Towards Collaborative Neural-Symbolic Graph Semantic Parsing via Uncertainty

Anonymous ACL submission

Abstract

Recent work in task-independent graph semantic parsing has shifted from grammar-based symbolic approaches to data-intensive neural approaches, and has shown strong performance on different types of meaning representations. However, it is still unclear that what are the limitations of these neural parsers, and whether these limitations can be compensated by collaborating with symbolic parsers. In this paper, we attempt to answer these questions by taking English Resource Grammar (ERG) parsing as a case study. Specifically, we first develop a state-of-the-art neural ERG parser, and then conduct detailed analyses on fine-grained linguistic phenomena. The results suggest that the neural parser's performance degrades significantly on long-tail examples, while the symbolic parser performs more robustly. To address this, we further propose a collaborative neural-symbolic semantic parsing framework. Specifically, we improve the beam search strategy by designing a decision criterion that incorporates both the model uncertainty about the testing data distribution and the prior knowledge from a symbolic parser. Experimental results show that this collaborative parsing framework can outperform the single neural parser and concretely improve the model's performance on long-tail examples.

1 Introduction

All things semantic are receiving heightened attention in recent years, and graph-structured semantic representations, which encode rich semantic information in the form of semantic graphs, have played an important role in natural language processing (Oepen et al., 2019). Parsing natural language sentences into these semantic graphs (e.g., Figure 1) has been extensively studied recently.

Work in this area has shifted from the symbolic (grammar-based) approach to the neural approach. Thanks to the flourishing of deep learning technologies, sequence-to-sequence (seq2seq) models have shown great performance on data sampled from the training distribution. These neural semantic parsers reduce the need for domain-specific assumptions, grammar learning, and more generally extensive feature engineering. However, this flexibility comes at a cost because it is no longer possible to interpret how the meaning composition is performed, given that the target outputs are structured and compositional graphs. Moreover, these neural models often generalize poorly to out-of-distribution samples especially for compositional generalization (Lake and Baroni, 2018), and previous work has shown that combining high-precision symbolic approaches with neural models can address this issue for task-oriented semantic parsing (Shaw et al., 2021; Kim, 2021; Cheng et al., 2019). However, this type of approach requires complex architecture engineering to incorporate the grammar formalism. The underlying grammar is usually simple, and was not tested beyond simple datasets such as SCAN (Lake and Baroni, 2018) or GEOQUERY (Zelle and Mooney, 1996). Therefore they are likely not sufficient for handling complex graph-based meaning representations.

Generally, there is no guarantee that all linguistic knowledge can be learned purely from the input training data. This is partly because natural language always has long tail phenomena that are hard to be captured by data-driven models. In this work, we aim to develop a more principled neural-symbolic approach for graph semantic parsing to address tail generalization, which leverages the information from an *a priori* grammar parser while maintaining the simplicity of neural seq2seq training. We take neural graph semantic parsing for English Resource Grammar (ERG) as our case study, which is a compositional semantic representation explicitly coupled with the syntactic structure. Compared to other graph-based meaning representations, ERG has high coverage of English text and strong transferability across domains

(Adolphs et al., 2008; Flickinger et al., 2010, 2012; Copestake and Flickinger, 2000; Ivanova et al., 2013), rendering itself has an attractive target formalism for automated semantic parsing.

However, symbolic parsing for ERG cannot guarantee full coverage for test samples due to the limitation of the grammar, e.g., incomplete categorization of lexical items and multi-word expression (Baldwin et al., 2004). Some data-driven parsers were proposed to address this issue (Buys and Blunsom, 2017; Chen et al., 2018, 2019; Cao et al., 2021). Their approaches either require pipeline settings or external tools such as aligners, part-of-speech taggers, and named entity recognizers, while the performance of the previous step or automatic tools will significantly impact the final results. This motivates us to build a pure end-to-end neural parser for ERG parsing that directly maps the input sentences to target graphs.

First, we present an end-to-end seq2seq model based on T5 (Raffel et al., 2020) that achieves the state-of-the-art results for ERG parsing. This model goes beyond the conventional multi-step predictions for node and edge in previous work, and does not have access to the underlying ERG or syntactic structures from which the annotation was originally derived.

Second, we conduct a series of fine-grained linguistic phenomena tests. By comparing the results of our neural parser and a symbolic parser ACE, we find that they exhibit complementary strengths on different types of linguistic phenomena. Particularly, the neural model suffers from long-tail examples in the test set. This motivates us to develop a collaborative parsing framework where the neural parser can collaborate with the grammar parser when it abstains from unreliable predictions.

The key lies in how to find an effective collaborative strategy. We address this by finding a solution in a Bayesian framework. The basic idea is to utilize uncertainty estimates of the neural parser to find unreliable predictions at the inference time. For those unreliable predictions, in addition to minimizing the loss functions conventionally, we will also minimize the distance with the prior distribution provided by the symbolic parser at the inference time.

Specifically, we improve the model's beam search strategy by designing a decision criterion that incorporates both the model uncertainty about the testing data distribution and the prior knowledge from a symbolic parser. This uncertainty-based collaborative parsing framework achieves stronger results compared to the original neural parser, especially in the tail linguistic categories. This also suggests that sometimes the limitation of the neural approach lies not necessarily in the model architecture or the training method, but in a sub-optimal inference procedure that naively maximize the *a posteriori* likelihood (e.g., the beam search) without questioning the reliability of the prediction. For two different uncertainty estimations used in our experiment, we also show that uncertainty with better calibration will lead to better collaborative results.

