All Entities are Not Created Equal: Examining the Long Tail for Ultra-Fine Entity Typing

Advait Deshmukh   Ashwin Umadi   Dananjay Srinivas   Maria Leonor Pacheco
University of Colorado Boulder
{advait.deshmukh, ashwin.umadi, dananjay.srinivas}@colorado.edu

Abstract

Due to their capacity to acquire world knowledge from large corpora, pre-trained language models (PLMs) are extensively used in ultra-fine entity typing tasks where the space of labels is extremely large. In this work, we explore the limitations of the knowledge acquired by PLMs by proposing a novel heuristic to approximate the pre-training distribution of entities when the pre-training data is unknown. Then, we systematically demonstrate that entity-typing approaches that rely solely on the parametric knowledge of PLMs struggle significantly with entities at the long tail of the pre-training distribution, and that knowledge-infused approaches can account for some of these shortcomings. Our findings suggest that we need to go beyond PLMs to produce solutions that perform well for infrequent entities.

1 Introduction

Ultra-fine entity typing (UFET) is the task of inferring the type of an entity with high specificity (Choi et al., 2018). For example, in the sentence *"Barack Obama attended Biden's presidential inauguration."*, the entity "Barack Obama" would have a coarse entity-type of "person", but a more specific, ultra-fine entity type such as "politician", "democrat" or "ex-president". Previous work on UFET relies mostly on pre-trained language models (PLMs) to elicit entity types (Li et al., 2022, 2023; Dai et al., 2021; Pan et al., 2022). The common approach is to position the UFET task close to the learning objective of the PLM. For example, Pan et al. (2022) exploit the masked language modeling objective by appending an entity-mention and mask to a sentence, allowing BERT to fill in the mask to elicit a type (e.g., Barack Obama attended Biden's presidential inauguration, Barack Obama is a [MASK]). Such approaches are attractive because they benefit from the large amount of pre-

training data the PLM has seen to make common associations, thus helping them determine the most likely type of entity based on the context provided.

Albeit successful, PLMs are limited by the data that they have been exposed to, so their reliability can be affected when working with infrequently used language constructions such as rarely/never seen entities. Since PLMs rarely encounter these entities, they have fewer opportunities to capture knowledge about them compared to frequent entities. Past work has mostly overlooked this issue, assuming that the embedding space will be enough to capture similarities between rare and less rare entities and generalize across them. Studies that have looked at infrequent entities usually characterize them in terms of their availability in task-specific training data (Choi et al., 2018; Schouten et al., 2022), rather than the data used to pre-train PLMs.

In this paper, we investigate the extent to which PLM-based methods struggle when determining the ultra-fine grained type of entities that occur infrequently in their pre-training data. This is a challenging undertaking, as the data used for pre-training PLMs are often unavailable (Shi et al., 2024). To address this challenge, we set out to answer the following research questions:

RQ1: Do internet search hits provide a good proxy for estimating relative entity frequency? To answer this question, we follow the conventional wisdom that modern PLMs have been trained on "all of the Internet", and we estimate the frequency of an entity by querying the Google search API to obtain the number of documents containing that entity. We validate this proxy by correlating it with real-world datasets known to be used in PLM pre-training, acknowledging that such disclosures are limited to only a few models.

RQ2: Do internal model representations reflect the entity frequencies in the pre-training data? To answer this question, we measure the correlation between the number of Google search

189

hits obtained for an entity and the model's probability of eliciting that entity in numerous contexts. We perform this test across various PLMs, including both masked LMs and causal LMs, and find a strong correlation across the board.

RQ3: Do PLMs struggle to type entities that are in the long tail of the pre-training distribution? Finally, we design a benchmark to answer this question.[*] To do this, we divide entities in various UFET datasets into bins based on their resulting frequencies from our Google search. Our hypothesis is that the bins consisting of the least frequent entities "in the wild" will be much harder to predict than the rest. We test this assertion by looking at the performance of two types of published approaches for UFET; PLM-only approaches, which we expect to struggle at the long tail of the pre-training distribution, and knowledge-infused approaches, which we expect can level the playing field across frequencies by leveraging external sources of information. Our results confirm our hypothesis, suggesting that we need to go beyond PLMs to produce solutions that perform well for rare, new, or infrequent entities.

2 Related Work

UFET was proposed to generate free-form noun phrases that appropriately describe the type of a target entity (Choi et al., 2018). UFET has found several downstream applications, such as coreference resolution (Durrett and Klein, 2014), entity linking (Onoe and Durrett, 2020) and relation extraction (Yaghoobzadeh et al., 2017).

PLM solutions for UFET. PLMs have been shown to capture world knowledge in their parameter space (Roberts et al., 2020; Jiang et al., 2020). This ability has allowed them to perform well on 'fill-in-the-blank' problems, where the goal is to elicit an answer to a query based on the context provided. Many UFET approaches have capitalized on this ability; Dai et al. (2021) looked at introducing Hearst patterns with [MASK] tokens to describe entity types using BERT, while Pan et al. (2022) primed BERT to produce an ultra-fine type by appending the entity mention and a [MASK] token to a sentence. While these methods report some of the best metrics for UFET, they do not explore how their approaches fare on infrequent, rare entities that PLMs may not have been sufficiently

---

exposed to. In this work, we explore the effect of entity representation strength on a variety of PLMs in order to determine whether rarer entities cause significant issues for PLM-based approaches.

