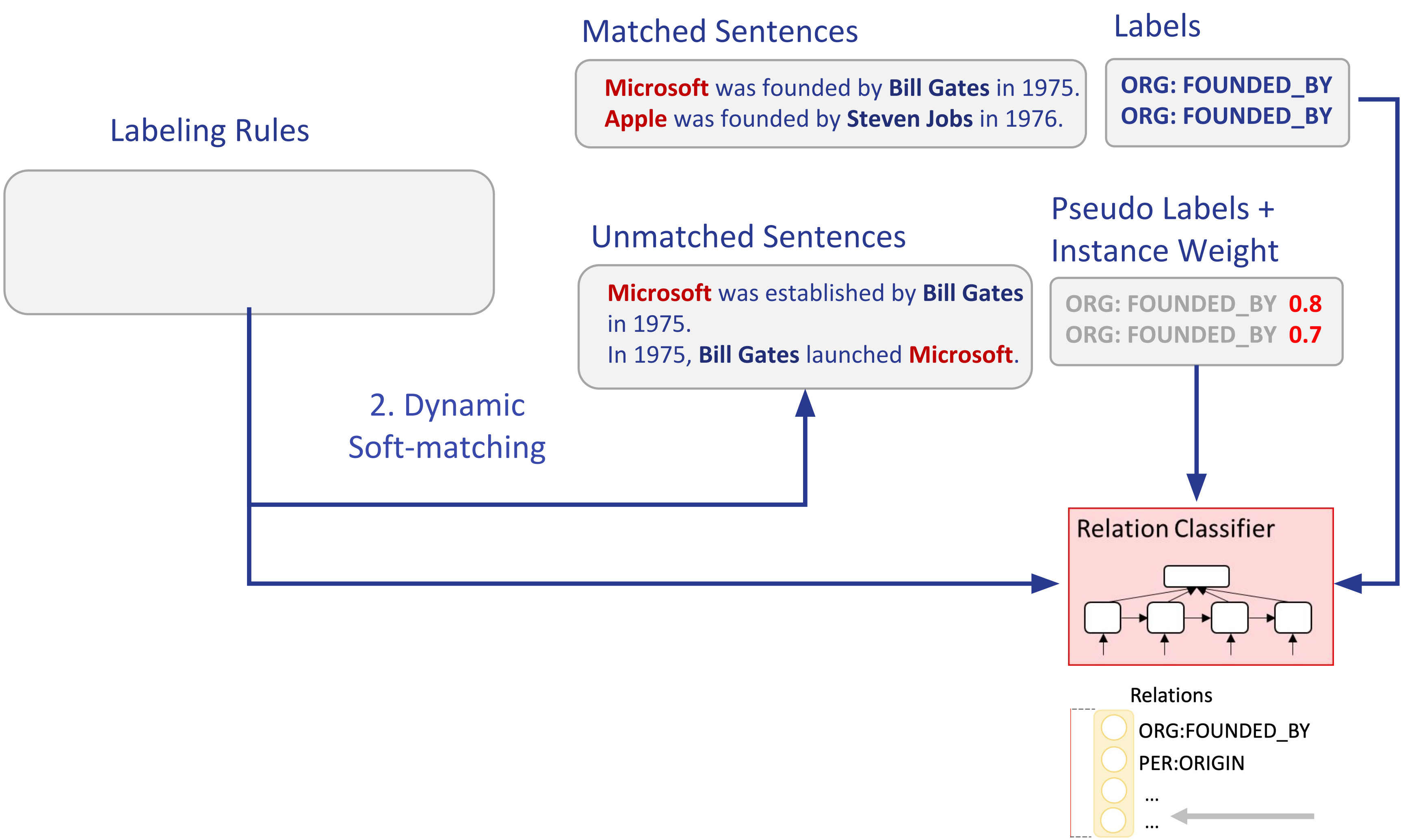
Rethinking the Matching Process



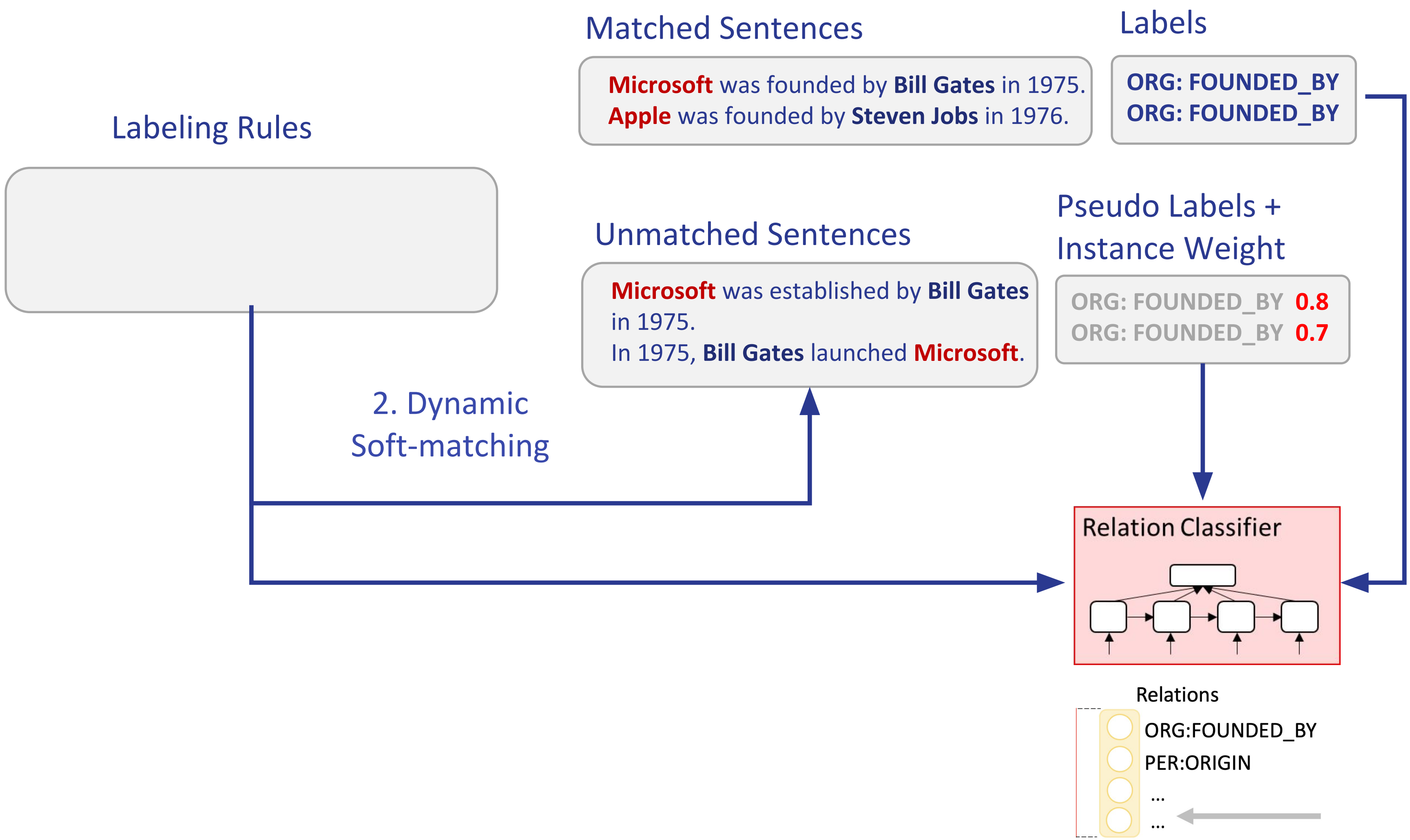
1. soft-matching of rule-sentence pairs

2. weighting labels by matching scores

