

# SPEAR

SemisuPervisEd dAta pROgramming

Presented

By:-

Aziz Shameem(20d070020)

Vishal Jorwal(200070089)

Kalp Vyas(200070030)

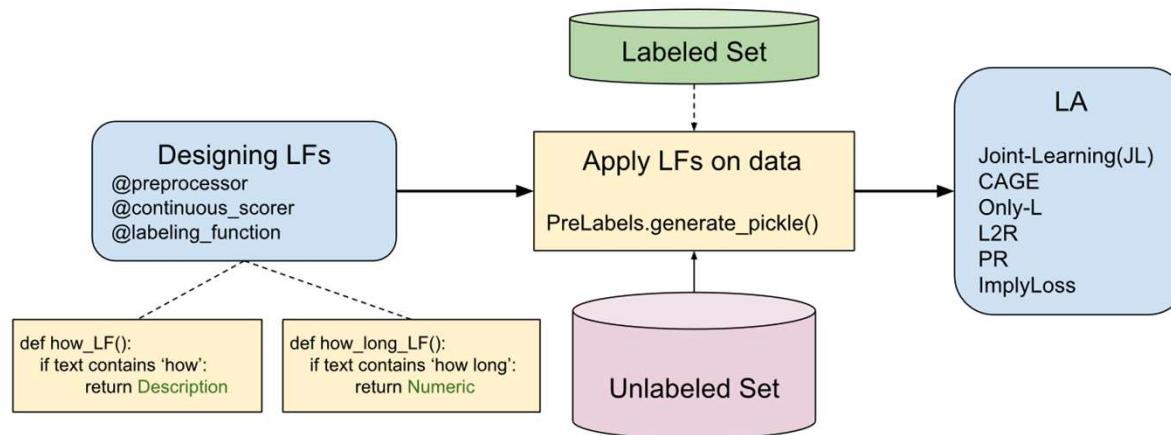
# **SPEAR : Semi-supervised Data Programming in Python**

In this presentation, we will introduce SPEAR, which stands for Semi-supervised Data Programming in Python. We discuss what SPEAR is, how it uses semi-supervised learning and data programming to create accurate models with limited labeled data, and how it can be used in Python.

# SEMI-SUPERVISED DATA PROGRAMMING FOR DATA EFFICIENT MACHINE LEARNING

## Pipeline

- Design Labeling functions(LFs)
- Generate pickle file containing labels by passing raw data to LFs
- Use one of the Label Aggregators(LA) to get final labels



# Automatic LFs generation

Automatic LF (Labelling Function) generation is the process of transforming natural language text into a formal representation of its meaning that can be processed by a computer. This formal representation is often in the form of a logical expression or a semantic representation.

Transforming natural language text into a formal representation, allows computers to reason about the meaning of the text and perform various tasks based on that understanding.

# How it works?

Automatic rule induction (ARI) approaches to circumvent this problem by automatically inducing rules from the data. ARI methods use a small labeled set to extract rules either by using decision tree approaches or weights of a classifier. The above approaches initially find a list of patterns and filter them to find the top k patterns. These patterns are transformed into rules that yield noisy labels. The rules are then fed to the unsupervised or semi-supervised aggregation approaches to aggregate noisy labels

# Joint Learning in SPEAR

The loss function considers six different types of losses. These include the cross entropy on the labeled set, an entropy SSL term on the unlabeled dataset, a cross entropy term to ensure consistency between the feature model and the LF model, the LF graphical model terms on the labeled and unlabeled datasets, a KL divergence again for consistency between the two models, and finally a regularizer.

$$L_{ss}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$

# Joint Learning in SPEAR

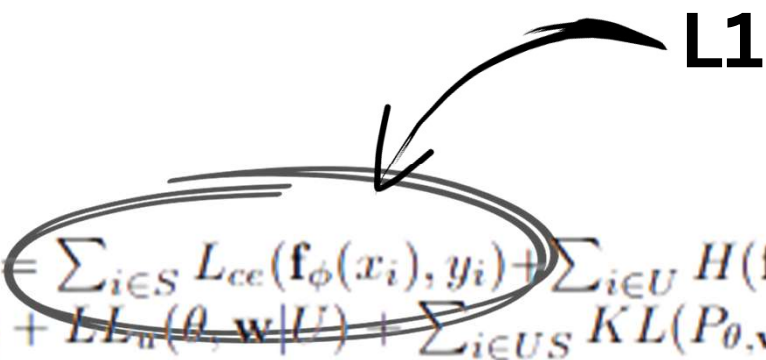
SPEAR has a feature-based classification model  $f_{\theta}(x)$  which takes the features as input and predicts the class label.

