Instilling Type Knowledge in Language Models via Multi-Task QA

Shuyang Li
UC San Diego
shl008@ucsd.edu

Mukund Sridhar
Amazon Alexa AI
harakere@amazon.com

Chandana Satya Prakash
Amazon Alexa AI
chanprak@amazon.com

Jin Cao
Amazon Alexa AI
jincao@amazon.com

Wael Hamza
Amazon Alexa AI
waelhamz@amazon.com

Julian McAuley
UC San Diego
jmcauley@ucsd.edu

Abstract

Understanding human language often necessitates understanding entities and their place in a taxonomy of knowledge—their *types*. Previous methods to learn entity types rely on training classifiers on datasets with coarse, noisy, and incomplete labels. We introduce a method to instill fine-grained type knowledge in language models with text-to-text pre-training on type-centric questions leveraging knowledge base documents and knowledge graphs. We create the **WikiWiki** dataset: entities and passages from 10M Wikipedia articles linked to the Wikidata knowledge graph with 41K types. Models trained on WikiWiki achieve state-of-the-art performance in zero-shot dialog state tracking benchmarks, accurately infer entity types in Wikipedia articles, and can discover new types deemed useful by human judges.

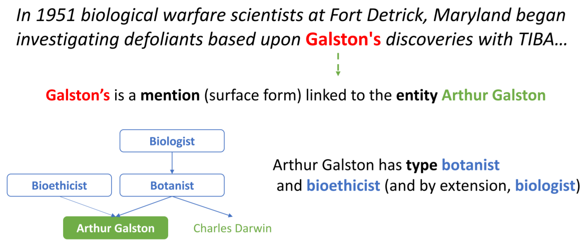

Figure 1: Via the **WikiWiki** dataset, we train a model to answer questions about entities mentioned in Wikipedia articles (top) and WIkidata *types* that such entities are an *instance of* (P31) or *subclass of* (P279).

1 Introduction

Entities can be categorized by their *types*, which indicate where they belong in a taxonomy of knowledge. For example, Venus is a planet and thus also an astronomical body. Much like how knowledge acquisition in cognitive development progresses from recognizing concrete objects to gradually understanding their relations to one another (Lucariello et al., 1992), we aim to extend language models' existing rough understanding of entities (Heinzerling and Inui, 2021) to the types that govern how entities are related. Instilling type knowledge in multi-purpose models can improve performance in tasks like entity linking (Onoe and Durrett, 2020), question-answering (Févry et al., 2020a), and semantic parsing (Thirukovalluru et al., 2021).

While language models can memorize some facts (Petroni et al., 2019), they frequently hallucinate false information (Logan IV et al., 2019; Shuster et al., 2021). Current attempts to learn to infer types for entities are hampered by 1) the difficulty of collecting diverse, large-scale typing datasets; and 2) how existing corpora assume independence between types (Choi et al., 2018), while in reality types sit at levels of granularity that are useful in different settings: a pizza store may care whether a user likes Cheese Pizza; a restaurant recommender might care if the user wants Pizza; finally, a general dialog agent might only care if a user wants Food.

We address both issues by proposing a simple and effective approach for pre-training generative language models to answer questions about entities, types, and surface forms (mentions) in a large public knowledge graph (KG) consisting of Wikipedia articles and Wikidata nodes. We leverage high quality type labels in a large corpus of knowledge-rich text *and* an ordered, hierarchical type ontology.

To summarize our main contributions: 1) We create the new **WikiWiki** dataset comprising 10M Wikipedia articles linked to nodes from Wikidata; 2) We propose a pre-training scheme for generative language models using type-centric question-answering based on WikiWiki; 3) We achieve state-of-the-art (SOTA) performance in zero-shot domain adaptation for dialog state tracking using our type-instilled models, with average per-domain gains of 14.9% (49.4% relative) joint accuracy; and 4) We show that our models can precisely infer types for seen and unseen entities in new articles from WikiWiki, and propose novel types that hu-

Findings of the Association for Computational Linguistics: NAACL 2022, pages 594 - 603
July 10-15, 2022

	Training	Test	Test (New Ent)
Documents	10 M	5.0 K	5.0 K
Unique Entities	2.2 M	14.1 K	6.0 K
Unique Types	40.6 K	4.0 K	1.2 K
Num. of Mentions	38.7 M	19.3 K	6.4 K
Type References	43.8 M	21.5 K	6.5 K

Table 1: Unique documents/entities/types and number of mentions in each split of **WikiWiki**. *Test (New Ent)* comprises entities not seen in the training split.

mans judge to be accurate and appropriate.

