Diagnosing Transformers in Task-Oriented Semantic Parsing

Shrey Desai **Ahmed Aly**
Facebook
{shreyd, ahhegazy}@fb.com

Abstract

Modern task-oriented semantic parsing approaches typically use seq2seq transformers to map textual utterances to semantic frames comprised of intents and slots. While these models are empirically strong, their specific strengths and weaknesses have largely remained unexplored. In this work, we study BART (Lewis et al., 2020) and XLM-R (Conneau et al., 2020), two state-of-the-art parsers, across both monolingual and multilingual settings. Our experiments yield several key results: transformer-based parsers struggle not only with disambiguating intents/slots, but surprisingly also with producing syntactically-valid frames. Though pre-training imbues transformers with syntactic inductive biases, we find the ambiguity of copying utterance spans into frames often leads to tree invalidity, indicating span extraction is a major bottleneck for current parsers. However, as a silver lining, we show transformer-based parsers give sufficient indicators for whether a frame is likely to be correct or incorrect, making them easier to deploy in production settings.

1 Introduction

Task-oriented semantic parsing—mapping textual utterances to semantic frames—is a critical component of modern conversational AI systems (Gupta et al., 2018; Aghajanyan et al., 2020). Recent methodology casts parsing as transduction, using seq2seq pre-trained transformers to produce linearized parse trees (Aghajanyan et al., 2020; Chen et al., 2020; Li et al., 2021); here, each frame token is either *copied* from the utterance or *generated* from an ontology. Compared to explicit grammar-based approaches (Gupta et al., 2018), this plug-and-play of transformers simplifies the learning objective and scales to multilingual settings, but the lack of provenance makes it challenging to understand model behavior "under the hood."

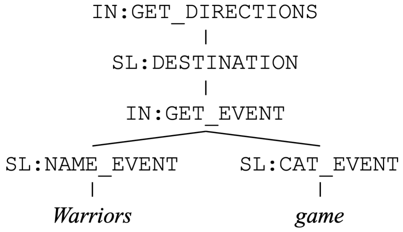

Figure 1: Example decoupled semantic frame representation (Aghajanyan et al., 2020) for the utterance *Directions to the Warriors game*.

In this work, we investigate the strengths and weaknesses of transformer-based semantic parsers and provide modeling directions based on data-driven insights. Specifically, we study BART (Lewis et al., 2020) and XLM-R (Conneau et al., 2020), two state-of-the-art conversational semantic parsers, on both monolingual (TOP/TOPv2; (Gupta et al., 2018; Chen et al., 2020)) and multilingual (MTOP; (Li et al., 2021)) datasets. The compositionality of utterances in these datasets provide a strong testbed for resolving both complex syntactic structure and semantic ambiguity, mirroring the types of challenges our parsers are likely to encounter in practice.

We design our experiments around three main questions. First, broadly speaking, what types of errors do transformer-based parsers make? We begin by annotating 500+ predicted frames across 6 languages and categorize them with fine-grained types. We find transformer-based parsers struggle not only with *classification* (i.e., disambiguating intents/slots) but also *planning* (i.e., switching between copying/generating). Planning errors are more egregious: misplacing close brackets, for example, can violate tree constraints, rendering the entire frame unusable.

Next, we investigate transformer-based parsers' abilities to generate syntactically-valid trees. Specifically, are planning mistakes caused by general uncertainty, or worse, a pathology of seq2seq learning? To address this, we devise an oracle set-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 57–62
August 1–6, 2021. ©2021 Association for Computational Linguistics

split	TOP	TOPv2	MTOP
train	31,279	124,579	73,956
dev	4,462	17,160	10,852
test	9,042	38,785	30,541

Table 1: Dataset splits for TOP, TOPv2, and MTOP.

N	d_{model}	d_{ff}	h	d_k	d_v
6	1024	4096	16	64	64

Table 2: Dimensions of transformer decoder added to XLM-R for MTOP fine-tuning. Notation is borrowed from Vaswani et al. (2017).

ting where a model conditions on partially gold information (either utterance spans or syntactic structure) and predicts the remaining parts of the frame. Surprisingly, we find conditioning on gold spans—not gold structures—results in near-perfect trees at most depths, pointing towards span extraction as a major bottleneck for current parsers.

Finally, though transformer-based parsers are susceptible to error, ideally, we should be able to proactively diagnose mistakes. Using features from model generations (e.g., confidence), can we intrinsically judge if a sequence is correct or incorrect? Encouragingly, we show that a confidence estimation system combining a transformer-based parser and feature-based classifier can detect *correct* frames with 90%+ F1, indicating usability in production settings.

