Entity Decoupling and Alignment Mechanism for First-Order Logic Parsing

Anonymous EACL submission

Abstract

Semantic parsing is the task of obtaining machine-interpretable representations from natural language. We consider FOL and model its parsing as a seq2seq mapping where given a sentence, it is encoded using an LSTM followed by a decoder which sequentially generates the predicates. We show the effectiveness of decoupling the FOL entity and predicting its *category* - Unary, Binary, Variables and Scoped, at each decoder step as an auxiliary task. We further improve upon it by introducing a variable alignment mechanism to align variables across predicates. We perform extensive evaluations & ablations to explore the capability of neural models in mimicking the ground truth FOL generated by *Boxer*. We also release our code to aid further research in logic-based parsing and inference in NLP.

1 Introduction

Semantic parsing aims at mapping natural language to structured meaning representations. This enables a machine to understand unstructured text better which is useful in many tasks like question answering (Berant et al., 2013; Pasupat and Liang, 2015), robot navigation (MacMahon et al., 2006; Artzi and Zettlemoyer, 2013), database querying (Zelle and Mooney, 1996) etc. A variety of logical forms and meaning representations have been proposed including graph-based formalisms (Banarescu et al., 2013; Abend and Rappoport, 2013; Oepen et al., 2014; Kollar et al., 2018) where text is represented as a typed graph. The entities and action events are represented as nodes with labeled edges depicting relations between them. AMR (Abstract Meaning Representation) graphs (Banarescu et al., 2013) use variables to annotate nodes following neo-Davidsonian style (Davidson, 1969). In this work, we focus on first-order logic (FOL) (Smullyan, 2012) as the language formalism for text. FOL represents entities and actions in natural language through quantified variables and consists of functions (called predicates) which take variables as arguments. The predicates attach semantics to variables and express relations between objects (Blackburn, 2005). For instance, sentence - "a man is eating" can be represented through FOL

$$\exists A(\exists B(man(A) \wedge eat(B) \wedge agent(B,A)))$$

The FOL is interpreted with the context that entities in sentence make the universe. Natural language concepts as in sentence "the man and woman are seated facing each other" can be expressed as

$$\exists A(\exists B(\exists D(\exists C(man(A) \wedge woman(B) \wedge seat(D) \wedge subset\_of(A,C) \wedge subset\_of(B,C) \wedge theme(D,C) \wedge not(\exists E(other(E) \wedge not(\exists F(face(F) \wedge theme(F,E) \wedge agent(F,C))))))))))$$

where "man" and "woman" are represented together through shared variable C and "facing each other" is represented by negating the existence of a thing E for which C is not facing E holds true.

The success of encoder-decoder framework based neural approaches in NLP tasks like machine translation (Cho et al., 2014; Sutskever et al., 2014; Vaswani et al., 2017) and logical inference (Kim et al., 2019) has motivated their use in semantic parsing (Kočiskỳ et al., 2016; Buys and Blunsom, 2017; Cheng et al., 2017; Liu et al., 2018; Li et al., 2018). Some of these are designed to solve question answering tasks for specific domains (Jia and Liang, 2016; Dong and Lapata, 2016) on confined logical formalism capturing limited concepts.

To parse sentences to FOL, we train our model on a large corpus of text-FOL pairs for SNLI Dataset (Bowman et al., 2015) parsed using C&C parser (Clark and Curran, 2007) and Boxer (Bos, 2008)[1]. FOL parsing can enable neural models

[1]https://github.com/valeriobasile/candcapi

to: 1) capture complex relationships between entities resulting in richer embeddings(can be useful in NLP tasks); 2) perform reasoning in a logical way(like theorem provers). Since this is one of the first exploration for neural text to FOL, we use Seq2Seq coupled with attention mechanism (Bahdanau et al., 2014; Cohn et al., 2016) as baseline. We disentangle prediction of different types of FOL entities (unary and binary predicates, variables etc) and show improvements through performing category type prediction as an auxiliary task. We further show major improvements by explicitly constraining the decoder to align variables across predicates to maintain consistency. Even though the ground truth for our model has been generated using Boxer and is therefore somewhat noisy, it provides a reasonable ground for testing the neural models' capability.

Our contributions can be enumerated as: 1) We analyse the capability of neural models to parse sentences to FOL (ground truth) generated by Boxer; 2) We show effectiveness of disentangled FOL entity type prediction & argument variable alignment mechanism through extensive ablations; 3) We aim to release our code and models to aid further research in neural FOL parsing.