2 Related Work

Knowledge Grounding in Language Models Large pre-trained language models have been shown to memorize some facts (Petroni et al., 2019). One recent line of work aims to explicitly condition generation on knowledge bases by combining a retrieval module and a language model (Majumder et al., 2020; Guu et al., 2020; Lewis et al., 2020b; Mazaré et al., 2018). Peters et al. (2019) propose instead to align token representations from pre-trained language models with entity embeddings to reason over a limited set of entities. Yamada et al. (2020) explicitly denote entity tokens with a learned input embedding. Specific entity embeddings have also been learned jointly by using knowledge graphs as auxiliary inputs during language model pre-training (Sun et al., 2020a; Févry et al., 2020b; Zhang et al., 2021). Another line of work aims to model specific factual statements from knowledge bases (Wang et al., 2021) for reading comprehension (Lu et al., 2021) and trivia QA (Agarwal et al., 2021). We propose text-to-text pre-training on knowledge recovery tasks to instill *type-awareness*. Our models learn type knowledge that transfers to the type-adjacent downstream task of dialog state tracking and can infer unseen types.

Entity Representation Learning Many SOTA systems for knowledge retrieval and QA rely on learned dense embeddings of individual entities or types to perform multi-class classification (Ganea and Hofmann, 2017; Karpukhin et al., 2020; Wu et al., 2020a). Several recent frameworks aim to learn entity knowledge during language model pre-training via entity masking (Sun et al., 2020b) or contrastive learning (Qin et al., 2021). Systems for entity typing (Dai et al., 2021) and disambiguation (Yamada et al., 2019) also learn dense vector encodings that are later matched via dot-product scoring. Cao et al. (2021) aim to address some downsides of the above approaches—the linearly increasing space required to store learned representations and difficulties in negative sampling—by casting the task as generative language modeling: predict the name of an entity to be linked. We generalize this approach from entity names (which appear verbatim) to include types, which require a more nuanced understanding of a context.

Context: These included carbon dioxide by burning diamond, and mercuric oxide by heating mercury. This type of experiment contributed to the discovery of "dephlogisticated air" by Priestley, which became better known as oxygen, following Lavoisier's investigations.

Entity/Type Discovery (20%): List all concepts and types mentioned here.
Answer: Priestley (chemist), Lavoisier (chemist), mercuric oxide (chemical compound), mercury (chemical element), and dephlogisticated air (superseded scientific theory)

Entity Typing (30%): What is dephlogisticated air an example of?
Answer: superseded scientific theory

Entity Recognition (20%): What does Priestley refer to?
Answer: Joseph Priestley (chemist)

Slot Filling (30%): Which chemists are mentioned here?
Answer: Joseph Priestley and Antoine Lavoisier

Table 2: In pre-training, the model reads a Wikipedia article and answers questions from four tasks involving entities and types. It must generate answers containing terms not found verbatim in the text. Surface forms (mentions) in **green**, entities in **red**, and types in **blue**.

3 Type-Centric Multitask Modeling

WikiWiki Corpus To train an entity- and type-aware language model, we build the **WikiWiki** dataset by combining Wikipedia articles with the Wikidata KG (Vrandecic, 2012). Wikipedia articles have been used to enrich corpora for dialog (Dinan et al., 2019), coreference resolution (Singh et al., 2012), and QA (Liu et al., 2020). KGs have been used for entity typing and relation extraction (Sakor et al., 2020). Yao et al. (2019) use Wikipedia pages as context for relation triples mined from Wikidata.

We link articles, entities, and types as in Figure 1: like Wu et al. (2020b), we take Wikipedia hyperlinks as links between *entities* (target page) and their *mentions* (link text); we link pages to Wikidata nodes via ID; and for each node we extract types T from Wikidata where $t \in T$ is an *instance*/*subclass* of the node (discarding entities with no types).[1] To address sparsity of hyperlinks,

[1]All humans on Wikidata are an instance of 'human'; we thus use the 'occupation' relation to determine their types.

we follow Yao et al. (2019) and use spaCy to identify additional entities. We sample 10M articles for training, with two disjoint 5K-article splits for evaluation, containing seen and unseen (New Ent) entities respectively (Table 1). The ontology of Wikidata types forms a directed acyclic graph with 41K type nodes applying to 2.2M entities. Previous entity typing datasets rely on annotations from small groups of crowd-workers and include a small type ontology in the hundreds (Ling and Weld, 2012) and/or sacrifice label accuracy (Choi et al., 2018). We instead rely on the cumulative, cross-checked annotations from tens of thousands of active Wikidata users.