2 Experimental Setup

We conduct experiments on the following task-oriented semantic parsing datasets: (1) **TOP:** parallel corpus consisting of English utterances and corresponding semantic frames (Gupta et al., 2018); (2) **TOPv2:** monolingual extension of TOP to 6 domains (Chen et al., 2020); (3) **MTOP:** multilingual extension of TOP spanning English, Spanish, French, German, Hindi, and Thai (Li et al., 2021). Table 1 shows train, dev, and test splits for the datasets.

Each dataset sample consists of a textual utterance x and (linearized) semantic frame y. Here, frames are in decoupled form (Aghajanyan et al., 2020), as each token is derived either from *copying* from the utterance or *generating* from the ontology (see Figure 1). Following prior work, we fine-tune seq2seq transformers to maximize the log likelihood of the gold frame token at each timestep: $\sum_{(x,y)} \sum_t \log P(y_t | y_{<t}, x; \theta)$.

split	TOP	TOPv2	MTOP
dev	85.41	87.53	76.00
test	85.74	87.52	77.20

Table 3: Exact match (EM) of BART and XLM-R on TOP/TOPv2 and MTOP, respectively.

setting	TOP	TOPv2	MTOP
model	BART	BART	XLM-R
dropout	8.68e-2	1.82e-1	0
batch size	16	16	16
epochs	50	50	50
optimizer	Lamb	Lamb	Lamb
lr	3.72e-4	4.88e-4	6.91e-4
weight decay	6.25e-7	6.26e-7	6.25e-7
swa lr	2.08e-4	1.86e-4	3.96e-4
swa start	8945	18876	19450
swa freq	219	233	185
scheduler	exp	exp	exp
warmup	5000	5000	5000
gamma	0.95	0.95	0.95

Table 4: Hyperparameters for fine-tuning models on TOP, TOPv2, and MTOP.

On TOP/TOPv2, we fine-tune BART (Lewis et al., 2020), a seq2seq transformer pre-trained with a denoising autoencoder objective on monolingual corpora, and on MTOP, we fine-tune XLM-R (Conneau et al., 2020) (equipped with a randomly-initialized decoder), a transformer encoder pre-trained with a masked language modeling objective on multilingual corpora. For XLM-R, specifically, we attach a randomly-initialized decoder (see Table 2). Table 3 shows model performance as judged by exact match. Hyperparameters for all models are listed in Table 4.

3 Error Analysis

In this section, we seek to better understand the types of errors transformer-based parsers make across both monolingual and multilingual settings.

3.1 Error Types

To standardize our analysis, we categorize model errors under the following types: **intent** (incorrect intent prediction), **slot** (incorrect slot prediction), **out-of-domain** (incorrect out-of-domain intent prediction), **mode** (confusion between copying an utterance token or generating an ontology token), and **leaf** (incorrect span in a frame leaf slot). In addition, we report the syntactic **validity** of parse trees separately, though we note mode errors typically result in invalid constructions.

	Exact Match			Tree Validity		
d	TOP	TOPv2	MTOP	TOP	TOPv2	MTOP
1	78.03	86.58	84.75	98.65	94.57	91.23
2	92.30	90.67	85.73	96.97	96.82	93.80
3	90.94	88.50	74.56	97.10	96.35	90.85
4	88.24	86.32	64.53	95.93	95.47	85.73
5	83.39	83.63	44.29	94.29	94.85	69.55
6	83.06	84.54	44.44	94.00	94.45	62.50

Table 5: Benchmarks of BART and XLM-R on TOP/TOPv2 and MTOP, respectively, according to exact match and tree validity at increasing tree depths (d).

One complicating factor is that a predicted sequence may potentially contain several errors, and because decoding is conducted autoregressively, a given error may be influenced by earlier errors (if any such exist). Therefore, to reduce the number of confounding variables, we only consider settings where an incorrect prediction has gold history $\text{argmax}_{y_i} P(y_i|y^*_{<t}, x) \neq y^*_i$; put another way, we only count the *first* error in a sequence.

Using the framework discussed above, we annotate 700 errors across BART and XLM-R on TOP and MTOP, respectively; 100 errors are from TOP and 6×100 errors are from MTOP (100 per language).

