Data programming for PubAnnotation via Snorkel

JUAN M. BANDA

STANFORD CENTER FOR BIOMEDICAL INFORMATICS RESEARCH, STANFORD, CA

Biomedical Linked Data Hackathon – BLAH3

Tokyo, Japan

January 16th

About Me

- Research Scientist at Nigam Shah's lab at Stanford Center for Biomedical Informatics Research

- Hackathon enthusiast (attended 50+, 20 in 2016)

- Research Interests:
  - Electronic Phenotyping
  - Machine Learning:
    - Predictive Modeling
    - Noisy Labeling
  - Pharmacovigilance
  - Semantic Web
  - Image Retrieval/Classification (past)
  - CBIR systems (past)

BLAH3 Project: Data programming for PubAnnotation via Snorkel

Why is this needed?

As the volumes of medical data being generated continues to increase with wide adoption of electronic health records, cheaper sequencing technology, etc. and their subsequent use in medical publications increases, we are faced with the challenge of producing high-quality labeled datasets for research

Data programming, have been proposed to rely instead on 'lower quality' auto generated noisy data to train weakly supervised models and then use generative models to denoise the noisy training data

What is Snorkel? (1)

- Developed as a ‘light’ version of Deep Dive by Chris Ré’s Hazy Research group at Stanford University

- Snorkel is framework for developing **structured information extraction applications** for domains in which large labeled training sets are not available or easy to obtain, using the *data programming* paradigm

- In **Data programming** approach to developing a machine learning system, the developer focuses on writing a set of *labeling functions*, which create a large but noisy training set.

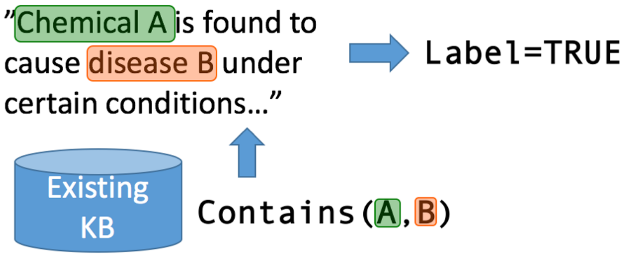
- Snorkel then learns a generative model of this noise—learning, essentially, which labeling functions are more accurate than others—and uses this to train a discriminative classifier.

What is Snorkel? (2)

- Labeling functions:

```
def lf1(x):
  cid = (x.chemical_id, x.disease_id)
  return 1 if cid in KB else 0
```

```
def lf2(x):
  m = re.search(r'.*cause.*', x.between)
  return 1 if m else 0
```

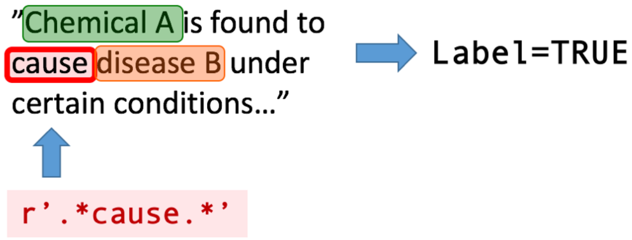


"Chemical A is found to cause disease B under certain conditions…"

Label=TRUE

Existing KB

Contains(A,B)

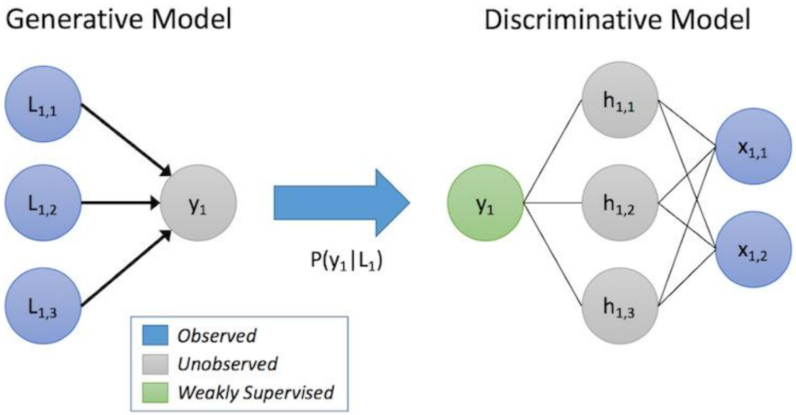


"Chemical A is found to cause disease B under certain conditions…"

