Metadata Shaping: Natural Language Annotations for the Long Tail

Anonymous ACL submission

Abstract

Language models (LMs) struggle to capture knowledge about rare entities. To better capture entity knowledge, a common procedure in prior work is to start with a base LM such as BERT and to modify the LM architecture or objective function to produce a *knowledge-aware LM*. Proposed knowledge-aware LMs perform well compared to base LMs on entity-rich tasks; however deploying, understanding, and maintaining many different specialized architectures is challenging, and they also often introduce additional computational costs. Thus we ask: to what extent we can match the quality of these architectures using a base LM and only changing the data? We propose *metadata shaping*, a method which inserts readily available entity metadata, such as descriptions and categorical tags, into examples at train and inference time based on mutual information. Intuitively, if metadata corresponding to popular entities overlap with metadata for rare entities, the LM may be able to better reason about the rare entities using patterns learned from similar popular entities. On standard entity-rich tasks (TACRED, FewRel, OpenEntity), metadata shaping exceeds the BERT-baseline by an average of 4.3 F1 points and achieves state-of-the-art results. We further show the gains are on average 4.4x larger for the slice of examples containing tail vs. popular entities.

1 Introduction

While recent language models (LMs) are remarkable at learning patterns that are seen frequently during training, they still suffer from performance degradation over rare "tail" patterns. In this work, we propose metadata shaping, a method to address the tail challenge by encoding properties shared between popular and tail examples *in the data itself*.

Tail degradation is particularly apparent in *entity-rich tasks*, since users care about very different entities and facts when using popular entity-centric applications such as search and personal assistants (Bernstein et al., 2012; Poerner et al., 2020; Orr et al., 2020). Given the importance of entity-rich applications, considerable effort has been devoted to developing methods for capturing entity knowledge more reliably. Many of the proposed approaches proceed by changing the model architecture.

While early LMs relied heavily on feature engineering (Lafferty et al., 2001; Mintz et al., 2009; Zhang and Nivre, 2011), recent neural models could learn useful features during training (Collobert et al., 2011), and shifted the focus to model engineering. As such, given data annotated with entity tags, recent methods for reasoning about entities typically enrich *base LMs* (e.g., Devlin et al. (2019)) to produce *knowledge-aware LMs*. This is accomplished by modifying the base LM architecture or introducing auxiliary objectives to better capture entity properties (Zhang et al., 2019; Peters et al., 2019; Yamada et al., 2020; Wang et al., 2020b; Xiong et al., 2020; Su et al., 2021), or by combining multiple learned modules, which are each specialized to handle fine-grained reasoning patterns or subsets of the data distribution (Chen et al., 2019; Wang et al., 2020a).

These knowledge-aware LMs have led to impressive gains compared to base LMs on entity-rich tasks. However they also raise a few challenges: additional pretraining is expensive, deploying and maintaining multiple specialized LMs can be memory and time-intensive, each LM needs to be optimized for efficient use (e.g., on-device), and the methods require training to adapt to changing facts or new entities. Further, these LMs learn about an entity from its individual occurrences during training, rather than explicitly encoding that patterns learned for one entity may be useful for reasoning about other similar entities. Implicitly learning entity similarities may be data-inefficient as the Wikidata knowledge base alone holds $\sim$ 100M entities and unfortunately, 89% of the Wikidata entities do not appear in Wikipedia, a popular source