Knowledge Infused UFET. The UFET dataset has 9 general, 121 fine, and 10, 201 ultra-fine types. While a large type vocabulary and the scarcity of annotated examples per type make this task especially challenging, type labels often consist of rich semantics. Li et al. (2022) leverage the type semantics and formulate the task as an NLI problem. Others have exploited the dependencies between labels (Liu et al., 2021) and hierarchies within types (Onoe et al., 2021) to supplement the PLM objectives. Such techniques often perform significantly better than PLM-only approaches.

Approaches to Estimate Entity Frequencies in Large Corpora. While there have been efforts to efficiently index large pre-training corpora to better estimate entity frequencies (Liu et al., 2024; Xu et al., 2025), we argue that there are two clear limitations with this approach: (1) the assumption that training data for models is always available and (2) the need to re-index datasets as they are updated over time. Therefore, we need simple proxies that can approximate the entity distribution "seen" by PLMs without direct access to their training data.

3 Experimental Design

Our study is composed of three experiments. First, we evaluate a proxy to establish the relative frequency of entities in pre-training data using Search Engine API indexing. Then, we examine whether parametric representations match these frequencies by correlating the likelihood of a PLM suggesting a target entity to how the term is indexed on the Web. Lastly, we select seven models to perform ultra-fine entity typing; three based on PLM objectives, three that leverage some additional knowledge, and one simple baseline. We measure typing performance across entity groups of varying frequency across the Internet data.

Establishing the Long Tail (RQ1, RQ2) The long tail of an entity distribution is characterized by a large number of entities that are rare to encounter in the real world. Assuming that the Internet is a fairly balanced representation of a real-world distribution, we use the Internet search hits as a proxy for frequency. We quantify the occurrence "in the wild" of each entity in our dataset by performing strict searches using the Google Search API.

Bin	# of Examples	Representative Entity	Avg. # of Tokens
1	301	the Baton Rouge police chief and the serial murder task force	11.63
2	301	Left fielder Carl Crawford	4.35
3	300	The Polish government	2.67
4	1095	the film	1.18

Table 1: Distribution of entities across UFET test bins

Figure 1: Entity distribution across **UFET** test bins

As most datasets used to pre-train modern PLMs come from a subset of the Internet, we hypothesize that such "long-tail" entities would be underrepresented in the parameter space of these models. To test this hypothesis, we query models and assess whether an entity is underrepresented based on its ability to be predicted or reconstructed accurately. We then measure the correlation between Internet frequencies and model-estimated probabilities.

To perform model estimations reliably, we must balance different considerations. First, given that PLMs learn word representations from in-context examples, we need a representative set of in-context examples for each of the entities that we want to estimate. To do this, we prompt *Llama3-8B-Instruct* to generate 10 different sentences that include the target entity. Second, we need to consider the training objective of the PLM when querying it for the probability of a given entity in a given context. To recover entity probabilities with Masked LMs, we first replace entity tokens with [MASK] tokens. Then, we generate a probability distribution over all candidate tokens. To deal with Causal LMs, we reframe the task as a *fill-in-the-blank* problem. We provide the model with a prompt (see App E) and compute the probabilities through a conditional generation process.

In both cases, we use the probability that the model assigns to the tokens of the target entity to represent its salience in the model's parametric knowledge. More details about the probability estimation process can be found in App. D.

Measuring Impact on Typing (RQ3) We examine how the real-world distribution of entities affects the performance of entity-typing models. To define rare entities, we calculate frequency scores for all target UFET entities using the Google Custom Search API and group them into four bins based on quartiles. Bin 1 includes the rarest entities, while Bin 4 contains the most frequent; based
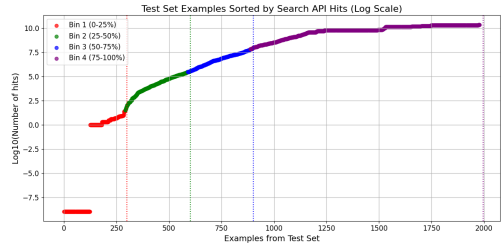
on their occurrence on the Internet. Representative examples are presented in Tab. 1. The entity distribution across the bins is visualized in Fig. 1.

We select seven representative models to test against the bin splits. The models can be roughly divided into two categories. (1) Naive PLM-based approaches that rely on PLM objectives such as MLM (BERT family of models) or Causal LM (Llama3 and Qwen3) to predict types. We include implementation details in App. G. (2) Knowledge-infused approaches that exploit additional information embedded within or relevant to the type labels. LITE (Li et al., 2022) formulates entity typing as an NLI problem. LRN (Liu et al., 2021) exploits intrinsic and extrinsic dependencies between label types. Box4Types (Onoe et al., 2021) relies on box embeddings to capture hierarchies of types. More details about these systems can be found in App G. For completeness, we also benchmark an LSTM model. This is a supervised model built on top of pre-trained word embeddings, which consider similar representation objectives to those of PLMs (word co-occurrence).
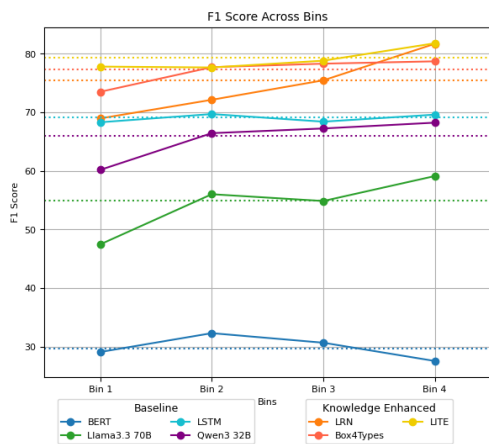
We use the crowd-annotated portion of the UFET dataset (Choi et al., 2018) for our experiments. This dataset contains entity mentions with their surrounding context and the ultra-fine types associated with them. The dataset of 5,994 tuples is divided into train/test/dev splits, each containing 1,998 tuples. We use OntoNotes (Gillick et al., 2016) as a secondary dataset, which has a train/dev/test split of 250k/2k/9k examples.

