NAACL 2021 follow-up

Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach

https://www.aclweb.org/anthology/2021.naacl-main.84/

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, Chao Zhang

Some slides copied from authors' presentation at NAACL-2021

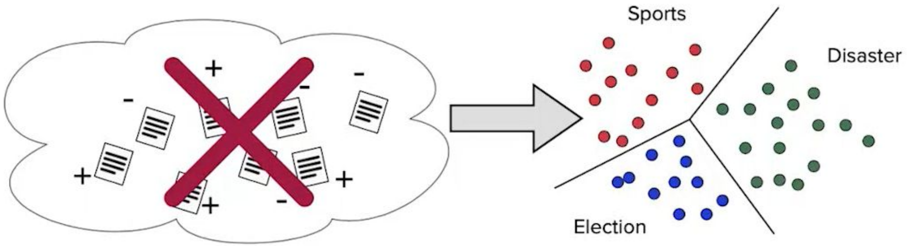
Background

1. Deep learning model is label hungry
2. Labeled data is expensive to obtain.

Our Goal: Fine-tuning Language Models with Weak Supervision

Traditional methods rely on manual annotations from domain experts – Time Consuming and Expensive

Unlabeled Text Data



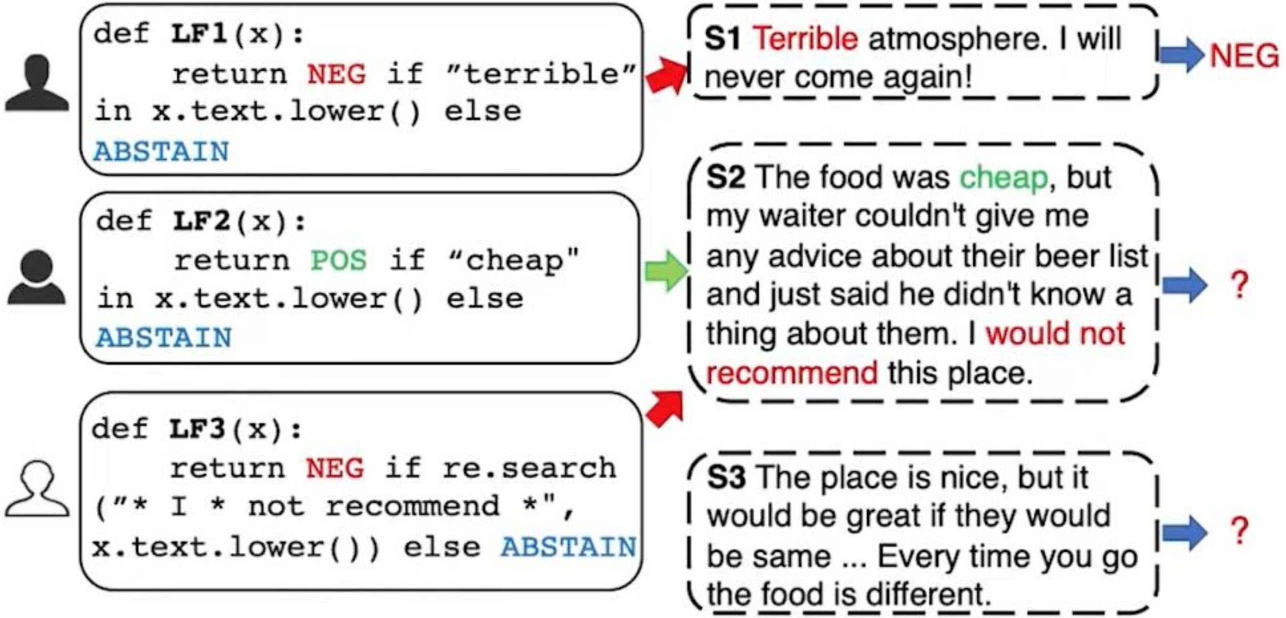
WIKIPEDIA Die freie Enzyklopädie Freebase DBpedia yago select knowledge

Structured Knowledge & Insights

We aim to only use existing knowledge base/heuristics as weak supervision to automatically perform downstream NLP tasks

Weak Supervision Sources

- Labeling Function – a unified ways to represent weak supervision



Drawbacks of Weak Supervision Sources

- Limited Coverage
  - Weak supervisions are often too specific to cover all cases
  - Many training data cannot be labeled