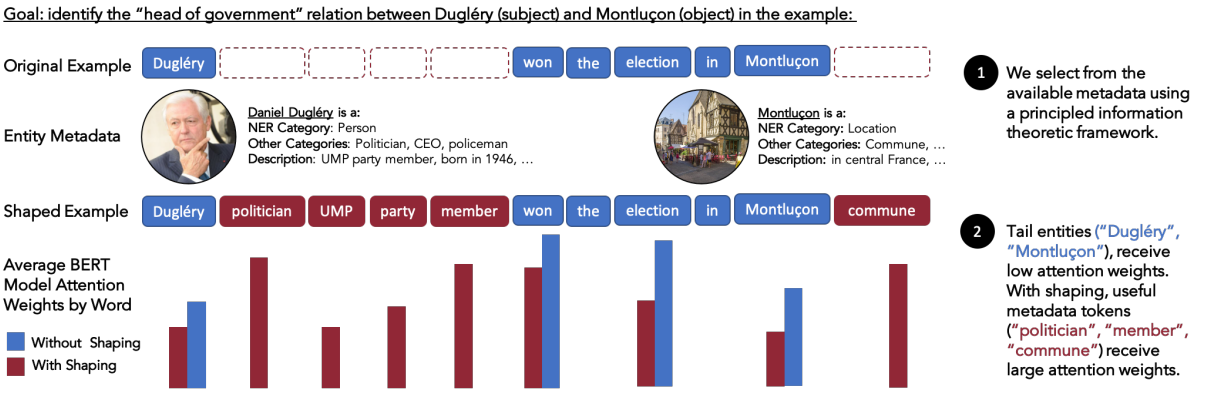

Figure 1: Metadata shaping inserts metadata (e.g., entity categories and descriptions) into the data at train and test time. The FewRel benchmark involves identifying the relation between a subject and object string. The above subject and object are unseen in the FewRel training data and the tuned base LM places low attention weights on those tokens. A base LM trained with shaped data places high attention weights on useful metadata tokens such as "politician". Weights are shown for words which are not stop-words, punctuation, or special-tokens.

of unstructured training data for the LMs, at all. [1]

Thus, we ask the question: to what extent can we match the quality of these architectures using a base LM architecture and only changing the data? We propose some simple modifications to the data at train and test time, which we call *metadata shaping*, and find the method is surprisingly quite effective. Given unstructured text, there are several readily available tools for generating annotations at scale (e.g., Manning et al. (2014); Honnibal et al. (2020)), and knowledge bases contain entity metadata including categorical tags (e.g., Barack Obama is a "politician") and descriptions (e.g., Barack Obama "enjoys playing basketball"). Metadata shaping entails explicitly inserting retrieved entity metadata in examples as in Figure 1 and inputting the resulting *shaped examples* to the LM. Our contributions are:

Simple and Effective Method We propose metadata shaping and demonstrate its effectiveness on entity-rich benchmarks. Metadata shaping, with simply an *off-the-shelf* base LM, exceeds the base LM trained on unshaped data by by an average of 4.3 F1 points and is competitive to state-of-the-art methods, which do modify the LM. Metadata shaping thus enables re-using well-studied and optimized base LMs (e.g., Sanh et al. (2020)).

Tail Generalization We show that metadata shaping improves tail performance — the observed gain from shaping is on average 4.4x larger for the slice of examples containing tail entities than for the slice containing popular entities. Metadata establish "subpopulations", groups of entities sharing similar properties, in the entity distribution (Zhu et al., 2014; Cui et al., 2019; Feldman, 2021). For example on the FewRel benchmark (Han et al., 2018), "Daniel Dugléry" (a French politician) appears 0 times, but "politician" entities in general appear > 700 times in the task training data. Intuitively, performance on a rare entity should improve if the LM has the explicit information that it is similar to other entities observed during training. This is data-efficient compared to learning the properties of an entity from its individual occurrences.

Explainability Prior knowledge-aware LMs use metadata (Peters et al., 2019; Alt et al., 2020), but do not explain when and why different metadata help. Broadly there are limited tools for reasoning about model architectural changes. This work instead takes a data-centric perspective and, inspired by classic feature selection techniques (Guyon and Elisseeff, 2003), we conceptually explain the effect of different metadata on generalization error.

We hope this work motivates further research on addressing the tail challenge through the data.

2 Method

This section introduces metadata shaping, including the set up and conceptual framework.

2.1 Objective

The goal of metadata shaping is to improve tail performance using properties shared by popular and rare examples. For instance, this work investigates

[1] Orr et al. (2020) finds that a BERT based model needs to see an entity in on the order of 100 samples to achieve 60 F1 points when disambiguating the entity in Wikipedia text.

improving tail performance in entity-rich tasks using entity metadata (e.g., "politician") which are readily available for *tail* (e.g., "Daniel Dugléry") and popular *head* (e.g., "Barack Obama") entities. Tail entities are those seen < 10 times during training and head entities are seen ≥ 10 times, consistent with Orr et al. (2020); Goel et al. (2021).

Metadata are easily sourced using off-the-shelf models such as those for named entity (NER, NEL) or part-of-speech (POS) tagging (Manning et al., 2014; Honnibal et al., 2020), heuristic rules, and knowledge bases (KBs) (e.g., Wikidata, Wordnet (Miller, 1995), and domain-specific KBs (Bodenreider, 2004)) allowing us to annotate unstructured text at scale. We draw inspiration from prior work that aligns text to metadata (Mintz et al., 2009), though instead of using predefined feature schemas, we consider using an unrestricted set of metadata.