4 Evaluation and Discussion

Long Tail Analysis (RQ1, RQ2) We compare average PLM probability estimates with the number of hits the Search Engine API has for the target entity. We do this for three PLMs with both MLM and Causal LM learning objectives: BERT (Devlin et al., 2019), BART (Lewis et al., 2020) and Qwen3 (Yang et al., 2025). We calculate the probability assigned by the PLMs per in-context example
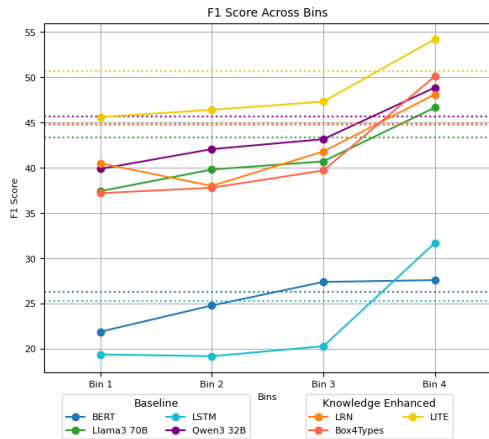
(a) UFET

(b) OntoNotes

Figure 2: Baseline vs. Knowledge-enhanced Performance across test bins

Training Corpus	Spearman Correlation
BookCorpus	0.583
C4-train	0.957
Pile-train	0.954
RedPajama	0.957
Dolma-v1.7	0.961

Table 2: Spearman correlation coefficient of **UFET** entity frequencies as estimated by search API vs counts from actual pre-training corpora
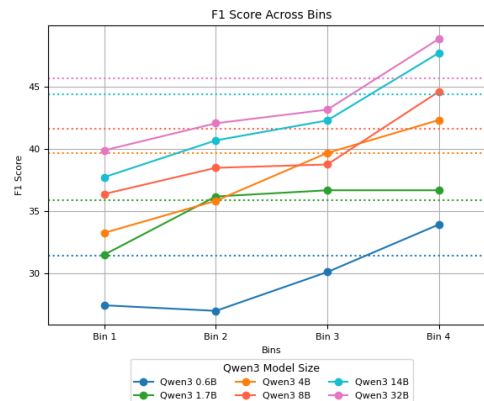

Figure 3: Effect of scaling on performance across **UFET** bins

using the method described in Section 3, and take the average across all sentences for its respective entity. We perform a correlation analysis between PLM probability estimates and the entity frequencies obtained from the Google Search API and observe high Spearman correlation coefficients (0.885 for BERT, 0.716 for BART and 0.897 for Qwen3). To visualize this, we plot the hits from the Search Engine API against the average probability for the entity obtained by each model in App. F. The high correlation between the PLM probability estimates and the number of API hits supports our hypothesis: entities that occur more/less frequently in the real world are more/less salient in PLMs.

Temporal shifts. We recognize that the Internet is constantly evolving and that temporal dynamics could potentially alter the distribution of entities. For this reason, we performed our analysis with API data capped at 2018 and at 2024, and found the results to be consistent over time, with minor changes in correlation coefficients (See App. B).

Search API vs. real pre-training datasets. We test the validity of using Internet counts to estimate the pre-training distribution by performing a correlation study between search API hits and real-world pre-training datasets such as BookCorpus (Zhu et al., 2015), C4 (Raffel et al., 2020), Pile (Gao et al., 2020; Biderman et al., 2022), RedPajama (Weber et al., 2024), Dolma-v1.7 (Soldaini et al., 2024). We estimate the entity counts for BookCorpus by performing strict searches. For Pile, we rely on the Infini-gram-mini hosted API (Xu et al., 2025). For the remaining datasets (C4, RedPajama, Dolma); we use the Infini-gram hosted API (Liu et al., 2024). N-gram counts in Infini-gram and Infini-gram-mini are case-sensitive and therefore noisy. Regardless, we see high spearman coefficients (>0.9) for all of them. While the correlation coefficient for BookCorpus is significantly lower, we find that the largest contributing factor is its much smaller scale. As a result, many entities are not present in BookCorpus. When restricting only to entities that are present in BookCorpus, we find that the correlation jumps as high as 0.88.

For completeness, we also report the Spearman correlation coefficients between the frequency of UFET entities in different pre-training datasets and the average entity recovery probability given surrounding context (see App. A).

Typing Performance (RQ3) We benchmark the seven models introduced in Sec. 3: UFET-LSTM (Choi et al., 2018), few-shot Llama3 (Dubey et al., 2024)/Qwen3 (Yang et al., 2025), BERT (PLM-based), and LITE (Li et al., 2022), LRN (Liu et al., 2021) and Box4Types (Onoe et al., 2021) (Knowledge-infused), and plot results in Fig. 2.

We find that most of the examined approaches perform better in determining the ultra-fine entity type when the target entity is more frequent (Bin4), and worse when the entity is rarer (Bin1). We note that most PLM approaches show a major decline in performance as we go down the bins (or underperforming completely, as in the case of BERT for OntoNotes). This performance decline is most underlined when moving from Bin 4 to Bin 3. Looking at the knowledge-infused approaches, we find that the enrichment from auxiliary tasks or label dependencies helps overcome the overreliance on learned entity representations for Bin 1. LITE performs the best, beating the rest by achieving an $F1$ score of 45.6 on infrequent entities and 54.2 on frequent entities for UFET, and an $F1$ score of 77.8 on infrequent entities and 81.8 on frequent entities for OntoNotes. This further highlights the advantage of the auxiliary task (NLI), which allows the model to transfer more knowledge than approaches that only rely on the PLMs parameters.