- Noisy
  - Weak supervisions are often too simple to capture the rich context information
  - Pre-trained language models are usually giant models, which are especially vulnerable to the label noise

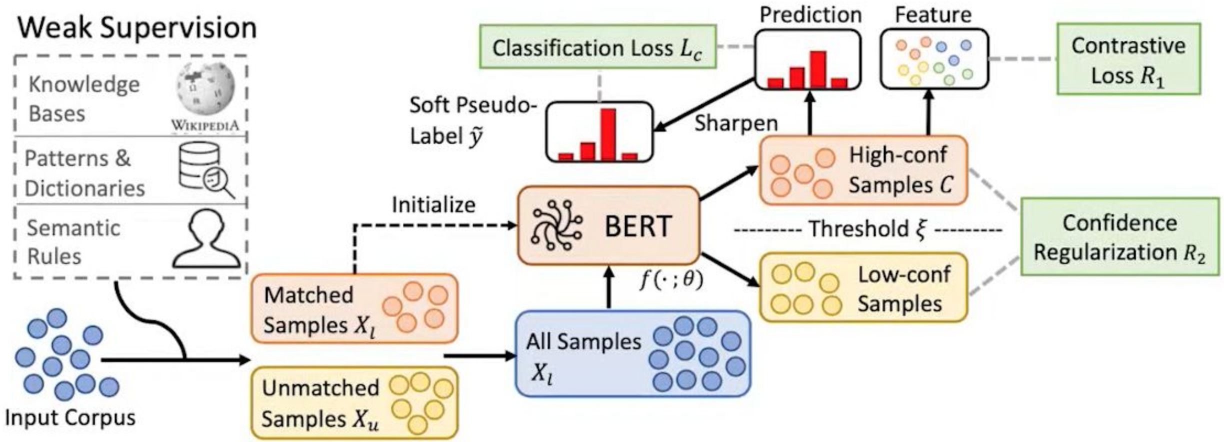
Our Framework: Self-training for LM Fine-tuning

- How to fine-tune pre-trained language models with weak supervision only, without any external knowledge?
  - Our solution: use self-training for denoising weak labels

- Self training can …
  - Generate pseudo labels for unlabeled examples to augment the training set
  - Denoise the noisy labels via gradually refining the pseudo labels

Our Framework: Self-training for LM Fine-tuning

- Overall framework



  - Initialize with *weakly labeled* data
  - Self-training with *both labeled and unlabeled data*

Self-training: Initialization with Weak Labels

- Directly fine-tune pre-trained language model $f(\theta)$ with weakly labeled data

$$\min L = \frac{1}{|X_L|} \sum_{(x_i,y_i)\in X_L} \ell(f(x_1;\theta), y_i)$$

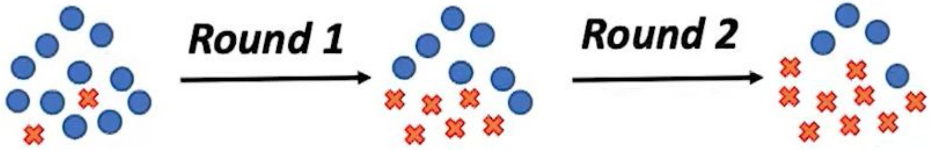
- Early Stopping
  - Prevent the LM for *overfitting* to label noise

Self-training: Learning with All Data

- Generate pseudo labels for all data

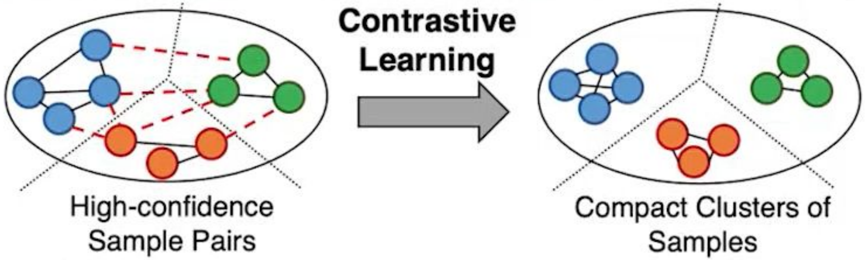
$$\min L = \frac{1}{|X_L|} \sum_{(x_i, y_i) \in X_L} KL(f(x_i;\theta), \widetilde{y_i})$$