Entities in Wikidata on average are assigned 1.28 types; for entities with multiple types, not all types are necessarily relevant to a context. For example, take the following passage: *"Obama was elected to the Illinois Senate in 1996, succeeding Democratic State Senator Alice Palmer from Illinois's 13th District, which, at that time, spanned Chicago South Side neighborhoods from Hyde Park–Kenwood south to South Shore and west to Chicago Lawn."*

While Wikidata entities may have 5+ types, many are not directly relevant to a context. For example, while Barack Obama has types including *Politician*, *Jurist*, *Political Writer*, *Community Organizer*, and *Podcaster*, the latter is not relevant to the context. To teach our models to infer types relevant to the context at hand, in pre-training data we take only types that are shared between Barack Obama and other entities in the document (e.g. Alice Palmer—Politician). We have made the WikiWiki dataset publicly available on Github.[2]

Pre-training Tasks To instill type-centric knowledge from WikiWiki, we train our models to answer four types of knowledge-based questions conditioned on a passage from Wikipedia (examples in Table 2). In *entity/type discovery*, the model is tasked to recover all surface forms (mentions) that reference an entity, along with their types—this is analogous to simultaneous entity recognition and typing. *Entity typing* consists of assigning types to an entity of interest. For *entity recognition*, we follow Cao et al. (2021) by training our model to respond with an entity's full name and type when queried with a surface form. In *slot filling* we ask our model to return all entities mentioned in the passage belonging to a certain type. For multi-type entities, we use a subset of relevant types given other entities in the context (Appendix A). We treat QA as a universal format for diverse NLU tasks (McCann et al., 2018), and adopt the framework of Raffel et al. (2020) to treat each of our tasks as text-to-text generative modeling. We create 50M questions for pre-training.

[2] https://github.com/amazon-research/wikiwiki-dataset/

User:	I'm looking for a place to stay during my upcoming trip to Cambridge.
System:	I can definitely help you with that! What area are you staying in, and what is the price range you are looking for?
User:	It should be located in the west and it should be cheap.
Belief State:	[hotel price range]: cheap; [hotel area]: west

Table 3: In Dialog State Tracking (DST), a model infers the belief state of a user given the dialog history thus far, comprising slots (red) and their values (blue). In Zero-shot DST, the model must infer the correct values for slots that it has not seen during training, requiring the agent to rely on general type knowledge.

Model Architecture We use an encoder-decoder (Sutskever et al., 2014) model initialized from BART—a Transformer (Vaswani et al., 2017) language model pre-trained via de-noising autoencoding (Lewis et al., 2020a). Our model generates an answer a as a text sequence given a document D of length t_d and question q. The document is encoded via the encoder—consisting of l Transformer layers of hidden dimensionality h, each applying 16-headed self-attention—to produce $z := \text{Enc}(D) \in \mathbb{R}^{t_d \times h}$.

We train the model to perform QA via conditional language modeling. Instead of concatenating the question with the context in encoder input (Lin et al., 2021), the decoder generates a sequence consisting of the question and answer: $x = [q; a]$. We can thus cache the document encoding at inference to answer multiple questions. At training time we perform next-token prediction, calculating cross-entropy loss by maximizing the log likelihood of the question and answer conditioned on the document: $P(q, a|D) = \prod_t^T P(x_t|x_{<t}, D)$. We assess the impact of our pre-training on Base (l=12, h=768) and Large (l=24, h=1024) models.

4 Experiments

We demonstrate the effectiveness of our pre-training on two tasks that require type understand-

	# Params	R	H	A	T	X
TRADE	90M	12.6	14.2	20.1	22.4	59.2
MA-DST	90M	13.6	16.3	22.5	22.8	59.3
SUMBT	355M	16.5	19.8	22.6	22.5	59.5
GPT2-DST	355M	26.2	24.4	31.3	29.1	59.6
BART	139M	27.9	31.9	38.4	34.3	70.5
Ours (Base)	139M	*40.4*	*36.5*	*39.8*	*36.1*	*70.9*
Ours (Large)	406M	**46.7**	**38.8**	**49.8**	**37.7**	**72.1**

Table 4: Zero-shot domain adaptation JGA (%) on MultiWOZ 2.1 test set on the (R)estaurant, (H)otel, (A)ttraction, (T)rain, and Ta(X)i domains. We achieve SOTA results on all domains by significant margins.

ing: zero-shot domain generalization in dialog state tracking (DST), and fine-grained entity typing.