Our contribution are four-fold:

- We propose the first end-to-end model that achieves the state-of-the-art results for ERG parsing on the DeepBank benchmark. Specifically, we improve the best known SMATCH scores 95.67 to 96.54;
- We conduct a thorough analysis of the neural parser in terms of both generalization and uncertainty calibration. Specifically, we compared the predictive performance of neural parser with the state-of-the-art symbolic parser in various important linguistic categories, showing that both parsers exhibit complementary strengths, validating the potential to build a neural-symbolic parsing framework.
- We further conduct the first known investigation on the calibration quality of model uncertainty of a seq2seq neural parser, revealing that the choice of uncertainty estimator is critical for performance. Specifically, we found predictive margin, a simple uncertainty estimator, exhibits a surprisingly strong correlation with the model's test SMATCH score, while some more well-known uncertainty metrics (e.g., predictive entropy) are poorly calibrated.
- Leveraging the above two contributions, we propose a general-purpose collaborative neural-symbolic parsing framework that is inspired by the Bayesian formalism and incorporates model uncertainty. The resulting framework further boosted the performance of the neural model by 0.5 SMATCH score, while also robustly improving parser performance in various tail linguistic categories.

Reproducibility. We will release the code on Github[1].

[1] https://github.com/anonymous

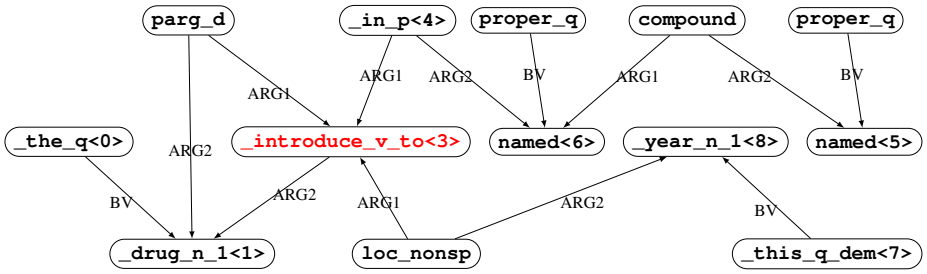


Figure 1: An example of semantic graph for ERG. Some nodes are surface concepts, meaning that they are related to a single lexical unit, e.g. `_introduce_v_to` (the number in the angle brackets indicates their token alignments in the sentence), while others are abstract concepts representing grammatical meanings, e.g. `compound` (multiword expression), `parg_d` (passive) and `loc_nonsp` (temporal). Color red indicates the root of this semantic graph. It also supports light-weight named entity recognition (e.g., "West Germany" is labeled as two `named` in the graph).

2 Background and Related Work

2.1 English Resource Grammar (ERG)

In this paper, we take the representations from English Resource Grammar (ERG; Flickinger et al., 2014) as our target meaning representations. A brief introduction to other meaning representations can be found in Appendix A. ERG is an open-source, domain-independent, linguistically precise, and broad-coverage grammar of English, which is rooted in the general linguistic theory of Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag, 1994). ERG can be presented into different types of annotation formalism (Copestake et al., 2005). In this work, we consider the Elementary Dependency Structure (EDS; Oepen and Lønning, 2006) which converts ERG into variable-free dependency graphs, and is more compact and interpretable when compared to other types of annotation schemes, e.g., DMRS (Buys and Blunsom, 2017; Chen et al., 2018).

Figure 1 shows an example graph. The semantic structure is a directed graph $G = \langle N, E \rangle$, where N denotes nodes labeled with semantic predicates/relations (e.g., `_drug_n_1, compound`), and E denotes edges labeled with semantic argument roles (e.g., `ARG1, ARG2`).

2.2 Parsing to Semantic Graphs

In this section, we present a summary of different parsing technologies for graph-based meaning representations, with a focus on English Resource Grammar (ERG).

Grammar-based approach. In this type of approach, a semantic graph is derived according to a set of lexical and syntactico-semantic rules. For ERG parsing, sentences are parsed to HPSG derivations consistent with ERG. The nodes in the derivation trees are feature structures, from which MRS is extracted through unification. However, this approach fails to parse sentences for which no valid derivation is found. It is implemented in the PET (Callmeier, 2000) and ACE[2] parser. Chen et al. (2018) also proposed a Synchronous Hyperedge Replace Grammar (SHRG) based parser by relating synchronous production rules to the syntacto-semantic composition process.

Factorization-based approach. This type of approach is inspired by graph-based dependency tree parsing (McDonald, 2006). A factorization-based parser explicitly models the target semantic structures by defining a score function that can evaluate the probability of any candidate graph. For ERG parsing, Cao et al. (2021) implemented a two-step pipeline architecture that identifies the concept nodes and dependencies by solving two optimization problems, where prediction of the first step is utilized as the input for the second step. Chen et al. (2019) presented a four-stage pipeline to incrementally construct an ERG graph, whose core idea is similar to previous work.

Transition-based approach. In these parsing systems, the meaning representations graph is generated via a series of actions, in a process that is very similar to dependency tree parsing (Yamada and Matsumoto, 2003; Nivre, 2008), with the difference being that the actions for graph parsing need to allow reentrancies. For ERG parsing, Buys and Blunsom (2017) proposed a neural encoder-decoder transition-based parser, which uses stack-based embedding features to predict graphs jointly with unlexicalized predicates and their token alignments.

[2] http://sweaglesw.org/linguistics/ace/

Composition-based approach. Following a principle of compositionality, a semantic graph can be viewed as the result of a derivation process, in which a set of lexical and syntactico-semantic rules are iteratively applied and evaluated. For ERG parsing, based on Chen et al. (2018), Chen et al. (2019) proposed a composition-based parser whose core engine is a graph rewriting system that explicitly explores the syntactico-semantic recursive derivations that are governed by a synchronous SHRG.

