Zero and Few-shot Semantic Parsing with Ambiguous Inputs

Elias Stengel-Eskin[1] **Kyle Rawlins**[2] **Benjamin Van Durme**[1,2]
`{elias,kgr,vandurme}@jhu.edu`

[1]Department of Computer Science [2]Department of Cognitive Science

Abstract

Despite the ubiquity of ambiguity in natural language, it is often ignored or deliberately removed in semantic parsing tasks, which generally assume that a given surface form has only one correct logical form. We attempt to address this shortcoming by introducing AMP, a framework, dataset, and challenge for parsing with linguistic ambiguity. We define templates and generate data for five well-documented linguistic ambiguities. Using AMP, we investigate how several few-shot semantic parsing systems handle ambiguity, introducing three new metrics. We find that large pre-trained models perform poorly at capturing the distribution of possible meanings without deliberate instruction. However, models are able to capture distribution well when ambiguity is attested in their inputs. These results motivate a call for ambiguity to be explicitly included in semantic parsing, and promotes considering the distribution of possible outputs when evaluating semantic parsing systems. [1]

1 Introduction

Formalizing the meaning of natural language into a symbolic representation has been attempted across a variety of domains, from philosophy and linguistics (Wittgenstein, 1921; Montague, 1970) to artificial intelligence (Winograd, 1972; Zelle and Mooney, 1996). Attempts at formalization have often faced a shared challenge: many natural language statements have multiple possible meanings, i.e. they are ambiguous. Past work (e.g. Zipf, 1949; Piantadosi et al., 2012) has argued that this is a natural feature of a communication system, resulting from competing pressures on speakers and listeners. Specifically, Piantadosi et al. contend that ambiguity allows speakers to minimize their efforts. Rather than exactly specify their intended meaning (resulting in a long and expensive message), speakers can send shorter, cheaper messages and rely on listeners to resolve any ambiguities. However, this resolution in turn relies on *commmonsense knowledge* and *conversational context*: most speakers of English would infer from the utterance, *"I ate spaghetti with a fork"* that someone used a fork as a utensil, but commonsense knowledge would preclude this parse of *"I ate spaghetti with meatballs"*. Similarly, conversational context can provide clues to help us choose between interpretations.

Language can be used not only to communicate with other people, but also to interact with AI agents. One common method for interaction is semantic parsing, whereby natural language is translated into a formal and symbolic representation of its meaning (e.g. code, logic, graphs, etc.). However, the tools humans use for ambiguity resolution may not be available to these non-human translation systems. Models of language lack the embodied world experience which underlies much of our commonsense knowledge. Furthermore, we often interact with models in unnatural ways and without providing a full conversational context.

Ideally, given an ambiguous input, we would like our parsing models to capture a *distribution* over interpretations with some uncertainty across plausible items in the distribution. This would allow robust handling of ambiguous utterances – for example by enabling smart follow-up interactions (Stengel-Eskin and Van Durme, 2023) – getting us closer to the goal of using language as a general-purpose API for interaction. Given that semantic parsing systems are typically based on language models, which represent distributions over strings, it could be that models already capture ambiguity. However, this hypothesis is hard to test given current semantic parsing datasets, which typically commit to a single interpretation for each utterance. To this end, we introduce an extensible framework

[1]Data and code: `https://github.com/esteng/ambiguous_parsing`

and dataset for investigating ambiguity in semantic parsing. Our framework consists of templates covering five well-documented types of natural language ambiguity: prepositional-phrase attachment, scope and inverse scope, pronominal coreference, and conjunctions. For each type, our templates can generate large numbers of ambiguous and unambiguous utterances. Each ambiguous utterance is paired with two possible interpretations, or logical forms (LFs); see Fig. 1 for an example. LFs can be represented as first-order-logic (FOL) formulae or as programs in Lisp. We use our framework to create a benchmark dataset we call AMP (**Am**biguous **P**arsing). Unlike past efforts which have grounded ambiguous utterances in answers to questions (Stengel-Eskin et al., 2023), language inferences (Liu et al., 2023), videos (Berzak et al., 2015), or images (Mehrabi et al., 2022), we focus our dataset on semantic parses. This choice follows from several motivating factors. For one, semantic parsing has a long tradition of use in interactive systems, including in robotics (Kate et al., 2005; Tellex et al., 2011; Artzi and Zettlemoyer, 2013; Tellex et al., 2020), question-answering (Zelle and Mooney, 1996; Berant et al., 2013; Yu et al., 2018), and digital assistants (Semantic Machines et al., 2020; Damonte et al., 2019). Ensuring that these systems capture ambiguity and that their confidence reflects appropriate uncertainty about the user's intent is crucial, as misunderstandings could have negative real-world consequences (Stengel-Eskin and Van Durme, 2023). Furthermore, semantic parsing not only allows people to access computation, but also provides a way for models to use external tools: for example, simple forms of semantic parsing have been employed to augment large language models (LLMs) (Parisi et al., 2022; Schick et al., 2023; Mialon et al., 2023). Finally, longer-form code prediction from language also involves many of the same challenges.

Using our generated AMP data, we introduce a pair of challenging tasks designed for LLMs using in-context learning (ICL). In ICL, rather than explicitly training models to predict LFs, we provide models with instructive examples in a prompt, which is prepended to the test input. This parsing setting has become increasingly popular in semantic parsing (Shin et al., 2021; Shin and Van Durme, 2022; Roy et al., 2022). Our tasks aim to quantify how well existing models model ambiguity and to provide a framework for improving their ability to predict multiple meanings. We develop three metrics to measure models' abilities to capture ambiguity in two settings: **zero-shot** and **mixed prompt**.

In the **zero-shot** setting, we provide models with the "ingredients" to produce both possible derivations of a given ambiguity type, but we provide no examples of that ambiguity type; see Fig. 2 for an example. In this unique compositional generalization challenge, the model must combine structures into novel derivations and *also* recognize that the structures can be selected and combined in two ways to produce different derivations. We also annotate a subset of our data with crowdsourced judgements, comparing these to our model's predictions. Models struggle to predict parses correctly in this setting. When they do compose parses correctly, while models and people tend to choose similar interpretations, models generally fail to predict both possible parses.

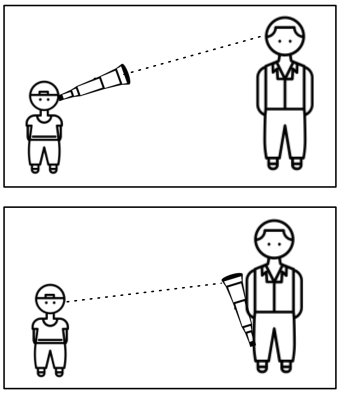
In the **mixed prompt** setting, we examine how model distributions and outputs change when varying the number of examples for each interpretation in the prompt. For each ambiguity type, we construct "mixed prompts" consisting of conflicting examples. Some examples shown to the model pair utterances of an ambiguity type with one kind of LF, and others pair the same kinds of inputs with the alternative LF. This setting is motivated by a case in which ambiguity in language might lead to conflicting annotations in a training dataset; when examples are retrieved from that data to construct a prompt for a test example, the resulting prompt will contain conflicting parses. Here, our metrics measure to what extent a model represents the distribution in its input given conflicting evidence. Some models perform remarkably well here, capturing the prompt distribution across ambiguity types.