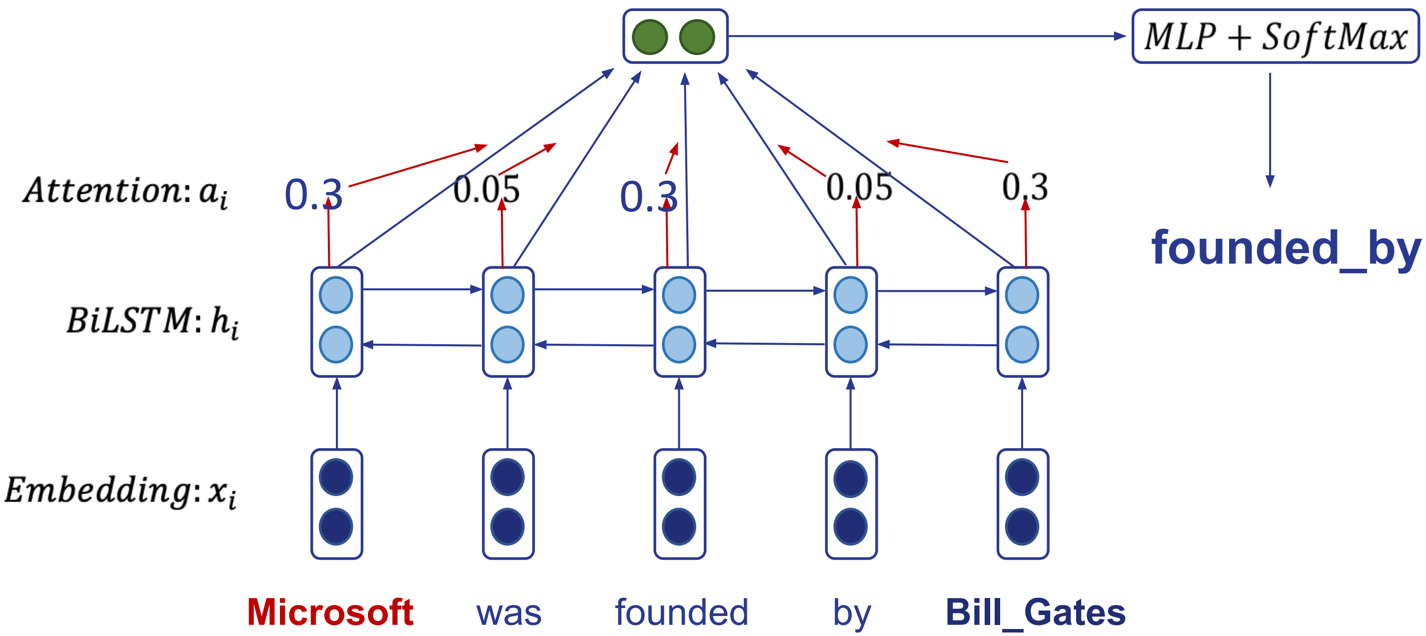
Relation Classifier



Relation Classifier



Neural Model for Relation Extraction



$$\{\mathbf{h_t}\}_{t=1}^{n} = \text{BiLSTM}\left(\{\mathbf{x_t}\}_{t=1}^{n}\right)$$

