Language to Code: Learning Semantic Parsers for If-This-Then-That Recipes

Chris Quirk
chrisq@microsoft.com

Raymond Mooney
mooney@cs.utexas.edu

Michel Galley
mgalley@microsoft.com

GOAL: ENABLE LANGUAGE-TO-CODE RESEARCH.
If-this-then-that is a small programming language with *lots* of data http://research.microsoft.com/lang2code/

TASK AND DATA

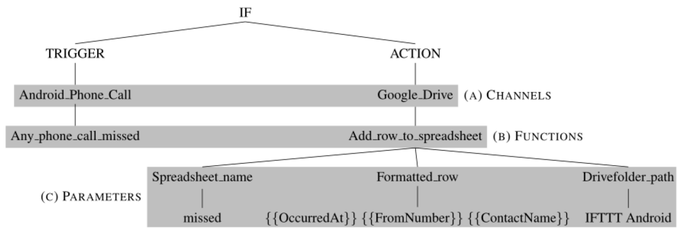
if-this-then-that (ifttt.com) programs automate many common activities:

- If your facebook profile picture changes then update your twitter profile picture
- If I star an email in Gmail, create a reminder to take care of it
- Upload new iOS photos to Google drive
- Keep a phone call log in Google drive

Currently users must author these recipes using a GUI programming environment.

Our goal is to automatically convert natural language descriptions into a executable programs. Interpreting and executing these programs is clearly very important, but not addressed here.

Recipes with descriptions as language-code pairs



Archive your missed calls from Android to Google Drive

These recipes can be seen as programs in a simple programming language with a single top-level construct and a number of triggers or actions. Each trigger or action refers to a single Channel with a single Function and an optional list of Parameters.

Data crawled from http://ifttt.com

		Language	Code
Train	Recipes	77,495	77,495
	Tokens	527,368	1,776,010
	Vocabulary	58,102	140,871
Dev	Recipes	5,171	5,171
	Tokens	37,541	110,074
	Vocabulary	7,741	14,804
Test	Recipes	4,294	4,294
	Tokens	28,214	94,367
	Vocabulary	6,782	13,969

METHODOLOGY

Baselines

1. Retrieval
Given a test description, find the closest description in the training set, and return the program used for that training example.

2. Phrasal
Apply a conventional phrasal statistical machine translation system to the pairs (Andreas et al. 2013)

3. Sync
Learn a synchronous context free grammar on the training data; search for the best derivation according to this grammar (cf. Andreas et al. 2013)

Novel methods

4. Classifier
Learn a feature rich distribution over the productions in the formal grammar

5. Position-based classifier
Augment feature rich distribution with latent alignment variable (cf. Kate and Mooney 2006)

Mechanical Turk

Mechanical turk workers also solved this task, though partially. They were presented with descriptions of recipes, and asked to select the channel and function for the trigger and action. Five-way coverage of 4K test recipes completed in 9 hours. Agreement with Krippendorff's α was good, especially over recipes that were marked as English and intelligible:

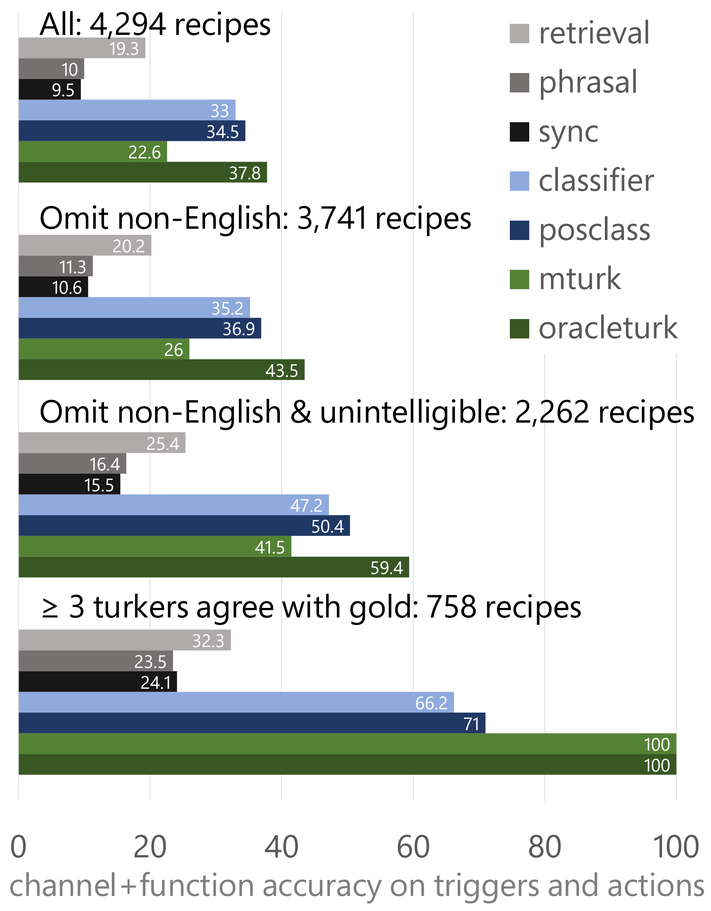
	Trigger		Action	
	C	C+F	C	C+F
# of categories	*128*	*552*	*99*	*229*
All	.592	.492	.596	.532
Intelligible English	.687	.528	.731	.627

We added two more systems based on this result:

6. Mturk
Take the majority decision by the turkers, separately for the trigger and the action.

7. Oracleturk
If at least one of the turkers agreed with the gold standard, use the gold standard trigger. Otherwise pick the majority.

EVALUATION



The task is surprisingly difficult. Systems dependent on word alignment struggled on this dataset; word alignment quality looked poor. Discriminative approaches fared much better.

Some disagreement comes from overlapping channels: both Android and iOS can provide location. When we use cases where humans agree, systems perform reasonably well.

CONCLUSIONS

We've presented a broad and interesting dataset for evaluating semantic parsers. Although the data is somewhat constrained, the data is very natural. We hope this dataset will drive new research in natural language programming.

We are investigating approaches that hopefully perform better on the first interaction. We are also interested in data-driven dialog systems to that use multiple interactions for improved understanding.