The output of this model is  $P_{\theta}(y|x)$ , i.e., the probability of the classes given the input features.

$$P_{\theta}(\mathbf{l}_i, y) = \frac{1}{Z_{\theta}} \prod_{j=1}^m \psi_{\theta}(l_{ij}, y)$$
$$\psi_{\theta}(l_{ij}, y) = \begin{cases} \exp(\theta_{jy}) & \text{if } l_{ij} \neq 0 \\ 1 & \text{otherwise.} \end{cases}$$

# Joint Learning in SPEAR

**First Component (L1):** The first component (L1) of the loss is the standard crossentropy loss on the labeled dataset L for the model.



The diagram shows the equation for the loss function  $L_{ss}(\theta, \phi, \mathbf{w})$ . The first term,  $\sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i)$ , is circled with a double-lined oval. An arrow labeled "L1" points from the text above to this circled term.


$$L_{ss}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$



# Joint Learning in SPEAR

**Second Component (L2):** The second component L2 is the semi-supervised loss on the unlabelled data U. In our framework, we can use any unsupervised loss function. It acts as a form of semi-supervision by trying to increase the confidence of the predic

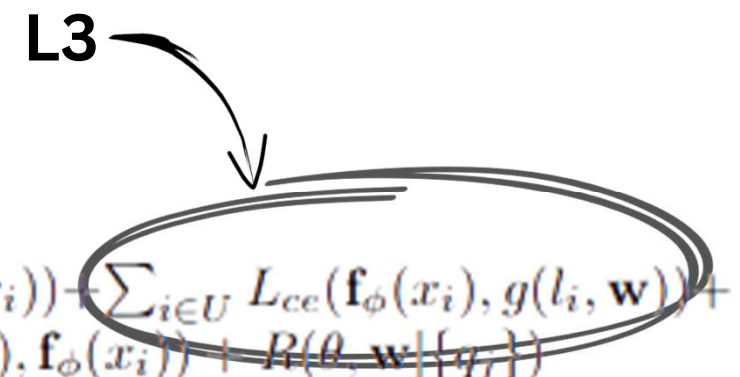
L2



$$L_{ss}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$

# Joint Learning in SPEAR

**Third Component (L3):** The third component is the cross-entropy of the classification model using the hypothesised labels from CAGE on U.

$$L_{ss}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$
A diagram consisting of a curved arrow pointing from the label 'L3' to a double-lined oval that encircles the term  $\sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w}))$  in the equation above.

# Joint Learning in SPEAR

**Fourth Component (L4):** The fourth component  $LL_s(\theta|L)$  is the (supervised) negative log likelihood loss on the labelled dataset  $L$ .

$$L_{SS}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$



**L4**

# Joint Learning in SPEAR

**Fifth Component (L5):** The fifth component  $LL_u(\theta|U)$  is the negative log likelihood loss for the unlabelled dataset  $U$ . Since the true label information is not available, the probabilities need to be summed over  $y$ .

$$L_{ss}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + \underbrace{LL_u(\theta, \mathbf{w}|U)}_{\text{L5}} + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$

# Joint Learning in SPEAR

**Sixth Component (L6):** The sixth component is the Kullback-Leibler (KL) divergence between the predictions of both the models, viz., feature-based model  $f_\phi$  and the LF-based graphical model  $P_\theta$  summed over every example  $x_i$ . Through this term, we try and make the models agree in their predictions over the union of the labelled and unlabelled datasets.

$$L_{ss}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_\phi(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_\phi(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_\phi(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_\phi(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$

**L6**



# Joint Learning in SPEAR

**Quality Guides (QG):** As the last component in our objective, we use quality guides  $R(\theta|\{q_j\})$  on LFs, to stabilise the unsupervised likelihood training while using labelling functions.

$$L_{ss}(\theta, \phi, \mathbf{w}) = \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\})$$

QG



# SNUBA

Snuba is a three-step approach that

- Generates candidate rules using a labeled set
- Filters heuristics based on precision on the labeled set and coverage on the unlabeled set
- Finds uncovered points or abstained points in the labeled set.

# Classifier weights

A linear model classifier  $C$  on the small labeled set. Suppose for  $N$  instances in our dataset, each instance  $x_i$  is denoted by its feature matrix  $X_i$  of size  $K$ . The classifier model  $C(x_i) = \sigma(W X_i)$  where  $W$  is a  $K \times N$  is a weight matrix and  $\sigma$  represents an element-wise sigmoid function. Then, their approach finds  $P$  features corresponding to the largest weights in  $W$ .



# WISDOM

WISDOM is a system for reliable aggregation of automatically generated labeling functions. It has two components: SNUBA for generating accurate and diverse labeling functions, and CAGE for re-weighting them based on their reliability. The weights are optimized using a bi-level optimization algorithm that jointly learns the feature-based classifier and labeling function aggregator's parameters.