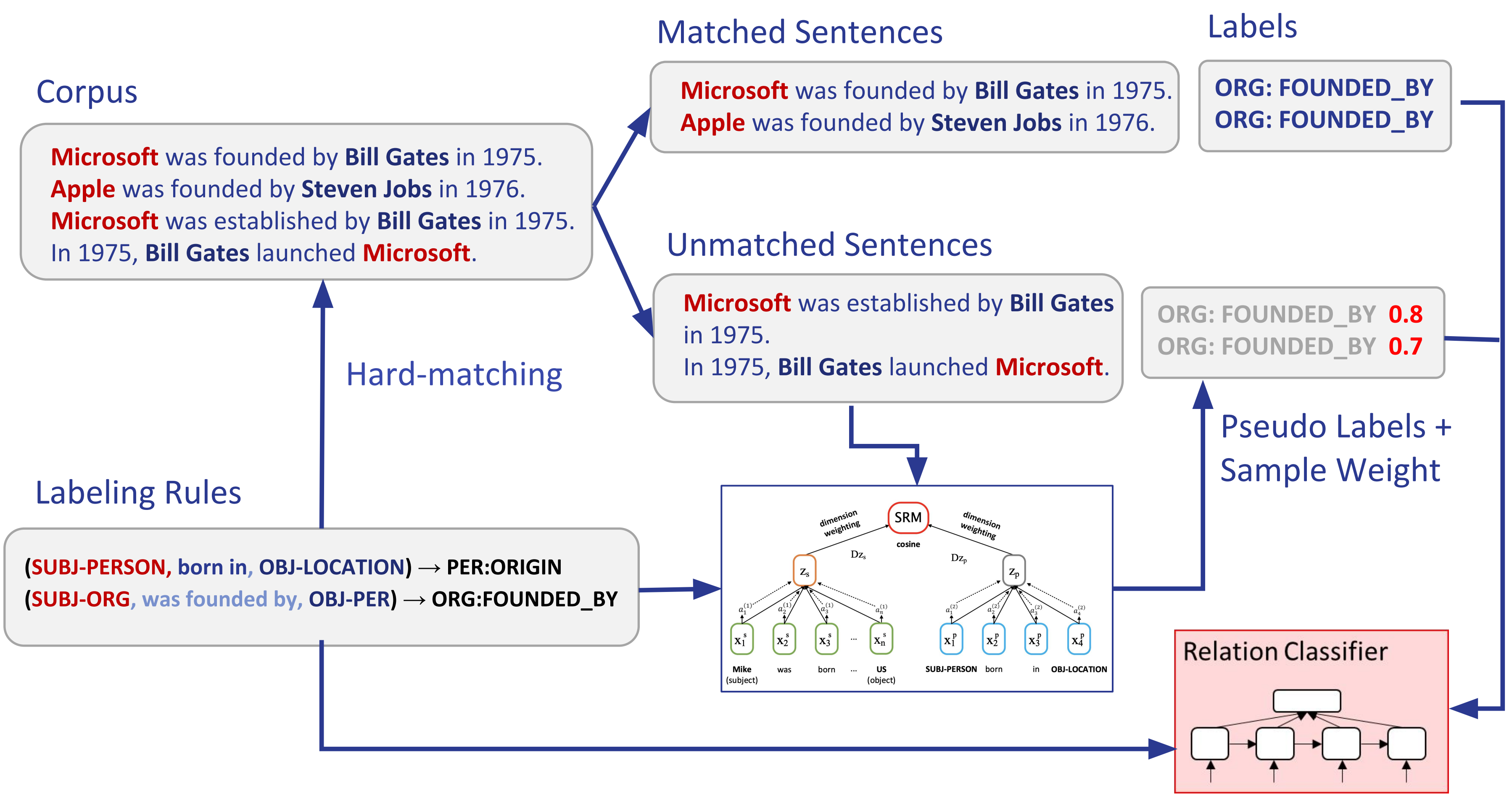
$$\alpha_t = \frac{\exp(\mathbf{v}^{\mathrm{T}} \tanh(\mathbf{A}\mathbf{h_t}))}{\sum_{t'=1}^{n} \exp(\mathbf{v}^{\mathrm{T}} \tanh(\mathbf{A}\mathbf{h_{t'}}))}$$

$$\mathbf{c} = \sum_{t=1}^{n} \alpha_t \mathbf{h}_t$$

$$\text{RC}(s, e_{\text{subj}}, e_{\text{obj}}) = \text{SoftMax}(\mathbf{W}_{\text{rc}}\mathbf{c})$$

$$\mathbb{P}_{\theta_{RC}}(r = i|s) = \text{RC}(s, e_{\text{subj}}, e_{\text{obj}})[i]$$

Joint Parameter Learning:
Relation Extractor + Soft Rule Matcher



Joint Parameter Learning:
Relation Extractor + Soft Rule Matcher

$$L_{\text{matched}}(\theta_{RC}) = \mathbb{E}_{s \sim \mathcal{S}_{\text{matched}}} \left[-\log \mathbb{P}_{\theta_{RC}}(r = r_s | s) \right]$$

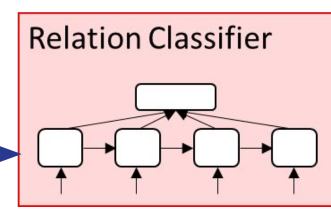
$$L_{\text{rules}}(\theta_{RC}) = \mathbb{E}_{p \sim \mathcal{P}} \left[-\log \mathbb{P}_{\theta_{RC}}(r = r_p | p) \right]$$