- $\widetilde{y_j} = \frac{[f(x;\theta)]_j^2 / f_i}{\sum_{j'} [f(x;\theta)]_{j'}^2 / f_{j'}}$ is the soft label associated with x
- One potential drawback: Self-training suffers from error-propagation – *More and more wrong examples are created!*
- One Example:



Robust Self-training with Contrastive Regularization

- Contrastive Learning on *Feature Space* with High-confidence Samples



- Similarity between samples

$$W_{ij} = \begin{cases} 1, & \text{if } \underset{k\in\mathcal{Y}}{\operatorname{argmax}}[\widetilde{\boldsymbol{y}}_i]_k = \underset{k\in\mathcal{Y}}{\operatorname{argmax}}[\widetilde{\boldsymbol{y}}_j]_k \\ 0, & \text{otherwise} \end{cases}$$

- Contrastive Regularization

$$\ell = W_{ij} d_{ij}^2 + (1 - W_{ij})[\max(0, \gamma - d_{ij})]^2$$

Other Techniques for Improving Self-training

- Confidence-based Sample Reweighting
  - Reweight different samples based on prediction accuracy

$$\omega = 1 - \frac{H(\tilde{y})}{\log(C)}, H(\tilde{y}) = -\sum_{i=1}^{C} \tilde{y_i} \log \widetilde{y_i}$$

- Confidence-based regularizer encouraging *smoothness* over predictions

$$\ell = KL\big(u \| f(x;\theta)\big)$$

- Final Loss

$$L = L_1 + R_1 + R_2$$

Self-training Loss

Contrastive Loss

Confidence Regularization

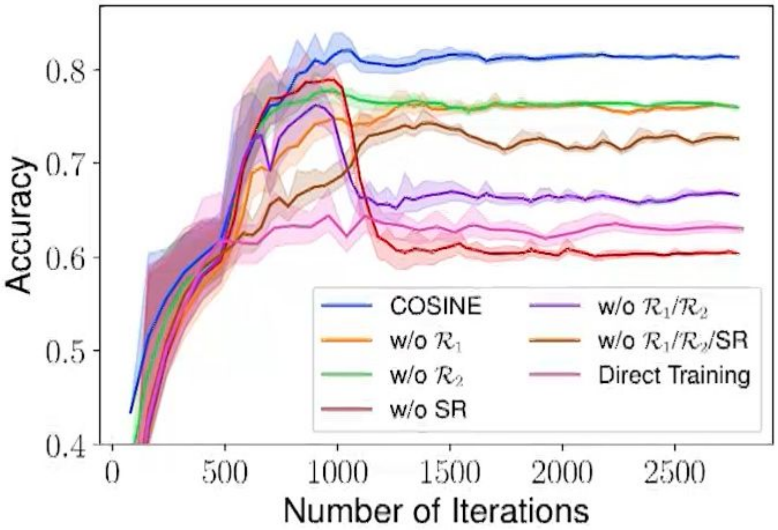
Evaluation on Various Benchmarks

Dataset	Agnews	IMDB	Yelp	TREC	MIT-R	ChemProt	WiC
Task	Text Classification				Slot Filling	Relation extraction	Word Sense Disambiguation
Fully supervised	92.54	94.26	97.27	96.68	88.51	79.66	70.53
w/ Weak Labels	82.25	74.89	74.89	62.25	70.95	44.80	59.36
Previous SOTA	86.28	88.04	92.05	80.20	74.41	53.48	64.88
Ours	**87.52**	**90.54**	**95.97**	**82.59**	**76.61**	**54.36**	**67.71**

- Our framework achieves better performance on all datasets compared w/ SOTA weakly-supervised baselines and fine-tuning baselines.
- Our performance is much closer to the fully-supervised result.

Ablation Study

Dataset	Agnews	IMDB	Yelp	TREC	MIT-R
Ours	**87.52**	**90.54**	**95.97**	**82.59**	**76.61**
w/o R_1	86.04	88.32	94.64	78.28	70.95
w/o R_2	85.91	89.32	93.96	77.11	74.11
w/o SR	86.72	87.10	93.08	79.77	74.29
w/o R_1/R_2	86.33	84.44	92.34	76.95	73.67
w/o Soft Label	86.07	89.72	93.73	71.91	73.05