Zero-Shot DST The goal of Dialog State Tracking (DST) is to infer user intent and goals from conversations by filling in belief slots (Lemon et al., 2006; Wang and Lemon, 2013). In many real-world settings, DST models must be able to predict new slot values (i.e. new entities that are not present in the training corpus) and new slot types (e.g. requirements for applications in new domains). This problem setting is known as zero-shot DST (Table 3). We follow the zero-shot setting in Campagna et al. (2020): train a model on multi-domain DST data and evaluate on a held-out domain. We measure domain generalization performance via joint goal accuracy (JGA): the percent of turns in which a model successfully predicts values for all slots in the target domain. We use the MultiWOZ 2.1 benchmark (Eric et al., 2019), evaluating zero-shot JGA for the Restaurant, Hotel, Attraction, Train, and Taxi domains. At each turn, we ask the model a question about the preference for each slot. We compare against recent systems that can perform zero-shot DST: TRADE (Wu et al., 2019), MA-DST (Kumar et al., 2020), SUMBT (Lee et al., 2019), and GPT2-DST (Li et al., 2021). Our method is complementary to systems for creating synthetic in-domain dialogs (Kim et al., 2021).

As seen in Table 4, *our* type-centric pre-training allows a model to answer questions about unseen slots. BART-base itself achieves SOTA JGA across all domains, and our pre-training significantly increases the gain to 10.6% absolute / 34.8% relative JGA—despite only using one-third of the parameters. Our Large model achieves 14.9% absolute and 49.4% relative gain in JGA compared to previous SOTA. The most significant gains come in the Hotel and Restaurant domains, which contain the

	100%	50%	20%
Base (139M)	13.7	14.7	39.0
Large (406M)	0.9	1.6	4.8

Table 5: Relative gain (%) in JGA for models trained on WikiWiki vs standard BART pre-training. Our method helps more in low-data regimes and for smaller models.

most categorical slots that resemble types (e.g. cuisine, hotel type). In Table 5 we compare our models against same-size BART models at different levels of training data availability to demonstrate the additive utility of our method. Our method is particularly helpful with less fine-tuning data (low-data regimes), with average gains of 39% for small models and 4.8% for large models at 20% data availability. Gains are magnified for smaller models, affirming that our method can effectively instill type knowledge in lightweight language models.

Ultra-Fine Entity Typing Our method improves generalization in type-adjacent tasks; we next aim to infer entity types in unseen documents. In preliminary experiments on the UltraFine dataset with 11K types (Choi et al., 2018), our models underperform SOTA (24.0 vs. 49.1 F1). Manual inspection of gold labels reveals two main causes for error: 1) inaccurate labels—e.g. "rare plants" as type "bird"; and 2) inconsistent usage of gold labels: different spellings (*organization* / *organisation*) or synonyms (*car* / *automobile*) are treated as distinct and often do not collocate. This suggests that label noise in UltraFine may make it unsuitable for assessing granular, hierarchical type knowledge.

We examine these annotation errors via **human evaluation**, presenting crowd-workers with 200 contexts from UltraFine (10% of the test set). Only 68% of gold type labels were judged accurate, and 21% inaccurate. We compare gold labels against zero-shot predictions from our model in a second trial with 200 pairs. Judges preferred our predictions 51% of the time compared to 29% for gold. We observed moderate inter-annotator agreement of κ=0.4044 (Fleiss, 1971). This suggests that our models can accurately infer types, but current benchmarks do not suitably measure typing quality.

Entity Typing on WikiWiki We turn to WikiWiki to evaluate fine-grained entity typing, leveraging type labels verified by active users of Wikidata. To verify the accuracy of ground-truth type labels in the WikiWiki test set, we asked human evalua-

Entities	Model	Precision	Recall	F1
Seen	RoBERTa	62.35	59.38	60.82
	Ours	**78.13**	**72.39**	**75.15**
Unseen	RoBERTa	48.88	47.96	48.41
	Ours	**66.65**	**63.71**	**65.14**

Table 6: P/R/F1 of pred. vs. gold types on WikiWiki Test (seen) and Test New Ent (unseen entities) splits.

tors to judge the accuracy of 443 type labels from 200 randomly sampled contexts. We confirm that WikiWiki is a high-quality benchmark for entity typing, with 85% type precision assessed by human judges (compared to 68% for UltraFine).

We found that multi-label classifiers built on RoBERTa (Liu et al., 2019) that perform well on UltraFine require significant hyper-parameter tuning to output non-trivial predictions to classify our large and sparse (41K) type ontology. To perform entity typing with our model, we generate comma-delimited text sequences of types (Yang et al., 2018). This allows our models to infer and generate novel types while classifiers remain restricted to the training ontology. We confirm that our pre-training helps models better infer types for both seen (+14.3 F1) and unseen entities (+16.7 F1) in new contexts compared to classifiers (Table 6).