Labels

ORG: FOUNDED_BY
ORG: FOUNDED_BY

Labeling Rules

(SUBJ-PERSON, born in, OBJ-LOCATION) → PER:ORIGIN
(SUBJ-ORG, was founded by, OBJ-PER) → ORG:FOUNDED_BY



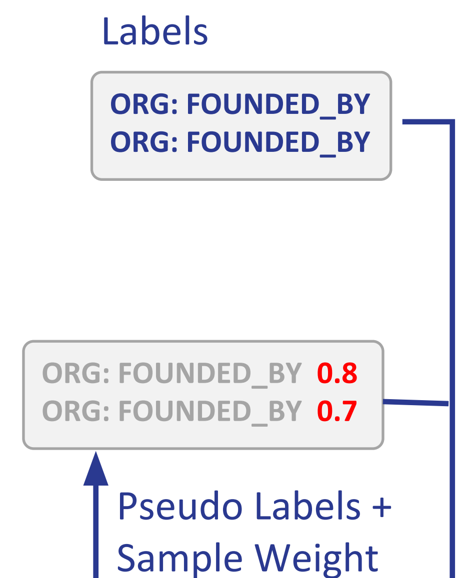
Joint Parameter Learning:
Relation Extractor + Soft Rule Matcher

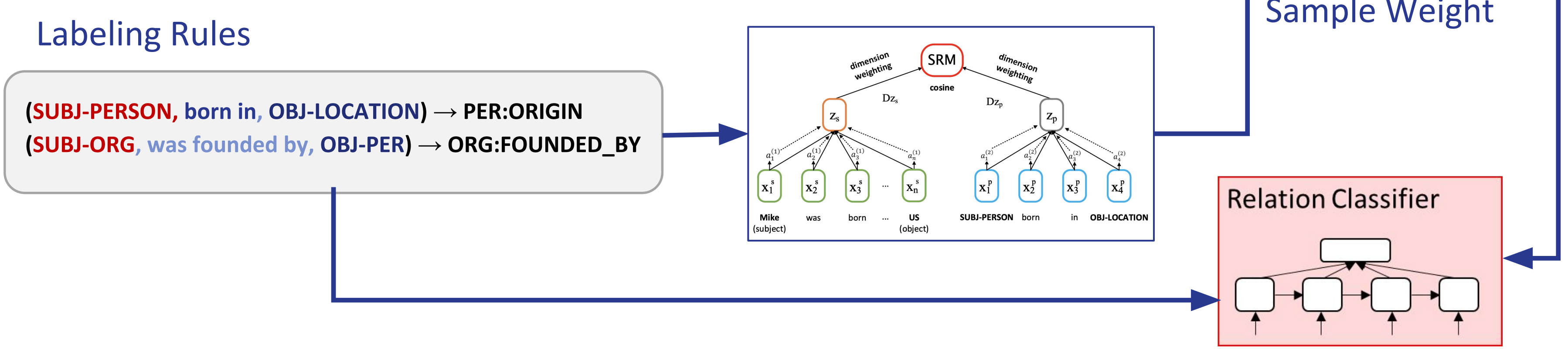
$$L_{\text{matched}}(\theta_{RC}) = \mathbb{E}_{s \sim \mathcal{S}_{\text{matched}}} \left[-\log \mathbb{P}_{\theta_{RC}}(r = r_s | s) \right]$$

$$L_{\text{rules}}(\theta_{RC}) = \mathbb{E}_{p \sim \mathcal{P}} \left[-\log \mathbb{P}_{\theta_{RC}}(r = r_p | p) \right]$$

$$w_s = \frac{\exp(\sigma \mathrm{SRM}(s, \hat{p}_i))}{\sum_{s' \in \mathcal{B}_{\text{u}}} \exp(\sigma \mathrm{SRM}(s', \hat{p}_j))},$$

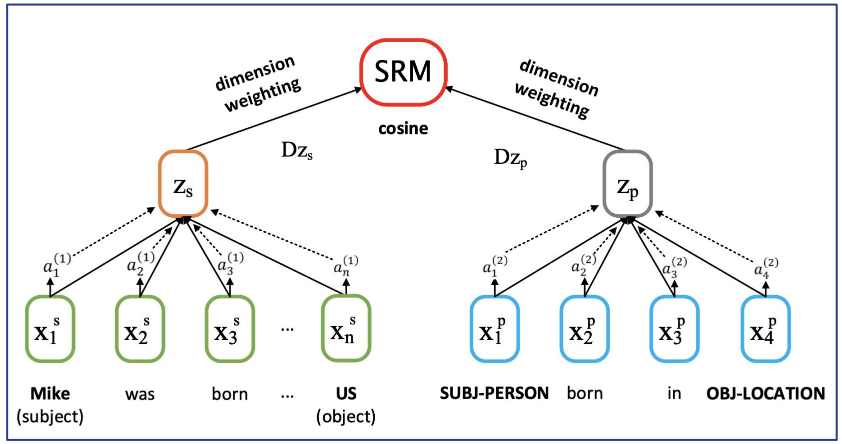
$$L_{\text{unmatched}}(\theta_{RC}) = \frac{1}{|\mathcal{B}_{\text{u}}|} \sum_{s \in \mathcal{B}_{\text{u}}} \left[-w_s \log \mathbb{P}_{\theta_{RC}}(r = r_{\hat{p}} | s) \right]$$