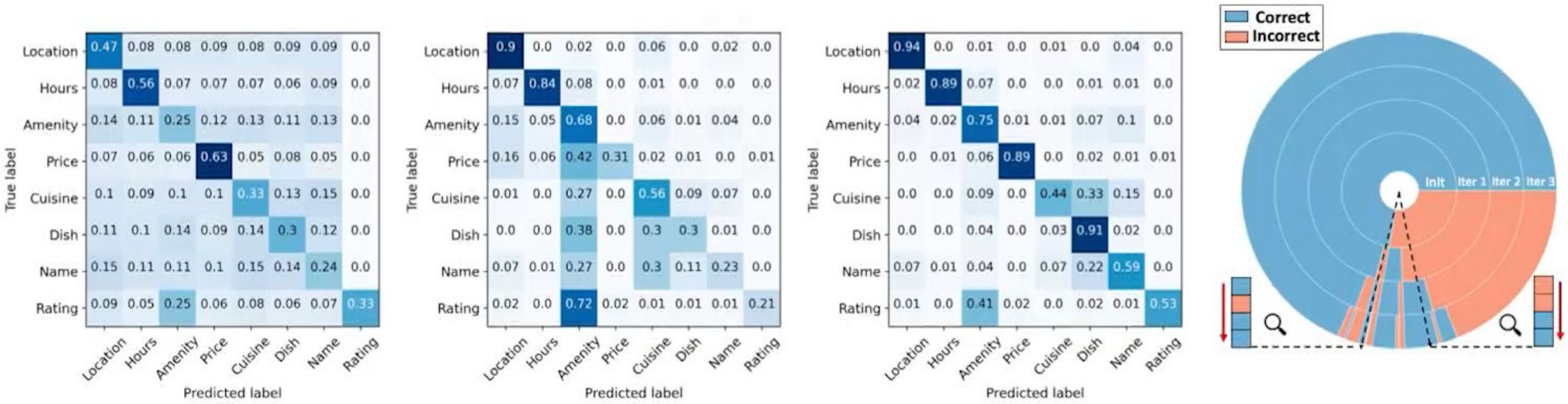
- All components in our framework are useful for down-stream tasks.
- With contrastive regularization and sample reweighting, the self-training becomes more stable

Extension to Semi-supervised Learning

Model	Dev	Test	#Params
Human Baseline	80.0		–
BERT (Devlin et al., 2019)	–	69.6	335M
RoBERTa (Liu et al., 2019)	70.5	69.9	356M
T5 (Raffel et al., 2019)	–	76.9	11,000M
Semi-Supervised Learning			
SenseBERT (Levine et al., 2020)	–	72.1	370M
RoBERTa-WL† (Liu et al., 2019)	72.3	70.2	125M
w/ MT† (Tarvainen and Valpola, 2017)	73.5	70.9	125M
w/ VAT† (Miyato et al., 2018)	74.2	71.2	125M
w/ COSINE†	**76.0**	**73.2**	125M
Transductive Learning			
Snorkel† (Ratner et al., 2020)	80.5	–	1M
RoBERTa-WL† (Liu et al., 2019)	81.3	76.8	125M
w/ MT† (Tarvainen and Valpola, 2017)	82.1	77.1	125M
w/ VAT† (Miyato et al., 2018)	84.9	79.5	125M
w/ COSINE†	**89.5**	**85.3**	125M

- **Semi-Supervised** Learning: augment the original training data with sentence pairs extracted from lexical KB (wordnet)
- **Transductive** Setting: Have access to train data (w/o labels) and augment them to training set.
- Our framework can achieve best performance compared with other semi-supervised learning and transductive learning baselines.

Case Study



From left to right: (1) visualization of Exact Match, (2) results after the initialization step, (3) results after contrastive self-training, (4) wrong-label correction after self-training.

Our framework can gradually correct the wrong annotated examples.