2 Dataset

We take a neo-Davidsonian event semantics approach to our logical forms (LFs) (Parsons, 1990), and express our logical forms in quantified first-order logic (FOL). Events are represented as variables, with event-type predicates applied to them. For example, the statement *a woman walks* would be represented as $\exists x.\exists e.woman(x) \land walk(e) \land agent(e, x)$, allowing for an arbitrary number of semantic roles. Generic noun phrases (e.g. *a dog*) are existentially quantified: $\exists x.dog(x)$.[2] Proper nouns

[2]Here, we differ from Kim and Linzen (2020), who use the ι notation for definite articles to denote a uniqueness clause.

The boy saw the man with the telescope



$\exists x.\exists y.\exists z.\exists a.boy(x) \wedge man(y) \wedge telescope(z) \wedge saw(a) \wedge agent(a, x) \wedge patient(a, y) \wedge instrument(a, z)$

```
(exists x (exists y (exists z ( exists a
  (AND (boy x) (man y) (telescope z)
       (saw a) (agent a x) (patient a y)
       (instrument a z))))))
```

$\exists x.\exists y.\exists z.\exists a.\exists e.boy(x) \wedge man(y) \wedge telescope(z) \wedge saw(a) \wedge agent(a, x) \wedge patient(a, y) \wedge have(e) \wedge agent(e, x) \wedge patient(e, z)$

```
(exists x (exists y (exists z ( exists a
(exists e (
  (AND (boy x) (man y) (telescope z)
       (saw a) (agent a x) (patient a y)
       (have e) (agent e x) (patient e z)
)))))))
```

Figure 1: An example of prepositional phrase (PP) attachment ambiguity. The statement is compatible with two possible interpretations, represented visually, in first-order logic, and in Lisp format.

are assumed to have a single referent, and are not quantified, e.g. *Mary walks* $\rightarrow \exists e.walks(e) \wedge agent(e, Mary)$.

We can canonicalize our LFs, so that logically equivalent formulae with varying syntax are treated as identical: we transform LFs into binary trees, where nodes are ordered alphabetically, and we anonymize variables. Note that when prompting our model, we do use a standard variable set and order, where the variables x, y, z are used for nouns, and a, e, i are used for events. Our canonicalization process also allows us to render our LFs in different formats. In addition to a standard FOL format, we experiment with a Lisp format (cf. Fig. 1). To make tokenization easier and more natural, we always render logical connectives in plaintext, i.e. $\exists$ becomes `exists`, $\wedge$ becomes `AND`, etc.. We consider five types of syntactic and semantic ambiguities, given in Table 1.[3]

Extending to new templates The framework we release allows for the addition of new templates and lexical items. To add a new template, the user specifies a surface-form template and an LF template, and provides the set of lexical items that can be used to fill slots in the templates. The framework enumerates all possible combinations of lexical items which respect the template constraints and produces paired inputs and LFs.

This is implemented as a existential quantifier at the widest scope, which can be ignored in all cases except scope ambiguity, where we only have indefinitely-quantified NPs. Similarly, Artzi et al. (2015) introduce Skolem terms (Steedman, 2011) for definite NPs, which are also globally scoped. Thus, we do not differentiate between definite and indefinite NPs in AMP. This has no impact on ambiguity.

[3]We can also generate unambiguous utterances.

3 Methods

Semantic parsing tasks are often framed as sequence transduction tasks where a model learns to translate text into LFs by training on large amounts of paired data (Dong and Lapata, 2016; Zhang et al., 2019). It has become clear that neural models can capture distributions they are trained on; thus, if we were to train on ambiguous data, it would not be surprising if the model captured ambiguity, and vice-versa. Rather than directly training models, we instead consider several models for in-context learning (ICL), focusing on large pre-trained autoregressive language models. In both settings, we use the Codegen series of models (Nijkamp et al., 2022), which are based on the GPT-2 architecture (Radford et al., 2019) and pre-trained on large amounts of code and text. Past work (Shin and Van Durme, 2022) has shown that code pre-training improves over text pre-training on other ICL semantic parsing tasks. We consider multiple sizes of Codegen models: 350 million (M) parameters, 2 billion (B), 6B, and 16B. We also consider LLMs pre-trained on text; here, we examine Llama-13B (Touvron et al., 2023), an open-source autoregressive transformer trained on text and some code. To examine the impact of instruction tuning (Wei et al., 2022), we also consider Vicuna-13B (Chiang et al., 2023), which uses prompts distilled from ChatGPT to instruction-finetune Llama-13B. All models above 350M were run at `fp16` precision. In the zero-shot setting, we also consider OpenAI's gpt-3.5-turbo (which underlies ChatGPT). While little is known about the model's details, it is a large pre-trained transformer model which has undergone both instruction tuning and fine-tuning from

Type	Ex. Input	LF_0	LF_1
Prepositional phrase attachment (PP)	*The man saw the boy with the telescope*	$\exists x.\exists y.\exists z.\exists a.\exists e.man(x) \wedge boy(y) \wedge saw(a) \wedge agent(a,x) \wedge patient(a,y) \wedge telescope(z) \wedge have(e) \wedge agent(e,y) \wedge patient(e,z)$ *Interpretation*: the man saw the boy, who was holding a telescope.	$\exists x.\exists y.\exists z.\exists a.man(x) \wedge boy(y) \wedge telescope(z) \wedge saw(a) \wedge agent(a,x) \wedge patient(a,y) \wedge instrument(a,z)$ *Interpretation*: the man used a telescope to see the boy.
Quantifier scope (Scope)	*every cow saw a dog*	$\exists x.\forall y.\exists a.cow(y) \wedge dog(x) \wedge saw(a) \wedge agent(a,y) \wedge patient(a,x)$ *Interpretation*: there is *exactly* one dog.	$\forall x.\exists y.\exists a.cow(x) \wedge dog(y) \wedge saw(a) \wedge agent(a,x) \wedge patient(a,y)$ *Interpretation*: there may be more than one dog.
Reversed, or inverse scope (revscope)	*a cow saw every dog*	$\exists x.\forall y.\exists a.cow(x) \wedge dog(y) \wedge saw(a) \wedge agent(a,x) \wedge patient(a,y)$ *Interpretation*: there is *exactly* one cow.	$\forall x.\exists y.\exists a.cow(y) \wedge fish(x) \wedge saw(a) \wedge agent(a,y) \wedge patient(a,x)$ *Interpretation*: there may be more than one cow.
Pronoun coreference (bound)	*Mary saw the woman and she smiled*	$\exists x.\exists a.\exists e.woman(x) \wedge saw(a) \wedge agent(a,Mary) \wedge patient(a,x) \wedge smiled(e) \wedge agent(e,Mary)$ *Interpretation*: Mary smiled.	$\exists x.\exists a.\exists e.woman(x) \wedge saw(a) \wedge agent(a,Mary) \wedge patient(a,x) \wedge smiled(e) \wedge agent(e,x)$ *Interpretation*: the woman smiled.
Conjunction (conj.)	*the man drank and ate or swam*	$\exists x.\exists a.\exists e.\exists i.man(x) \wedge ((drank(a) \wedge agent(a,x) \wedge ate(e) \wedge agent(e,x)) \vee (swam(i) \wedge agent(i,x)))$ *Interpretation*: the man either drank and ate or he swam.	$\exists x.\exists a.\exists e.\exists i.man(x) \wedge (drank(a) \wedge agent(a,x) \wedge ((ate(e) \wedge agent(e,x)) \vee (swam(i) \wedge agent(i,x))))$ *Interpretation*: the man drank, and he either ate or swam.

