QALF: Information Extraction for the Long Tail via Question Answering

Braden Hancock, Robin Jia, Percy Liang, Christopher Ré
Stanford University

Summary

Problem:
Information extraction targets in the long tail have few distant supervision resources, and traditional supervision is slow and inefficient. How can we more quickly train an extractor in new domains?

Solution:
Generate weak supervision sources by asking relevant questions to a Question Answering (QA) model. Use the probabilistically labeled training set from these sources to train the powerful discriminative classifier you need.

Result:
Using as few as a dozen QA-based labeling functions (LFs) outperforms direct application of a QA model by an average of 26 F1 points.

Motivation

1) The efficiency of weak supervision

- Weak supervision resources are a higher bandwidth form of supervision than individually labeled examples.
- On many tasks, as few as 10 QA-based LFs can perform as well as thousands of labels.

2) QA Models are flexible

- Via word embeddings and training on massive datasets, QA models are relatively robust to paraphrasing differences between questions and documents.
- A natural language interface makes these systems extremely accessible.

3) QA Models will keep improving

- Major companies/universities across the globe are actively competing in SQuAD and other QA competitions, improving the quality of QA models on an almost weekly basis.
- QALF uses QA models as a black box; any improvements in the latter further strengthen the former.

Main Idea: Instead of training an information extraction system with individually labeled examples, weakly supervise it by writing questions that have positive/negative associations with the concept you are trying to extract. Convert the questions into functions for automatically labeling a training set.

Task: Create a database of spouses

User Provides:

Documents

"John Stamos and new bride Caitlin McHugh celebrated their honeymoon at Disney World Resort."

"Kevin Edwards was spotted on his way to the meeting with his newly hired lawyer, Emily Grant."

Schema

Subject: X (PERSON),
Relation: is spouse of,
Object: Y (PERSON)

(+) Positive Qs

"Who is the bride of X?"
"Who has a child with X?"
"Who tied the knot with X?"
"Who lives with X?"
"Who is X's husband?"

(-) Negative Qs

"Who works with X?"
"Who is X's daughter?"
"Who played X in a movie?"
"Who is X's boss?
"Who played against X?"

QALF:

Step 1: Extract candidates

Use an off-the-shelf NER tagger and the provided schema.

x1: (X=John Stamos, Y=Caitlin McHugh)

x2: (X=Kevin Edwards, Y=Emily Grant)

Step 2: Convert Qs into Labeling Functions (LFs)

If a QA model has high confidence that Y is the answer to the question about X, label with the polarity of the Q.

```
def QA_LF1(x):
  question = "Who is the bride of {}?".format(x.X)
  return 1 if QAModel(x.doc, question, x.Y) > 0.8
    else 0

def QA_LF2(x):
  question = "Who works with {}?".format(x.X)
  return -1 if QAModel(x.doc, question, x.Y) > 0.8
    else 0
```

Step 3: Apply QA_LFs to candidates

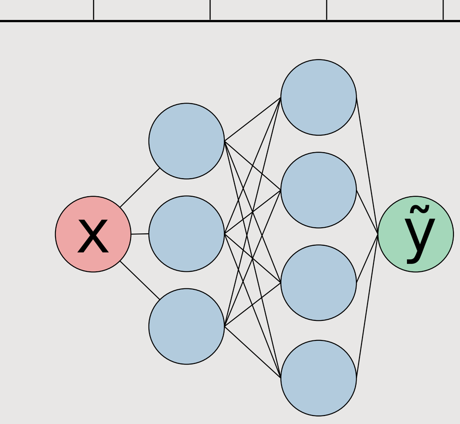
Apply each QA_LF to each candidate to make a matrix of noisy labels.

	x_1	x_2	x_3	x_4	...	$x_{1000000}$
QA_LF_1	1			1		
QA_LF_2		−1	−1			−1
QA_LF_3		1		1		1
QA_LF_4		−1		−1		
⋮						

Step 4: Train a Classifier

Aggregate the noisy labels to make a large training set with probabilistic labels for your task-specific classifier.

	x_1	x_2	x_3	x_4	...	$x_{1000000}$
$\tilde{y}$	.85	.33	.09	.69		.52



Interface

Candidate:

"John Stamos and new bride Caitlin McHugh celebrated their honeymoon at Disney World Resort."

Label:

Is X the spouse of Y?

✓ ✕ ⊘

Related Question:

Write a question whose answer is Y.

Who went on a honeymoon with X?

SUBMIT

Resulting QA-based LF:

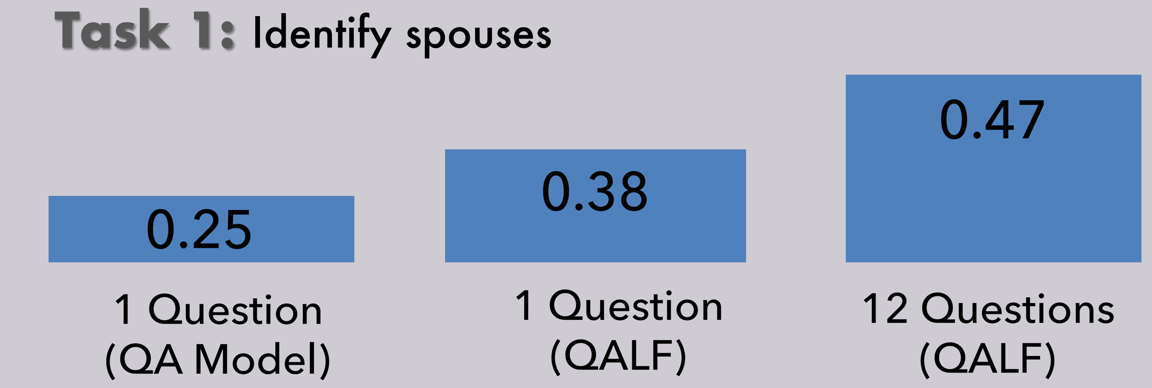
```
def QA_LF1(x):
  question = "Who went on a honeymoon with {}?".format(x.X)
  return 1 if QAModel(x.doc, question, x.Y) > 0.8 else 0
```

Results

Summary:

Over two relation extraction tasks, using a small number of questions as QA-based LFs to generate a training set for a classifier provides a boost of 26 F1 points on average over using a trained QA model directly on the test set.

Task 1: Identify spouses



Task 2: Identify official titles