

Training Classifiers with Natural Language Explanations

Braden Hancock, Paroma Varma and so on

ACL 2018
Stanford

Contents

- **Introduction**
- **Methods**
- **Experiments**
- **Conclusions and Future Work**

Contents

- **Introduction**
- Methods
- Experiments
- Conclusions and Future Work

Introduction

- Annotator provides a natural language explanation for each labeling decision.
- A semantic parser converts explanations into programmatic labeling functions
- Generating noisy labels for an arbitrary amount of unlabeled data.
- Training classifiers with comparable F1 scores from 5-100 faster by providing explanations instead of just labels.

Introduction

Example

Both cohorts showed signs of optic nerve toxicity due to ethambutol.

Label

Does this chemical cause this disease?



Explanation

Why do you think so?

Because the words "due to" occur between the chemical and the disease.


Labeling Function

```
def lf(x):  
    return (1 if "due to" in between(x.chemical, x.disease)  
            else 0)
```

Contents


- Introduction
- **Methods**
- Experiments
- Conclusions and Future Work

Methods

Unlabeled Examples + Explanations	Labeling Functions	Filters	Label Matrix																																										
<p>Label whether person 1 is married to person 2</p> <p>X₁ Tom Brady and his wife Gisele Bündchen were spotted in New York City on Monday amid rumors of Brady's alleged role in Deflategate.</p> <p>True, because the words "his wife" are right before person 2.</p>	<pre>def LF_1a(x): return (1 if "his wife" in left(x.person2, dist==1) else 0)</pre> <pre>def LF_1b(x): return (1 if "his wife" in right(x.person2) else 0)</pre>	<p>Correct</p> <p>Semantic Filter (inconsistent)</p>	<table><tr><td></td><td>X₁</td><td>X₂</td><td>X₃</td><td>X₄</td><td>...</td></tr><tr><td>LF_{1a}</td><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>LF_{2b}</td><td></td><td>-1</td><td></td><td></td><td></td></tr><tr><td>LF_{3a}</td><td>-1</td><td></td><td>-1</td><td></td><td></td></tr><tr><td>LF_{4c}</td><td>1</td><td></td><td>1</td><td>1</td><td></td></tr><tr><td>⋮</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>\tilde{y}</td><td>+</td><td>-</td><td>-</td><td>+</td><td>...</td></tr></table>		X ₁	X ₂	X ₃	X ₄	...	LF _{1a}	1					LF _{2b}		-1				LF _{3a}	-1		-1			LF _{4c}	1		1	1		⋮						\tilde{y}	+	-	-	+	...
	X ₁	X ₂	X ₃	X ₄	...																																								
LF _{1a}	1																																												
LF _{2b}		-1																																											
LF _{3a}	-1		-1																																										
LF _{4c}	1		1	1																																									
⋮																																													
\tilde{y}	+	-	-	+	...																																								
<p>X₂ None of us knows what happened at Kane's home Aug. 2, but it is telling that the NHL has not suspended Kane.</p> <p>False, because person 1 and person 2 in the sentence are identical.</p>	<pre>def LF_2a(x): return (-1 if x.person1 in x.sentence and x.person2 in x.sentence else 0)</pre> <pre>def LF_2b(x): return (-1 if x.person1 == x.person2) else 0)</pre>	<p>Pragmatic Filter (always true)</p> <p>Correct</p>																																											
<p>X₃ Dr. Michael Richards and real estate and insurance businessman Gary Kirke did not attend the event.</p> <p>False, because the last word of person 1 is different than the last word of person 2.</p>	<pre>def LF_3a(x): return (-1 if x.person1.tokens[-1] != x.person2.tokens[-1] else 0)</pre> <pre>def LF_3b(x): return (-1 if not (x.person1.tokens[-1] == x.person2.tokens[-1]) else 0)</pre>	<p>Correct</p> <p>Pragmatic Filter (duplicate of LF_{3a})</p>	<p>Noisy Labels</p> <p>(X₁, \tilde{y}_1) (X₂, \tilde{y}_2) (X₃, \tilde{y}_3) (X₄, \tilde{y}_4)</p> <p>Classifier</p> 																																										


The semantic parser converts natural language explanations into a set of logical forms representing labeling functions(LFs)