To investigate if our model can discover novel types, we perform another **human evaluation** over 557 such predictions from 300 contexts, with inter-annotator agreement of κ=0.4086. Our model accurately extrapolates its type knowledge beyond the training ontology—we observe 73.3% precision when inferring new types (compared to 74.5% precision for seen types), demonstrating that our pre-training enables models to reason about types beyond simple memorization. Our model discovers complex and specific scientific types, correctly proposing that anorthosite (an aluminum silicate rock) is a *metallurgical rock*[3] and that speckled tortoises are *monotrophs*.[4] This reflects the robust taxonomy of types in scientific disciplines. Our model also proposes granular categories of events, and is judged to correctly type the 2015 Tour of Taiwan as an instance of the *Tour de Taiwan* cycling race. In the future, we seek methods to automatically assess the factual accuracy of new types.

[3]rocks containing metallic compounds and properties

[4]has diet comprising one type of food (Herrera, 1976)

5 Conclusion

In this paper, we 1) propose a text-to-text pre-training scheme to instill type knowledge in language models via QA and 2) release the **WikiWiki** dataset built from Wikipedia articles and the Wikidata KG. We show that WikiWiki is larger-scale and more accurate than existing fine-grained type recognition datasets. We demonstrate that our type-centric pre-training framework allows us to train language models that can better generalize to unseen domains, entities, and types—which in turn lead to improved model performance on downstream tasks like dialog state tracking (achieving SOTA results on zero-shot DST with average gains of 14.9% joint accuracy). Our models can extrapolate type knowledge and infer novel types that humans judge to be useful and precise. As the body of human knowledge grows, we see an opportunity to use life-long learning (Parisi et al., 2019) on news and publications to expand and model the taxonomy of knowledge.

Acknowledgements

We would like to thank Stephen Rawls, Ryan Gabbard, and anonymous reviewers for providing valuable feedback on this work. We also thank Nicolas Guénon des Mesnards and Victor Soto for their help setting up MTurk for human evaluations. Work was performed during first author's internship at Amazon Alexa AI. Findings and observations are of the authors only, and do not necessarily reflect the views of Amazon or UCSD.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *NAACL-HLT*, pages 3554–3565.

Giovanni Campagna, Agata Foryciarz, Mehrad Moradshahi, and Monica S. Lam. 2020. Zero-shot transfer learning with synthesized data for multi-domain dialogue state tracking. In *ACL*, pages 122–132.

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *ICLR*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *ACL*, pages 87–96.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *ACL/IJCNLP*, pages 1790–1799.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR*. OpenReview.net.

Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, and Dilek Hakkani-Tür. 2019. Multiwoz 2.1: Multi-domain dialogue state corrections and state tracking baselines. *CoRR*, abs/1907.01669.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020a. Entities as experts: Sparse memory access with entity supervision. In *EMNLP*, pages 4937–4951.

Thibault Févry, Livio Baldini Soares, Nicholas FitzGerald, Eunsol Choi, and Tom Kwiatkowski. 2020b. Entities as experts: Sparse memory access with entity supervision. In *EMNLP*. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *EMNLP*, pages 2619–2629.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *EMNLP (Findings)*, pages 3356–3369.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: retrieval-augmented language model pre-training. *CoRR*, abs/2002.08909.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *EACL*, pages 1772–1791.

Carlos M Herrera. 1976. A trophic diversity index for presence-absence food data. *Oecologia*, 25(2):187–191.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *ACL*, pages 328–339. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP*, pages 6769–6781.

Sungdong Kim, Minsuk Chang, and Sang-Woo Lee. 2021. Neuralwoz: Learning to collect task-oriented dialogue via model-based simulation. In *ACL*, pages 3704–3717.

Adarsh Kumar, Peter Ku, Anuj Kumar Goyal, Angeliki Metallinou, and Dilek Hakkani-Tür. 2020. MA-DST: multi-attention-based scalable dialog state tracking. In *AAAI*, pages 8107–8114.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: slot-utterance matching for universal and scalable belief tracking. In *ACL*, pages 5478–5483.

Oliver Lemon, Kallirroi Georgila, James Henderson, and Matthew N. Stuttle. 2006. An ISU dialogue system exhibiting reinforcement learning of dialogue policies: Generic slot-filling in the TALK in-car system. In *EACL*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, pages 7871–7880.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*.

Shuyang Li, Jin Cao, Mukund Sridhar, Henghui Zhu, Shang-Wen Li, Wael Hamza, and Julian J. McAuley. 2021. Zero-shot generalization in dialog state tracking through generative question answering. In *EACL*, pages 1063–1074.