Translation-based approach. This type of approach is inspired by the success of seq2seq models which are the heart of modern Neural Machine Translation. A translation-based parser encodes and views a target semantic graph as a string from another language. In a broader context of graph semantic parsing, simply applying seq2seq models is not successful, in part because effective linearization (encoding graphs as linear sequences) and data sparsity were thought to pose significant challenges (Konstas et al., 2017). Alternatively, some specifically designed preprocessing procedures for vocabulary and entities can help to address these issues (Konstas et al., 2017; Peng et al., 2017). These preprocessing procedures are very specific to a certain type of meaning representation and are difficult to transfer to others. However, we show that by devising proper linearization and tokenization (Section 3.1), we can successfully transfer the ERG parsing problem into a translation problem, which can be solved by a state-of-the-art seq2seq model T5 (Raffel et al., 2020). This linearization and tokenization can be applied to any meaning representations.

2.3 Neural-Symbolic Semantic Parsing

While seq2seq models excel at handling natural language variation, they have been shown to struggle with out-of-distribution compositional generalization (Lake and Baroni, 2018; Shaw et al., 2021). This has motivated new specialized architectures with stronger inductive biases for the compositional generalization, especially for task-oriented semantic parsing like SCAN (Lake and Baroni, 2018) and GEOQUERY. Some examples include NQG-T5 (Shaw et al., 2021), a hybrid model combining a high-precision grammar-based approach with a pretrained seq2seq model; seq2seq learning with latent neural grammars (Kim, 2021); a neural semantic parser combining a generic tree-generation algorithm with domain-general grammar defined by the logical language (Cheng et al., 2019).

However, there are not so much progress regarding neural-symbolic parsing for graph meaning representations. Previous work has shown that the utility of context-free grammar for graph semantic parsing was somewhat disappointing (Peng et al., 2015; Peng and Gildea, 2016). This is mainly because the syntax-semantics interface encoded in those graph meaning representations is much more complicated than pure syntactic rules or logical formalism, and is difficult to be exploited in data-driven parsing architecture.

3 A Collaborative Neural-Symbolic Parsing Framework

In this section, we design and implement a new collaborative neural-symbolic parsing framework for ERG parsing. The framework takes the neural parser's uncertainty as a trigger to the collaborative process with the symbolic parser. This requires the neural parser to model uncertainty based on the optimization problem given observed sentence s:

$$\arg\max_{N,E} p(G = \langle N, E \rangle | s)$$

Previous data-driven work on ERG parsing either requires pipeline settings (predict nodes N and edges E separately) or external tools such as aligners, part-of-speech taggers and named entity recognizers, while the performance of the previous step or automatic tools will significantly impact the final results. In this paper, we aim to build an end-to-end seq2seq parser that directly maps the input sentences to target ERG graphs. Specifically, our parser is a translation-based parser that encodes and views the target semantic graphs as a string from another language. Despite the fact that seq2seq parsers for graph semantic parsing require specific engineering for meaning representations, we show that by devising proper linearization and tokenization (Section 3.1), we can successfully transfer the ERG parsing problem into a translation problem that can be solved by a state-of-the-art seq2seq model T5 (Raffel et al., 2020). This linearization and tokenization can be applied to any meaning representations. The experimental results show that our model improves significantly in comparison with the previously reported results (Table 1).

3.1 Linearization and Tokenization

Variable-free top-down linearization. A popular linearization approach is to linearize a directed

graph as the pre-order traversal of its spanning tree. Variants of this approach have been proposed for neural constituency parsing (Vinyals et al., 2015) and AMR parsing (Barzdins and Gosko, 2016; Peng et al., 2017). AMR (Banarescu et al., 2013) uses the PENMAN notation (Kasper, 1989), which is a serialization format for the directed, rooted graphs used to encode semantic dependencies. It uses parentheses to indicate nested structures. Since nodes in the graph get identifiers (initialized randomly) in PENMAN notation that can be referred to later to establish a reentrancy, e.g., `_drug_n_1` in Figure 1, and will confuse the model to learn the real meaningful mappings, we remove the identifiers and use star markers instead to indicate reentrancies. For example, our variable-free linearization for graphs in Figure 1 can be written as:

```
( _introduced_v_to
  :ARG2 ( _drug_n_1 * :BV-of ( _the_q ) )
  :ARG1-of ( parg_d :ARG2 ( _drug_n_1 * ) )
  :ARG1-of ( loc_nonsp
    :ARG2 ( _year_n_1 :BV-of ( _this_d_dem ) ) )
  :ARG1-of ( _in_p
    :ARG2 ( named
      :BV-of ( proper_q )
      :ARG1-of ( compound
        :ARG2 ( named :BV-of ( proper_q ) ) ) ) ) )
```

More details about the implementation of linearization can be found in Appendix B.

Compositionality-aware tokenization. Tokenization has always been seen as a non-trivial problem in Natural Language Processing (Liu et al., 2019). In the case of graph semantic parsing, it is still a controversial issue which unit is the most basic one that triggers conceptual meaning and semantic construction (Chen et al., 2019). While previous work can customize some off-the-shelf tokenizers to correspond closely to the ERG tokenization, there are still some discrepancies between the tokenization used by the system and ERG (Buys and Blunsom, 2017). Moreover, using customized tokenization means we need to pretrain our model from scratch, and this will cost lots of time and computation.

We address this issue by replacing the non-compositional part of ERG graphs with some non-tokenizable units in the T5 vocabulary. This will let the model learn the compositionality of ERG units by giving the signal of which type of units are tokenizable. More details can be found in Appendix C. This process is crucial since it not only reflects the original design of ERG vocabulary, but also dramatically reduces the sequence length of the output (around 16%). Additionally, it can be applied to any meaning representations by simply identifying the set of non-compositional units in semantic graphs.

3.2 A Decision-theoretic Framework for Collaborative Neural-Symbolic Parsing

It is known that the performance of a neural model tends to suffer in situations under-represented in training data, e.g., tail categories or out-of-domain (OOD) examples. Indeed, when analyzing our neural parser, we find the naive T5 parser's performance degrades significantly in the tail linguistic categories, while the symbolic parser performs more robustly (see Section 5.1). This motivates us to explore principled strategies to exploit the complementary strengths of both parsers. Specifically, we improve the beam search strategy by designing a decision criteria that incorporates both the model uncertainty about the testing data distribution and the prior information from a symbolic parser, thereby concretely improving the model performance at the tail.