Table 1: Ambiguity types considered with example inputs and LFs. **Lexical items**: For PP ambiguity, we pair visual verbs (e.g. *see, observe, spot, etc.*) with visual instruments (e.g. *telescope, binoculars, etc.*) and tactile verbs (e.g. *grab, pick up, etc.*) with things that can be worn/possessed and used for manipulation (e.g. *gloves, ovenmitts, tongs, etc.*). For scope and reverse scope, we use common nouns and visual and tactile verbs. For pronoun coreference, the lexical items used here are gendered names (e.g *Mary, John, etc.*) and gendered nouns (e.g. *woman, man, boy, girl*). Conjunction examples use intransitive verbs.

human feedback (Ouyang et al., 2022). It is often the most performant model; however, the API does not provide access to logit scores, precluding analyses of uncertainty. As such, we only consider it for our zero-shot experiments, where our main metric is accuracy-based rather than uncertainty-based.

Constrained decoding For locally-run models, we use grammar-constrained decoding (Shin et al., 2021; Shin and Van Durme, 2022; Roy et al., 2022) to ensure that the model produces syntactically-correct formulae. During decoding, we use the BenchCLAMP framework (Roy et al., 2022) to restrict the model's output vocabulary according to a context-free grammar, such that the model can only produce strings accepted by the grammar. [4] This allows us to separate the model's semantic performance from its syntactic abilities. We decode with beam search, using a beam of 5.

Computing probability under a forced decode In our analyses, we would like to compare the probabilities the model assigns to LF_0 and LF_1. While one could compare the product of probabilities under the model for each LF, we find that in practice, this results in very low scores for either LF. We instead use Stengel-Eskin and Van Durme (2022)'s sequence confidence estimate to obtain $P_\theta(LF_0)$, renormalizing at the end:

1. We obtain token-level probabilities under a forced decode of LF_0 and LF_1. For each token t_i in an LF with tokens $t_1 \dots t_N$, we compute $P_\theta(t_i|x;t_{1:i-1})$, where $t_{1:i-1}$ is the *gold* token prefix and x is the input prompt.
2. We take $\hat{p} = \min_{i=1}^{N} P_\theta(t_i|x;t_{1:i-1})$, the minimum probability across all tokens.[5]
3. We normalize the probabilities: $P_\theta(LF_0) = \frac{\hat{p}_{LF_0}}{\hat{p}_{LF_0}+\hat{p}_{LF_1}}$ and set $P_\theta(LF_1) = 1 - P_\theta(LF_0)$

3.1 Metrics

Zero-shot metrics In the zero-shot setting, we aim to measure the degree to which the model captures both possible interpretations of an ambiguous statement. Intuitively, when given the "ingredients" to make both interpretations, a model that robustly captures ambiguity should allocate some probability to both. We measure this by computing the proportion of elements for which the model has both

[4]We release our FOL and Lisp grammars.

[5]We also experimented with averaging, which resulted in similar results. We chose min to be consistent with Stengel-Eskin and Van Durme (2022).

The boy saw the man	The boy saw with the telescope	The boy with the telescope
$\exists x.\exists y.\exists a.boy(x)$	$\exists x.\exists y.\exists a.boy(x)$	$\exists x.\exists y.\exists a.boy(x)$
$\wedge\, man(y) \wedge saw(a)$	$\wedge\, telescope(y) \wedge saw(a)$	$\wedge\, telescope(y) \wedge have(a)$
$\wedge\, agent(a, x) \wedge patient(a, y)$	$\wedge\, agent(a, x) \wedge instrument(a, y)$	$\wedge\, agent(a, x) \wedge patient(a, y)$

Figure 2: Example of prompt sentences used for PP attachment ambiguities.

interpretations in its top-k predictions. Note that we remain agnostic here to the exact probability of each interpretation; we aim instead to quantify whether it predicts both interpretations at all. Note also that as we increase k, this metric becomes less stringent. Let T_k be the top k most probable predictions from the model under some sampling method (e.g. beam search), and $\mathbb{I}$ be an indicator function. The zero-shot metric ZM_k is given by Eq. (1). This metric counts how often both LFs are found in the top k outputs; it ranges from 0 to 100, and higher is better.

$$ZM_k = \frac{\sum_{i=1}^{N} \left(\mathbb{I}[LF_0 \in T_k] * \mathbb{I}[LF_1 \in T_k]\right)}{N} * 100 \quad (1)$$

$$FDM = \frac{1}{|R|} \sum_{r \in R} \Big(\big| \big(\frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[y_i = LF_0]\big) - r \big| + \big| \big(\frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[y_i = LF_1]\big) - (1 - r) \big| \Big) \quad (2)$$

$$FIM = \frac{1}{|R|} \sum_{r \in R} \big(\frac{1}{N} \sum_{i=1}^{N} (P_\theta(y_i = LF_0) - r)^2 \big) \quad (3)$$

Few-shot metrics In the few-shot setting, we are concerned about the level to which the model is capturing the distribution given in the prompt. A core assumption here is that an ideal model would perfectly capture the uncertainty in the given distribution. We consider metrics at two levels of granularity to evaluate this behavior. The first metric we consider measures model performance at the level of the dataset. Intuitively, as we sweep across ratios $r \in R$, we expect the proportion of predicted LFs to match r. For example, when $r = 0.10$ (meaning that 10% of the prompt examples are LF_0 and 90% are LF_1) we would expect the model to produce LF_0 in roughly 10% of instances. Let y_i be the predicted LF for input instance x_i. Then the fewshot dataset metric FDM is given by Eq. (2). Intuitively, this measures the difference between the accuracy on each LF and the ratio of that LF; lower is better for FDM, which ranges from 1.0 to 0.0. The second metric measures model performance at the level of individual datapoints. If the model is capturing the distribution in the prompt, then the probability assigned to LF_0 should roughly match r, e.g. if $r = 0.10$, the model should assign $P(LF_0) \approx 0.10$. The few-shot instance metric FIM is given by Eq. (3). FIM resembles a Brier score (Brier et al., 1950) and measures the error between the predicted probability and the ratio; it also ranges from 1.0 to 0.0 and lower is better.