Zhaojiang Lin, Bing Liu, Seungwhan Moon, Paul A. Crook, Zhenpeng Zhou, Zhiguang Wang, Zhou Yu, Andrea Madotto, Eunjoon Cho, and Rajen Subba. 2021. Leveraging slot descriptions for zero-shot cross-domain dialogue state tracking. *CoRR*, abs/2105.04222.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *AAAI*. AAAI Press.

Dayiheng Liu, Yeyun Gong, Jie Fu, Yu Yan, Jiusheng Chen, Daxin Jiang, Jiancheng Lv, and Nan Duan. 2020. Rikinet: Reading wikipedia pages for natural question answering. In *ACL*, pages 6762–6771.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Robert L. Logan IV, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *ACL*, pages 5962–5971.

Yinquan Lu, Haonan Lu, Guirong Fu, and Qun Liu. 2021. KELM: knowledge enhanced pre-trained language representations with message passing on hierarchical relational graphs. *CoRR*, abs/2109.04223.

Joan Lucariello, Amy Kyratzis, and Katherine Nelson. 1992. Taxonomic knowledge: What kind and when? *Child development*, 63(4):978–998.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian J. McAuley. 2020. Interview: Large-scale modeling of media dialog with discourse patterns and knowledge grounding. In *EMNLP*, pages 8129–8141.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *EMNLP*, pages 2775–2779.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *CoRR*, abs/1806.08730.

Yasumasa Onoe and Greg Durrett. 2020. Interpretable entity representations through large-scale typing. In *Findings of EMNLP*, pages 612–624.

German Ignacio Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71.

Matthew E. Peters, Mark Neumann, Robert L. Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *EMNLP*, pages 43–54.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *EMNLP*, pages 2463–2473.

Bharadwaj Pudipeddi, Maral Mesmakhosroshahi, Jinwen Xi, and Sujeeth Bharadwaj. 2020. Training large neural networks with constant memory using a new execution algorithm. *CoRR*, abs/2002.05645.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: improving entity and relation understanding for pre-trained language models via contrastive learning. In *ACL*, pages 3350–3363. Association for Computational Linguistics.

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. *JMLR*, 21:140:1–140:67.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Schema-guided dialogue state tracking task at DSTC8. *CoRR*, abs/2002.01359.

Ahmad Sakor, Kuldeep Singh, Anery Patel, and Maria-Esther Vidal. 2020. Falcon 2.0: An entity and relation linking tool over wikidata. In *CIKM*, pages 3141–3148. ACM.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *EMNLP (Findings)*, pages 3784–3803.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Technical Report UM-CS-2012-015, University of Massachusetts, Amherst.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuanjing Huang, and Zheng Zhang. 2020a. Colake: Contextualized language and knowledge embedding. In *COLING*. International Committee on Computational Linguistics.

Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020b. ERNIE 2.0: A continual pre-training framework for language understanding. In *AAAI*. AAAI Press.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Raghuveer Thirukovalluru, Mukund Sridhar, Dung Thai, Shruti Chanumolu, Nicholas Monath, Sankaranarayanan Ananthakrishnan, and Andrew McCallum. 2021. Knowledge informed semantic parsing for conversational question answering. In *RepL4NLP*, pages 231–240, Online.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.

Denny Vrandecic. 2012. Wikidata: a new platform for collaborative data collection. In *WWW*, pages 1063–1064. ACM.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. KEPLER: A unified model for knowledge embedding and pre-trained language representation. *TACL*, 9:176–194.

Zhuoran Wang and Oliver Lemon. 2013. A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *SIGDIAL*, pages 423–432.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William S. Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and social risks of harm from language models. *CoRR*, abs/2112.04359.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *ACL*, pages 808–819.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020a. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*, pages 6397–6407.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020b. Scalable zero-shot entity linking with dense entity retrieval. In *EMNLP*, pages 6397–6407.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. In *EMNLP*. Association for Computational Linguistics.

Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Global entity disambiguation with pretrained contextualized embeddings of words and entities. *CoRR*, abs/1909.00426.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: sequence generation model for multi-label classification. In *COLING*, pages 3915–3926.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. Docred: A large-scale document-level relation extraction dataset. In *ACL*, pages 764–777.

Yang You, Jing Li, Sashank J. Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training BERT in 76 minutes. In *ICLR*. OpenReview.net.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2021. DKPLM: decomposable knowledge-enhanced pre-trained language model for natural language understanding. *CoRR*, abs/2112.01047.

A Data

We use the June 2021 Wikidata database file from `https://www.wikidata.org/wiki/Wikidata:Database_download` for raw KG data. We use English Wikipedia article HTML crawled from the same time period. While Wikidata contains multilingual definitions and labels for each node, in this paper we use only English entity and type names.