Self-Training with Weak Supervision

https://www.aclweb.org/anthology/2021.naacl-main.66.pdf

Giannis Karamanolakis, Ahmed Hassan Awadallah, Subhabrata Mukherjee, Guoqing Zheng

Some slides copied from authors' presentation at NAACL-2021

Dominant Supervised Learning Paradigm: A Labeled Data Bottleneck

Task Specification
(e.g., document-level, binary sentiment classification)

"labeled data bottleneck"

Standard Benchmarks
- Fixed task specifications
- Large-scale labeled data

Real-World Applications
- Dynamic task specifications
- Limited or no labeled data

Weak Supervision Via Domain-Specific Rules

- Rules: heuristic labeling functions written by **domain experts**
- Rules are used to automatically annotate **unlabeled** data

---

Example: regular expression patterns

Spam classification

```
def regex_check_out(x):
    return SPAM if re.search("check.*out", x) else ABSTAIN
```

Question type classification

```
def numeric_question(x):
    return NUMERIC if x.startswith("when") else ABSTAIN
```

Example: heuristic functions based on lexicons / models / knowledge bases

Sentiment classification

```
def sentiment_lexicon_score(x, sentiwordnet):
    if sentiwordnet(x) > 0.8:
        return POSITIVE
    elif sentiwordnet(x) < 0.2:
        return NEGATIVE
    else:
        ABSTAIN
```

Challenges in Learning with Weak Rules

(1) Noise

$\text{rule}(x_i) \rightarrow \text{SPAM}$ ✘

True label: HAM

---

(2) Coverage

$\text{rule}(x_i) \rightarrow \text{ABSTAIN}$

---

(3) Conflicts

x_i:
- $\text{rule1} \rightarrow \text{SPAM}$
- $\text{rule2} \rightarrow \text{HAM}$
- $\text{rule3} \rightarrow \text{SPAM}$
- $\text{rule4} \rightarrow \text{HAM}$

➡ ???

Our ASTRA Framework for Weak Supervision

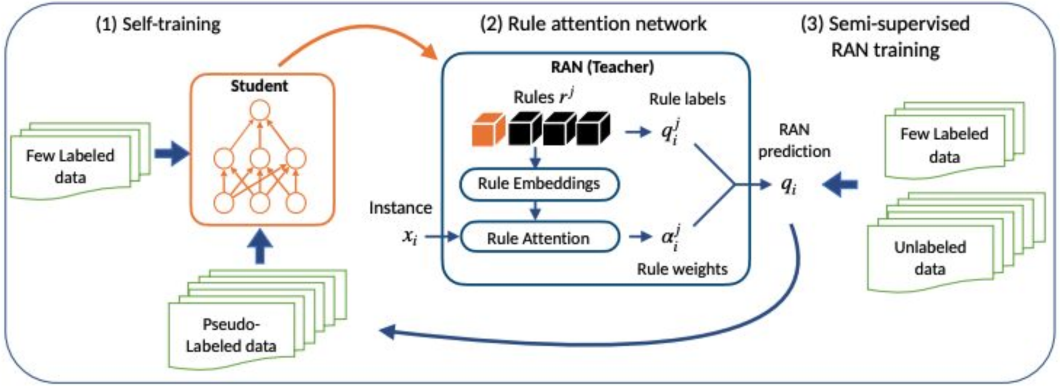


Figure 2: Our ASTRA framework for self-training with weak supervision.

Our Contributions:

1. Present an **iterative self-training** mechanism for training deep neural networks (Student) with weak supervision
2. Present a **rule attention network** (RAN Teacher) for aggregating multiple weak sources with instance-specific weights and construct an **SSL objective**
3. Show the effectiveness of ASTRA on **six benchmarks** for text classification

Limitation of Previous Methods for Weak Supervision

- Previous work **ignore unlabeled instances** that are **not** covered by rules

[Ratner et al., 2017; Bach et al., 2019; Awasthi et al., 2020]

- Expert-defined rules are usually **sparse**:

6 real-world datasets
45 rules / dataset

- **just 33%** of instances covered by > 1 **rule**
- **40%** of instances are **not** covered by **any rules**

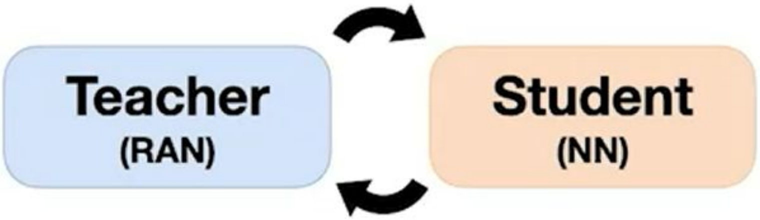
Filtered-out

Don't throw them away!