2.2 Set Up

Let input $x \in \mathcal{X}$ and label $y \in \mathcal{Y}$, and consider the classification dataset $\boldsymbol{D} = \{(x_i, y_i)\}_{i=1}^n$ of size n. Let $m \in \mathcal{M}$ denote a metadata tag and let $\boldsymbol{M(x_i)}$ be the set of metadata collected for example x_i. A shaping function $f_s : \mathcal{X} \rightarrow \mathcal{X}_s$ accepts an original example $x_i \in \mathcal{X}$ and produces a shaped example $s_i \in \mathcal{X}_s$ by inserting a subset of $\boldsymbol{M(x_i)}$ into x_i (see Figure 1). The downstream classification model $\hat{p}_\phi$ is learned from shaped train examples and infers y_i from the shaped test examples.

This work uses the following representative metadata shaping functions for all tasks to insert a range of *coarse-grained* signals associated with groups of examples to *fine-grained specific* signals associated with individual examples:

Categorical tokens establish subpopulations of entities (e.g., *Dugléry* falls in the coarse grained category of "person" entities, or finer grained category of "politician" entities). NER and POS tags are coarse grained categories, and knowledge bases contain finer-grained categories (i.e., entity types and relations). Categories are consistent and frequent compared to words in the original examples.

Description tokens give cues for rare entities and alternate expressions of popular entities (e.g., *Dugléry* is a "UMP party member"). Descriptions are likely unique across entities, and can be viewed as the finest-grained category for an entity.

2.3 Conceptual Framework

Next we want to understand if inserting $m \in \boldsymbol{M(x_i)}$ for $x_i \in \boldsymbol{D}$ can improve tail performance. We measure the generalization error of the classification model $\hat{p}_\phi$ using the cross-entropy loss:

$$\mathcal{L}_{\text{cls}} = \mathbb{E}_{(x,y)}\big[-\log(\hat{p}_\phi(y|x))\big]. \quad (1)$$

Let $\Pr(y|x_i)$ be the true probability of class $y \in Y$ given x_i. Example x_i is composed of a set of patterns $\boldsymbol{K_i}$ (i.e., subsets of tokens in x_i). We make the assumption that a pattern $k \in \boldsymbol{K_i}$ is a useful signal if it informs $\Pr(y|x_i)$. We thus parametrize the true distribution $\Pr(y|x_i)$ using the principle of maximum entropy (Berger et al., 1996):

$$\Pr(y|x_i) = \frac{1}{Z(x_i)} \exp\Big(\sum_{k \in \boldsymbol{K_i}} \lambda_k \Pr(y|k)\Big). \quad (2)$$

where λ_k represents learned parameters weighing the contributions of patterns (or events) k and $Z(x_i)$ is a partition function that ensures $\Pr(y|x_i)$ represents a probability distribution. Therefore when evaluating $\hat{p}_\phi$, achieving zero cross-entropy loss between the true probability $\Pr(y|k)$ and the estimated probability $\hat{p_\phi}(y|k)$, for all k, implies zero generalization error overall.

Unseen Patterns Our insight is that for a pattern k that is unseen during training, which is common in entity-rich tasks,[2] the class and pattern are independent ($y \perp k$) under the model's predicted distribution $\hat{p}_\phi$, so $\hat{p}_\phi(y|k) = \hat{p}_\phi(y)$. With the assumption of a well-calibrated model and not considering priors from the base LM pretraining stage,[3] this probability is $\hat{p}_\phi(y) = \frac{1}{|\mathcal{Y}|}$ for $y \in \mathcal{Y}$.

Plugging in $\hat{p}_\phi(y) = \frac{1}{|\mathcal{Y}|}$, the cross-entropy loss between $\Pr(y|k)$ and $\hat{p}_\phi(y|k)$ is $\Pr(k) \log |Y|$. Our idea is to effectively replace k with another (or multiple) *shaped* pattern k', which has non-uniform $\hat{p}_\phi(y|k')$ and a lower cross-entropy loss with respect to $\Pr(y|k')$, as discussed next.

Inserting Metadata Consider the shaped example, $s_i = f_s(x_i)$, which contains new tokens from $\boldsymbol{M(x_i)}$, and thus contains a new set of patterns $\boldsymbol{K_i^s}$. Let $k_m \in \boldsymbol{K_i^s}$ be a pattern containing some $m \in \boldsymbol{M(x_i)}$. For a rare pattern (e.g., a mention of a rare entity in x_i) k, if an associated pattern k_m (e.g., a metadata token for the rare entity) occurs non-uniformly across classes during training,

[2] For example, on the FewRel benchmark used in this work, 90.7%/59.7% of test examples have a subject/object span which are unseen as the subject/object span during training.