2 Proposed Approach

We convert the output FOL comprising of existential quantifiers and disjunction of atomic formulas into an equivalent mapping as a sequence of predicates, argument variables, scoping symbols (such as "fol(", ")", "not(") and train our models to predict the sequence. We arrange scope symbols in accordance to their nesting level (top most appearing first in the sequence) with further ordering that entities that are part of same scope are arranged as sequence of unary predicates, followed by binary predicates and other nested scoped entities. We model parsing a given sentence into FOL as a sequence to sequence *transduction* task.

2.1 Baseline

Our baseline consists of Encoder-Decoder model. The Encoder E is a bidirectional LSTM (biLSTM) which encodes a sequence of input tokens $X : \{x_0, x_1..., x_m\}$ into a sequence of hidden states $H_e : \{h_{e_0}, h_{e_1}, ..., h_{e_m}\}$, $h_{e_i} \in R^{d_h}$. The decoder D consists of an LSTM which uses the outputs of encoder E along with previously decoded outputs, provided as embeddings $E_d : \{e_{d_0}, e_{d_1}, ..., e_{d_n}\}$ to it as input, to generate a sequence of hidden states $H_d : \{h_{d_0}, h_{d_1}, ..., h_{d_n}\}$. We also use attention (Bahdanau et al., 2014) to calculate context c_{e_i} at i^{th} decoding step. We train the model on the standard cross-entropy objective L_{CE} while adopting *teacher forcing* methodology *i.e.* giving the inputs to the decoder from ground truth instead of previously decoded tokens while training

2.2 Separate Heads

The output tokens in an FOL sequence do not all belong to the same token category unlike majority of sequence to sequence problems. The output tokens can be divided into four major types - Unary Predicates U, Binary Predicates B, Variables V, and Scoped Entities S. We create separate vocabularies for each category. The variables V(which do not have standalone semantics) have one-hot embedding while other types of output tokens have *dense* embeddings. Building on above intuition, we use five different softmax *heads* on top of Decoder LSTM. We train the model to predict which type of token to choose at each step (using head T) as an *auxiliary* task while other heads decode the probability of the four different types of tokens. Hence, the overall cross-entropy objective becomes as follows, t_i^t being the target token type to be predicted at step i:

$$L_{sep}(\theta, \phi) = L_{CE}(\theta, \phi) - \frac{1}{n}\sum_{i=0}^{n} t_i^t * log(t_i)) \qquad (1)$$

2.3 Alignment Mechanism

One of the key challenges for the model is to identify relationship between *variables*. A variable A which is argument in a binary predicate should be aligned with the argument variable in a unary predicate where it was used previously. This could be achieved through decoder self-attention which did not perform well (shown in experiments) since the model does not receive any explicit signal and have to such discover relations. To mitigate this, we introduce **Alignment Mechanism**. At each step where a *variable* is decoded, an additional linear classifier calculates probability A_{d_i} whether current variable aligns with any previously decoded variable and with which variable, in case it does align. The probability α_{i_j} that a particular step j aligns with the current decoding step i, $i > j$ is calculated with a dot product based attention over the already decoded tokens and a context vector c_{a_i} is thus generated as probability weighted sum. For other category of tokens but *variables*, the decoder heads remain the same. For *variables*, we determine aligned hidden state as

$$h_{d_i}^{ali} = A_{d_i} * c_{a_i} + (1 - A_{d_i}) * h_{d_i} \qquad (2)$$

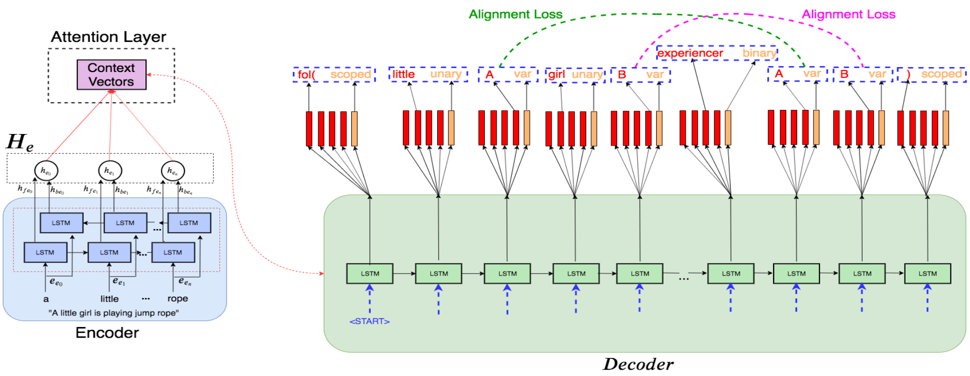

Figure 1: Overview of our architecture showing separate heads (red), category prediction (orange) and alignment mechanism (green and pink). Input to Decoder LSTM (blue) depicts the output of last step being fed at next step. Red arrow between Attention Layer and Decoder depicts standard encoder-decoder attention.