Wikipedia data was collected under the original terms of release which allow free usage of such materials for non-commercial purposes.[5] We will release WikiWiki under the same license.

When creating questions for pre-training tasks, if a question has multiple answers (e.g. multiple chemists in Table 2), the answers are a comma- and `and`-delimited sequence, in order of appearance in the context. For the entity typing question, we use the order that types appear in the Wikidata page.

B Experimental Settings

We train all of our models on a node with eight Nvidia V100 GPUs (comprising 256 GB total VRAM) and 768 GB of RAM. We optimize using Deepspeed Stage 1 (Pudipeddi et al., 2020) using FP16 and the Lamb optimizer (You et al., 2020). Experimental results, where applicable, are reported as median of 3 experiments.

Hyperparameters For pre-training, we use a learning rate of 1e-4 with a linear warm-up for the first 10% of training iterations, using an effective batch size of 960. Our models were trained on a single pass of our pre-training dataset of 50M questions, totaling 52K steps. We fine-tune models using the same learning rate schedule, using an effective batch size of 2560 and early stopping for a maximum of 10 epochs based on validation loss. We aim to establish the general ability of our pre-training scheme to instill type awareness, and thus fix hyperparameters for generative language models trained with our method without hyperparameter tuning.

As mentioned in Section 4, the RoBERTa-based classifier for entity typing on WikiWiki required significantly more hyperparameter tuning; we performed a hyperparameter sweep on batch size (512 to 2048), learning rate (1e-3 to 1e-5), optimizer

[5] `https://en.wikipedia.org/wiki/Wikipedia:Copyrights`

	# Params	R	H	A	T	X
GPT2-DST	355M	26.2	24.4	31.3	29.1	59.6
+ SGD	355M	27.7	24.9	*42.4*	**41.1**	60.3
Ours (Base)	139M	*40.4*	*36.5*	39.8	36.1	*70.9*
Ours (Large)	406M	**46.7**	**38.8**	**49.8**	*37.7*	**72.1**

Table 7: Zero-shot domain adaptation JGA (%) on MultiWOZ 2.1 test set on the (R)estaurant, (H)otel, (A)ttraction, (T)rain, and Ta(X)i domains. Compared to GPT2-DST (Li et al., 2021) augmented with out-of-domain DST data (+SGD), our Base model out-performs the augmented model in 3/5 domains and our Large model out-performs it in 4/5 domains.

(Adam vs. Lamb), and whether to freeze the encoder. We achieved best performance (as in Table 6) with a learning rate of 1e-4, the Adam optimizer, an effective batch size of 960, and with gradual unfreezing (Howard and Ruder, 2018) over 5K steps. We found gradual unfreezing to be critical for model performance, with fully frozen and fully unfrozen RoBERTa models achieving entity typing F1 scores of $\leq$ 10.0.

C Dialog State Tracking Notes

As discussed in Section 4, our method is orthogonal to and thus can be used simultaneously with techniques for creating synthetic in-domain training data for DST (Campagna et al., 2020; Kim et al., 2021). For slot queries, we use templated questions of the form: `What [domain] [slot] is the user interested in?`.

We compare our models against SOTA models for zero-shot DST on MultiWOZ 2.1. We affirm the observations of Lin et al. (2021) that while T5-DST achieves strong DST performance on the 2.0 version of the dataset, performance degrades on the 2.1 benchmark.

Li et al. (2021) also present results for GPT2-DST when training is augmented with additional DST data from a wider pool of domains—the Schema-Guided Dialog dataset (Rastogi et al., 2020). In the interest of fairness, we do not compare this setting in Table 4 as our models do not have access to *any* conversational data in pre-training and—like the other baseline models—cannot access additional DST data in fine-tuning. Despite the lack of exposure to conversational data, in Table 7 we show that our Small and Large models out-perform GPT2-DST + SGD in 3/5 domains (with absolute per-domain gain of 5.5% and relative gain of 18.3%) and 4/5 domains (with absolute gains of 9.7% and relative gains of 30.6%), respectively. We additionally present zero-shot DST performance (JGA) on the MultiWOZ 2.1 validation set in Table 8.

	# Params	R	H	A	T	X
BART-base	139M	29.6	31.5	38.7	35.0	70.5
Ours (Base)	139M	41.3	33.6	42.5	36.6	71.9
Ours (Large)	406M	46.4	37.6	52.3	38.0	72.1

Table 8: Zero-shot domain adaptation JGA (%) on MultiWOZ 2.1 ***validation*** set on the (R)estaurant, (H)otel, (A)ttraction, (T)rain, and Ta(X)i domains.