- We leverage **all unlabeled instances** for weak supervision via **self-training**

ASTRA: Weakly-Supervised Self-Training

1. Student
2. Teacher

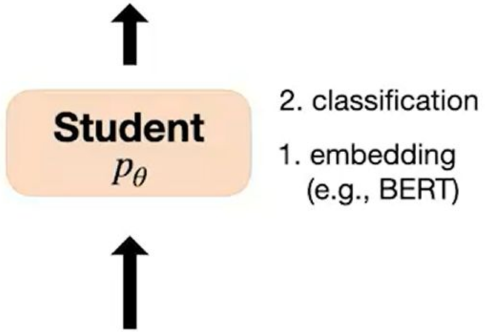


Student: An Embedding-Based Neural Network

- Represents input x using contextualized representations

Example: Question Type Classification (in TREC)

Question type $y =$ "NUMERIC"

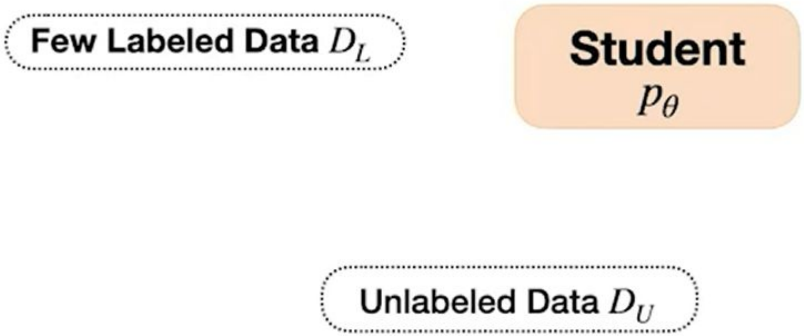


input x: "What is the percentage of water content in the human body?"

Student: An Embedding-Based Neural Network

- Represents input x using contextualized representations
- Large-scale labeled data is expensive to obtain

Self-Training Paradigm



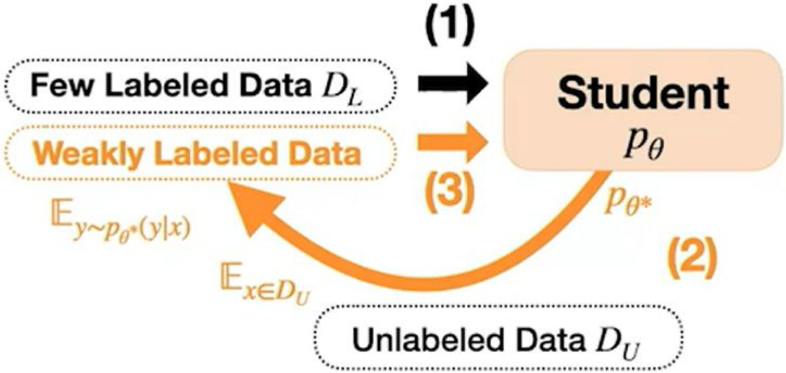
Student: An Embedding-Based Neural Network

- Represents input x using contextualized representations
- Large-scale labeled data is expensive to obtain

<u>Self-Training Paradigm</u>

$$\min_{\theta} \mathbb{E}_{x,y \in D_L} - \log \, p_\theta(y \mid x) \;+\; \lambda \mathbb{E}_{x \in D_U} \mathbb{E}_{y \sim p_{\theta^*}(y|x)} - \log \, p_\theta(y \mid x)$$