Methods

Unlabeled Examples + Explanations	Labeling Functions	Filters	Label Matrix																																										
<p>Label whether person 1 is married to person 2</p> <p>X₁ Tom Brady and his wife Gisele Bündchen were spotted in New York City on Monday amid rumors of Brady's alleged role in Deflategate.</p> <p>True, because the words "his wife" are right before person 2.</p>	<pre>def LF_1a(x): return (1 if "his wife" in left(x.person2, dist==1) else 0)</pre> <pre>def LF_1b(x): return (1 if "his wife" in right(x.person2) else 0)</pre>	<p>Correct</p> <p>Semantic Filter (inconsistent)</p>	<table><tr><td></td><td>X₁</td><td>X₂</td><td>X₃</td><td>X₄</td><td>...</td></tr><tr><td>LF_{1a}</td><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>LF_{2b}</td><td></td><td>-1</td><td></td><td></td><td></td></tr><tr><td>LF_{3a}</td><td>-1</td><td></td><td>-1</td><td></td><td></td></tr><tr><td>LF_{4c}</td><td>1</td><td></td><td>1</td><td>1</td><td></td></tr><tr><td>⋮</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>\tilde{y}</td><td>+</td><td>-</td><td>-</td><td>+</td><td>...</td></tr></table>		X ₁	X ₂	X ₃	X ₄	...	LF _{1a}	1					LF _{2b}		-1				LF _{3a}	-1		-1			LF _{4c}	1		1	1		⋮						\tilde{y}	+	-	-	+	...
	X ₁	X ₂	X ₃	X ₄	...																																								
LF _{1a}	1																																												
LF _{2b}		-1																																											
LF _{3a}	-1		-1																																										
LF _{4c}	1		1	1																																									
⋮																																													
\tilde{y}	+	-	-	+	...																																								
<p>X₂ None of us knows what happened at Kane's home Aug. 2, but it is telling that the NHL has not suspended Kane.</p> <p>False, because person 1 and person 2 in the sentence are identical.</p>	<pre>def LF_2a(x): return (-1 if x.person1 in x.sentence and x.person2 in x.sentence else 0)</pre> <pre>def LF_2b(x): return (-1 if x.person1 == x.person2) else 0)</pre>	<p>Pragmatic Filter (always true)</p> <p>Correct</p>																																											
<p>X₃ Dr. Michael Richards and real estate and insurance businessman Gary Kirke did not attend the event.</p> <p>False, because the last word of person 1 is different than the last word of person 2.</p>	<pre>def LF_3a(x): return (-1 if x.person1.tokens[-1] != x.person2.tokens[-1] else 0)</pre> <pre>def LF_3b(x): return (-1 if not (x.person1.tokens[-1] == x.person2.tokens[-1]) else 0)</pre>	<p>Correct</p> <p>Pragmatic Filter (duplicate of LF_{3a})</p>	<p>Noisy Labels</p> <p>(X₁, \tilde{y}_1) (X₂, \tilde{y}_2) (X₃, \tilde{y}_3) (X₄, \tilde{y}_4)</p> <p>Classifier</p> 																																										