The output, then, is $o_v^i = W_o * [h_{d_i}^{ali}; c_{e_i}]$. To provide explicit signal during training, we train α (attention weights) and A_{d_i} on target alignment positions and decisions with a cross entropy objectives

$$L_{dec}(\theta, \phi, \zeta) = -\frac{1}{n}\sum_{i=0}^{n}(A_{d_i}^t * log(A_{d_i}) + (1 - A_{d_i}^t) * log(1 - A_{d_i})) \quad (3)$$

$$L_{pos}(\theta, \phi, \zeta) = -\frac{1}{n}\sum_{i=0}^{n}\sum_{j=0}^{i-1} A_{d_i}^t * \alpha ij^t * log(\alpha_{ij}) \quad (4)$$

where $A_{d_i}^t$ and α_{ij}^t refer to ground truth decision and alignment position values and ζ refers to additional parameters introduced due to alignment mechanism. Therefore our overall loss becomes sum of L_{sep}, L_{dec} and L_{pos}.

3 Experiments

We used a subset of SNLI (Bowman et al., 2015) corpus by eliminating duplicates and create 255,501 training instances, 10,691 dev and 10,633 test instances(from SNLI development & test sets). All the models were trained with an Adam Optimizer(Kingma and Ba, 2014) initialized with a learning rate of 0.001 with a decay rate of 10^{-4}. We use an embedding size $D = 100$ for both encoder and decoder. Unary and Binary predicates have an embedding size of 100 each while variable and type tokens have one-hot embeddings. All the embeddings were initialized randomly.

3.1 Results and Discussion

3.1.1 Evaluation Framework

We evaluate degree of match between two FOLs following intuition behind D-match and Smatch (Cai and Knight, 2013). We align two FOLs beginning with variables where two variables are aligned if corresponding predicates' name (of which they are arguments) and argument positions match. While aligning two predicates, we check if their arguments are aligned and names are same. Finally we align nested scope symbols(eg. "not("). For scoped entity, we determine predicted scoped entity having maximum alignment with it based on count of other aligned predicates and scoped entities that are contained inside them. We decompose FOL into related pairs of the form (n_1, n_2) such that n_2 appears inside the scope of n_1. For instance, a variable that is an argument in a predicate or a predicate appearing inside a scoped entity. We estimate number of pairs in expected FOL that can be matched with pairs in predicted FOL based on whether corresponding entities in the pairs are aligned.

3.1.2 Comparison and Ablation Studies

We evaluate different ablations of our method and also compare them with the baseline in Table 1. Vanilla is baseline model, Separate Heads model predicts different tokens based on category(with auxiliary task), Align refers to configuration where we add our alignment mechanism. Evidently, our final model **Separate Heads + Align** convincingly outperforms all described models and improves the baseline by ~ 7 F-1 points. Decoder self-attention, even though, improves **Vanilla** Model does not provide any improvements when used with **Separate Heads**. This can be attributed to its inability to incorporate decoder level information which probably becomes factorized automatically during training through using separate heads. However, it provides improvements over **Vanilla** by a good margin but still only matches or remains inferior to the standalone **Separate Heads** model. **Align Mechanism** manages to provide a huge boost to the **Sep-**

arate Heads model by improving it by ~ 5 F-1 points. However, performance deteriorates when used with **Vanilla** model since its ability to align variables *only* vanishes in this setup which we find critical for its working.

Models	Valid			Test		
	P	R	F-1	P	R	F-1
Vanilla (Baseline)	65.48	65.15	65.31	66.43	65.84	66.13
+ Self Attention	67.86	66.70	67.28	68.56	67.23	67.89
+ Align Mechanism	62.13	61.06	61.59	61.74	60.60	61.17
Separate Heads	68.48	66.81	67.64	69.18	67.63	68.40
+ Self Attention	67.11	65.87	66.48	66.92	65.69	66.30
Sep Heads + Align	**73.68**	**72.17**	**72.92**	**74.05**	**72.53**	**73.28**

Table 1: Results on both validation and test split

Variables Consistency: We consider binary predicate-variable argument pairs and not entire FOL. Our test set comprises of 92844 such pairs, baseline yields recall(R) of 58.4%(F1:58.9), Sep-Heads - R of 60.8%(F1:62.0) while Sep heads + align improves R to 65.7%(F1:66.8) since more variables get aligned now.