(-) Prone to error propagation



Student: An Embedding-Based Neural Network

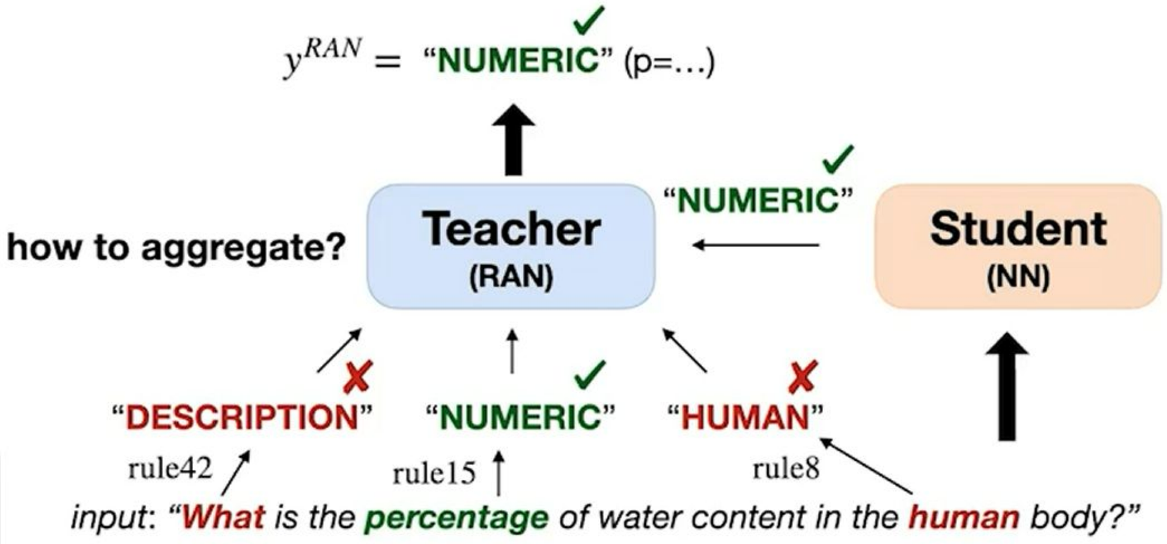
- Represents input x using contextualized representations
- Large-scale labeled data is expensive to obtain
- We train Student using Teacher's labels

Weakly-Supervised Self-Training

$$\min_{\theta} \ \mathbb{E}_{x,y \in D_L} - \log \ p_\theta(y \mid x) \ + \ \lambda \mathbb{E}_{x \in D_U} \underset{\mathbb{E}_{y \sim q_{\phi^*}(y|x)}}{\cancel{\mathbb{E}_{y \sim p_{\theta^*}(y|x)}}} - \log \ p_\theta(y \mid x)$$

Teacher: Rule Attention Network (RAN)

- RAN aggregates weak labels predicted by **rules** and **Student**
  - **Heuristic rules** cover only a subset of the data
  - **Student** covers more data via contextualized embeddings



Teacher: Rule Attention Network (RAN)

- RAN aggregates weak labels predicted by **rules** and **Student**
- RAN learns to predict **instance-specific** weights using **rule attention**
- RAN does **not** require rule supervision: we employ a **SSL objective**

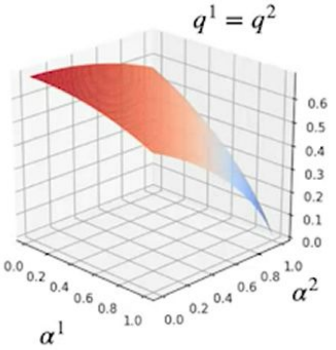
RAN label

$$q_i = \frac{1}{Z} \sum_{j \in R} a_i^j q_i^j + (1 - a_i^j) u$$

Semi-Supervised Training Objective: $\mathscr{L}^{RAN} = - \sum_{(x_i, y_i) \in D_L} y_i \log q_i - \sum_{x_i \in D_U} q_i \log q_i .$

Cross-Entropy (labeled data)

Min-Entropy (unlabeled data)

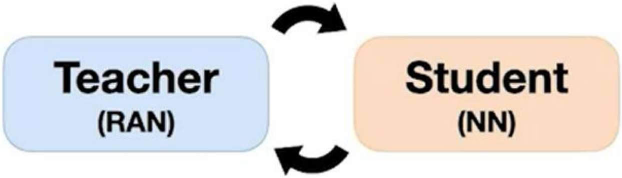


high weights $a^j = 1$ for rules j that agree in predictions q^j

more details in our paper!

Summary of our ASTRA Framework

1. Train **Student** using few labeled data
2. Iterate:
    1. Train **RAN Teacher** to aggregate weak rules and Student
    2. Train **Student** using Teacher's labels



Access to rules during test time?

- YES -> use **Teacher** (Student + Rules)
- NO -> use **Student**

Experiments: Learning with Weak Supervision

Benchmark	# Rules	Rule Coverage
TREC (question classification)	68	46%
SMS (spam classification)	73	9%
YouTube (spam classification)	10	48%
CENSUS (income classification)	83	94%
MIT-R (slot filling)	15	1%
Spouse (relation classification)	9	8%

- **Rule types:** keywords, regular expressions, lexicons, knowledge bases
- Rules are **sparse**:
  - 66% of the examples are covered by **fewer than 2 rules**
  - 40% of the examples are **not covered** by any rule