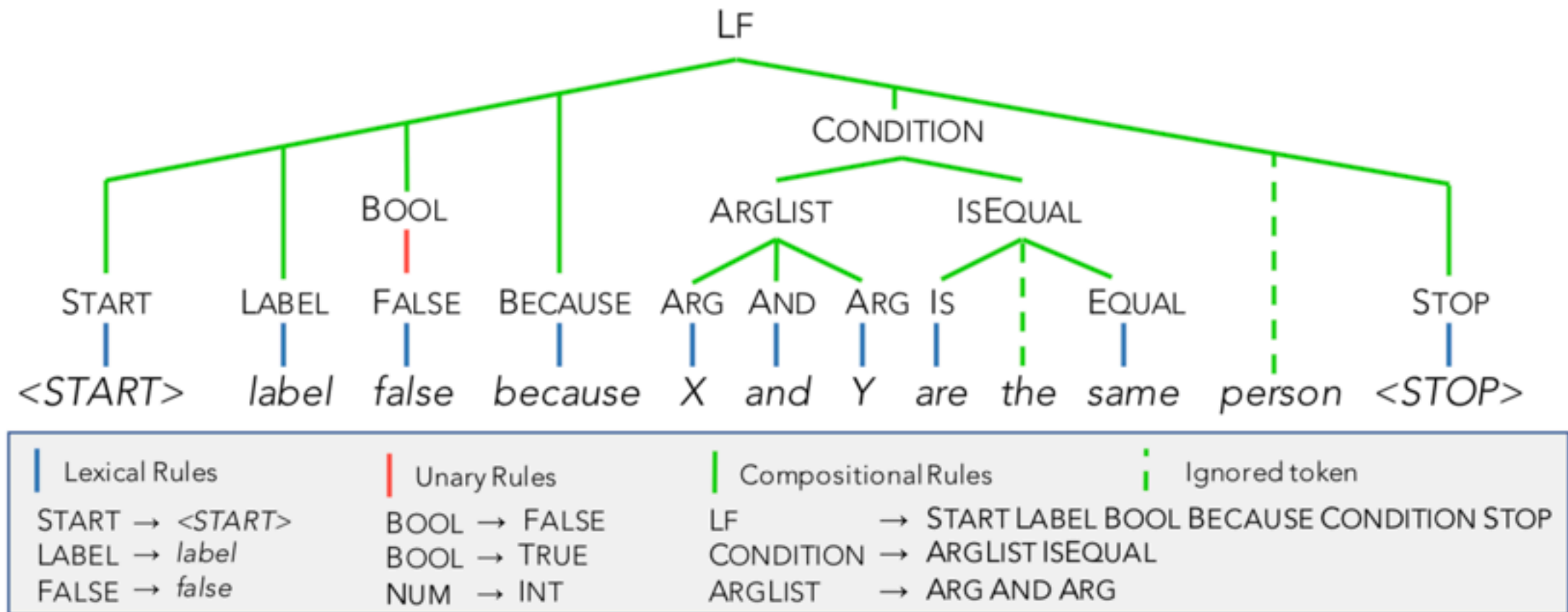
The filter bank removes as many incorrect LFs as possible without requiring ground truth labels.

Methods

Unlabeled Examples + Explanations	Labeling Functions	Filters	Label Matrix																																										
<p>Label whether person 1 is married to person 2</p> <p>X₁ Tom Brady and his wife Gisele Bündchen were spotted in New York City on Monday amid rumors of Brady's alleged role in Deflategate.</p> <p>True, because the words "his wife" are right before person 2.</p>	<pre>def LF_1a(x): return (1 if "his wife" in left(x.person2, dist==1) else 0)</pre>	Correct	<table><tr><td></td><td>X₁</td><td>X₂</td><td>X₃</td><td>X₄</td><td>...</td></tr><tr><td>LF_{1a}</td><td>1</td><td></td><td></td><td></td><td></td></tr><tr><td>LF_{2b}</td><td></td><td>-1</td><td></td><td></td><td></td></tr><tr><td>LF_{3a}</td><td>-1</td><td></td><td>-1</td><td></td><td></td></tr><tr><td>LF_{4c}</td><td>1</td><td></td><td>1</td><td>1</td><td></td></tr><tr><td>⋮</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>\tilde{y}</td><td>+</td><td>-</td><td>-</td><td>+</td><td>...</td></tr></table>		X ₁	X ₂	X ₃	X ₄	...	LF _{1a}	1					LF _{2b}		-1				LF _{3a}	-1		-1			LF _{4c}	1		1	1		⋮						\tilde{y}	+	-	-	+	...
	X ₁	X ₂		X ₃	X ₄	...																																							
LF _{1a}	1																																												
LF _{2b}		-1																																											
LF _{3a}	-1		-1																																										
LF _{4c}	1		1	1																																									
⋮																																													
\tilde{y}	+	-	-	+	...																																								
<p>X₂ None of us knows what happened at Kane's home Aug. 2, but it is telling that the NHL has not suspended Kane.</p> <p>False, because person 1 and person 2 in the sentence are identical.</p>	<pre>def LF_1b(x): return (1 if "his wife" in right(x.person2) else 0)</pre>	Semantic Filter (inconsistent)																																											
<p>X₃ Dr. Michael Richards and real estate and insurance businessman Gary Kirke did not attend the event.</p> <p>False, because the last word of person 1 is different than the last word of person 2.</p>	<pre>def LF_2a(x): return (-1 if x.person1 in x.sentence and x.person2 in x.sentence else 0)</pre>	Pragmatic Filter (always true)	<p>Noisy Labels</p> <p>Classifier</p> 																																										
	<pre>def LF_2b(x): return (-1 if x.person1 == x.person2) else 0)</pre>	Correct																																											
	<pre>def LF_3a(x): return (-1 if x.person1.tokens[-1] != x.person2.tokens[-1] else 0)</pre>	Correct																																											
	<pre>def LF_3b(x): return (-1 if not (x.person1.tokens[-1] == x.person2.tokens[-1]) else 0)</pre>	Pragmatic Filter (duplicate of LF _{3a})																																											

The label combines these potentially conflicting and overlapping labels into one label.

