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

**Abstract:**

An approach that learns to map natural language descriptions of “if-then” rules to executable code. Training and testing on large data of naturally occurring programs are called recipes. This paper compares a number of semantic parsing approaches on the highly noisy training data collected from ordinary users and find that loosely synchronous systems perform best.

**Introduction:**

Work in semantic parsing has explored mapping natural language to some formal domain-specific programming languages such as database queries, commands to robots, operating systems, smartphones and spreadsheets.

Several services, such as Tasker and IFTTT, allow users to create simple programs with “triggers” and “actions.” For example, one can program their Phillips Hue light bulbs to flash red and blue when the Cubs hit a home run.

Have collected 114,408 recipe description pairs from the <http://ifttt.com> website. Because users often provided short or incomplete English descriptions, the resulting data is extremely noisy for the task of training a semantic parser.

Therefore, have constructed semantic-parser learners that utilize and adapt ideas from several previous approaches to learn an effective interpreter from such noisy training data.

**Background:**

Given supervised training data in the form of natural-language sentences each paired with their corresponding IFTTT recipe, we learn to introduce productions from the formal-language grammar into the derivation of the target program based on expressions in the natural-language input.

This approach originated with the SILT system and was further developed in

the WASP and KRISP systems.

**WASP** casts semantic parsing as a syntax-based statistical machine translation (SMT) task, where a synchronous context-free grammar is used to model the translation of natural language into a formal meaning representation. It uses statistical models developed for syntax-based SMT for lexical learning and parse disambiguation. Productions in the formal-language grammar are used to construct

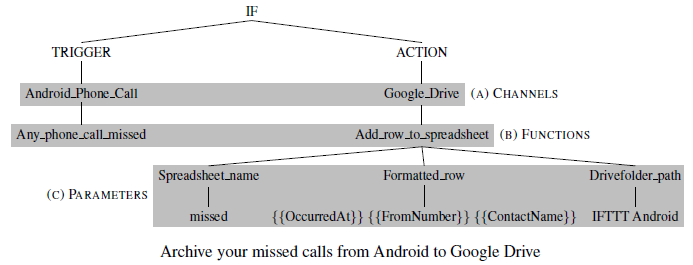
synchronous rules that simultaneously model the generation of the natural language.

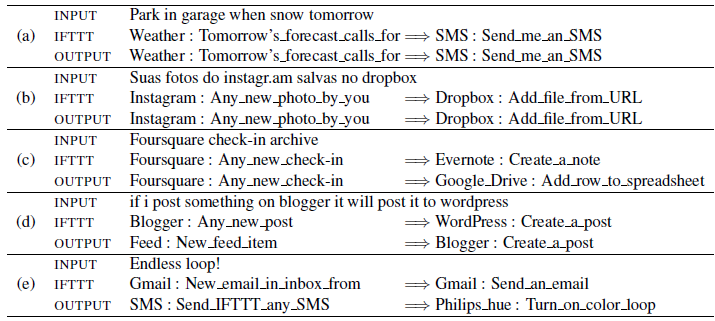
**KRISP** uses classifiers trained using a Support-Vector Machine (SVM) to introduce productions in the derivation of the formal translation. During semantic parsing, these classifiers are employed to estimate probabilities on different substrings of the sentence to compositionally build the most probable meaning representation for the sentence.

**IFTTT** is extremely noisy, we have taken an approach similar to KRISP, but we use a probabilistic log-linear text classifier rather than an SVM to recognize productions.

**If-this-then-that recipes:**

Each recipe always contains exactly one trigger and one action. Whenever the conditions of the trigger are satisfied, the action is performed.





From the above image based on few keywords the output is predicted correctly. But the last example (e) is wrong prediction since its email loop situation and prediction becomes wrong due to the keyword ‘loop’.

**Summary:**

Semantic parsing systems have accuracy nearly as high or even higher than turkers (Human annotators on Amazon Mechanical Turk) in certain conditions.