3.1.3 Analysis

Variation with Input Length: The proposed changes make model relatively much more robust to increase in length in the input sentence as shown in Fig. 2. This is due to many factors - increased model capacity as well as its ability to process different categories of output tokens separately giving better long range dependencies and less confusion in generating many variables over FOL owing to better alignment across the sequence.

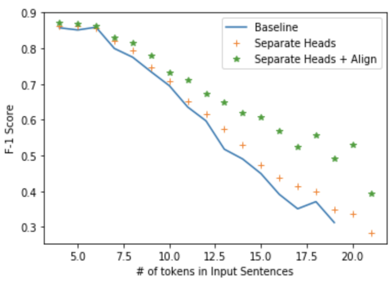

Figure 2: F-1 variation with input length on Test Data

Perturbed Training: It has been noticed ((Jia and Liang, 2017; Niven and Kao, 2019) that neural models sometimes exploit trivial patterns in outputs/inputs to fool and provide pseudo-improved results. One such pattern could be presence of variables like *A* and *B* with some specific unary and binary predicates. In order to disturb such patterns, we randomly permute the presence of such variables in the ground truth during training. Our **baseline** model indeed shows a significant drop in results (Table 2). On the other hand, our other two main models do not show such large drop proving their robustness to such disturbances.

Model	Precision	Recall	F-1
Vanilla(Baseline)	63.75	62.32	63.03
Separate Heads	67.39	65.59	66.48
Sep Heads + Align	72.90	71.95	72.42

Table 2: Results on Test set with perturbed training

4 Related Work

Early semantic parsers were majorly rule based (Johnson, 1984; Woods, 1973; Thompson et al., 1969) followed by statistical techniques (Kwiatkowski et al., 2010; Zettlemoyer and Collins, 2012). Data driven neural approaches (Buys and Blunsom, 2017; Cheng et al., 2017; Li et al., 2018) have alleviated need for manually defining lexicon. Some of these produce sequential parses (Jia and Liang, 2016; Kočiský et al., 2016) while others leverage hierarchical parse structure (Alvarez-Melis and Jaakkola, 2016; Rabinovich et al., 2017). (Dong and Lapata, 2016) proposed SEQ2TREE to recursively generate sub-trees for domain-specific hierarchical logical form. (Liu et al., 2018) parse DRSs using dedicated hierarchical decoders to generate partial structure first before semantic content. We instead make model disambiguate syntactic types through type prediction as auxiliary task and multiple prediction heads. Constrained decoding using target language grammar rules has been explored (Yin and Neubig, 2017; Xiao et al., 2016). Copy-mechanism (Gu et al., 2016) and decoder self-attn (Zhang et al., 2019) have been used previously, but our variable alignment mechanism is different since it enforces variable consistency through explicit decision based supervised loss on whether to align and where to align.

5 Conclusion and Future Work

We examined capability of neural models on task of parsing First-Order Logic from natural language sentences. We proposed to disentangle the representations of different token and perform category prediction as an auxiliary task. We utilized entity decoupling to build an alignment mechanism to capture relationship between variables across predicates and show effectiveness through various ablations. Despite the limitations of the dataset, the alignment and disentanglement concepts proposed are generic and can pave way for future explorations over different natural language parsing datasets.

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.

David Alvarez-Melis and Tommi S Jaakkola. 2016. Tree-structured decoding with doubly-recurrent neural networks.

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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

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.

Patrick Blackburn. 2005. Representation and inference for natural language: A first course in computational semantics.

Johan Bos. 2008. Wide-coverage semantic analysis with boxer. In *Proceedings of the 2008 Conference on Semantics in Text Processing*, pages 277–286. Association for Computational Linguistics.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Jan Buys and Phil Blunsom. 2017. Robust incremental neural semantic graph parsing. *arXiv preprint arXiv:1704.07092*.

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.

Jianpeng Cheng, Siva Reddy, Vijay Saraswat, and Mirella Lapata. 2017. Learning structured natural language representations for semantic parsing. *arXiv preprint arXiv:1704.08387*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Stephen Clark and James R Curran. 2007. Wide-coverage efficient statistical parsing with ccg and log-linear models. *Computational Linguistics*, 33(4):493–552.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. *arXiv preprint arXiv:1601.01085*.

Donald Davidson. 1969. The individuation of events. In *Essays in honor of Carl G. Hempel*, pages 216–234. Springer.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. *arXiv preprint arXiv:1601.01280*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *arXiv preprint arXiv:1707.07328*.