Semantic Parser



- Lexical: converting tokens into symbols
- Unary: converting one symbol into another symbol
- Compositional: combining many symbols into a single higher-order symbol.

Semantic Parser

Predicate	Description
bool, string, int, float, tuple, list, set	Standard primitive data types
and, or, not, any, all, none	Standard logic operators
=, ≠, <, ≤, >, ≥	Standard comparison operators
lower, upper, capital, all_caps	Return True for strings of the corresponding case
starts_with, ends_with, substring	Return True if the first string starts/ends with or contains the second
person, location, date, number, organization	Return True if a string has the corresponding NER tag
alias	A frequently used list of words may be predefined and referred to with an alias
count, contains, intersection	Operators for checking size, membership, or common ele-

Filter Bank

Semantic Filter: any LF f for which $f(x_i) \neq y_i$ is discarded.

Unlabeled Examples + Explanations

Label whether person 1 is married to person 2
x_1 Tom Brady and his wife Gisele Bündchen were spotted in New York City on Monday amid rumors of Brady's alleged role in Deflategate.
True , because the words "his wife" are right before person 2 .

Labeling Functions

<pre>def LF_1a(x): return (1 if "his wife" in left(x.person2, dist==1) else 0)</pre>	Correct
<pre>def LF_1b(x): return (1 if "his wife" in right(x.person2) else 0)</pre>	Semantic Filter (inconsistent)
<pre>def LF_2a(x): return (1 if "person 1" in left(x.person2, dist==1) else 0)</pre>	Pragmatic

Filter Bank

Pragmatic Filter: removes LFs that are constant, redundant, or correlated

<p>are right before person 2.</p>	<pre>def LF_2a(x): return (-1 if x.person1 in x.sentence and x.person2 in x.sentence else 0)</pre>	Pragmatic Filter (always true)
<p>X₂ None of us knows what happened at Kane's home Aug. 2, but it is telling that the NHL has not suspended Kane.</p> <p>False, because person 1 and person 2 in the sentence are identical.</p>	<pre>def LF_2b(x): return (-1 if x.person1 == x.person2) else 0)</pre>	Correct

LF 2a is constant, as it labels every example positively (since all examples contain two people from the same sentence)

Filter Bank

Pragmatic Filter: removes LFs that are constant, redundant, or correlated

<p>X₃ Dr. Michael Richards and real estate and insurance businessman Gary Kirke did not attend the event.</p> <p>False, because the last word of person 1 is different than the last word of person 2.</p>	<table><tr><td data-bbox="1262 968 2189 1195"><pre>def LF_3a(x): return (-1 if x.person1.tokens[-1] != x.person2.tokens[-1] else 0)</pre></td><td data-bbox="2189 968 2551 1195">Correct</td></tr><tr><td data-bbox="1262 1195 2189 1441"><pre>def LF_3b(x): return (-1 if not (x.person1.tokens[-1] == x.person2.tokens[-1]) else 0)</pre></td><td data-bbox="2189 1195 2551 1441">Pragmatic Filter (duplicate of LF_3a)</td></tr></table>	<pre>def LF_3a(x): return (-1 if x.person1.tokens[-1] != x.person2.tokens[-1] else 0)</pre>	Correct	<pre>def LF_3b(x): return (-1 if not (x.person1.tokens[-1] == x.person2.tokens[-1]) else 0)</pre>	Pragmatic Filter (duplicate of LF_3a)
<pre>def LF_3a(x): return (-1 if x.person1.tokens[-1] != x.person2.tokens[-1] else 0)</pre>	Correct				
<pre>def LF_3b(x): return (-1 if not (x.person1.tokens[-1] == x.person2.tokens[-1]) else 0)</pre>	Pragmatic Filter (duplicate of LF_3a)				