3.2 Results

Table 5 benchmarks overall model performance and Figure 2 categorizes errors with fine-grained types; from these results, we draw the following conclusions:

Transformer-based parsers typically struggle with both classification and planning. In the seq2seq formulation, models must jointly *classify* (i.e., provide intent and slot labels) and *plan* (i.e., switch between copying and generating) when producing a semantic frame. Our results show intent/slot and mode errors, which generally fall under the theme of classification and planning, respectively, account for nearly 70-80% of errors. A key observation, however, is that classification and planning error statistics are relatively consistent across languages, suggesting our models may not need language-specific fine-tuning to address these particular errors.

Nearly 40% of incorrectly predicted frames are syntactically invalid. Surprisingly, a large percentage of incorrectly predicted frames violate tree constraints; for linearized frames, this implies the number of open brackets (`[in` or `[sl`) do not match the number of close brackets (`]`). Though well-formedness *is* correlated with depth, we see tree validity (1) is not substantially improved by increasing the number of monolingual samples (TOP → TOPv2) and (2) drops off quite rapidly for multilingual samples (TOP/TOPv2 → MTOP).

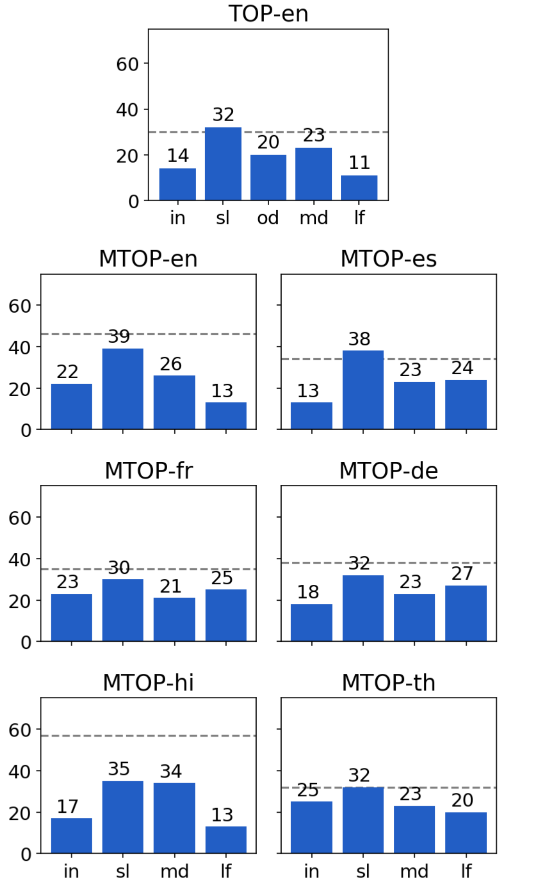

Figure 2: Distribution of errors across TOP and MTOP categorized by intent (in), slot (sl), out-of-domain (od), mode (md), and leaf (lf). Dashed lines indicate the percentage of trees which are syntactically valid.

Span extraction is more challenging in multilingual settings. Leaf errors in English (`TOP|MTOP)-en` are typically twice as lower compared to those in non-English languages `MTOP-(es|fr|de|hi|th)`. Upon closer inspection, we find most leaf errors in English are relatively benign; the model may drop a preposition when copying a span (e.g., *Monday* as opposed to *on Monday*). However, for languages beyond English, extracted spans in leaf slots typically consist of hallucinated or duplicated subwords, which are much more serious in nature. Finally, though languages with non-projective structures (e.g., German) can populate leaf slots with non-contiguous spans, we noticed errors on these types of samples were infrequent.

Out-of-domain detection is also a significant source of error. TOP, in particular, mixes the canonical semantic parsing task with out-of-domain detection by assigning such utterances the frame `[in:unsupported ]`.[1] Though well-motivated, roughly 20% of errors are related to incorrect out-of-domain predictions, suggesting our models have not precisely learned the boundary between in-domain and out-of-domain utterances. If high detection accuracy is preferred, multi-tasking parsers in this fashion may not be an effective use of parameters (assuming more data is not available); instead, out-of-domain detection can be conducted independently with alternate methodology (Gangal et al., 2019).

4 Syntactic Structure

Our case study above demonstrates transformer-based parsers can produce syntactically-invalid frames at a high rate. These structural errors are more serious than disambiguation errors since they render the frame unusable, potentially causing cascading failures in a task-oriented dialog system. Therefore, in this section, we dive deeper into why tree constraints are not satisfied and question the possibility of achieving perfect tree validity.