Results Summary Across 6 Benchmarks

Method	Learning to Weight Rules	Learning to Weight Instances	Unlabeled (no rules)	Average Accuracy
PosteriorReg (Hu et al., 2016)	✓	-	-	82.6
Snorkel (Ratner et al., 2017)	✓	-	-	82.9
L2R (Ren et al., 2018a)	-	✓	-	82.8
Standard self-training	-	-	✓	83.5
ImplyLoss (Awasthi et al., 2020)	✓	✓	-	85.2
ASTRA	✓	✓	✓	**88.0** (+3.3%)

- **Self-training** outperforms weak supervision approaches
- **ASTRA** outperforms all previous approaches:
  - (+) Learns **instance-specific** rule weights
  - (+) Leverages **all unlabeled data**
  - (+) Does **not** require rule supervision ("rule exemplars" in Awasthi et al., 2020)

Multi-Style Transfer with Discriminative Feedback on Disjoint Corpus

https://www.aclweb.org/anthology/2021.naacl-main.275.pdf

Navita Goyal, Anadhavelu Natarajan, Abhilasha Sancheti, Balaji Vasan Srinivasan

Some slides copied from authors' presentation at NAACL-2021

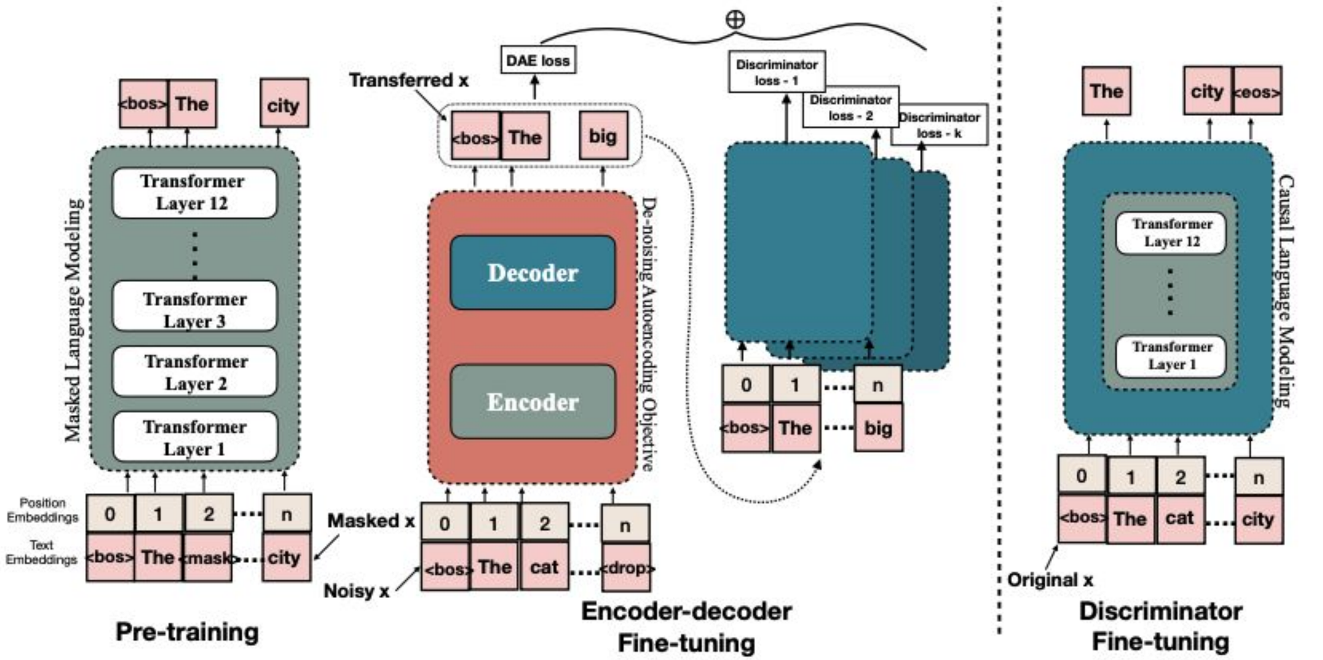


Figure 1: Model Architecture - Left: Generative pre-training using MLM objective, and Fine-tuning encoder-decoder LM with multiple discriminative losses and Right: Discriminator fine-tuning with language modeling (next token prediction) objective. Color for model blocks represents the pre-trained model used for initialization prior to fine-tuning.