Formally, consider a sequence learning problem where the input and target sequences $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X} \times \mathcal{Y}$ are generated from an underlying distribution $\mathcal{D} = p^*(\boldsymbol{y}|\boldsymbol{x})p^*(\boldsymbol{x})$. Given the neural parser $p(\boldsymbol{y}|\boldsymbol{x})$ trained on the in-domain examples $\boldsymbol{x} \in \mathcal{X}_{ind}$ and a symbolic parser prior $p_0(\boldsymbol{y}|\boldsymbol{x})$ that encodes *a priori* linguistic knowledge, our goal is to produce a decision criteria $\mathcal{L}(\boldsymbol{y}|\boldsymbol{x})$ for beam candidates that incorporates uncertainty form the neural model p and leverages information from the symbolic prior p_0. Under the Bayesian formulation, a (naive) decision criteria that incorporates the prior information p_0 is the negative log posterior likelihood (Bissiri et al., 2016): $\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}) = -log\, p(\boldsymbol{y}|\boldsymbol{x}) - log\, p_0(\boldsymbol{y}|\boldsymbol{x})$ where $p_0(\boldsymbol{y}|\boldsymbol{x}) \propto exp(-\frac{d(\boldsymbol{y},\boldsymbol{y}_0)}{\lambda})$ is the generalized Boltzmann distribution centered around the output of the symbolic parser $\boldsymbol{y}_0$. Here λ is the temperature parameter, and $d(y, y')$ is a suitable divergence metric for the space of ERG graphs, which we choose to be the SMATCH metric in this work (Cai and Knight, 2013). This leads to the naive decision criteria:

$$\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}) = -log\, p(\boldsymbol{y}|\boldsymbol{x}) + \frac{\text{SMATCH}(\boldsymbol{y}, \boldsymbol{y}_0)}{\lambda} \quad (1)$$

A caveat of the above criteria is not accounting for whether $\boldsymbol{x}$ is from the in-domain ($\mathcal{X}_{ind}$) or the out-of-domain ($\mathcal{X}/\mathcal{X}_{ind}$) regions of the input space. When $\boldsymbol{x}$ is in-domain, (1) can be too conservative since minimizing the beam score $-log\, p(\boldsymbol{y}|\boldsymbol{x})$

alone is known to generalize well in the i.i.d. situations. When $\boldsymbol{x}$ is from a region that is under-respresented in the training data, however, (1) can be overly optimistic since the neural model $p(\boldsymbol{y}|\boldsymbol{x})$ may generalize poorly in the under-represented regions, and a more prudent strategy is to revert to the prior by focus on minimizing $p_0(\boldsymbol{y}|\boldsymbol{x})$. To handle this challenge, we consider an improved criteria that accounts for model uncertainty:

$$\mathcal{L}(\boldsymbol{y}|\boldsymbol{x}) = \alpha(\boldsymbol{x}) * -log\, p(\boldsymbol{y}|\boldsymbol{x}) + (1-\alpha(\boldsymbol{x})) * \frac{\textsc{Smatch}(\boldsymbol{y}, \boldsymbol{y}_0)}{\lambda} \quad (2)$$

where $\alpha(\boldsymbol{x}) = \text{sigmoid}(-\frac{1}{T} * (\mathcal{H}(\boldsymbol{x}) - b))$ is a monotonic transformation of model uncertainty $\mathcal{H}(\boldsymbol{x})$ which is known as the Platt calibration (Platt et al., 1999). whose parameters (T, b) can be estimated using a small amount of validation data. As shown, depending on the value of $\mathcal{H}(x)$, the proposed criteria (2) approaches the original beam score $-log\, p(\boldsymbol{y}|\boldsymbol{x})$ when the model is confident, and reverts to the prior likelihood $-log\, p_0(\boldsymbol{y}|\boldsymbol{x})$ when the model is uncertain and $\mathcal{H}$ is high.

A comment regarding the choice of $\mathcal{H}(\boldsymbol{x})$ is in order. For the proposed criteria (2) to perform robustly in practice, the uncertainty estimator $\mathcal{H}(\boldsymbol{x})$ should be *well calibrated*, i.e., the magnitude of $\mathcal{H}$ is indicative of the model's predictive error. To this end, we notice that a reliable uncertainty measure for sequence prediction tasks is still an open research challenge (Malinin and Gales, 2020). In this work, we experiment with several well-known estimators of model uncertainty:

Margin probability. The simplest estimator for model uncertainty is the predictive margin, i.e., the difference in probability of the top 1 prediction minus the likelihood of the top 2 prediction based on the beam score:

$$\mathcal{H}_{\text{margin}}(p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})) = p(\boldsymbol{y}^{(1)}|\boldsymbol{x}, \mathcal{D}) - p(\boldsymbol{y}^{(2)}|\boldsymbol{x}, \mathcal{D})$$

Weighted entropy. Considering that our model uses beam-search for inference, and with regards to the Monte-Carlo estimators, beam-search can be interpreted as a form of importance-sampling which yields hypotheses from high-probability regions of the hypothesis space. We can estimate uncertainty which is importance-weighted in proportion to $p(\boldsymbol{y}^{(b)}|\boldsymbol{x}, \mathcal{D})$ such that

$$\mathcal{H}_{\text{entropy}}(p(\boldsymbol{y}|\boldsymbol{x}, \mathcal{D})) = -\sum_{b=1}^{B} \frac{\pi_b}{L^{(b)}} \ln p(\boldsymbol{y}^{(b)}|\boldsymbol{x}, \mathcal{D}),$$

where $\pi_b = \frac{p(\boldsymbol{y}^{(b)}|\boldsymbol{x}, \mathcal{D})}{\sum_k^B p(\boldsymbol{y}^{(k)}|\boldsymbol{x}, \mathcal{D})}$ is the estimated importance weight for each beam candidate (Malinin and Gales, 2020).

