CASPER
Controllable Semantic Parsing via Retrieval Augmentation

Ice Pasupat, Yuan Zhang, Kelvin Guu
Google Research
Goal: Modify the behavior of the semantic parser at test time.

Train
Training data:

- make a call to Jay Brown's mom
- send a text message to Jack

Test
Produce MESSAGING parses given a few examples

Parser

[IN create call =
 [SL contact =
 [IN get contact =
 [SL contact related = Jay Brown]
 [SL type relation = mom]]]]

[IN send message =
 [SL recipient = Jack]]
Motivation

**Goal:** Modify the behavior of the semantic parser at test time.

We want to modify the behavior without additional training.

- Less computation resources.
- Stability: Avoid model churn.
- Faster development: Update the parser and immediately see the result.
- Customization: Clients can modify the parser without touching the model's params on the server.
Given the query, a retriever retrieves related exemplars (e.g., training examples with similar queries).
**CASPER** ControllAble Semantic Parser via Exemplar Retrieval

Given the query, a **retriever** retrieves related **exemplars** (e.g., training examples with similar queries). A seq2seq **generator** then takes the exemplar-augmented query and produces a parse.
**CASPER** ControllAble Semantic Parser via Exemplar Retrieval

- The generator learns to use (or ignore) additional information given by the exemplars.
The generator learns to use (or ignore) additional information given by the exemplars. At test time, we can modify the parser's behavior by manipulating the retrieval process.
Increasing faithfulness toward exemplars

- The generator learns to use (or ignore) additional information given by the exemplars.
- At test time, we can modify the parser's behavior by manipulating the retrieval process.
  - This can be done without additional model training.
  - But for this to work, we want the parser to lean toward using instead of ignoring the exemplars.
Increasing faithfulness toward exemplars

**Method 1:** Mix in *anonymized* training data.
- Teach the generator to rely on the exemplars when producing semantic labels.
Increasing faithfulness toward exemplars

Method 2: Add manual control via **guiding tags**.

**Test**

```
call me maybe
@@ poker face ## [IN play music = …
@@ bad guy ## [IN play music = …
```

The parser ignores the exemplars.

```
call me maybe
@@ PLATINUM poker face ## [IN play music = …
@@ PLATINUM bad guy ## [IN play music = …
```

The parser follows the exemplars more closely.
Increasing faithfulness toward exemplars

**Method 2:** Add manual control via **guiding tags**.
- Teach the model about guiding tags by mixing in **oracle examples**

---

**Train**

- Make a call to Jay Brown’s mom

**Retrieval Index**

- Call Zoey’s wife
- [IN create call = ...]
- Make a call to ...
- [IN create call = ...]
- Get number ...
- [IN get number = ...]

**Retriever**

**Generator**

- Make a call to Jay Brown’s mom
  - @@ PLATINUM call Zoey’s wife
  - @@ PLATINUM Make a call ...
  - @@ Get number ...

**Exemplars**

- [IN create call = ...
  - [SL contact = ...
  - [IN get contact = ...
  - [SL contact related = Jay Brown]
  - [SL type relation = mom]]]]

---

**Only use exemplars with the same template as the gold parse.**
Experiments

**Dataset:** English portion of MTOP (Li et al., 2021)

- **CASPER improves the accuracy** in the standard train-test setup.
- We can **control CASPER’s behavior** at test time without additional training in 3 different setups.
Experiments

(0) Standard train-test setup

- Improve upon SotA (Li et al., 2021) by 2%
- Adding anonymized training data slightly hurts, but will pay off in other experiments.

<table>
<thead>
<tr>
<th></th>
<th>Test accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SotA</td>
<td>84.3</td>
</tr>
<tr>
<td>seq2seq (T5)</td>
<td>85.1</td>
</tr>
<tr>
<td>CASPER (-anon)</td>
<td>86.4</td>
</tr>
<tr>
<td>CASPER (+anon)</td>
<td>85.5</td>
</tr>
</tbody>
</table>

\( x: \) What’s the biggest story today?
\( x_1^t: \) what’s the top story for today?
\( y_1: \) [IN get stories news = [SL news reference = top]
 [SL news type = story] [SL date time = for today]]
\( x_4^t: \) Tell me the biggest news story of the day.
\( y_4: \) [IN get stories news = [SL news type = news story]]

T5: [IN get stories news = [SL news type = story]
 [SL date time = today]]
C_0: [IN get stories news = [SL news reference = biggest]
 [SL news type = story] [SL date time = today]]
Experiments

(1) Domain bootstrapping
- Remove 1 domain (out of 11) from the training data.
- At test time, add 100 examples of that removed domain to the retrieval index.

The accuracy on other domains remain roughly the same.

![Graph showing Dev accuracy on the new domain for seq2seq (T5), CASPER (-anon), and CASPER (+anon).](image)

← averaged over 5 bootstrapped domains, some are easier (event: acc = 68.29) and some are harder (music: acc = 8.21)

The accuracy on other domains remain roughly the same.
Experiments

(2) Parse guiding

- Override the retriever by manually supplying oracle exemplars.

(same template as the gold parse)

<table>
<thead>
<tr>
<th>Dev accuracy when supplying ...</th>
<th>retrieved exemplars</th>
<th>oracle exemplars</th>
<th>oracle exs + guiding tags</th>
</tr>
</thead>
<tbody>
<tr>
<td>C (CASP)</td>
<td>84.3</td>
<td>88.2</td>
<td>88.3</td>
</tr>
<tr>
<td>C+anon (CASP)</td>
<td>83.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C+anon+guide (CASP)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Oracle exemplars → increased accuracy in general.

But the model that learned about guiding tags can become extra faithful toward the exemplars when the guiding tags are present → even higher accuracy.

Practical applications: Overriding persistent model errors or sensitive queries
Experiments

(3) Schema refactoring

- Split 10 semantic labels into 2 each at test time.
- Update the retrieval index accordingly.

Both adding anonymized training examples and guiding tags on the affected exemplars lead to improved post-refactoring accuracy.
Summary

We proposed **CASPER**: ControllAble Semantic Parser via Exemplar Retrieval.

The parser's behavior can be modified **without additional model training** by manipulating the retrieval process at test time.

See the paper for more:
- Accounting for bad retrievals
- Ablation studies
- Error analysis
- Comparison with fast update methods

Thank you!