4 Experiment 1: Zero-shot parsing

For each ambiguity type, we construct a prompt that provides the ingredients for deriving both LFs. The order of the component sentences is shuffled to avoid biasing the model towards one interpretation or the other. Crucially, the prompt contains no examples of the types of sentences being tested. Fig. 2 shows an example for PP attachment.

Here, the model is given an example of how to parse transitive verbs, instruments, and possessives, each in isolation. To successfully generalize, the model has to overcome two challenges: first, it must compositionally generalize to compose the ingredients in the prompt into a valid derivation. Secondly, it must recognize the ambiguity and reflect both derivations in its output. For each ambiguity type, we test 200 examples.

4.1 Zero-shot results and analysis

In Fig. 3, we see that, when considering accuracy on each LF independently, smaller Codegen models (350M, 2B) struggle to predict either LF correctly. On some ambiguity types, larger Codegen models (6B, 16B) begin to predict one LF correctly. However, we rarely observe any models predicting both LFs correctly; one exception to this is conjunction ambiguity, where we see some models correctly predicting LF_0 for some examples and LF_1 for others. GPT-3.5 does well at predicting LF_1 for PP and scope ambiguities, and predicts both LFs for coreference and conjunctions. Interestingly, while Llama-13B is unable to correctly predict either LF for any of the ambiguities, Vicuna-13B (the instruction-tuned variant of Llama) is comparable to Codegen-16B. This suggests that instruction tuning helps the model predict LFs correctly (though not to capture ambiguity). Separately, we find that predicting FOL generally outperforms Lisp; for example, the Codegen-2B model on scope predicts LF_1 correctly 18% of the time when using FOL and only 11% when using Lisp; we report only FOL results moving forward.

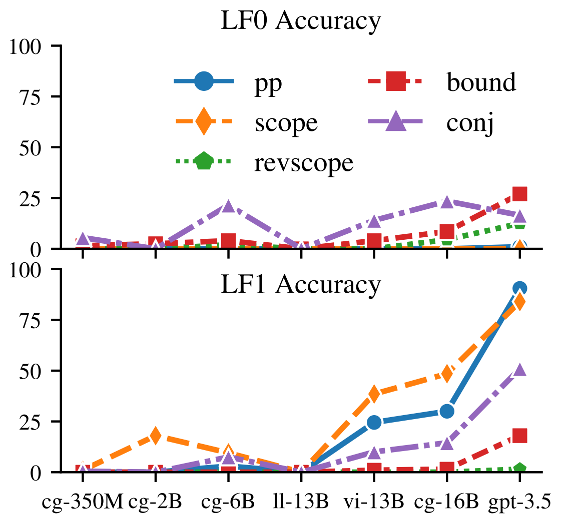


Figure 3: Zero-shot exact-match accuracy on ambiguity types. Cg = Codegen, Ll = Llama, Vi = Vicuna. Models increasing in size from left to right. Most models produce neither LF in the majority of instances.

These results are further underscored in Table 2 showing the ZM_5 values for all models. We see that all models tested perform very poorly on this metric, with most models scoring 0.0 in most settings. Qualitatively, we find that the model's top 5 outputs tend to include variations of the same LF. This finding aligns with the probability results seen in Fig. 4, where models tend to assign extreme probabilities to LFs on 3 out of 5 ambiguity types. Notable exceptions here are conjunction and bound pronoun types, where in Fig. 4 models assign closer to 0.5 probability to each parse; we see this also reflected in Fig. 3 and Table 2, where models predict both LF_0 and LF_1 correctly some of the time. As a whole, these results underscore the difficulty of the compositional task we have proposed; while some models are able to obtain high accuracy on one LF in isolation (GPT-3.5 predicts LF_1 for PP attachment and scope almost perfectly) no model is able to consistently predict both interpretations.

We can also ask whether token-level confidence reflects the ambiguity in the space of possible parses. Examining task-oriented semantic parsing models, Stengel-Eskin and Van Durme (2022) find that many models (including Codegen) are relatively well-calibrated at the token level, meaning their confidence aligns with their average accuracy. Taking token-level probabilities as confidence scores, we follow their analysis and ask whether models are well-calibrated w.r.t. alternative parses. Specifically, we compare the model's confidence on tokens at the points where the predicted and alternative parse diverge. This is visualized for each ambiguity type in Fig. 5. Here, we take the first correctly-predicted LF (either LF_0 or LF_1) from Codegen-16B, predicted via beam search with grammar-constrained decoding (not forced decoding). We overlay the confidence onto the token as the background color (darker is more confident). Below each predicted parse, we give the alternative parse. For scope and inverse scope, the model assigns low confidence to the quantifier tokens at the start of the formula, which are in the reverse order in the alternative parse. Similarly, the tokens involving the quantified variables have lower confidence. We also see low confidence around the area of divergence for conjunction. However, pronominal coreference and PP attachment lack such interpretable confidence changes. These results are promising: for some ambiguity types, the model's confidence reflects the alternative parse.

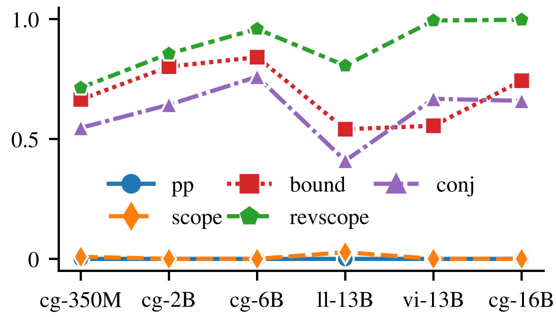


Figure 4: $P(LF_0)$ per model.

Model	PP	Scope	Revscope	Bound	Conj.
cg-350M	0.00	0.00	0.00	0.00	1.00
cg-2B	0.00	0.00	0.00	0.50	0.00
cg-6B	0.00	0.00	0.00	0.00	3.50
cg-16B	0.00	0.00	0.00	3.50	15.00
ll-13B	0.00	0.00	0.00	0.00	0.00
vi-13B	0.00	0.00	0.00	4.00	9.50
gpt-3.5	0.00	0.00	0.00	0.00	0.00

Table 2: ZM_5 for all models (cg = Codegen). Models generally fail to predict both LFs.