D Human Evaluation Details

We perform our evaluation using the Amazon Mechanical Turk platform.[6] To ensure high quality annotations, we recruit only crowd workers with Master qualification—indicating a history of high quality accepted work—and who are native English speakers.[7] Crowd-workers remained anonymous outside of their qualifications and we did not collect any additional demographic information. Workers were informed that their type accuracy judgements were to be used in an academic research setting, with an option to opt-out and reject the task.

As both gold types and predicted types could be complex and require domain knowledge, evaluators were instructed to search any relevant additional material (textbooks, sites, papers) to ensure they made a high confidence judgment of type accuracy. Based on the average time spent evaluating each article, our pay rate worked out to above Federal minimum wage in the United States.

In Figure 2 we display the example instructions given to a human evaluator for assessing the accuracy of a type for an entity referenced in a context. In Figure 3 we show sample instructions given to a human evaluator to choose which of two types (predicted or gold label in random order) is more suitable / applies more accurately to the referenced entity.

E Ethics

As with all models capable of generating arbitrary text sequences, models trained with our framework and tasks run the risk of outputting toxic or offensive text (Gehman et al., 2020). However, our training aims to instill type knowledge for type-

[6] `https://www.mturk.com/`
[7] `https://www.mturk.com/worker/help`

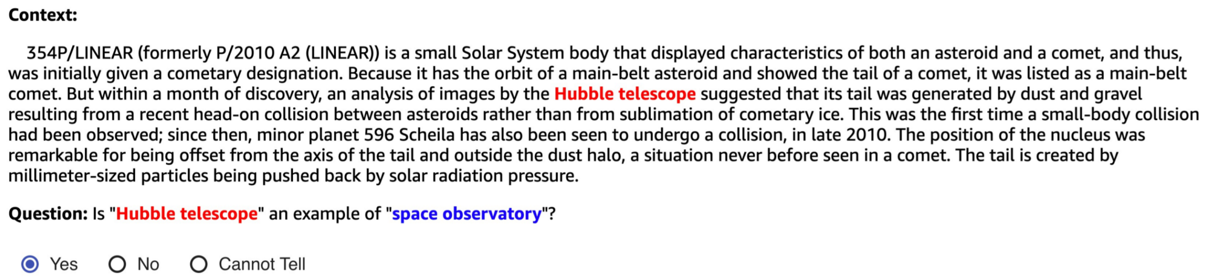

Figure 2: Example of human evaluation question where the judge is asked to assess whether a predicted / ground truth type accurately applies to the entity referenced.

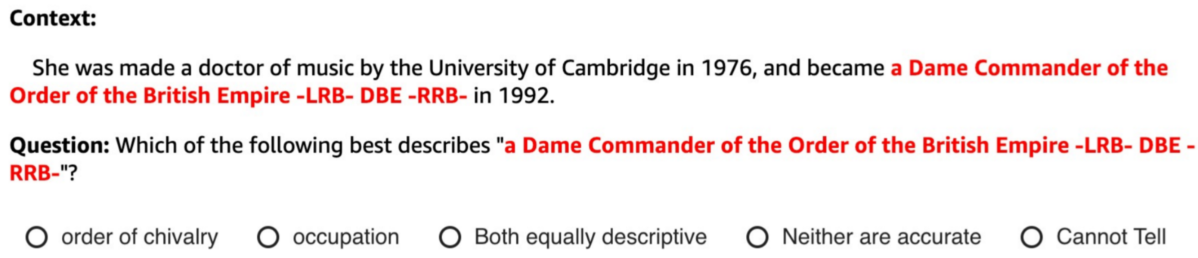

Figure 3: Example of human evaluation question where the judge is asked to assess to determine the relative suitability and quality of two different types for the entity referenced.

and concept-reliant downstream tasks. As such, we expect that our pre-training does not heighten the risk of offensive outputs compared to other general-purpose pre-training schemes on wide internet corpora.

The primary risk of instilling models with type knowledge lies in the potential for misinformation (Weidinger et al., 2021). For example, if our model is used to extend existing taxonomies, it runs the risk of hallucinating false types. We observe in Table 6 that while our model achieves high typing precision and recall for seen and unseen types in new documents, we are not at the point where it can be used in isolation to discover and add knowledge to existing knowledge graphs. In parallel with developing better methods for verifying type ontologies and assignments, it is important to incorporate domain experts or crowd-source verification when language models are used to discover facts or type relationships in new documents.

We also advocate for more careful inspection of racial, gender, and socioeconomic biases in existing type ontologies, as it is possible for type-aware models to propagate such biases (e.g. associating people with certain patterns of names with specific occupations).