In our experiment, we investigate the calibration of the above uncertainty estimations (Section 5.2), and experiment with their respective efficacy in improving the collaborative parsing system's predictive performance (Table 2).

4 Experiments

Dataset. We conduct experiments on DeepBank v1.1 that correspond to ERG version 1214, and adopt the standard data split. The Pydelphin[3] library is leveraged to extract EDS graphs and transfer them into PENMAN format.

Implementation Details. T5 (Raffel et al., 2020) is a pre-trained sequence-to-sequence Transformer model that has been widely used in many NLP applications. We use the open-sourced T5X [4], which is a new and improved implementation of T5 codebase in JAX and Flax. Specifically, we use the official pretrained T5-Large (770 million parameters) and finetuned it on DeepBank. Despite the general fact that larger model size will lead to better performance on finetuning for some tasks, our empirical results show that adopting model sizes larger than T5-Large will not lead to further gain for ERG parsing.

For the collaborative neural-symbolic parsing, we set the beam size to 5, which means our combined predictions will be selected from the top 5 predictions produced by the model. We set λ and T for the monotonic transformation $\alpha(\boldsymbol{x})$ to 0.1 and 0.1.

Evaluation Metrics. For evaluation, following previous work, we adopt the SMATCH metric (Cai and Knight, 2013), which was originally proposed for evaluating AMR graphs. It measures graph overlap, but does not rely on sentence alignments to determine the correspondences between graph nodes. Specifically, SMATCH is computed by performing inference over graph alignments to estimate the maximum F1-score obtainable from a one-to-one matching between the predicted and gold graph nodes. This is also ideal for measuring the divergence between predicted and prior graphs in our collaborative framework.

[3] https://github.com/delph-in/pydelphin

[4] https://github.com/google-research/t5x

	Node			Edge			Graph
	P	R	F	P	R	F	SMATCH
w/o preprocess	96.29	91.72	93.95	93.86	88.66	91.19	92.57
w/ preprocess	97.67	96.93	97.30	97.71	96.85	95.81	96.54

Table 1: Comparision of precision, recall, and F1-score for node and edge prediction and SMATCH scores on the test set under the settings of with/without tokenization preprocessing.

Impact of Tokenization. To validate the effectiveness of our proposed tokenization process, we report the performance of node and edge prediction and the SMATCH scores with and without the process on the test set in Table 1, which indicates that after this process, the SMATCH score is improved by 4.29% on the test set. We can find that the recall score for node prediction has significant improvement, and this is because that the sequence without tokenization preprocessing will lead to longer sequence length, and many output graphs have reached the max decoding sequence length and thus are incomplete.

Model	Node	Edge	SMATCH
ACE[5]	93.18	88.76	90.94
Transition-based (Buys and Blunsom, 2017)	89.06	84.96	87.00
SHRG-based (Chen et al., 2018)	94.51	87.29	90.86
Composition-based (Chen et al., 2019)	95.63	91.43	93.56
Factorization-based (Chen et al., 2019)	97.28	94.03	95.67
Factorization-based (Cao et al., 2021)	96.42	93.73	95.05
ACE-T5 (following Shaw et al. (2021))	93.46	89.19	91.30
Translation-based (Ours)	97.30	95.81	96.54
Collaborative w/ margin probability	**97.64**	**96.41**	**97.01**
Collaborative w/ weighted entropy	97.27	96.14	96.70

Table 2: F1 score for node and edge predictions and the SMATCH scores on the test set.

Comparison w/ Existing Parsers. We compared our parser with the ERG-guided ACE parser and other data-driven parsers in Table 2. The baseline models also include a similar practice with Shaw et al. (2021), which takes T5 as a backup for grammar-based parser. Our model outperforms all previous work, and achieves a SMATCH score of 96.54, which is significantly better than existing parsers. After applying the collaborative parsing framework, we further improve the parser's performance to 97.01.

[5]The results for ACE are lower than those reported in previous work, which are originally from Buys and Blunsom (2017). We use the same ACE parser and we have confirmed with other authors that those higher results are not reproducible. As the ACE parser fails to parse some of the sentences (more than 1%), we only evaluate sentences that are successfully parsed by ACE.

5 In-depth Analyses

5.1 Fine-grained Linguistic Evaluation

Though performs better than symbolic parser, we find that actually neural and symbolic parsers yield different distributions on the test set (see Appendix D for details). This has motivated us to dive deeply into more fine-grained evaluation for our models.

ERG provides different levels of linguistic information that is beneficial to many NLP tasks, e.g., named entity recognition, semantic role labeling, and coreference. This rich linguistic annotation can help us quantify different types of errors the model makes. We reported the detailed evaluation results in Table 3. Specifically, we consider:

Lexical construction. ERG uses the abstract node `compound` to denote compound words. The edge labeled with `ARG1` refers to the root of the compound word, and thus can help to further distinguish the type of the compound into (1) nominal with normalization, e.g., "flag burning"; (2) nominal with noun, e.g., "pilot union"; (3) verbal, e.g., "state-owned"; (4) named entities, e.g., "West Germany".

Argument structure. In ERG, there are different types of core predicates in argument structures, specifically, verbs, nouns and adjectives. We also categorize verb in to basic verb (e.g., `_look_v_1`) and verb particle constructions (e.g., `_look_v_up`). The verb particle construction is handled semantically by having the verb contribute a relation particular to the combination.

Coreference. ERG resolves sentence-level coreference, i.e., if the sentence referring to the same entity, the entity will be an argument for all the nodes that it is an argument of, e.g., in the sentence, "What we want to do is take a more aggressive stance", the predicates "want" (`_want_v_1`) and "take" (`_take_v_1`) share the same agent "we" (`pron`). As discussed before, this can be presented as reentrancies in the ERG graph, we notice that one important type of reentrancies is the passive construction (e.g., `parg_d` in Figure 1), so we also report evaluation on passive construction in Table 3.