Effects of tokenizers. If a specific word is unknown to a PLM, it follows that the tokenizer will split it into multiple tokens. For this reason, we investigate whether words being split into multiple tokens causes a degradation in the probability of recovering an entity by looking at the splitting ratio (number of tokens / number of words). While we see a marked difference between a ratio of 1 and a ratio $\neq 1$, our results do not suggest any incremental effect afterwards (see Fig. 4). This suggests that our API Search proxy is a better, more nuanced indicator of entity rarity than the PLM tokenizer.

Other levels of granularity. We look at performance across frequency bins at varying entity type granularity (i.e., fine, coarse) and find the same trend of performance decay across buckets within each granularity level (see App. H).

Effect of Scaling. To understand how the size of the LM affects the typing task across bins, we evaluate Qwen3 models of different sizes (Fig. 3). We chose Qwen3 given the availability of a large range of model sizes of the same family. We find that regardless of model size, the effect of entity frequency in performance degradation is still ob-
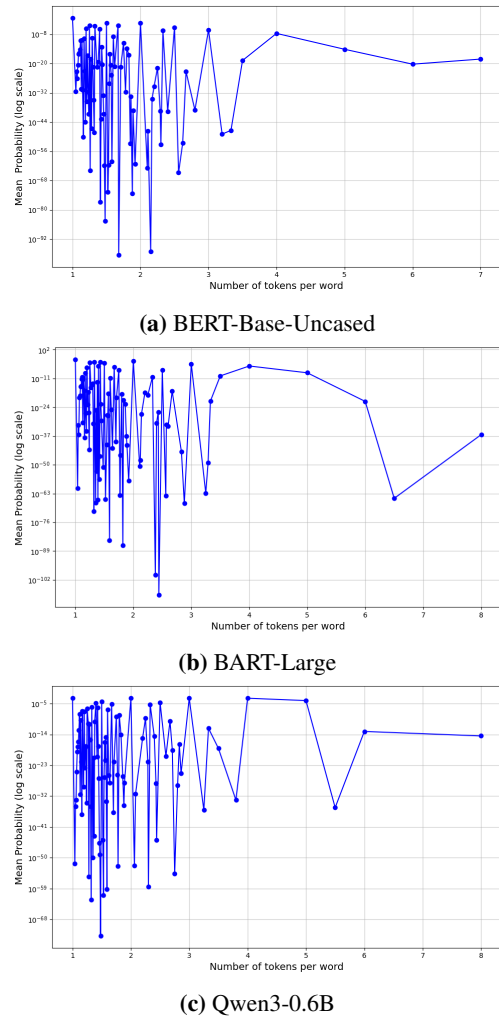

(a) BERT-Base-Uncased

(b) BART-Large

(c) Qwen3-0.6B

Figure 4: Average **UFET** entity recovery probability versus average number of tokens per word for three model tokenizers

served. We include results for Llama3 in App. G.

5 Conclusion and Future Work

We showed the effectiveness of internet search API hits as a proxy for entity frequency in large pretraining corpora (RQ1). We also showed that, as expected, this distribution significantly affects entity representations for different PLMs (RQ2). Finally, we measured the performance of several PLM-based and knowledge-infused entity typing systems on entities with different frequencies and found that all models performed worse on rarer/less-probable entities (RQ3). However, we found that all knowledge-infused systems are considerably more robust to frequency shifts than PLM-based approaches. Our results show that for PLMs to fulfill their promise for long-tail entity typing, we need better strategies to inject knowledge about rare entities into PLMs by using external resources and other forms of domain knowledge.

193

Limitations

Due to limited resources, we used quantized versions of the models when studying the impact of scaling on entity typing systems in Sec. 4. Previous work has shown that techniques such as quantization can impact the performance of LLMs, especially in low-resource settings (Diddee et al., 2022). We acknowledge that quantization may be further exacerbating the general struggle that PLMs have with entities at the long-tail of the distribution.

Ethical Considerations

To the best of our knowledge, this work does not incur any violation of the code of ethics. We used models that were Open Sourced by their authors, with code available online. All the information required to replicate our experiment is provided in the paper. We use Llama3/Qwen3, large language models whose weights may be updated by the model's creator. In such a case, we caution that some results may not be reproducible exactly, but believe that our findings will still hold.

In the interest of space, we moved some plots and details to the appendix.

References

Stella Biderman, Kieran Bicheno, and Leo Gao. 2022. Datasheet for the pile. *arXiv preprint arXiv:2201.07311*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. Ultra-fine entity typing with weak supervision from a masked language model. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1790–1799, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Harshita Diddee, Sandipan Dandapat, Monojit Choudhury, Tanuja Ganu, and Kalika Bali. 2022. Too brittle to touch: Comparing the stability of quantization and distillation towards developing low-resource MT models. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 870–885, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2016. Context-dependent fine-grained entity type tagging. *Preprint*, arXiv:1412.1820.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How Can We Know What Language Models Know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Bangzheng Li, Wenpeng Yin, and Muhao Chen. 2022. Ultra-fine entity typing with indirect supervision from natural language inference. *Transactions of the Association for Computational Linguistics*, 10:607–622.

Na Li, Zied Bouraoui, and Steven Schockaert. 2023. Ultra-fine entity typing with prior knowledge about labels: A simple clustering based strategy. *Preprint*, arXiv:2305.12802.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*.