Hand-coded GPT-3.5 responses Due to OpenAI's API choices, constrained decoding is not possible with GPT-3.5, so we use open decoding, generating without the grammar constraints used for open-source models like Codegen. Despite this, GPT-3.5 produces more correct LFs than any other model. Notably, under open-ended decoding, we

```
scope (a): forall x . exists y . exists a . boy(x) AND pyjamas(y) AND observed(a) AND agent(a, x) AND patient(a, y)
scope (b): exists x . forall y . exists a . boy(y) AND pyjamas(x) AND observed(a) AND agent(a, y) AND patient(a, x)
revscope (a): exists x . forall y . exists a . boy(x) AND sweater(y) AND spied(a) AND agent(a, x) AND patient(a, y)
revscope (b): forall x . exists y . exists a . boy(y) AND sweater(x) AND spied(a) AND agent(a, y) AND patient(a, x)
bound (a): exists x . exists a . exists e . girl(x) AND spied(a) AND agent(a, x) AND patient(a, Mary) AND smiled(e) AND agent(e, x)
bound (b): exists x . exists a . exists e . girl(x) AND spied(a) AND agent(a, x) AND patient(a, Mary) AND smiled(e) AND agent(e, Mary)
conj (a): exists x . exists a . exists e . exists i . cat(x) AND ( ( drank(a) AND agent(a, x) ) OR ( ate(e) AND agent(e, x) ) ) AND ( played(i) AND agent(i, x)
conj (b): exists x . exists a . exists e . exists i . cat(x) AND ( ( drank(a) AND agent(a, x) ) OR ( ate(e) AND agent(e, x) AND played(i) AND agent(i, x) ) )
pp (a): exists x . exists a . camera(x) AND saw(a) AND agent(a, Watson) AND instrument(a, x) AND patient(a, Galileo)
pp (b): exists x . exists a . exists e . camera(x) AND saw(a) AND agent(a, Watson) AND patient(a, Galileo) AND have(e) AND agent(e, Galileo) AND patient(e, x)
```

Figure 5: Zero-shot per-token probability (darker is more probable) for each ambiguity type. Alternative parse given below each predicted parse. Token probability sometimes reflects divergences between the parses.

observed that the model occasionally is able to explicitly identify ambiguous inputs, sometimes producing statements like, *"This sentence is ambiguous and could have two different meanings depending on the intended emphasis."* In some cases, the model then lists two possible interpretations, often paired with corresponding LFs. One author examined all such responses from GPT-3.5 across all ambiguity types and annotated the number of times the model identified that an input was ambiguous, whether it produced one correct LF, whether it produced both LFs correctly, and whether, even if it did not produce correct LFs, its output reflected the meanings of both LFs. These results are tallied in Table 3. While GPT-3.5 rarely identifies ambiguity, for conjunction and inverse scope, when it identifies ambiguity it often captures both meanings (but not LFs).

Amb.	Id'd	1 LF	2 LFs	2 mng.
conj	11.0	1.5	3.0	6.5
pp	0.5	0.5	0.0	0.0
scope	2.0	0.0	0.0	0.0
revscope	5.0	0.0	0.0	4.0
bound	0.0	0.0	0.0	0.0

Table 3: Hand-coded results for open decoding with GPT-3.5. Columns denote whether ambiguity was identified (Id'd), whether one LF was correctly predicted (1 LF), whether both LFs were correctly predicted (2 LFs), and if not, whether the model at least provided text that indicated both meanings (2 mng.).

4.2 Human validation

Fig. 3 indicates that models tend to produce one interpretation or the other – when models have non-zero accuracy on one interpretation, they tend to have zero accuracy on the other. Psycholinguistic research suggests that people have preferred interpretations; for example, AnderBois et al. (2012) show human preferences in scope ambiguities are influenced by several factors, including linear order. In cases of scope ambiguity, people tend to prefer a "wide scope" reading ($\forall x \exists y$) while for inverse scope they prefer a "narrow scope" reading ($\exists x \forall y$). This is mirrored at an aggregate level in Fig. 4: models assign high probability to LF_0 (narrow scope) for inverse scope and high probability to LF_1 (wide scope) for scope.

AnderBois et al. (2012) and Dwivedi (2013) describe strong lexical effects in scope ambiguity, meaning that the choice of words in the example has an effect on the interpretation taken. In order to further examine how the models tested compare with these results, we annotate a subset of our validation examples with human interpretations and confidence scores. This allows us to compare model predictions to humans at an item-level rather than only at an aggregate level.

To ensure the validity of our results, we conducted a pilot paraphrasing task, where we asked Mechanical Turk annotators to verify that they were native English speakers and paraphrase a short passage. All annotators who attempted the pilot passed. In the main annotation task, annotators were asked to choose between interpretations and provide a confidence score on a sliding scale, following the EASL protocol (Sakaguchi and Van Durme, 2018). The continuous confidence scale has 3 ticks: *not confident*, *somewhat confident*, *very confident*, and annotators can slide the indicator anywhere along the scale. Since annotators are unlikely to know FOL or Lisp, instead of showing annotators LFs, each LF is paired with a statement that clearly indicates the interpretation (as in Table 1). For example, for a PP attachment example like, *"the boy saw the man with the telescope"*, the verbalized interpretations are *"the boy saw the man, who had/was wearing a telescope"* and *"the boy saw the man and used a telescope to do so"*. For each ambiguity type except conjunctions, we randomly select

20 examples from our development splits.[6] Each example is annotated by 3 annotators, and each annotator performed exactly 20 annotations in a sequence. Each sequence has 5 tuples of the 4 ambiguity types. Annotator payment corresponded to $\sim$ \$14.00 per hour.

To transform the annotators' confidence scores into probabilities, we first min-max normalize raw confidence scores, following past work using sliding bars (Vashishtha et al., 2019). This accounts for the fact that different annotators may use the slider differently. We then take the lowest confidence value to correspond to $p(LF_c) = 0.5$, where LF_c is the LF corresponding to the chosen interpretation. Intuitively, if $p(LF_c)$ were less than 0.5, the annotator would have chosen the other LF. The highest confidence value corresponds to $p(LF_c) = 1.0$.

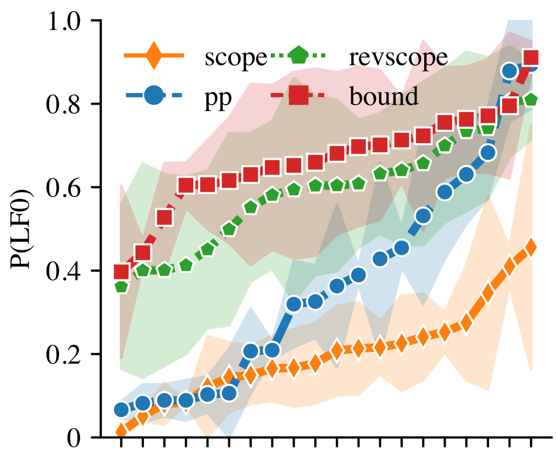


Figure 6: Per-example probabilities derived from human confidence scores on LFs. Examples are sorted by probability. For each ambiguity type, some examples are more or less compatible with each LF.