As shown, the T5 parser performs much better than ACE, especially for compound recognition. This indicates that local semantic information such as compound constructions or named entities can be easily captured by those pretrained embedding-based models. For argument structure, though performs better than ACE in most cases, the T5 parser

Type	#	ACE	T5	Collob.
Compound	2,266	80.58	**90.46**	90.36
Nominal *w/ nominalization*	22	85.71	**89.66**	82.76
Nominal *w/ noun*	1,044	85.28	90.96	**91.42**
Verbal	23	75.00	77.27	**81.82**
Named entity	1,153	82.92	**91.36**	90.40
Argument structure	7,108	86.98	90.68	**91.66**
Total verb	4,176	85.34	89.75	**90.50**
Basic verb	2,356	85.79	89.97	**90.90**
ARG1	1,683	90.25	93.40	**93.94**
ARG2	1,995	90.48	92.95	**93.79**
ARG3	195	**85.63**	83.08	84.62
Verb-particle	1,761	84.69	89.47	**90.00**
ARG1	1,545	89.57	93.50	**94.05**
ARG2	923	86.27	91.10	**91.26**
ARG3	122	87.88	86.75	**88.08**
Total noun	394	92.41	91.84	**92.63**
Total adjective	2,538	89.05	92.09	**93.25**
Reentrancy	2,343	77.29	87.88	**88.43**
passive	522	84.89	91.54	**92.72**

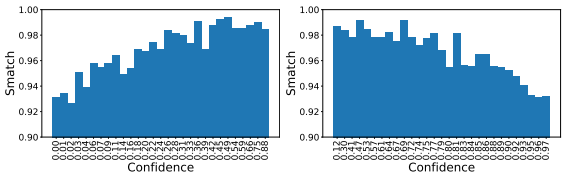
Table 3: Comparing ACE, T5 parsers and collaborative parsing (Collab.) on fine-grained linguistic categories. All scores are reported in accuracy. The underlined denotes the best in ACE and T5, and the bold denotes the best in ACE, T5 and Collab.

still has relatively low accuracy for ARG3 and noun structure recognition. This is mainly due to their relatively low frequency in the training set (1.94% for ARG3 and 5.54% for noun argument structures).

Our analysis in this section is consistent with previous work: the T5 parser, similar to many other neural parsers, is fragile to certain tail instances that do not have sufficient representation in the training data. We also further report the evaluation results for our collaborative neural-semantic parsing framework (Collab.), where we can see that it brings improvement for the issues above, which validates the effectiveness of the collaborative framework.

5.2 Model Calibration

A common approach to evaluate a model's uncertainty quality is to measure its *calibration* performance, i.e., whether the model's predictive uncertainty is indicative of the predictive error (Guo et al., 2017). To understand how well the T5 parser's neural uncertainty correlates with its prediction reliability, we plot the diagrams for the model's confidence verses SMATCH scores on the test set in Figure 2. As shown, comparing to the weighted entropy, margin probability is qualitatively much better calibrated. [6] Correspondingly, Table 2 shows that the collaborative result using margin probability yields much strongly performance, confirming the connection between a uncertainty model's calibration quality and its effectiveness is collaborative prediction (Kivlichan et al., 2021).



(a) Margin Probability (b) Weighted Entropy

Figure 2: Diagrams for the model's confidence verses SMATCH scores on the test set. Each bin contains 50 examples.

6 Conclusions and Future Work

In this paper, we present a simple, uncertainty-based approach to collaborative neural-symbolic parsing for graph-based meaning representations. In contrary to the prior neural-symbolic approaches, we maintain the simplicity of the seq2seq training, and design a decision-theoretic inference criteria for beam candidate selection, incorporating model uncertainty and prior knowledge from an existing symbolic parser.

Remarkably, despite the simplicity of the method, our approach strongly outperform all the previously-known approach on the DeepBank benchmark (Table 2), and attains strong performance even in the tail linguistic categories (Table 3). Our study revealed that the commonly observed weakness of the neural model may root from a sub-optimal inference procedure. Therefore, developing a more calibrated neural semantic parser and developing principled inference procedure may be a fruitful avenue for addressing the generalization issues of neural parsers.

In the future, we plan to apply this approach to a broader range of graph meaning representations, e.g., AMR (Banarescu et al., 2013) and UCCA (Abend and Rappoport, 2013), and build a more advanced uncertainty estimation approach to quantify model uncertainty about sub-components of the graph, thereby allowing more fine-grained integration between neural prediction and symbolic derivations.

[6] We hypothesize that the inferior performance of entropy is due to the well-known "length bias" (Yang et al., 2018), i.e., shorter predictions tend to have higher beam score, which also tend to have lower SMATCH score

Ethical Consideration

This paper focused on collaborative neural-symbolic semantic parsing for the English Resource Grammar (ERG). Our architecture are built based on open-source models and datasets (all available online). We do not anticipate any major ethical concerns.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria. Association for Computational Linguistics.

Peter Adolphs, Stephan Oepen, Ulrich Callmeier, Berthold Crysmann, Dan Flickinger, and Bernd Kiefer. 2008. Some fine points of hybrid natural language parsing. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Timothy Baldwin, Emily M. Bender, Dan Flickinger, Ara Kim, and Stephan Oepen. 2004. Road-testing the English Resource Grammar over the British National Corpus. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Guntis Barzdins and Didzis Gosko. 2016. RIGA at SemEval-2016 task 8: Impact of Smatch extensions and character-level neural translation on AMR parsing accuracy. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1143–1147, San Diego, California. Association for Computational Linguistics.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. 2016. A general framework for updating belief distributions. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 78(5):1103.