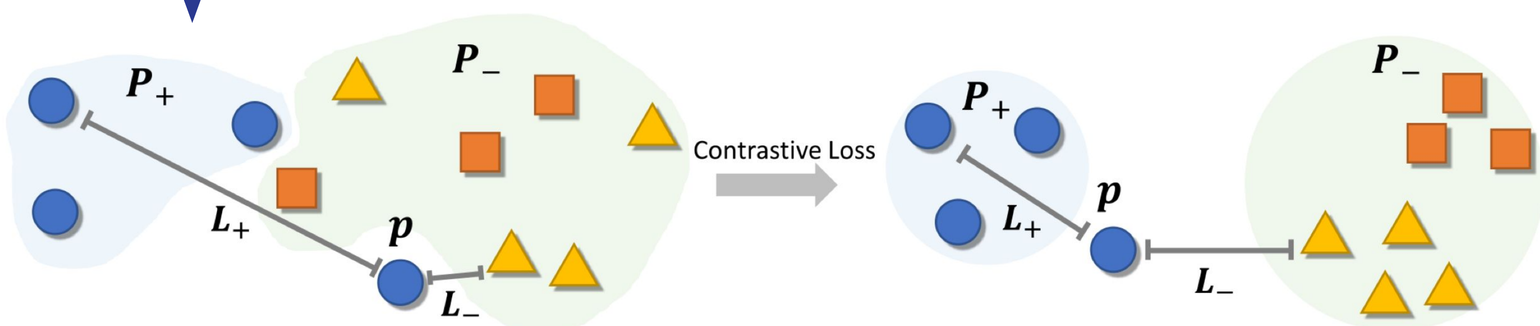




Soft Rule Matcher Clustering

Labeling Rules





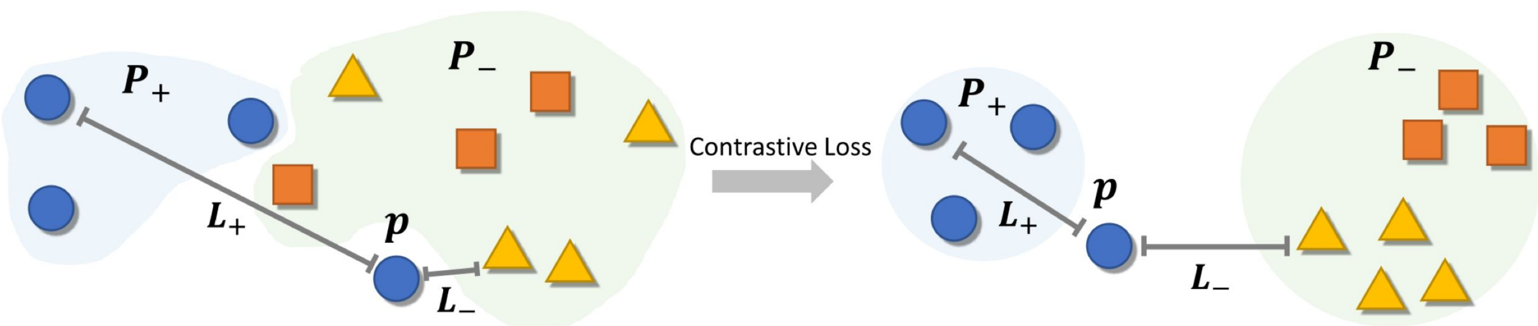
Contrastive loss for discriminating rule bodies

Soft Rule Matcher Clustering

$$L_{\text{clus}} = \mathbb{E}_{p \sim \mathcal{P}} \left[\max_{p_i \in \mathcal{P}_+(p)} \text{dist}_+(p, p_i) - \min_{p_j \in \mathcal{P}_-(p)} \text{dist}_-(p, p_j) \right]$$

$$\text{dist}_+(p, p_i) = \max\left(\tau - \text{SRM}(p, p_i),\ 0\right)^2$$

$$\text{dist}_-(p, p_j) = 1 - \max\left(\text{SRM}(p, p_j),\ 0\right)^2$$



Full Training Algorithm

Algorithm 1: Optimization of NERO model

Input: A raw corpus $\mathcal{S}$, pre-defined relations $\mathcal{R} \cup \{\text{NONE}\}$.
Output: A relation classifier $f : \mathcal{S} \rightarrow \mathcal{R} \cup \{\text{NONE}\}$.
Extract candidate rules from $\mathcal{S}$ with pattern mining tools.
Ask human annotators to select and label the candidate rules to get $\mathcal{P}$.
Partition $\mathcal{S}$ into $\mathcal{S}_{\text{matched}}$ and $\mathcal{S}_{\text{unmatched}}$ by hard-matching with $\mathcal{P}$.
while *L in* Eq. *3.5 not converge* **do**
    Sample batch $\mathcal{B}_m = \{(s_i, r_i)\}_{i=1}^{n}$ from $\mathcal{S}_{\text{matched}}$.
    Update L_{matched} by *Eq.* 7.
    Sample batch $\mathcal{B}_u = \{s_j\}_{j=1}^{m}$ from $\mathcal{S}_{\text{unmatched}}$.
    foreach $s \in \mathcal{B}_u$ **do**
        Find highest-scored rule $\hat{p}$ and pseudo label $r_{\hat{p}}$ by SRM.
    Update $L_{\text{unmatched}}$ by *Eq.* 12.
    Update L_{rules} by *Eq.* 8.
    foreach $p \in \mathcal{P}$ **do**
        Calculate SRM(p, p') for each $p' \in \mathcal{P} - \{p\}$.
        Update L_{clus}.
    $L = L_{\text{matched}} + \alpha \cdot L_{\text{rules}} + \beta \cdot L_{\text{clus}} + \gamma \cdot L_{\text{unmatched}}$.
    Update model parameters *w.r.t.* L.