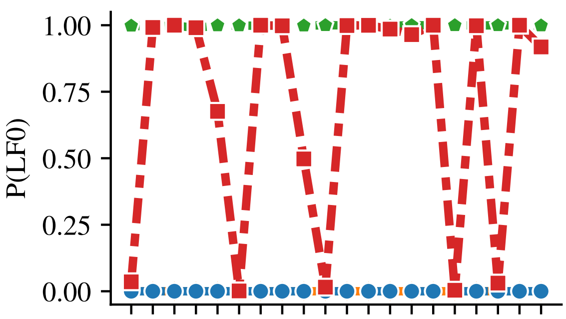


Figure 7: Model probabilities on the same set examples (ordered by the *human*-annotated $P(LF_0)$) as Fig. 6. Model probabilities do not vary according to the same lexical principles as human probabilities.

[6]Conjunction ambiguities were excluded due to difficulties in verbalizing their interpretations fluently.

We find that annotators disagree almost as often as they agree: 38 examples have disagreement while 42 have all 3 annotators agreeing. This indicates that our examples are highly ambiguous. Fig. 6 shows confidence scores (averaged across 3 annotators) for each item (sorted separately by mean confidence for each ambiguity type). On the whole, there are general preferences for all ambiguity types except PP attachment: bound and inverse scope tend to be matched to LF_0, and scope tends to be matched to LF_1. The scope findings align with some past findings (AnderBois et al., 2012; Caramazza et al., 1977), although other work has found linear order to have a negligible effect and pointed to additional factors influencing interpretations (Kurtzman and MacDonald, 1993; Dwivedi, 2013). For PP attachment, some examples are confidently matched to LF_0 and others to LF_1. Qualitatively, we find that visual verbs and nouns (e.g. *saw-telescope*, *observed-glasses*) are matched more to LF_1, where the PP is an instrument, while tactile verbs and nouns (e.g. *held-gloves*, *picked up-mittens*) yield a possessive interpretation.

In Fig. 7, we contrast the human results with the output of the Codegen-16B model. While the probabilities generally match in direction to the human annotations (with the exception of PP-attachment) we do not see the same kind of item-level sensitivity. For most PP, scope, and inverse scope examples, we see the model assigning all examples the same probability. For bound pronouns, we see more variation, with the model switching between LF_0 and LF_1; however, we still see fairly extreme predictions. Taken together, these results suggest that the model is not well-calibrated w.r.t. ambiguity at the item level.

5 Experiment 2: Few-shot parsing

Ambiguities may lead to similar inputs being paired with different logical forms. Increasingly, it is common to retrieve examples from a training set to compose a prompt for ICL. If that training set has ambiguity in it, it is likely that the retrieved prompt would contain conflicting examples, e.g. some examples pairing an utterance type with LF_0 and others pairing it with LF_1. In our few-shot experiments, we seek to fill this gap by investigating how model confidence and accuracy changes at different prompt ratios. Crucially, by investigating mixed prompts with ambiguous inputs, we are ensuring that the disagreement in the prompt is not

Model	PP		Scope		Revscope		Bound		Conj.	
–	FDM	FIM	FDM	FIM	FDM	FIM	FDM	FIM	FDM	FIM
codegen-350M	0.76	0.08	0.19	0.05	0.36	0.06	0.62	0.16	0.60	0.06
codegen-2B	0.51	0.09	0.19	0.03	0.18	0.03	0.48	0.10	0.39	0.05
codegen-6B	0.45	0.08	0.21	0.05	0.16	0.06	0.43	0.10	0.41	0.06
codegen-16B	0.35	0.08	0.20	0.03	0.21	0.03	0.38	0.10	0.27	0.06
llama-13B	1.00	0.06	1.00	0.04	1.00	0.04	1.00	0.06	1.00	0.05
vicuna-13B	0.50	0.08	0.17	0.05	0.20	0.06	0.26	0.07	0.35	0.09

Table 4: Few-shot metrics for all models (lower is better). FDM (Eq. (2)) measures the extent to which the model's accuracy on each LF across the whole dataset matches the percentage of that LF in the prompt. FIM (Eq. (3)) measures how well the model's uncertainty captures the uncertainty in the prompt.

due to simple mistakes; one could imagine a mixed prompt arising from noisy data, where instances are mislabeled. In such cases, a strong enough model may even learn to ignore mislabeled data in the prompt. However, in the case of ambiguity, there are multiple *legitimate* interpretations.

For each ambiguity type, we construct prompts by pairing sentences of the same type with LF_0 in some cases, and LF_1 in others; we run 100 examples per type, per ratio. Each prompt contains 10 input-LF pairs, and a different prompt is constructed for each test sentence. We vary the percentage of LF_0 sentences from 0 to 100 in increments of 10 (1 sentence at a time) and shuffle the prompt sentences to ensure that there is no positional bias.

5.1 Few-shot results and analysis

Fig. 8 shows the accuracy of models on LF_0 as we increase the percentage of LF_0 in the prompt for the Codegen-2B model. We see that for scope and inverse scope, the accuracy tracks almost perfectly with the percentage. For other ambiguities, the accuracy is correlated with the percentage but never reaches 100%. Table 4 shows the FIM and FDM scores for all models across all ambiguity types. Recall that FDM measures how well the

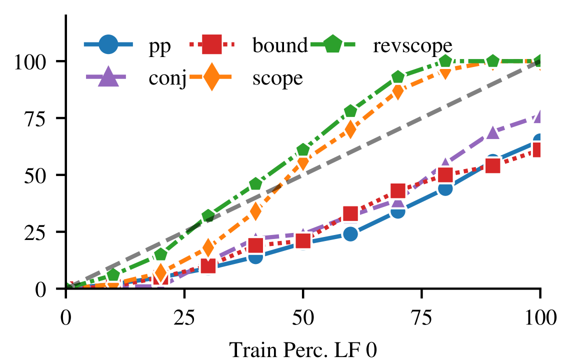


Figure 8: Fewshot accuracy increases according to the ratio of the LF in the prompt. LF_1 accuracy (not pictured) decreases accordingly.

model's accuracy aligns with the percentage of examples for each LF in the prompt: to obtain a lower FDM score, the model needs to predict each LF at roughly the rate that it is seen in the prompt. Note that a model that fails to predict either LF correctly will have a high FDM score. For example, because Llama-13B fails to predict any of the LFs correctly, it has an FDM score of 1.00 for all ambiguities. For Codegen models, FDM generally improves with model size on most ambiguity types. Overall, several models achieve fairly low FDM, especially on scope and inverse scope.

We also see that all FIM scores are relatively low. Because FIM computes the probability of each LF manually (i.e. using the gold LF to extract the sequence probablility) even models that have poor accuracy on both LFs can have a fairly low FIM. In other words, FIM presents the model with a forced choice between two parses, rather than evaluating the model's most probable generations like FDM and ZM do. For example, Llama-13B never produces a correct LF under beam-search decoding with constraints, but when probabilities are extracted with a forced decode of the gold LFs, its probabilities align fairly well with those in the prompt, and it often attains lower FIM scores than other models, including Vicuna. FIM remains fairly constant with model size: Codegen-16B is tied with smaller models on 4/5 ambiguity types.