While transduction models do not explicitly impose tree constraints, there is precedent that strong neural representations do implicitly model tree structures; recent studies demonstrate large-scale pre-training, in particular, imbues strong notions of syntax (Goldberg, 2019; Jawahar et al., 2019; Tenney et al., 2019). Taking these results together, we hypothesize that transformer representations may be "good enough", but instead there exist ambiguous aspects of our task-oriented semantic parsing task which cause tree invalidity.

Previously, we saw transformer-based semantic parsers largely struggled with classification- and planning-related errors. Therefore, the question we pose is: if we resolve these ambiguities by creating oracle models, can we achieve perfect tree validity? This setup also enables us to gain a deeper understand of the upper-bound performance of transformer-based semantic parsers, even as their representations get stronger.

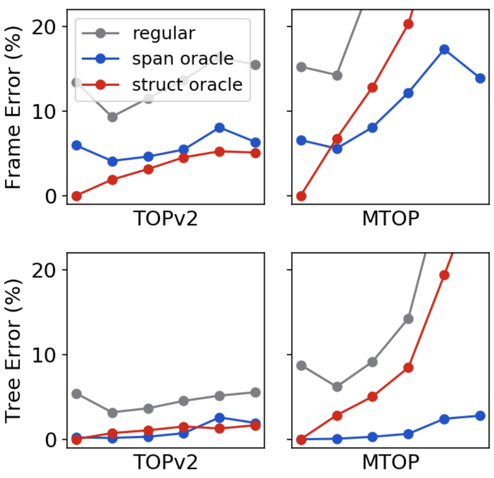

Figure 3: Exact match (EM) and tree validity (TV) error (%) of the regular, span oracle, and structure oracle models on TOPv2 and MTOP. Dots from left → right indicate increasing frame compositionality (the graph depths of 1 → 6).

Oracle Models. Because classification and planning target inherently different phenomena, creating an oracle that simultaneously makes both less ambiguous is challenging. Instead, we experiment with two separate oracles—**span oracle** and **structure oracle** models for classification and planning, respectively—which map an utterance x along with a "partially gold" snippet z to generate the frame y, inducing the objective $\sum_{(x,y,z)} \sum_t \log P(y_t|y_{<t}, x, z; \theta)$.

For example, given an utterance x *Where can I see fireworks tonight?* and frame y `[in [sl` *fireworks* `[sl` *tonight* `] ]`, the span oracle model defines z as `[span1]` *fireworks* `[span2]` *tonight* and the structure oracle model defines z as `[in [sl [span1] [sl [span2] ] ]`.[2] Here, providing z as input helps the model learn $y \setminus z$; span oracle models optimize for correct structure and structure oracle models optimize for correct spans. Table 6 shows example source and target pairs for the regular, span oracle, and structure oracle models.

Results. Figure 3 shows the oracle model results; we measure both exact match and tree validity error. **A key phenomenon we observe is that conditioning on gold spans results in near-zero tree validity error at most depths.** Surprisingly, we see conditioning on gold structures (to stress, the *exact* syntactic structure) never consistently results in well-formed trees, especially as the depth in-

[1] There also exist more fine-grained out-of-domain categories, such as `[in:unsupported-event ]`.

[2] Fine-grained intent/slot labels are omitted for visual clarity, but are included during model training.

model type	utterance x (+ snippet z)	frame y
regular	Where can I see fireworks tonight?	`[in [sl` fireworks `[sl` tonight `] ]`
span oracle	+ `[span1]` fireworks `[span2]` tonight	
struct oracle	+ `[in [sl [span1] [sl [span2] ] ]`	

Table 6: Example source and target pairs for oracle experiments. The span oracle specifies the gold spans while the struct oracle specifies the gold structure. Note that `[in` and `[sl` are used for brevity.

creases. Structure oracle models still suffer from mode errors during generation: augmenting a leaf span with an extra word instead of placing a close bracket, for example, is a typical mistake. Furthermore, we see this problem is magnified in MTOP, which connects to the notion that span extraction tends to be difficult in multilingual settings.

Our experiments suggest seq2seq transformer-based parsers *can* achieve near-perfect tree validity—even at large depths—provided that span extraction is precise. Currently, however, this is a major source of ambiguity our parsers are not well-equipped to handle, especially when scaling to languages beyond English.