Qing Liu, Hongyu Lin, Xinyan Xiao, Xianpei Han, Le Sun, and Hua Wu. 2021. Fine-grained entity typing via label reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4611–4622, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yasumasa Onoe, Michael Boratko, Andrew McCallum, and Greg Durrett. 2021. Modeling fine-grained entity types with box embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2051–2064, Online. Association for Computational Linguistics.

Yasumasa Onoe and Greg Durrett. 2020. Fine-grained entity typing for domain independent entity linking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8576–8583.

Weiran Pan, Wei Wei, and Feida Zhu. 2022. Automatic noisy label correction for fine-grained entity typing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4317–4323. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Stefan Schouten, Peter Bloem, and Piek Vossen. 2022. Probing the representations of named entities in transformer-based language models. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 384–393, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 116462–116492. Curran Associates, Inc.

Hao Xu, Jiacheng Liu, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. Infini-gram mini: Exact n-gram search at the internet scale with fm-index. *Preprint*, arXiv:2506.12229.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1183–1194, Valencia, Spain. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.

A Analysis of UFET entities in pre-training datasets

We report the spearman correlation values for UFET Entity counts as obtained in pre-training datasets and their recovery probabilities in Tab. 2.

B Temporal dynamics of search API hits

We use the Google Search API to approximate the distribution of entity frequencies that models have seen during training. While convenient, this approach may potentially ignore the temporal changes that might occur in the distributions of such entities. This is especially important as models we discuss in our work (BERT, BART and Qwen3) have been trained at different points within the last decade. To ensure that these dynamics do

Model	BookCorpus	Pile-train	C4-train	Dolma-v1.7	RedPajama
Bert-base-uncased	0.571	0.883	0.886	0.891	0.886
Bart-large	0.359	0.705	0.707	0.715	0.708
Qwen3-0.6B	0.514	0.895	0.897	0.902	0.901

Table 3: Spearman correlation coefficient of **UFET** entity frequencies in pretrained datasets vs Average entity recovery probability given surrounding context

Model	API till 2018	API till 2024
Bert-base-uncased	0.883	0.885
Bart-large	0.711	0.716
Qwen3-0.6B	0.895	0.897

Table 4: Spearman correlation coefficient of API entity frequencies vs Average **UFET** entity recovery probability given surrounding context (API results capped at 2018 and 2024)
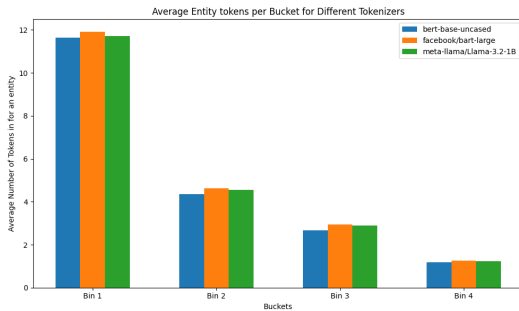

Figure 5: Average number of tokens for **UFET** test bins

not significantly impact our claims, we compare the correlation between the LM predictions and the API data capped at 2018 and 2024. We find results to be largely consistent across these two time periods (See Tab. 4). On further examination, we find that between 2018 and 2024 only 39 entities from the test set (<2%) change their bin classification. For our main results, we rank entities using the 2024 results.

C   UFET test bin distribution

To better visualize the distribution of entities across bins, we plot the log of API hits in Fig. 1. We also compare the average number of tokens as obtained with different tokenizers for each of our test bins in Fig. 5.

D   Method for Generating Masks and Calculating Entity Probability

An entity can be comprised of a single token or a multi-token phrase. For multi-token entities we employ a conditional generation approach where we generate the entity sequentially, one token at a time, moving from left to right. The probability

of the entire entity is determined by product of the conditional probability of each token t_i conditioned on all preceding tokens and the surrounding context C, where i represents the position of the token in the sequence and n is the total number of tokens comprising the entity.

$$P(t_1, t_2, \ldots, t_n \mid C) = \prod_{i=1}^{n} P\left(t_i \mid t_1, t_2, \ldots, t_{i-1}, C \right)$$

(1)

We use this approach to benchmark three models with comparable sizes and distinct pre-training objectives.

BERT (MLM): To recover an entity using BERT (bert-base-uncased), we first replace the entity with an equal number of [MASK] tokens. Then we calculate the probability the entity being recovered as described before, replacing one [MASK] per iteration.

BART (MLM): For BART (bart-large) (Lewis et al., 2020), we take a similar approach but with only a single <mask> token. We progressively expand the <mask>, one token at a time, calculating the probability for each subsequent token until the entire entity is recovered.

Qwen3 (Causal LM): Since Qwen3 (Yang et al., 2025) is not pre-trained with an MLM objective, we reframe the task as a *Fill-in-the-Blank* problem using *Qwen3-0.6B*. We provide the model with a prompt (See App. E) and compute the probabilities through a conditional generation process, one token at a time.

E   Prompt used to calculate the probability to recover an entity for Qwen3

The prompt used to calculate the probability to recover an entity for Qwen3 is:

Instruction: Fill in the appropriate entity that completes the sentence below.