Johan Bos, Stephen Clark, Mark Steedman, James R Curran, and Julia Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1240–1246.

Jan Buys and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1215–1226, Vancouver, Canada. Association for Computational Linguistics.

Shu Cai and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 748–752, Sofia, Bulgaria. Association for Computational Linguistics.

Ulrich Callmeier. 2000. Pet–a platform for experimentation with efficient hpsg processing techniques. *Natural Language Engineering*, 6(1):99–107.

Junjie Cao, Zi Lin, Weiwei Sun, and Xiaojun Wan. 2021. Comparing knowledge-intensive and data-intensive models for english resource semantic parsing. *Computational Linguistics*, 47(1):43–68.

Yufei Chen, Weiwei Sun, and Xiaojun Wan. 2018. Accurate SHRG-based semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 408–418, Melbourne, Australia. Association for Computational Linguistics.

Yufei Chen, Yajie Ye, and Weiwei Sun. 2019. Peking at MRP 2019: Factorization- and composition-based parsing for elementary dependency structures. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 166–176, Hong Kong. Association for Computational Linguistics.

Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2019. Learning an executable neural semantic parser. *Computational Linguistics*, 45(1):59–94.

Ann Copestake. 2009. **Invited Talk:** slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 1–9, Athens, Greece. Association for Computational Linguistics.

Ann Copestake and Dan Flickinger. 2000. An open source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Ann Copestake, Dan Flickinger, Carl Pollard, and Ivan A Sag. 2005. Minimal recursion semantics: An introduction. *Research on language and computation*, 3(2):281–332.

Dan Flickinger, Emily M. Bender, and Stephan Oepen. 2014. Towards an encyclopedia of compositional semantics: Documenting the interface of the English Resource Grammar. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 875–881, Reykjavik, Iceland. European Language Resources Association (ELRA).

Dan Flickinger, Stephan Oepen, and Gisle Ytrestøl. 2010. WikiWoods: Syntacto-semantic annotation for English Wikipedia. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Dan Flickinger, Yi Zhang, and Valia Kordoni. 2012. Deepbank. a dynamically annotated treebank of the wall street journal. In *Proceedings of the 11th International Workshop on Treebanks and Linguistic Theories*, pages 85–96.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Angelina Ivanova, Stephan Oepen, Rebecca Dridan, Dan Flickinger, and Lilja Øvrelid. 2013. On different approaches to syntactic analysis into bi-lexical dependencies. an empirical comparison of direct, PCFG-based, and HPSG-based parsers. In *Proceedings of the 13th International Conference on Parsing Technologies (IWPT 2013)*, pages 63–72, Nara, Japan. Assocation for Computational Linguistics.

Angelina Ivanova, Stephan Oepen, Lilja Øvrelid, and Dan Flickinger. 2012. Who did what to whom? a contrastive study of syntacto-semantic dependencies. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 2–11, Jeju, Republic of Korea. Association for Computational Linguistics.

Robert T Kasper. 1989. A flexible interface for linking applications to penman's sentence generator. In *Speech and Natural Language: Proceedings of a Workshop Held at Philadelphia, Pennsylvania, February 21-23, 1989*.

Yoon Kim. 2021. Sequence-to-sequence learning with latent neural grammars. *arXiv preprint arXiv:2109.01135*.

Ian Kivlichan, Zi Lin, Jeremiah Liu, and Lucy Vasserman. 2021. Measuring and improving model-moderator collaboration using uncertainty estimation. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 36–53, Online. Association for Computational Linguistics.

Ioannis Konstas, Srinivasan Iyer, Mark Yatskar, Yejin Choi, and Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–157, Vancouver, Canada. Association for Computational Linguistics.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning*, pages 2873–2882. PMLR.

Liyuan Liu, Zihan Wang, Jingbo Shang, Dandong Yin, Heng Ji, Xiang Ren, Shaowen Wang, and Jiawei Han. 2019. Raw-to-end name entity recognition in social media. *arXiv preprint arXiv:1908.05344*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Ryan McDonald. 2006. *Discriminative learning and spanning tree algorithms for dependency parsing*. University of Pennsylvania Philadelphia.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistics*, 34(4):513–553.

Stephan Oepen, Omri Abend, Jan Hajic, Daniel Hershcovich, Marco Kuhlmann, Tim O'Gorman, Nianwen Xue, Jayeol Chun, Milan Straka, and Zdenka Uresova. 2019. MRP 2019: Cross-framework meaning representation parsing. In *Proceedings of the Shared Task on Cross-Framework Meaning Representation Parsing at the 2019 Conference on Natural Language Learning*, pages 1–27, Hong Kong. Association for Computational Linguistics.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Silvie Cinková, Dan Flickinger, Jan Hajič, and Zdeňka Urešová. 2015. SemEval 2015 task 18: Broad-coverage semantic dependency parsing. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 915–926, Denver, Colorado. Association for Computational Linguistics.

Stephan Oepen and Jan Tore Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Xiaochang Peng and Daniel Gildea. 2016. UofR at SemEval-2016 task 8: Learning synchronous hyperedge replacement grammar for AMR parsing. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1185–1189, San Diego, California. Association for Computational Linguistics.

Xiaochang Peng, Linfeng Song, and Daniel Gildea. 2015. A synchronous hyperedge replacement grammar based approach for AMR parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 32–41, Beijing, China. Association for Computational Linguistics.

Xiaochang Peng, Chuan Wang, Daniel Gildea, and Nianwen Xue. 2017. Addressing the data sparsity issue in neural AMR parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 366–375, Valencia, Spain. Association for Computational Linguistics.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Carl Pollard and Ivan A Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Peter Shaw, Ming-Wei Chang, Panupong Pasupat, and Kristina Toutanova. 2021. Compositional generalization and natural language variation: Can a semantic parsing approach handle both? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 922–938, Online. Association for Computational Linguistics.