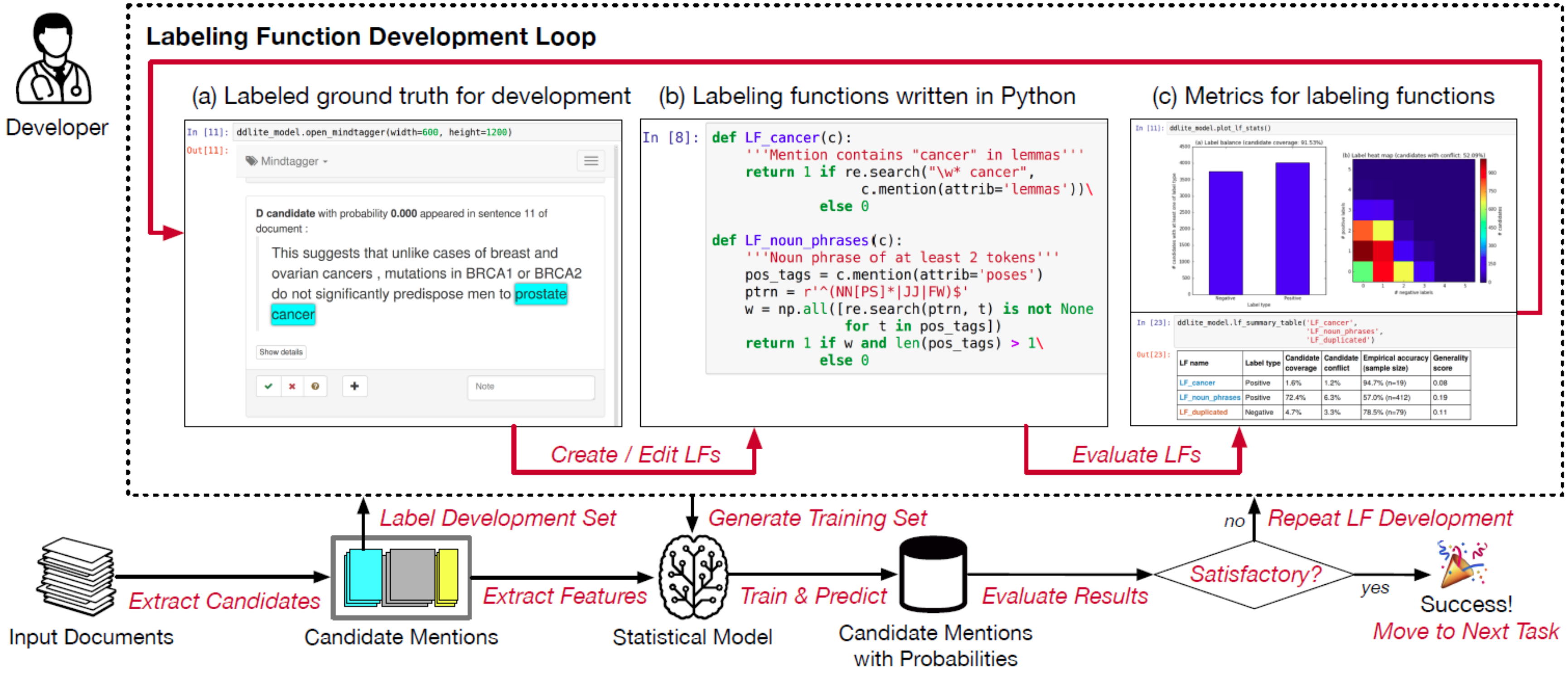
Label=TRUE

r'.*cause.*'

- Modeling steps:



Snorkle annotation procedure



```
In [11]: ddlite_model.open_mindtagger(width=600, height=1200)
Out[11]:
```

D candidate with probability 0.000 appeared in sentence 11 of document :

This suggests that unlike cases of breast and ovarian cancers , mutations in BRCA1 or BRCA2 do not significantly predispose men to prostate cancer

```
In [8]: def LF_cancer(c):
            '''Mention contains "cancer" in lemmas'''
            return 1 if re.search("\w* cancer",
                                  c.mention(attrib='lemmas'))\
                      else 0

        def LF_noun_phrases(c):
            '''Noun phrase of at least 2 tokens'''
            pos_tags = c.mention(attrib='poses')
            ptrn = r'^(NN[PS]*|JJ|FW)$'
            w = np.all([re.search(ptrn, t) is not None
                              for t in pos_tags])
            return 1 if w and len(pos_tags) > 1\
                      else 0
```

```
In [11]: ddlite_model.plot_lf_stats()
```

```
In [23]: ddlite_model.lf_summary_table('LF_cancer',
                                       'LF_noun_phrases',
                                       'LF_duplicated')
Out[23]:
```

LF name	Label type	Candidate coverage	Candidate conflict	Empirical accuracy (sample size)	Generality score
LF_cancer	Positive	1.6%	1.2%	94.7% (n=19)	0.08
LF_noun_phrases	Positive	72.4%	6.3%	57.0% (n=412)	0.19
LF_duplicated	Negative	4.7%	3.3%	78.5% (n=79)	0.11

From: Ehrenberg, H.R., Shin, J., Ratner, A.J., Fries, J.A., R´e, C.: Data programming with ddlite: Putting hum different part of the loop. In: Proceedings of the Workshop on Human-In-the-Loop Data Analytics. HILDA ’16, pp. 13–1136. ACM, New York, NY, USA (2016). http://doi.acm.org/10.1145/2939502.2939515

BLAH3 Goals

- Build extensions for Snorkle to produce all annotations in PubAnnotation format. This will allow us to take advantages of all of PubAnnotation’s capabilities and cross functionality with other tools like Brat and Tagtog

- Build extensions for Snorkle to extract annotations from PubAnnotation for validation/testing of the labeling functions

Maybe:

- Find a collaborator that is providing an annotated corpus to BLAH3 and wants to build labeling functions to validate the power and usability of Snorkle

Reach goals:

- Build a bare-bones web-interface to interact with a running instance of Snorkle. Essentially build a REST-like tool that allows rules to be constructed and tested via a browser with minimal interaction with Python

Acknowledgements

DBCLS/BLAH organizers

- Thanks for the travel support and the opportunity to work on this project

Stanford

- Nigam Shah’s lab
- Hazy Research Lab

QUESTIONS?

THANK YOU