LF 3b is redundant, since even though it has a different syntax tree from LF 3a, it labels the training set identically and therefore provides no new signal

Label Aggregator

Concretely, if m LFs pass the filter bank and are applied to n examples, the label aggregator implements a function $f : \{-1, 0, 1\}^{m \times n} \rightarrow [0, 1]^n$

$$\phi_{i,j}^{\text{Lab}}(\Lambda, Y) = \mathbb{1}\{\Lambda_{i,j} \neq 0\} \quad (1)$$

$$\phi_{i,j}^{\text{Acc}}(\Lambda, Y) = \mathbb{1}\{\Lambda_{i,j} = y_j\}. \quad (2)$$

$$p_w(\Lambda, Y) = Z_w^{-1} \exp \left(\sum_{j=1}^n w \cdot \phi_j(\Lambda, Y) \right), \quad (3)$$

$$\hat{w} = \arg \min_w - \log \sum_Y p_w(\Lambda, Y) \quad (4)$$

Discriminative Model

A discriminative model can incorporate features that were not identified by the user but are nevertheless informative.

- Unigrams
- Bigrams
- Trigrams
- Dependency labels
- POS
- Nodes between entities in the dependency parse of the sentence

Contents

- Introduction
- Methods
- **Experiments**
- Conclusions and Future Work

Data Sets

Spouse (person 1, person 2)	Example	They include Joan Ridsdale , a 62-year-old payroll administrator from County Durham who was hit with a €16,000 tax bill when her husband Gordon died.
	Explanation	True, because the phrase “her husband” is within three words of person 2.
Disease (chemical, disease)	Example	Young women on replacement estrogens for ovarian failure after cancer therapy may also have increased risk of endometrial carcinoma and should be examined periodically.
	Explanation	True, because “risk of” comes before the disease.
Protein (protein, kinase)	Example	Here we show that c-Jun N-terminal kinases JNK1 , JNK2 and JNK3 phosphorylate tau at many serine/threonine-prolines, as assessed by the generation of the epitopes of phosphorylation-dependent anti-tau antibodies.
	Explanation	True, because at least one of the words 'phosphorylation', 'phosphorylate', 'phosphorylated', 'phosphorylates' is found in the sentence and the number of words between the protein and kinase is smaller than 8."

Task	Train	Dev	Test	% Pos.
Spouse	22195	2796	2697	8%
Disease	6667	773	4101	20%
Protein	5546	1011	1058	22%

Experimental Result

# Inputs	BL	TS						
	30	30	60	150	300	1,000	3,000	10,000
Spouse	50.1	15.5	15.9	16.4	17.2	22.8	41.8	55.0
Disease	42.3	32.1	32.6	34.4	37.5	41.9	44.5	-
Protein	47.3	39.3	42.1	46.8	51.0	57.6	-	-
Average	46.6	28.9	30.2	32.5	35.2	40.8	43.2	55.0

Table 3: F1 scores obtained by a classifier trained with BabbleLabbble (BL) using 30 explanations or with traditional supervision (TS) using the specified number of individually labeled examples. BabbleLabbble achieves the same F1 score as traditional supervision while using fewer user inputs by a factor of over 5 (Protein) to over 100 (Spouse).

Experimental Result

	BL-FB	BL	BL+PP
Spouse	15.7	50.1	49.8
Disease	39.8	42.3	43.2
Protein	38.2	47.3	47.4
Average	31.2	46.6	46.8

Table 5: F1 scores obtained using BabbleLabble with no filter bank (BL-FB), as normal (BL), and with a perfect parser (BL+PP) simulated by hand.

Experimental Result

	BL-DM	BL	BL+PP	Feat	Feat+PP
Spouse	46.5	50.1	49.8	33.9	39.2
Disease	39.7	42.3	43.2	40.8	43.8
Protein	40.6	47.3	47.4	36.7	44.0
Average	42.3	46.6	46.8	37.1	42.3

Table 6: F1 scores obtained using explanations as functions for data programming (BL) or features (Feat), optionally with no discriminative model (-DM) or using a perfect parser (+PP).

Contents

- Introduction
- Methods
- Experiments
- **Conclusions and Future Work**

Conclusions and Future Work

Conclusions:

- (1) Natural language opens up a much higher-bandwidth communication channel;
- (2) Extend the framework to other tasks and more interactive settings.

Thanks!