Context: {sentence with the entity mention replaced by a [blank]}
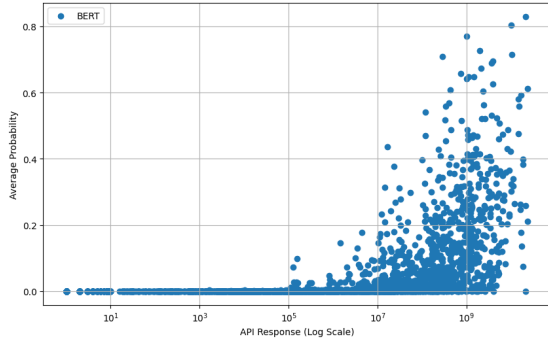
(a) BERT-Base-Uncased: Average probability vs. Log search-engine hits

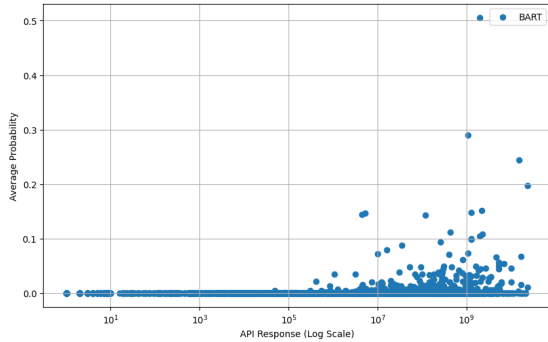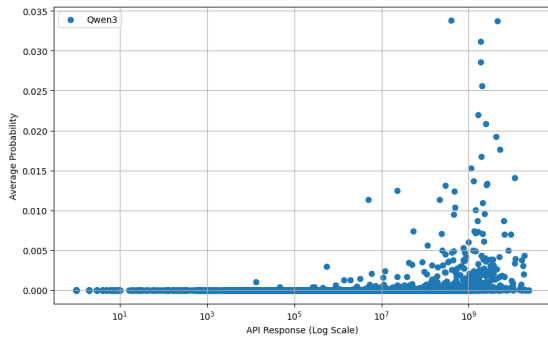**(b)** BART-Large: Average probability vs. Log search-engine hits

(c) Qwen3-0.6B: Average probability vs. Log search-engine hits

Figure 6: Average **UFET** Entity Recovery Probability vs Search API Hits

Response: `[blank]` can be replaced with:

F Scatter plots of Entity Recovery Probability in BERT, BART and Qwen3 against Search Engine API hits

To visualize the high Spearman correlation coefficient between entity recovery probabilities and search engine API hits, we plot Fig. 6.

G Models for entity typing

G.1 BERT - Baseline

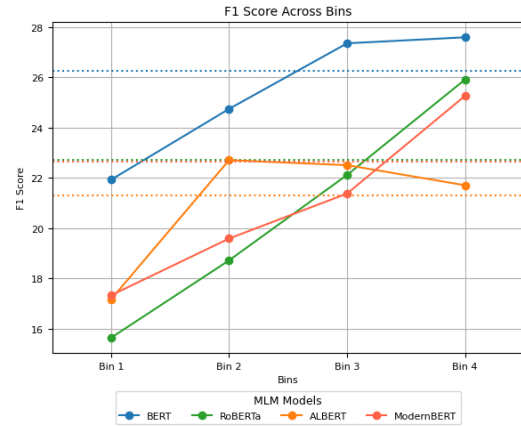Inspired by (Dai et al., 2021), we frame the typing problem as mask prediction task for BERT

Figure 7: Evaluation of MLM models across **UFET** test bins

(bert-base-uncased). We use hearst-like patterns ("[MASK] such as entity mention", "entity mention and any other [MASK]", "entity mention and some other [MASK]") and conduct experiments to find the optimal templates. The top 'n' predictions for the [MASK] (with plural → singular conversion, restricted to the type vocabulary) are considered as candidate labels. we find the optimal value for 'n' (number of labels) by experimenting with the development set.

{Hearst} n = 12, F1 = 0.0661

{Sentence} . {Hearst} n = 5, F1 = 0.2277

{Sentence} [SEP] {Hearst} n = 5, F1 = 0.2338

{Hearst} inserted in {sentence} n = 6, F1 = 0.2631

We experiment with different MLM models (See Fig. 7) and present the results from the best performing setting in Tab. 5. We find that the trend is largely consistent among them, with the exception of ALBERT which converges by Bin 3.

G.2 Llama3/Qwen3 - Baseline

We model the entity typing problem as a few-shot task for *Llama3* and *Qwen3* models to evaluate its efficacy in entity typing. We experiment with the number of examples (from the train set) in the prompts in increments of five examples. We found that the performance was optimal for 15-examples in the prompt and used that setting for the rest of our experiments.

We use the following system prompt:

```
# Entity-Typing Assistant
You are a precise entity-typing assistant.
```

197

```
Given a sentence in which **one entity mention
    ↪ is wrapped in '<ENT> ... </ENT>' tags**,
    ↪ produce **only** a JSON object whose
    ↪ single key is **"predicted_types"**.

## Guidelines
- The value must be a JSON array of strings.
- Include all the type labels that are relevant.
- Remove duplicates and keep each type concise
    ↪ (ideally a short noun phrase).
- Do not output any keys other than
    ↪ '"predicted_types"'.

## Input Format
- SENTENCE: The complete sentence with the
    ↪ target entity clearly marked with
    ↪ '<ENT>' tags
- ENTITY_MENTION: The target entity mention
    ↪ from the sentence

## Output Format
'''json
{
    "predicted_types": ["TypeA", "TypeB",
    ↪ "TypeC", ...]
}
'''
```

Followed by examples from the train set in this format:

```
# Example #{i}:
- INPUT:
- SENTENCE: '{sentence}'
- ENTITY_MENTION: '{entity_mention}'

- OUTPUT:
{{\"predicted_types\": [{types}]}}
```

With the input prompt as specified above, we generate the response from the model with the json schema for generation passed to the model (format parameter in ollama's client.chat method). Passing the generation schema ensures that the model adheres to the expected format and prevents malformed/incorrect json output. We filter out the generated types to match the type vocabulary for the respective dataset.