Model Inference

- Two ways to perform inference
- Relation Classifier obtains best performance
- Soft Matcher Module can be used for inference as well
  - Better interpretability (can present the most semantically similar rule which matched with sentence)
  - Predicting unseen relations using new labelling rules
  - Contextual information is missing and thus performance is worse

Experiments

Datasets

- Rules were generated and annotated for both datasets
- TACRED
  - 79.5% -> No Relation
  - 270 rules annotated
- SemEval
  - 17.4% -> No Relation
  - 164 rules annotated

Dataset	# Train / Dev / Test	# Relations	# Rules	# matched Sent.
TACRED [38]	75,049 /25,763 / 18,659	42	270	1,630
SemEval [11]	7,199 /800 / 1,864	19	164	1,454

Baselines

- Rule-based
  - Rules: Full Pattern Matching
  - CBOW (Soft-matching Cosine Distance)
  - BREDS: Rule Based Bootstrapping for Corpus Level RE
  - Neural Rule Engine
    - Soft matching: accumulates scores among parse tree structure
- Supervised (Supervised models only trained on matched sentences)
  - PCNN
    - Convolution and max pooling over positional and word embeddings
  - LSTM-ATT
  - PA-LSTM
    - Extends LSTM-ATT model with position information
  - Data Programming
    - Denoises conflicting rules by learning heir correlation structures
  - LSTM-ATT (Matched S + P)
    - Trains on small # of rules as well

Baselines

- Semi-Supervised
  - Pseudo-Labeling
    - Labels all unlabelled data with trained model
  - Self-Training
    - Iteratively trains and labels only most confident predictions in unlabelled data
  - Mean-Teacher
    - Self-training + perturbing unlabeled sentences and encouraging outputs to be similar
  - Dual RE
    - Jointly trains a model that retrieves unlabelled sentences for each relation along with RC
- Nero Variants
  - NERO w/o unmatched S
    - Removing unmatched loss (equivalent to LSTM-ATT (matched S + P) + Cluster loss)
  - NERO-SRM Inference
    - Inference performed with SRM modules
    - Context agnostic version of NERO

Main Results

Method / Dataset	TACRED			SemEval		
	Precision	Recall	F_1	Precision	Recall	F_1
Rules	85.0	11.4	20.1	81.2	17.2	28.5
BREDS [4]	53.8	20.3	29.5	62.0	24.5	35.1
CBOW-GloVe	27.9	45.7	34.6	44.0	52.8	48.0
NRE [17]	65.2	17.2	27.2	78.6	18.5	30.0
PCNN [36]	44.5 ± 0.4	24.1 ± 2.8	31.1 ± 2.6	59.1 ± 1.4	43.0 ± 0.7	49.8 ± 0.5
LSTM+ATT	38.1 ± 2.7	39.6 ± 2.7	38.8 ± 2.4	64.5 ± 2.8	53.3 ± 2.8	58.2 ± 0.8
PA-LSTM [38]	39.8 ± 2.5	40.2 ± 2.0	39.0 ± 0.6	64.0 ± 3.6	54.2 ± 2.5	58.5 ± 0.6
Data Programming [25]	39.2 ± 1.3	40.1 ± 2.0	39.7 ± 0.9	61.8 ± 2.1	54.8 ± 1.1	58.1 ± 0.7
LSTM+ATT ($\mathcal{S}_{\text{matched}} + \mathcal{P}$)	39.2 ± 1.7	45.5 ± 1.7	42.1 ± 0.9	63.4 ± 2.1	55.0 ± 0.3	58.8 ± 0.9
Pseudo-Labeling [16]	34.5 ± 4.1	37.4 ± 5.1	35.3 ± 0.8	59.4 ± 3.3	55.8 ± 2.1	57.4 ± 1.3
Self-Training [26]	37.8 ± 3.5	41.1 ± 3.1	39.2 ± 2.1	62.3 ± 2.0	53.0 ± 2.7	57.1 ± 1.0
Mean-Teacher [31]	46.0 ± 2.7	41.6 ± 2.2	43.6 ± 1.3	62.3 ± 1.5	54.5 ± 1.2	57.9 ± 0.5
DualRE [18]	40.2 ± 1.5	42.8 ± 2.0	41.7 ± 0.5	63.7 ± 2.8	54.6 ± 2.1	58.6 ± 0.8
NERO w/o $\mathcal{S}_{\text{unmatched}}$	41.9 ± 1.8	44.3 ± 3.8	42.9 ± 1.4	61.4 ± 2.4	56.2 ± 1.9	58.6 ± 0.6
NERO-SRM	45.6 ± 2.2	45.2 ± 1.2	45.3 ± 1.0	54.8 ± 1.6	55.2 ± 2.0	54.9 ± 0.6
NERO	54.0 ± 1.8	48.9 ± 2.2	**51.3 ± 0.6**	66.0 ± 1.5	55.8 ± 0.9	**60.5 ± 0.7**