[3] We ignore occurrences in the pretraining corpus and learned similarities between unseen k and seen k'. Future work can use these priors to refine the slice of unseen entities.

Algorithm 1 Metadata Token Selection

Precompute Train Statistics
Input: training data $\boldsymbol{D}_{train}$, metadata M
for each category $m \in M$ over $\boldsymbol{D}_{train}$ **do**
    Compute $\text{pmi}(y, m)$ for $y \in \mathcal{Y}$.
end for
for each class $y \in \mathcal{Y}$ over D_{train} **do**
    Compute frequency f_y.
end for

Select Metadata for Sentence
Input: x_i from $\boldsymbol{D}_{train}$ and $\boldsymbol{D}_{test}$, integer n.
Collect metadata $\boldsymbol{M(x_i)}$ for x_i.
for $m \in \boldsymbol{M(x_i)}$ **do**
    Compute $r_y = 2^{\text{pmi}(m,y)} f_y$ for $y \in \mathcal{Y}$.
    Normalize r_y values to sum to 1.
    Compute entropy H_m over r_y for $y \in \mathcal{Y}$.
end for
Rank $m \in \boldsymbol{M(x_i)}$ by H_m.
Return $\min(n, |\boldsymbol{M(x_i)}|)$ tokens with lowest H_m.

then the cross-entropy loss between $\hat{p}_\phi(y|k_m)$ and $\Pr(y|k_m)$ is lower than the cross-entropy loss between $\hat{p}_\phi(y|k)$ and $\Pr(y|k)$. If k_m provides a high quality signal, shifting $\hat{p}_\phi(y|x_i)$ in the correct direction, performance of $\hat{p}_\phi$ should improve.

We can measure the non-uniformity of k_m across classes using the conditional entropy $\hat{H}(\mathcal{Y}|k)$. When k is unseen and $\hat{p}_\phi(y|k) = \hat{p}_\phi(y, k) = \hat{p}_\phi(y) = \frac{1}{|\mathcal{Y}|}$ (uniform), so $\hat{H}(\mathcal{Y}|k)$ is maximized:

$$\hat{H}(\mathcal{Y}|k) = -\sum_{y \in \mathcal{Y}} \hat{p}_\phi(y, k) \log \hat{p}_\phi(y|k) = \log(|\mathcal{Y}|). \quad (3)$$

For non-uniform $\hat{p}_\phi(y|k_m)$, this conditional entropy decreases. Thus we connect the method of using metadata for the tail to classical feature selection methods (Guyon and Elisseeff, 2003); we seek the metadata providing the largest information gain. Next we discuss the practical considerations for selecting metadata.

Metadata Selection Entities are associated with large amounts of metadata $\boldsymbol{M(x_i)}$ — categories can range from coarse-grained (e.g., "person") to fine-grained (e.g., "politician" or "US president") and there are intuitively many ways to describe people, organizations, sports teams, and other entities. Since certain metadata may not be helpful for a task, and popular base LMs do not scale very well to long dependencies (Tay et al., 2020; Pascanu et al., 2013), it is important to understand *which* metadata to use for shaping.

We want to select k_m with non-uniform $\hat{p}_\phi(y|k_m)$ across $y \in \mathcal{Y}$, i.e. with lower $\hat{H}(\mathcal{Y}|k_m)$. Conditional probability $\Pr(y|k_m)$ is defined as:

$$\Pr(y|k_m) = 2^{\text{pmi}(y,k_m)} \Pr(y), \quad (4)$$

where we recall that the pointwise mutual information $\text{pmi}(y, k_m)$ is defined as $\log\left(\frac{\Pr(y,k_m)}{\Pr(y)\Pr(k_m)}\right)$. The pmi compares the probability of observing y and k_m together (the joint probability) with the probabilities of observing y and k_m independently. Class-discriminative metadata reduce $\hat{H}(\mathcal{Y}|k)$.

Directly computing the resulting conditional probabilities after incorporating metadata in $\boldsymbol{D}$ is challenging since the computation requires considering all patterns contained in all examples, generated by including m. Instead we use simplistic proxies to estimate the information gain. In Algorithm 1, we focus on the subset of $\boldsymbol{K_i^s}$ containing individual metadata tags m, and compute the entropy over $\hat{p}_\phi(y|m)$ for $y \in \mathcal{Y}$. Simple extensions to Algorithm 1, at the cost of additional computation, would consider a broader set of k_m (e.g., n-grams containing m for $n > 1$), or *iteratively* select tokens by considering the correlations in the information gain between different metadata tags.