We also studied the effect of scaling on typing performance. Specifically, it was of interest to us to understand if scaling can help bridge performance discrepancies between Bin 1 and Bin 4. Although we did not find significant trends, the gains in performance made by the models generally seemingly favor buckets with higher entity frequency, exacerbating the long-tail problem. (see Fig. 3 and Fig. 8)

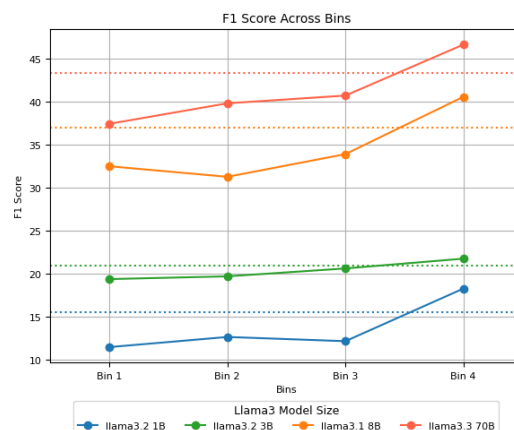Note: all the models we evaluate are the 4 bit quantized models (q4_K_M) available through Ollama.

Figure 8: Effect of scaling on model performance across **UFET** test bins

G.3 UFET-LSTM (Choi et al., 2018)

UFET-LSTM frames mention typing as predicting free-form labels (e.g., criminal, victim). Because the label set is so large and unconstrained, they adopt a neural model that represents both the mention and its context: a BiLSTM with attention for the sentence, plus a CNN with attention for the mention span. Although the authors' reported results rely on distant supervision and crowd sourced data, our analysis uses only the crowd sourced training data from UFET. Accordingly, we train a new LSTM-based model using that dataset and present the per-bin performance in Tab. 8.

G.4 LITE (Li et al., 2022)

LITE approaches the ultra-fine entity typing by reframing it as an NLI task: treat the original sentence (with the target entity) as a premise, and generate short textual descriptions of the entity as hypotheses using a predefined structure. A pretrained NLI model scores how strongly each description is entailed. Using a learning-to-rank objective, LITE distinguishes correct types from incorrect ones. For our experiments to find the performance across bins, we use the final results as shared by the authors on the test dataset of crowd sourced UFET. Table 11 uses these predictions while capturing its performance. For OntoNotes, we recreated the model to the best of our ability.

G.5 Box4Types (Onoe et al., 2021)

Box4Types deploys box embeddings to effectively capture the hierarchies of types. The model represents both types and entity mentions as boxes. Each mention and its context are fed into a BERT-based model to embed that mention in our box

space; essentially, the model leverages typological clues present in the surface text to hypothesize a type representation for the mention. This helps the model capture latent hierarchies better than the vector-based counterparts. We recreated the original approach described in paper and evaluated it against our bins in Tab. 10.

G.6 Label Reasoning Network (Liu et al., 2021)

Label Reasoning Network sequentially reasons about fine-grained entity labels by discovering and exploiting knowledge about label dependencies entailed by the data. These implicitly and explicitly entailed dependencies provide critical information which help the model overcome limitations of baseline LM approaches. The BERT based model leverages deductive and inductive reasoning. We recreated the models (without retrieval) as described in the paper and reported the results against our bins in Tab. 9.

H Fine grained evaluation of the models studied

We look at the performance of the discussed models across bins and label granularities (Coarse, Fine, Ultra-fine) as first proposed by (Choi et al., 2018).

The trend of decline in performance between Bin 4 to Bin 1 continues into the fine grained evaluation for the models. For each level of label granularity we find clear separation in performance levels as we move between the bins, suggesting that binning provides a measure of difficulty independent of the label granularity. This provides a unique opportunity to approach the entity typing task from a new perspective.

Table 5: BERT MLM (bert-base-uncased)

Subset	Overall			Coarse			Fine			Ultra-fine		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	23.7	29.5	26.3	64.9	43.0	51.7	34.2	40.7	37.1	16.8	22.5	19.2
Bin 1	19.2	25.5	21.9	59.0	41.8	48.9	22.4	34.3	27.1	14.1	19.0	16.2
Bin 2	22.5	27.7	24.8	57.6	34.2	43.0	31.7	43.2	36.6	17.3	23.2	19.8
Bin 3	24.6	31.0	27.4	60.1	41.0	48.7	34.4	42.1	37.9	18.8	25.1	21.5
Bin 4	25.1	30.8	27.6	69.1	45.8	55.1	38.2	41.1	39.6	16.8	22.5	19.2

Table 6: Llama3.3-70B

Subset	Overall			Coarse			Fine			Ultra-fine		
	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)
Full Test	45.9 ± 0.1	41.1 ± 0.1	43.4 ± 0.1	78.2 ± 0.3	68.7 ± 0.2	73.1 ± 0.3	58.0 ± 0.3	56.0 ± 0.2	57.0 ± 0.2	36.6 ± 0.2	30.0 ± 0.2	33.0 ± 0.2
Bin 1	38.3 ± 0.3	36.6 ± 0.7	37.4 ± 0.5	70.7 ± 0.5	54.8 ± 0.7	61.7 ± 0.4	48.5 ± 0.7	44.4 ± 1.0	46.4 ± 0.9	32.1 ± 0.3	30.5 ± 0.7	31.3 ± 0.5
Bin 2	43.2 ± 0.3	36.9 ± 0.3	39.8 ± 0.2	73.7 ± 0.8	50.9 ± 0.9	60.2 ± 0.9	55.0 ± 1.0	54.8 ± 0.8	54.9 ± 0.5	36.5 ± 0.7	30.0 ± 0.3	33.0 ± 0.4
Bin 3	44.4 ± 0.2	37.6 ± 0.3	40.7 ± 0.2	75.7 ± 1.0	59.5 ± 1.1	66.7 ± 1.1	59.4 ± 1.3	53.0 ± 0.9	56.0 ± 1.0	34.7 ± 0.4	28.4 ± 0.2	31.2 ± 0.2
Bin 4	49.0 ± 0.2	44.5 ± 0.1	46.6 ± 0.1	80.9 ± 0.3	78.2 ± 0.5	79.5 ± 0.4	60.4 ± 0.1	59.7 ± 0.4	60.1 ± 0.2	38.4 ± 0.2	30.4 ± 0.3	33.9 ± 0.3