Main Result Takeaways

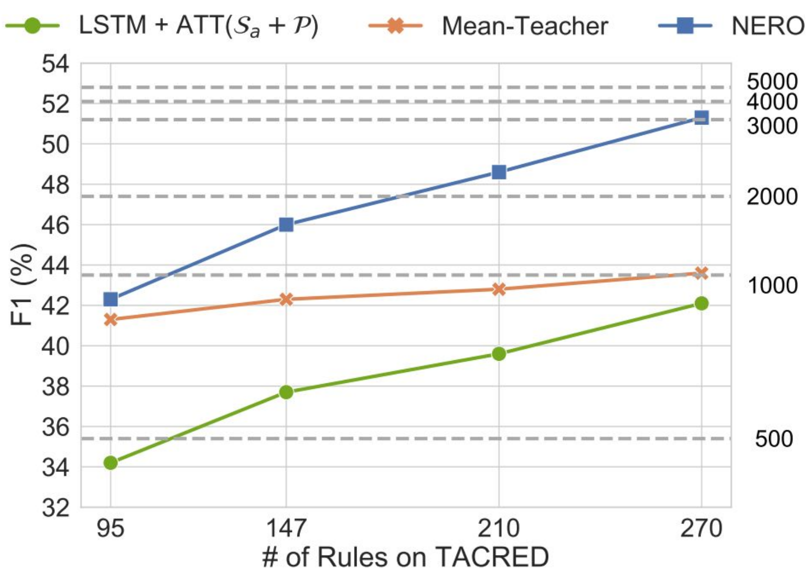
- Rule based models suffer from severe low recall problem
  - Best recall is 27% on TACRED and 24% on SemEval
  - CBOW soft matching has better recall but precision drops due to lack of context
- Supervised models
  - 4-5% improvement over CBOW soft-matching
  - Data programming does not help since rules are fairly independent
  - Important to note that these models are only training on sentences matched by hard-matching rules
  - 2020 SOTA using whole TACRED is much higher (74.8%)

Main Result Takeaways

- Adding Unlabelled Sentences
  - Self training performance drops compared to supervised model
    - Generated labels are too noisy due to low quality model
    - Mean Teacher obtains small improvements ~1%
  - NERO obtains ~9% improvement over base supervised model
  - This shows that using the rules directly for soft labelling reduces the noise in generated labels
- Difference in performance on SemEval is much smaller (~1.7%)
  - Supervised models do as well as all self-training except NERO
  - Authors hypothesize that this is due to SemEval having simpler rules and shorter sentences than TACRED

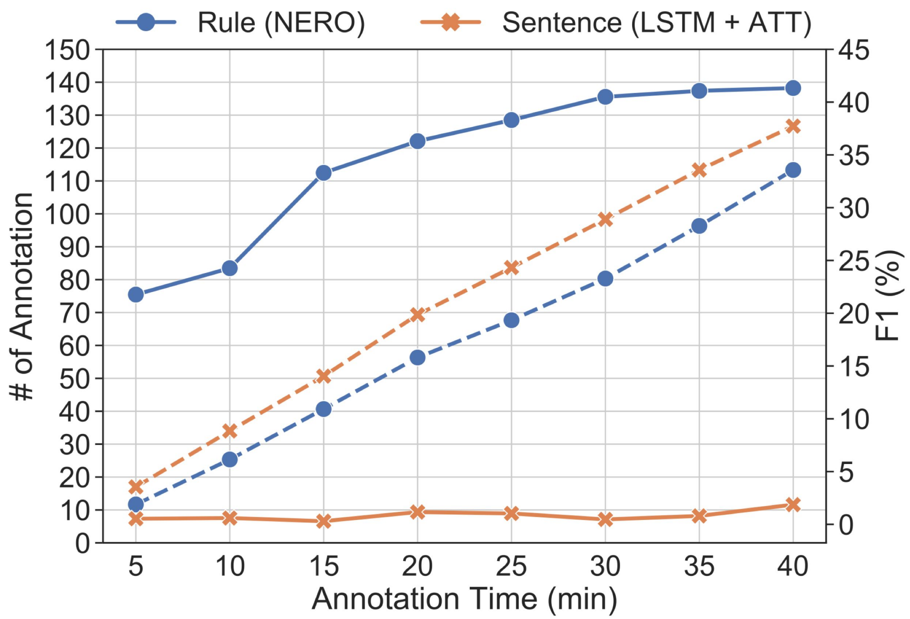
Rule Efficiency Study

- NERO performs as well as a supervised model with 3000 annotated labels using 270 rules
    - 10 x more efficient
- Even LSTM + ATT being trained on rules is 4 x more efficient than label annotations
- Takeaway:
    - Under constraints, consider using rule extraction instead of instance labelling