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. *Advances in neural information processing systems*, 28:2773–2781.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of the Eighth International Conference on Parsing Technologies*, pages 195–206, Nancy, France.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3054–3059, Brussels, Belgium. Association for Computational Linguistics.

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.

A Graph-based Meaning Representation

Considerable NLP research has been devoted to the transformation of natural language utterances into a desired linguistically motivated semantic representation. Such a representation can be understood as a class of discrete structures that describe lexical, syntactic, semantic, pragmatic, as well as many other aspects of the phenomenon of human language. In this domain, graph-based representations provide a light-weight yet effective way to encode rich semantic information of natural language sentences and have been receiving heightened attention in recent years. Popular frameworks under this umbrella includes Bi-lexical Semantic Dependency Graphs (SDG; Bos et al., 2004; Ivanova et al., 2012; Oepen et al., 2015), Abstract Meaning Representation (AMR; Banarescu et al., 2013), Graph-based Representations for English Resource Grammar (ERG; Oepen and Lønning, 2006; Copestake, 2009), and Universal Conceptual Cognitive Annotation (UCCA; Abend and Rappoport, 2013).

B Detailed Implementation of Linearization

The original PENMAN styled linearization for graph in Figure 1 can be written as:

```
(x0 / _introduced_v_to
  :ARG2 (x1 / _drug_n_1
    :BV-of (x2 / _the_q))
  :ARG1-of (e0 / parg_d
    :ARG2 x1)
  :ARG1-of (e1 / loc_nonsp
    :ARG2 (x3 / _year_n_1
      :BV-of (x4 / _this_d_dem)))
  :ARG1-of (x5 / _in_p
    :ARG2 (e2 / named
      :BV-of (e3 / proper_q)
      :ARG1-of (e4 / compound
        :ARG2 (e5 / named
          :BV-of (e6 / proper_q))))))
```

The term `-of` is used for reversing the edge direction for graph traversing. Nodes in the graph get identifiers (e.g., `x0`, `e0`), which can be referred to later to establish a reentrancy, e.g., the node `_drug_n_1` serves as `ARG2` of `_introduced_v_to` and `ARG2` of `parg_d` at the same time, so the identifier `x_1` appears twice in the notation. However, in our settings, these identifiers can be randomly set to any unique symbols, which will confuse the model to learn the real meaningful mappings. To tackle this issue and create a variable-free version of the PENMAN notation, we replace these identifiers with star markers to indicate reentrancy, e.g., replacing `x1` with `_drug_n_1 *`.

The rewriting process can be done by Algorithm 1. It is noted that there can be more than one reentrancy in the graph, and we use different numbers of star marks to indicate this (line 10 in Algorithm 1).

To illustrate more about reentrancies, we consider two different types of cases:

Algorithm 1 Variable-free PENMAN rewriting

Input: $G = \langle N, E \rangle$ is the EDS graph
Output: Variable-free PENMAN notations of G

$R \leftarrow \emptyset$ ▷ reenrancy set
$n_R \leftarrow 0$ ▷ number of of reenrancies
for $n \in N$ **do**
  if $\text{child}(n) \cap \text{child}(\text{parent}(n)) \neq \emptyset$ **then**
    $R' \leftarrow \text{child}(n) \cap \text{child}(\text{parent}(n))$
    $R \leftarrow R \cup R'$
  end if
end for
for $r \in R$ **do**
  $G \leftarrow \text{rewrite}(\text{G}, \text{r}, \text{r} +' *' \times (\text{n}_\text{R} + 1))$
  $n_R \leftarrow n_R + 1$
end for
return $\text{PENMAN}(G)$

(1) For cases where the second reentrancy still points back to the first `_drug_n_1`, e.g., in the sentence "the drug was introduced and used this year", the node will still be marked as `_drug_n_1 *`.

(2) For cases where the second reentrancy refers to another token span in the sentences, e.g., in the sentence "The drug was introduced this year, and another drug will be introduced next year", the second node reentrancy will be marked as `_drug_n_1 **`.

In other words, the max number of star markers $*$ indicates the total number of different reentrancies in the sentences. This will not confuse the model to do the reentrancy prediction as it can always refer to how many reentrancies have been predicted in the previous sequences.

C Details about Tokenization

ERG makes an explicit distinction between nodes with surface relations (prefixed by an underscore), and with grammatical meanings. The former, called the surface node, consists of a lemma followed by a coarse part-of-speech tag and an optional sense label. For example, for the node `_drug_n_1` in Figure 1, the surface lemma is drug (`_drug`), the part-of-speech is noun (`_n`), and `_1` here specifies that it is the first sense under the noun "drug". The later, called the abstract node, is used to represent the semantic contribution of grammatical constructions or more specialized lexical entries, e.g., `parg_d` (for passive), `proper_q` (for quantification of proper words), `compound` (for compound words), and `named` (for named entities).

It is noted that the set of abstract concepts and edges are fixed and relatively small (88 for abstract nodes and 11 for edges in the training set), while the surface nodes have high productivity, i.e., many different lemmas can fit into some fixed patterns such as `_n_1` and `_v_to`. Therefore, we rewrite those fixed abstract, concepts surface patterns and edges into some non-tokenizable tokens in the T5 vocabulary to inform the model that these units are non-compositional in ERG graphs.

D Distributions of the T5 and ACE Parsers

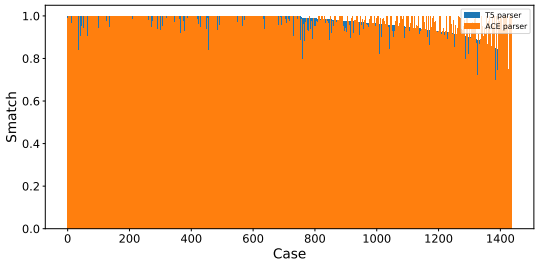


Figure 3: SMATCH scores of the T5 and ACE parsers across test examples