Table 7: Qwen3-32B

Subset	Overall			Coarse			Fine			Ultra-fine		
	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)	P($\pm\sigma$)	R($\pm\sigma$)	F1($\pm\sigma$)
Full Test	50.6 ± 0.6	41.6 ± 0.6	45.7 ± 0.3	77.6 ± 0.5	75.1 ± 0.9	76.4 ± 0.3	57.6 ± 0.8	55.9 ± 0.9	56.7 ± 0.4	39.5 ± 0.5	29.0 ± 0.5	33.5 ± 0.4
Bin 1	41.1 ± 1.0	38.7 ± 0.8	39.9 ± 0.7	67.0 ± 0.9	67.5 ± 0.8	67.2 ± 0.8	48.0 ± 1.7	48.5 ± 1.5	48.2 ± 1.4	33.2 ± 0.7	29.5 ± 0.9	31.3 ± 0.7
Bin 2	46.1 ± 0.6	38.7 ± 0.6	42.1 ± 0.3	74.6 ± 1.4	66.6 ± 1.1	70.4 ± 0.8	52.5 ± 0.7	53.9 ± 1.9	53.2 ± 0.7	37.8 ± 0.8	28.6 ± 0.3	32.6 ± 0.4
Bin 3	48.4 ± 1.1	39.0 ± 0.6	43.2 ± 0.4	71.7 ± 1.8	68.7 ± 0.6	70.1 ± 0.6	57.6 ± 1.2	54.6 ± 1.4	56.1 ± 1.2	38.5 ± 1.0	28.4 ± 1.0	32.7 ± 0.8
Bin 4	55.0 ± 0.6	44.0 ± 0.7	48.9 ± 0.3	82.2 ± 0.4	80.2 ± 1.0	81.2 ± 0.5	61.2 ± 0.9	58.3 ± 1.4	59.7 ± 0.7	42.1 ± 0.6	29.1 ± 0.7	34.4 ± 0.4

Table 8: LSTM

Subset	Overall			Coarse			Fine			Ultra-fine		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full test	41.7	18.2	25.3	57.3	52.8	55.0	41.7	16.2	23.4	27.3	7.7	12.0
Bin 1	33.4	13.7	19.4	42.3	39.9	41.0	18.2	8.1	11.2	24.4	5.7	9.2
Bin 2	34.3	13.3	19.2	42.4	38.4	40.3	45.6	19.6	27.4	29.2	7.0	11.4
Bin 3	32.6	14.7	20.3	46.4	40.9	43.5	43.8	15.4	22.8	25.0	7.2	11.1
Bin 4	47.5	23.8	31.7	69.4	63.9	66.6	45.6	21.0	28.8	28.0	10.5	15.3

Table 9: Label Reasoning Network

Subset	Overall			Coarse			Fine			Ultra-fine		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	57.1	33.7	42.4	75.7	76.6	76.1	57.5	46.8	51.6	44.0	19.8	27.3
Bin 1	52.7	27.4	36.0	65.8	69.5	67.6	52.8	42.1	46.9	43.4	14.5	21.8
Bin 2	49.2	26.8	34.7	65.6	67.7	66.7	47.4	38.8	42.7	37.1	15.7	22.0
Bin 3	52.3	27.9	36.4	67.3	68.7	68.0	55.0	37.6	44.7	38.8	17.3	23.9
Bin 4	61.5	38.9	47.7	82.6	82.0	82.3	61.4	52.3	56.5	47.0	23.2	31.1

Table 10: Box4Types

Subset	Overall			Coarse			Fine			Ultra-fine		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	52.8	38.9	44.8	70.5	82.9	76.2	52.9	53.4	53.2	45.4	24.5	31.8
Bin 1	46.3	31.0	37.2	58.9	77.7	67.0	42.9	43.6	43.2	38.8	16.0	22.7
Bin 2	47.7	31.4	37.8	60.0	76.7	67.3	45.3	46.3	45.8	44.5	20.5	28.0
Bin 3	47.8	33.9	39.7	60.2	79.9	68.7	49.6	53.8	51.6	42.8	20.6	27.8
Bin 4	57.4	44.4	50.1	79.4	86.2	82.6	58.3	57.3	57.8	47.7	29.0	36.1

Table 11: LITE

	Overall			Coarse			Fine			Ultra-fine		
Subset	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Full Test	54.8	47.1	50.7	74.5	81.7	77.9	61.4	57.2	59.3	44.3	35.5	39.4
Bin 1	50.7	41.4	45.6	64.7	76.3	70.0	53.8	42.4	47.4	41.0	31.8	35.8
Bin 2	49.7	43.5	46.4	65.7	76.0	70.4	61.1	56.2	58.6	41.9	34.1	37.6
Bin 3	49.2	45.6	47.3	63.9	73.9	68.6	57.1	53.7	55.4	43.3	36.3	39.5
Bin 4	58.8	50.2	54.2	81.7	86.0	83.8	64.2	61.7	62.9	46.1	36.7	40.8