Taken together, these results indicates that the models tested are surprisingly good at capturing the distribution in the prompt. The low FDM on some ambiguity types indicate that on the 100 examples as a whole, models like Codegen-16B and Vicuna-13B produce LF_0 roughly at the rate that it appears in the prompt. However, we see that low FIM does not imply low FDM, as to obtain low FDM the model must also be accurate.

Interestingly, the models seem to override the zero-shot tendencies seen in Fig. 4, where scope and inverse scope were strongly associated with one LF over the other. With direct evidence on how to parse scope sentences in the prompt – evidence we did not provide in the zero-shot setting – the model produces the interpretations seen in the prompt, and is especially close to the prompt distribution for scope and inverse scope. These results are promising: given that models seem to capture mixed prompts well, it could be that ambiguity poses less of a challenge in settings where such mixed prompts can be constructed, i.e. settings with ambiguity in the training data.

6 Related work

Ambiguity has been a longstanding topic of interest in linguistics and psycholinguistics. Past work has argued that it is a feature arising naturally from the trade-off between competing objectives (Zipf, 1949; Schutze, 1995; Piantadosi et al., 2012). However, providing a systematic overview of ambiguity in linguistics lies beyond the scope of this section.

Some work in NLP has focused on modeling ambiguity in visual contexts, where ambiguous questions and statements have been paired with images or videos depicting situations they refer to. Stengel-Eskin et al. (2023) introduce a dataset of linguistically ambiguous questions about images as well as a model for question disambiguation, and Futeral et al. (2022) examine ambiguous source sentences in machine translation, providing disambiguating contexts via images. More akin to our work, Berzak et al. (2015) introduce the LAVA corpus of syntatic, semantic, and pragmatic ambiguities paired with videos of scenes that are compatible with the multiple interpretations of an ambiguous statement. Follow-up work by Mehrabi et al. (2022) generates images for visualizable statements using text-to-image models. We use many of the same ambiguity types, but represent meaning with symbolic formulae instead of videos. This choice is motivated in part by the relative ease of checking the correctness of a logical formula over, for example, a generated image. Ambiguity has been studied in more general tasks such as question-answering (Min et al., 2020), natural language inference (NLI) (Liu et al., 2023), and coreference resolution (Yuan et al., 2023), where models have broadly been found lacking in their ability to resolve ambiguities. In parsing specifically, Rasmussen and Schuler (2020) introduce a lambda calculus dataset on 2,000 sentences of simple Wikipedia text, where roughly 50% contain quantifier scope ambiguity. We opt to use synthetic data instead, giving us greater control and allowing us to expand to more ambiguity types.

7 Discussion

In addition to semantic parsing's many applications in downstream tasks, it has often been used as a way to measure models' compositional generalization abilities through synthetic benchmarks like COGS (Kim and Linzen, 2020) and SCAN (Lake and Baroni, 2018). The ability to generalize systematically and compositionally to unseen combinations of symbols is a core component of human intelligence and language use (Fodor and Pylyshyn, 1988). Past efforts have generally made the simplifying assumption that there is a single correct LF for any given input, either implicitly (Lake and Baroni, 2018) or explicitly (Kim and Linzen, 2020, Appendix H). This assumption is not borne out in natural language, where ambiguous statements can have multiple compatible meanings. It is also violated in many common applications of semantic parsing models, such as text-to-code, where there are myriad ways of producing logically equivalent programs. Making this assumption can lead to semantic parsing tasks more akin to a syntactic parsing task – indeed, Rudinger and Van Durme (2014) found that one of the key features separating dependency-based syntax and event-based semantics was the ability to handle PP attachment ambiguity. In future work, we hope to improve on the challenging and novel compositional task we have proposed in Section 4, where all models struggle to capture both interpretations.

Section 5 offers a more hopeful takeaway: when ambiguity is present in the input, many models are able to capture the distribution of LFs. The caveat here is that for the ambiguity to be attested in the prompt, it must exist in the training data used to construct the prompt. This is discouraged by many annotation frameworks, which do not annotate items redundantly or exhaustively, meaning inputs are typically paired with a single output. Given that annotators often disagree on ambiguous examples (cf. Section 4.2) it is crucial to obtain multiple judgements per example. This has recently become more standard in other domains, such as NLI (Chen et al., 2020; Nie et al., 2020; Pavlick and Kwiatkowski, 2019). Even when items are annotated redundantly, disagreement is often discouraged and treated as noise. More recent work has begun to recognize that annotator disagreements can arise for valid reasons (Pavlick and Kwiatkowski, 2019) including ambiguity (Bhattacharya et al., 2019; Stengel-Eskin et al., 2023; Liu et al., 2023). Moving forward, we hope to see multi-way annotation with attention to annotator disagreement extended beyond single-label tasks like QA and NLI, to complex and sequential outputs like semantic parses.

7.1 Limitations

Firstly, we are limited by committing to a logical form with a particular grammar. To test the parsing abilities of models under ambiguity, we are forced to choose a form for our meaning representation (MR), which requires making abstractions and design choices that may be suboptimal. Motivated by Wu et al. (2023), who find that arbitrary choices in an MR's construction can hamper compositional generalization, we have tried to limit the number and difficulty of our choices, and mitigated them by offering two output formats.

It is also important to point out some key differences between the models we test and the relevant psycholingustic literature cited. Broadly, experiments on humans indicate that people maintain multiple interpretations during online processing, later settling on one interpretation (Lackner and Garrett, 1972; Rayner et al., 1983; Filik et al., 2004). Models generally do not process language in an online fashion, and receive input as text rather than audio. As such, our results should not be taken to reflect how people process language.

Methodologically, we are limited by our use of English only, and by our use of a fixed set of ambiguities. Our hope is that by introducing an extensible framework for creating templates and LFs, further languages and ambiguity types could be added in future work. Finally, we note that different decoding strategies might result in better results for the ZM metric. Our results indicate that models suffer from a lack of diversity in their outputs; this might in fact be compounded by constrained decoding, which limits the output space further. Alternatives to beam search that emphasize diversity may improve the ZM score.

8 Conclusion

By introducing a new benchmark for parsing under ambiguity, we are able to examine how modern semantic parsers handle cases where utterances can have multiple meanings. To this end, we introduced three new metrics for measuring the extent to which models capture the *distribution* of meanings. While we find that models struggle to compose symbols without explicit guidance, we also find that they are sensitive the ambiguity when given mixed prompts, suggesting that having ambiguity in the training data may be a sufficient condition for capturing it in the output. This motivates our call for explicitly allowing ambiguity in annotation interfaces.

References

Scott AnderBois, Adrian Brasoveanu, and Robert Henderson. 2012. The pragmatics of quantifier scope: A corpus study. In *Proceedings of Sinn und Bedeutung*, volume 16, pages 15–28.

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage ccg semantic parsing with amr. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.

Yevgeni Berzak, Andrei Barbu, Daniel Harari, Boris Katz, and Shimon Ullman. 2015. Do you see what I mean? Visual resolution of linguistic ambiguities. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1477–1487.