# Bi-Level

## objective:

WISDOM is a method for jointly learning the weights of labeling functions and the parameters of a feature classifier, with the objective of minimizing a cross-entropy loss on a validation set. This is achieved through a bi-level optimization problem, where the weights are updated through alternating gradient updates. WISDOM allows for filtering of labeling functions based on the feature model, and aims to learn weights that result in minimum validation loss on the jointly trained feature model.

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w}} \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathcal{L}_{ce}(f_{\phi^*}(x_i), y_i)$$

$$\text{where } \phi^*, \theta^* = \operatorname{argmin}_{\phi, \theta} \mathcal{L}_{ss}(\theta, \phi, \mathbf{w})$$

# Our Contributions...

We have worked on the integration of code performing Auto Induction of Labeling Functions.

This is built in the form of objects in the spear package and can be invoked as and when required.

We have also integrated packages with implementation of the WISDOM workflow, for re-weighting of generated LFs

# What is SPEAR?

SPEAR stands for Semi-supervised Data Programming in Python. It is a machine learning framework that combines the power of semi-supervised learning and data programming to build models with limited labeled data. SPEAR can be used to extract information from large datasets and automate the process of building accurate models. With SPEAR, users can train models with minimal supervision and significantly reduce the time and cost associated with manual labeling. Key features of SPEAR include its ability to handle noisy and incomplete data, its ability to improve model accuracy over time, and its easy integration with Python. Overall, SPEAR is a powerful tool for data scientists and machine learning practitioners looking to build accurate models with limited labeled data.

# Key features of SPEAR

- include:**
- **Semi-supervised learning:** SPEAR uses a combination of supervised and unsupervised learning techniques to build accurate models with limited labeled data.
  - **Data programming:** SPEAR enables users to label large amounts of data quickly and accurately using weak supervision techniques.
  - **Noise handling:** SPEAR is designed to handle noisy and incomplete data, which can improve model accuracy over time.
  - **Model improvement:** SPEAR allows users to continuously improve their models by iteratively labeling more data and retraining the model.
  - **Python integration:** SPEAR is easy to integrate with Python, which makes it accessible to a wide range of users.

***Overall, these key features make SPEAR a powerful tool for building accurate models with limited labeled data.***

# Why use

# SPEAR?

- **Limited labeled data:** If you have limited labeled data, SPEAR can help you build accurate models by leveraging semi-supervised learning and data programming techniques. This can save you time and money by reducing the need for manual labeling.
- **Noisy data:** If your data is noisy or incomplete, SPEAR can help you handle these challenges and improve model accuracy over time.
- **Continuous improvement:** If you want to continuously improve your models over time, SPEAR allows you to iteratively label more data and retrain the model, which can lead to better performance.
- **Python integration:** If you use Python for your machine learning workflows, SPEAR is easy to integrate and use within your existing codebase.
- **Wide range of applications:** SPEAR can be used in a wide range of applications, such as natural language processing, computer vision, and predictive analytics.

Overall, SPEAR can help you build more accurate models with limited labeled data and handle challenges such as noisy and incomplete data. It is also easy to use with Python and has a wide range of applications, which makes it a powerful tool for data scientists and machine learning practitioners.

# Overview of SPEAR

## workflow

- **Data ingestion:** The first step is to ingest your data into the SPEAR framework. This can include both labeled and unlabeled data.
- **Labeling functions:** Next, you create labeling functions that use weak supervision techniques to label your data. This can include rules, heuristics, or models that assign labels to your data.
- **Label model:** The labeling functions are combined to create a label model, which is used to label the remaining unlabeled data. This process is iterative, with the label model improving over time as more data is labeled.
- **Training and evaluation:** Once the data is labeled, you can train and evaluate your machine learning model. The labeled data is used as the training set, while the unlabeled data is used for evaluation.
- **Model improvement:** Finally, you can use the model to make predictions on new data, and iteratively label more data to further improve the model over time.

$$\begin{aligned} L_{ss}(\theta, \phi, \mathbf{w}) = & \sum_{i \in S} L_{ce}(\mathbf{f}_{\phi}(x_i), y_i) + \sum_{i \in U} H(\mathbf{f}_{\phi}(x_i)) + \sum_{i \in U} L_{ce}(\mathbf{f}_{\phi}(x_i), g(l_i, \mathbf{w})) + \\ & LL_s(\theta, \mathbf{w}|S) + LL_u(\theta, \mathbf{w}|U) + \sum_{i \in US} KL(P_{\theta, \mathbf{w}}(l_i), \mathbf{f}_{\phi}(x_i)) + R(\theta, \mathbf{w}|\{q_j\}) \end{aligned}$$