5 Confidence Estimation

Despite the criticism we have presented of state-of-the-art, transformer-based conversational semantic parsers, these models do demonstrate strong performance over prior baselines, and correctly parse a vast majority of samples. A property that can make these models easier to deploy in practice is if they "know what they don't know" (Desai and Durrett, 2020); besides interpretability, this is particularly useful for identifying and correcting errors in tail scenarios via active learning (Dredze and Crammer, 2008; Duong et al., 2018; Sen and Yilmaz, 2020). We frame this problem as confidence estimation (Blatz et al., 2004): given an utterance x, predicted frame y', and gold frame y, we seek to learn a binary classifier which uses target-side features $f(y')$ to estimate $P(y' = y) = \text{sigmoid}(w^\top f(y'))$.

To make our approach as generalizable as possible, we constrain $f(y')$ to be as model-agnostic and recall-oriented as possible. We select the following features: (1) **length:** $|y'|$; (2) **validity:** $\max(0, \sum_i \mathbb{1}[y'_i \in V^+] - \mathbb{1}[y'_i \in V^+])$ where V^+ and V^- are the set of open and close brackets, respectively; and (3) **confidence:** $\frac{1}{|y'|} \sum_i P(y'_t | y'_{<t}, x)$. Using our best transformer-based parsers, we obtain predictions on a held-out set D_{dev} and test set D_{test}. Then, we train and test a SVM on D_{dev} and D_{test}, respectively, using the features defined above.

	TOPv2			MTOP		
	P	R	F1	P	R	F1
SVM	97.2	85.7	91.2	95.0	85.2	89.8
–length	**97.7**	84.8	90.8	**95.1**	84.7	89.6
–validity	97.0	82.6	89.2	94.9	80.5	87.1
–confidence	91.6	**98.8**	**95.1**	85.3	**95.8**	**90.2**

Table 7: Precision (P), recall (R), and F1 of the SVM-based confidence estimator. –x indicates an ablation of feature x (i.e., it is omitted during learning).

In addition to the standard hinge loss, we also add a class imbalance penalty as positive examples are typically 5-8× as prevalent depending on the dataset. We chiefly evaluate the binary classifier's ability to identify semantic frames which are correct (i.e., the positive class). From an active learning standpoint, getting positive samples wrong is more serious than getting negative samples wrong; annotation resources are best directed towards boundary or incorrect predictions.

Table 7 shows the performance and ablations of our confidence estimator. **In both monolingual and multilingual settings, using transformer-based features, we can detect *correct* semantic frames with 90%+ F1.** In particular, we see length and validity largely capture the space of correct frames (recall) and confidence effectively distinguishes between correct and incorrect frames (precision). Practitioners may select an SVM variant depending on whether precision or recall is preferred.

6 Conclusion

In this work, we assess the strengths and weaknesses of seq2seq transformers for task-oriented semantic parsing. These models "know what they don't know", making them easier to depoy in practice, but cannot perfectly model compositional utterances, as indicated by the challenges of span extraction. We believe that modeling efforts in this direction—as opposed to simply annotating more data—can improve parsers substantially.

References

Armen Aghajanyan, Jean Maillard, Akshat Shrivastava, Keith Diedrick, Michael Haeger, Haoran Li, Yashar Mehdad, Veselin Stoyanov, Anuj Kumar, Mike Lewis, and Sonal Gupta. 2020. Conversational Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Xilun Chen, Ashish Ghoshal, Yashar Mehdad, Luke Zettlemoyer, and Sonal Gupta. 2020. Low-Resource Domain Adaptation for Compositional Task-Oriented Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shrey Desai and Greg Durrett. 2020. Calibration of Pre-trained Transformers. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mark Dredze and Kovy Crammer. 2008. Active Learning with Confidence. In *Proceedings of the Annual Meeting of the Conference on Computational Linguistics (ACL)*.

Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip Cohen, and Mark Johnson. 2018. Active Learning for Deep Semantic Parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Varun Gangal, Abhinav Arora, Arash Einolghozati, and Sonal Gupta. 2019. Likelihood Ratios and Generative Classifiers for Unsupervised Out-of-Domain Detection In Task Oriented Dialog. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. *arXiv preprint arXiv:1901.05287*.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic Parsing for Task Oriented Dialog using Hierarchical Representations. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pretraining for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A Comprehensive Multilingual Task-Oriented Semantic Parsing Benchmark. In *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*.

Priyanka Sen and Emine Yilmaz. 2020. Uncertainty and Traffic-Aware Active Learning for Semantic Parsing. In *Proceedings of the Workshop on Interactive and Executable Semantic Parsing (INTEXSEMPAR)*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*.