

# Learning from One Example using Logical Induction

Reference - *One-shot Information Extraction from Document Images using Neuro-Deductive Program Synthesis*

Learn a *program* to extract specific entities from document images that follow a template, from *one* sample. E.g. correspondence no., 186FDBC1802472 here:

```
TO
DNB NOR BANK ASA
TRADE FINANCE/GUARANTEE
DEPT. NO.0021
OSLO NORWAY
NO
```

```
Please Quote in all correspondence
186FDBC1802472
```

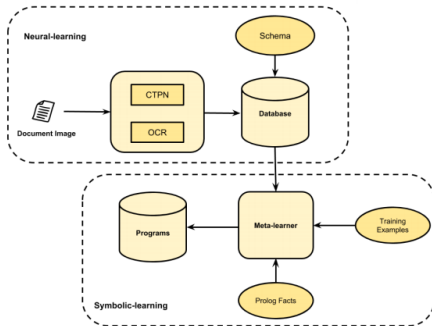
DL to extract to a database; then ILP.

A correct program for this example:

```
corr(A,B) :-
    has_keyword('Please', A,C),
    has_line_below(C,B).
```

But ILP will also find this incorrect one:

```
corr(A,B) :-
    has_keyword('Please', A,C),
    left_of('Please', C,D),
    right_of('ASA', D,C),
    has_line_below_word(C,B).
```

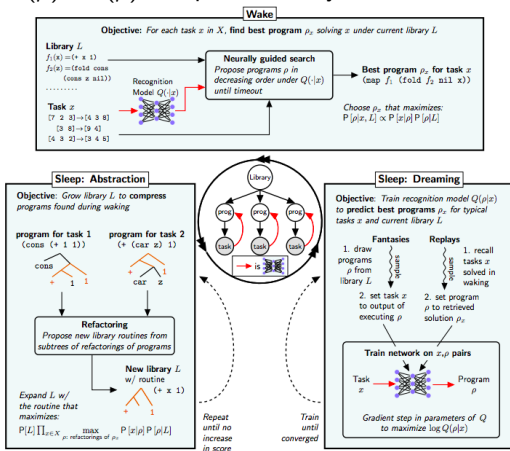


Key idea: perturb *all* entities in the DB to create a 'noisy clone' and feed both examples to ILP, which then learns the correct program.

# Logical Meta-learning

Reference - *DreamCoder: Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning*

For input-output task  $x$  learn a program  $\rho_x$  using library of primitive operations  $L$ .  
 $P(\rho) \sim I(\rho)$ . 'Improve' library and learn faster as more tasks are solved.



**Wake:**

$$\rho_x = \operatorname{argmax}_{\rho} P(\rho|x, L) \propto P(x|\rho)P(\rho|L) \text{ for task } x.$$

**Sleep:**  $L =$

$$\operatorname{argmax}_L \prod_x \max_{\rho=r(\rho_x)} P(x|\rho)P(\rho|L)$$

**Dream:** update  $Q_{\theta}(\rho|x)$ :

$$\operatorname{argmin}_{\theta} \|P(\rho|x, L) - Q_{\theta}(\rho|x)\|$$

Key: efficient algorithm to search over (potentially exponential) refactorings  $r(\rho_x)$ .