Tim Johnson. 1984. Natural language computing: the commercial applications. *The Knowledge Engineering Review*, 1(3):11–23.

Seonhoon Kim, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6586–6593.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tomáš Kočiskỳ, Gábor Melis, Edward Grefenstette, Chris Dyer, Wang Ling, Phil Blunsom, and Karl Moritz Hermann. 2016. Semantic parsing with semi-supervised sequential autoencoders. *arXiv preprint arXiv:1609.09315*.

Thomas Kollar, Danielle Berry, Lauren Stuart, Karolina Owczarzak, Tagyoung Chung, Lambert Mathias, Michael Kayser, Bradford Snow, and Spyros Matsoukas. 2018. The alexa meaning representation language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, Volume 3 (Industry Papers), pages 177–184.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2010. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1223–1233. Association for Computational Linguistics.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214.

Jiangming Liu, Shay B Cohen, and Mirella Lapata. 2018. Discourse representation structure parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 429–439.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. 2006. Walk the talk: Connecting language, knowledge, and action in route instructions. *Def*, 2(6):4.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*.

Stephan Oepen, Marco Kuhlmann, Yusuke Miyao, Daniel Zeman, Dan Flickinger, Jan Hajic, Angelina Ivanova, and Yi Zhang. 2014. Semeval 2014 task 8: Broad-coverage semantic dependency parsing. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 63–72.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*.

Ella Rabinovich, Noam Ordan, and Shuly Wintner. 2017. Found in translation: Reconstructing phylogenetic language trees from translations. *arXiv preprint arXiv:1704.07146*.

Raymond R Smullyan. 2012. *First-order logic*, volume 43. Springer Science & Business Media.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

Frederick B Thompson, PC Lockemann, B Dostert, and RS Deverill. 1969. Rel: A rapidly extensible language system. In *Proceedings of the 1969 24th national conference*, pages 399–417. ACM.

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 *Advances in neural information processing systems*, pages 5998–6008.

William A Woods. 1973. Progress in natural language understanding: an application to lunar geology. In *Proceedings of the June 4-8, 1973, national computer conference and exposition*, pages 441–450. ACM.

Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1341–1350.

Pengcheng Yin and Graham Neubig. 2017. A syntactic neural model for general-purpose code generation. *arXiv preprint arXiv:1704.01696*.

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.

Luke S Zettlemoyer and Michael Collins. 2012. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. *arXiv preprint arXiv:1207.1420*.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. Broad-coverage semantic parsing as transduction. *arXiv preprint arXiv:1909.02607*.

Entity Decoupling and Alignment Mechanism for First-Order Logic Parsing (Supplementary Material)

Anonymous EACL submission

1 Compute Details

Each of our models are implemented from scratch using Pytorch[1] and trained on a single V100 16 GB GPUs and 240 GB Memory machines. The models were trained till there was no validation loss improvements. On an average, the training for models took 6 hours each.

2 Evaluation Framework

Algorithm 1: Estimating alignment between variables (vars) and predicates (preds) in expected and predicted FOLs

Input : expected and predicted FOLs, FOL_{ex} and FOL_{pred} of an input sentence
Output : possible alignments between variables and predicates of input FOLs
$curr\_matching \leftarrow empty$
$var\_alignments \leftarrow$ ALIGN_ENTS(FOL_{ex}^{vars}, FOL_{pred}^{vars}, $curr\_matching$)
$pred\_alignments \leftarrow$ empty
for $curr\_matching \in var\_alignments$ **do**
    $pred\_alignments$.extend(ALIGN_ENTS(FOL_{ex}^{preds}, FOL_{pred}^{preds}, $curr\_matching$))
end
function ALIGN_ENTS (F_{ex}^{ents},F_{pred}^{ents},$matching$)
    $ext\_match \leftarrow$ empty
    $noChange \leftarrow$ True
    for $ent_{ex} \in F_{ex}^{ents}$, $ent_{pred} \in F_{pred}^{ents}$ **do**
        if $ent_{ex} \notin matching$ *and* $ent_{pred} \notin matching$ *and* ***EntsMatch***(ent_{ex}, ent_{pred}) **then**
            $noChange \leftarrow$ False
            $new\_matching \leftarrow$ copy($matching$).append((ent_{ex}, ent_{pred}))
            ext_match.extend(ALIGN_ENTS(F_{ex}^{ents},F_{pred}^{ents}, $new\_matching$))
        end
    end
    if $noChange$ **then**
        ext_match.append($matching$)
    end
    return ext_match
end_function

[1] https://pytorch.org/