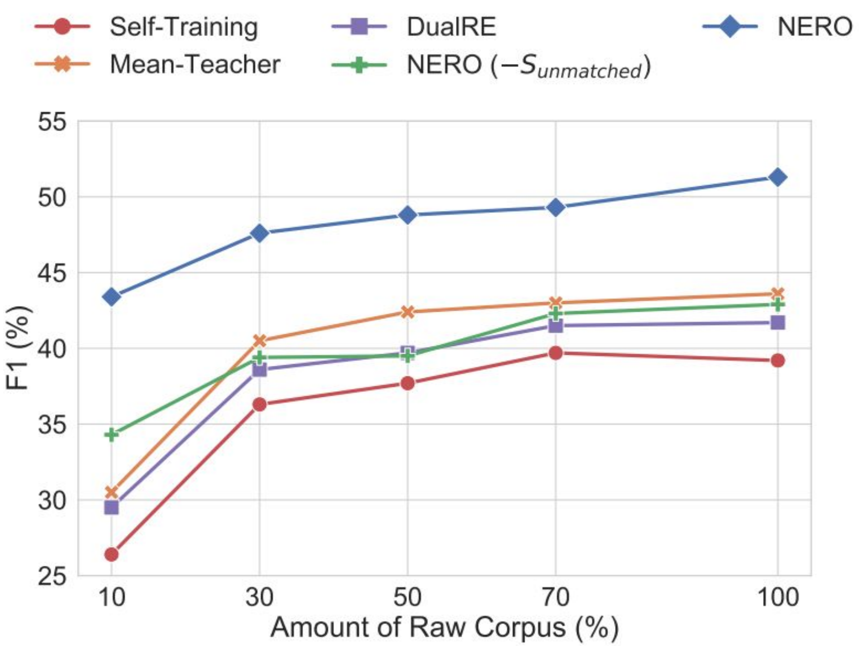
Label Efficiency Study

- 5 students spent 40 min labeling instances from TACRED
- Dashed: Avg # of **rules** / **sentences** labeled by annotators.
- Solid: Avg **model F1** trained with corresponding annotations
- Takeaways
  - With NERO it is possible to get much more reasonable performance with very minimal labelling investment



Raw Corpus Study

- This study shows that NERO leverages the TACRED unlabelled corpus more efficiently than all other self-training at all corpus sizes
- If trend continues, more unlabelled data might increase performance further



Unseen Relations Study

- 5 random relations removed from training data but not test data (10 different sets)
  - Test set contains only 5 relations and 'No relation' with same ratio as in original test
- For NERO we use the SRM module for inference of new relations with new rule set
- CBOW and BERT-base compare rule and sentence representations
- Authors claim that SRM learn more information about relation matching but frozen BERT is competitive does not support this idea

	TACRED			SemEval		
Method	P	R	F_1	P	R	F_1
Rule (exact match)	100	6.1	10.8	83.2	17.7	28.2
CBOW-GloVe	52.4	86.3	64.7	40.3	45.5	34.7
BERT-base (frozen)	66.2	76.8	**69.5**	37.8	33.2	35.3
NERO	61.4	80.5	68.9	43.0	54.1	**45.5**

Different SRM Modules Study

- Reported NERO performance using different SRM functions
- Surprisingly, non-contextual model performs better than both LSTM-ATT contextual model and fine tuned BERT
- Authors point out that rule BERT gave high scores to almost all sentence-rule pairs, making it harder to predict the most likely

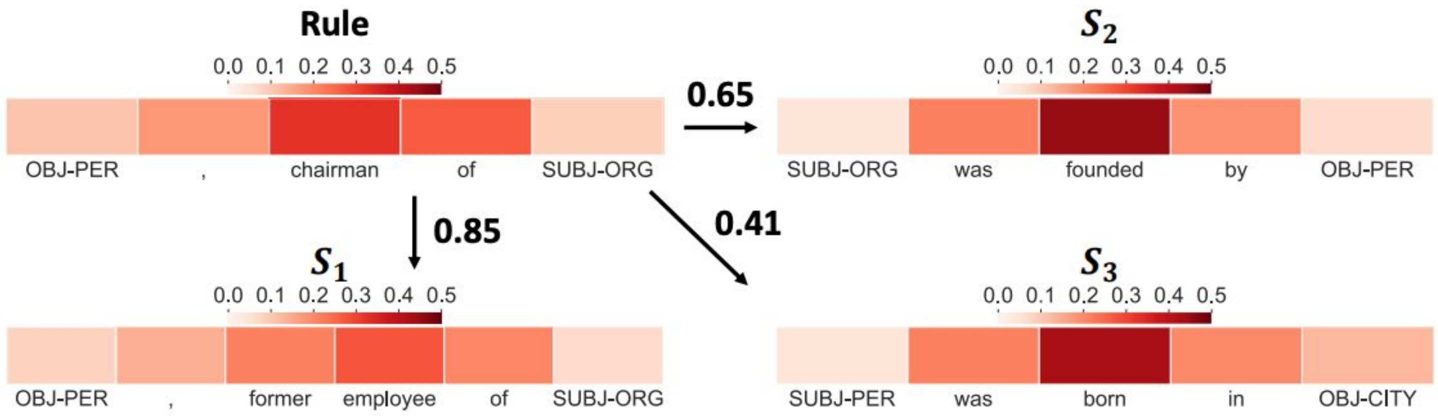
Objective	Precision	Recall	F_1
CBOW-Glove	49.4	43.5	46.2
LSTM+ATT	56.2	46.0	50.6
BERT-base (frozen)	45.6	47.6	46.5
BERT-base (fine-tuned)	50.3	45.8	47.9
Word-level attention (ours)	54.0	48.9	51.3

Model Ablation

- Removing different parts of the NERO framework
- Removing self-supervision training dropped performance to matched sentence supervised baseline
- Contrastive loss is also important for model performance
    - Directly training the rule representations to be discriminative in terms of relations is useful

Objective	Precision	Recall	F_1
L (ours)	54.0	48.9	51.3
$-L_{\text{rules}}$	50.0	47.7	49.0
$-L_{\text{clus}}$	50.9	43.0	46.4
$-L_{\text{unmatched}}$	41.9	44.3	42.9

SRM Interpretability Case Study



- Soft Rule Matcher is claimed to be more interpretable
- Qualitative study to show the weight of different sentences given a rule
- Labelling using the SRM gives access to the rule which labelled the sample
- Improves end user confidence and ability to verify model prediction

Conclusion

- Using rules directly for self-supervision in the relation extraction yields higher quality labels
- Rule labelling is much more efficient than instance annotation