Nilavra Bhattacharya, Qing Li, and Danna Gurari. 2019. Why does a visual question have different answers? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4271–4280.

Glenn W Brier et al. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Alfonso Caramazza, Ellen Grober, Catherine Garvey, and Jack Yates. 1977. Comprehension of anaphoric pronouns. *Journal of verbal learning and verbal behavior*, 16(5):601–609.

Tongfei Chen, Zhengping Jiang, Adam Poliak, Keisuke Sakaguchi, and Benjamin Van Durme. 2020. Uncertain natural language inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8772–8779, Online. Association for Computational Linguistics.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Marco Damonte, Rahul Goel, and Tagyoung Chung. 2019. Practical semantic parsing for spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pages 16–23.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43, Berlin, Germany. Association for Computational Linguistics.

Veena D Dwivedi. 2013. Interpreting quantifier scope ambiguity: Evidence of heuristic first, algorithmic second processing. *PloS one*, 8(11):e81461.

Ruth Filik, Kevin B Paterson, and Simon P Liversedge. 2004. Processing doubly quantified sentences: Evidence from eye movements. *Psychonomic Bulletin & Review*, 11(5):953–959.

Jerry A Fodor and Zenon W Pylyshyn. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71.

Matthieu Futeral, Cordelia Schmid, Ivan Laptev, Benoît Sagot, and Rachel Bawden. 2022. Tackling ambiguity with images: Improved multimodal machine translation and contrastive evaluation. *arXiv preprint arXiv:2212.10140*.

Rohit J Kate, Yuk Wah Wong, Raymond J Mooney, et al. 2005. Learning to transform natural to formal languages. In *AAAI*, volume 5, pages 1062–1068.

Najoung Kim and Tal Linzen. 2020. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9087–9105, Online. Association for Computational Linguistics.

Howard S Kurtzman and Maryellen C MacDonald. 1993. Resolution of quantifier scope ambiguities. *Cognition*, 48(3):243–279.

James R Lackner and Merrill F Garrett. 1972. Resolving ambiguity: Effects of biasing context in the unattended ear. *Cognition*, 1(4):359–372.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah A Smith, and Yejin Choi. 2023. We're afraid language models aren't modeling ambiguity. *arXiv preprint arXiv:2304.14399*.

Ninareh Mehrabi, Palash Goyal, Apurv Verma, Jwala Dhamala, Varun Kumar, Qian Hu, Kai-Wei Chang, Richard Zemel, Aram Galstyan, and Rahul Gupta. 2022. Is the elephant flying? resolving ambiguities in text-to-image generative models. *arXiv preprint arXiv:2211.12503*.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo

Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, et al. 2023. Augmented language models: a survey. *arXiv preprint arXiv:2302.07842*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.

Richard Montague. 1970. English as a formal language. *Linguaggi nella Societa e nella Tecnica*.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. *arXiv preprint arXiv:2203.13474*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Aaron Parisi, Yao Zhao, and Noah Fiedel. 2022. Talm: Tool augmented language models. *arXiv preprint arXiv:2205.12255*.

Terence Parsons. 1990. *Events in the Semantics of English: A Study in Subatomic Semantics*. MIT Press.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Steven T Piantadosi, Harry Tily, and Edward Gibson. 2012. The communicative function of ambiguity in language. *Cognition*, 122(3):280–291.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners.

Nathan Ellis Rasmussen and William Schuler. 2020. A corpus of encyclopedia articles with logical forms. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*.

Keith Rayner, Marcia Carlson, and Lyn Frazier. 1983. The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. *Journal of verbal learning and verbal behavior*, 22(3):358–374.

Subhro Roy, Sam Thomson, Tongfei Chen, Richard Shin, Adam Pauls, Jason Eisner, and Benjamin Van Durme. 2022. BenchCLAMP: A benchmark for evaluating language models on semantic parsing. *arXiv preprint arXiv:2206.10668*.

Rachel Rudinger and Benjamin Van Durme. 2014. Is the stanford dependency representation semantic? In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 54–58.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Hinrich Schutze. 1995. *Ambiguity in language learning: Computational and cognitive models*. Stanford University.

Semantic Machines, Jacob Andreas, John Bufe, David Burkett, Charles Chen, Josh Clausman, Jean Crawford, Kate Crim, Jordan DeLoach, Leah Dorner, Jason Eisner, Hao Fang, Alan Guo, David Hall, Kristin Hayes, Kellie Hill, Diana

Ho, Wendy Iwaszuk, Smriti Jha, Dan Klein, Jayant Krishnamurthy, Theo Lanman, Percy Liang, Christopher H. Lin, Ilya Lintsbakh, Andy McGovern, Aleksandr Nisnevich, Adam Pauls, Dmitrij Petters, Brent Read, Dan Roth, Subhro Roy, Jesse Rusak, Beth Short, Div Slomin, Ben Snyder, Stephon Striplin, Yu Su, Zachary Tellman, Sam Thomson, Andrei Vorobev, Izabela Witoszko, Jason Wolfe, Abby Wray, Yuchen Zhang, and Alexander Zotov. 2020. Task-oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, 8:556–571.

Richard Shin, Christopher Lin, Sam Thomson, Charles Chen Jr, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7699–7715.

Richard Shin and Benjamin Van Durme. 2022. Few-shot semantic parsing with language models trained on code. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5417–5425.

Mark Steedman. 2011. *Taking scope: The natural semantics of quantifiers*. Mit Press.

Elias Stengel-Eskin, Jimena Guallar-Blasco, Yi Zhou, and Benjamin Van Durme. 2023. Why did the chicken cross the road? Rephrasing and analyzing ambiguous questions in vqa. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Elias Stengel-Eskin and Benjamin Van Durme. 2022. Calibrated interpretation: Confidence estimation in semantic parsing. *arXiv preprint arXiv:2211.07443*.

Elias Stengel-Eskin and Benjamin Van Durme. 2023. Did you mean...? Confidence-based trade-offs in semantic parsing. *arXiv preprint arXiv:2303.16857*.

Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. 2020. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3:25–55.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew Walter, Ashis Banerjee, Seth Teller, and Nicholas Roy. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 1507–1514.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

Ludwig Wittgenstein. 1921. *Tractatus Logico-Philosophicus*. Annalen der Naturphilosophie.

Zhengxuan Wu, Christopher D Manning, and Christopher Potts. 2023. ReCOGS: How incidental details of a logical form overshadow an evaluation of semantic interpretation. *arXiv preprint arXiv:2303.13716*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921.

Yuewei Yuan, Chaitanya Malaviya, and Mark Yatskar. 2023. AmbiCoref: Evaluating human and model sensitivity to ambiguous coreference.

In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 993–1000.

John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence*, pages 1050–1055.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. Broad-coverage semantic parsing as transduction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3786–3798, Hong Kong, China. Association for Computational Linguistics.

George Kingsley Zipf. 1949. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley Press.