3 Experiments

In this section, we demonstrate that metadata shaping is general and effective.

3.1 Datasets

We evaluate on standard entity-typing and relation extraction benchmarks used by baseline methods. **Entity typing** involves predicting the the applicable *types* for a given substring in the input example from a set of output types. We use OpenEntity (9 output types) (Choi et al., 2018) for evaluation. **Relation extraction** involves predicting the relation between the two substrings in the input example, one representing a subject and the other an object. We use FewRel (80 output relations) and TACRED Revisited (42 output relations) for evaluation (Han et al., 2018; Zhang et al., 2017; Alt et al., 2020).

3.2 Experimental Settings

Model We fine-tune a BERT-base model on metadata shaped data for each task, taking the pooled

Model	FewRel			TACRED			OpenEntity		
	P	R	F1	P	R	F1	P	R	F1
BERT-base	85.1	85.1	84.9	66.3	78.7	72.0	76.4	71.0	73.2
K-BERT	83.1	85.9	84.3	-	-	-	76.7	71.5	74.0
ERNIE	88.5	88.4	88.3	74.8	77.1	75.9	78.4	72.9	75.6
E-BERT$_{concat}$	88.5	88.5	88.5	-	-	-	-	-	-
KnowBERT$_{Wiki}$	89.2	89.2	89.2	**78.9**	**76.9**	**77.9**	78.6	71.6	75.0
CokeBERT	89.4	89.4	89.4	-	-	-	78.8	**73.3**	75.6
Ours (BERT-base)	**90.2**	**90.2**	**90.2**	77.0	76.3	76.7	**79.3**	**73.3**	**76.2**

Table 1: Test scores on standard relation extraction and entity-typing tasks. "Ours (Base LM)" is metadata shaping. All methods use the same base LM (BERT-base) and external information (Wikipedia) for consistent comparison. A dash ("-") indicates the baseline method did not report scores for the task.

[CLS] representation and using a linear prediction layer for classification (Devlin et al., 2019). We use cross-entropy loss for FewRel and TACRED and binary-cross-entropy loss for OpenEntity. All test scores are reported at the epoch with the best validation score and we use the scoring implementations released by (Zhang et al., 2019). Additional training details are provided in appendix A.

Metadata Source We use the state-of-the-art pretrained entity-linking model from Orr et al. (2020) to link the text in each task to an October 2020 dump of Wikidata. We use Wikidata and the first sentence of an entity's Wikipedia page to obtain descriptions. Additional details for the source of metadata are in Appendix A. For certain examples in the tasks, there are no linked entities (e.g., several subject or object entities are simply pronouns or dates). Table 3 gives statistics for the number of examples with available of metadata for each task. Metadata tags are selected by Algorithm 1.

3.3 Baselines

Prior work proposes various knowledge-aware LMs, which are currently the state-of-the-art for the evaluated tasks. **ERNIE**, (Zhang et al., 2019) **LUKE** (Yamada et al., 2020), **KEPLER** (Wang et al., 2020b), **CokeBERT** (Su et al., 2021), and **WKLM** (Xiong et al., 2020) introduce auxiliary loss terms and require additional pretraining. Prior approaches also modify the architecture for example using alternate attention mechanisms (**KnowBERT** (Peters et al., 2019), **K-BERT** (Liu et al., 2020), **LUKE**) or training additional transformer stacks to specialize in knowledge-based reasoning (**K-Adapter** (Wang et al., 2020a)). **E-BERT** (Poerner et al., 2020) does not require additional pretraining and uses entity embeddings which are aligned to the word embedding space. In Table 1, we compare to methods which use the same base LM, BERT-base, and external information resource, Wikipedia, for consistency.

3.4 End-to-End Benchmark Results

We simply use an *off-the-shelf* BERT-base LM (Wolf et al., 2020), with no additional pretraining and fine-tuned on shaped data to exceed the BERT-base LM trained on unshaped data by 5.3 (FewRel), 4.7 (TACRED), and 3.0 (OpenEntity) F1 points. Metadata shaping is also competitive with SoTA baselines which *do* modify the BERT-base LM. Results are shown in Table 1. Table 3 reports the availability of metadata for each task. We observe that metadata shaping is effective both when most task examples have available metadata (e.g., FewRel) and when a small proportion of task examples have available metadata (e.g., OpenEntity), analyzed further in Section 4.

For the baselines, we give reported numbers when available, Su et al. (2021) reports two of the KnowBERT-Wiki and all K-BERT results, and we obtain remaining numbers using the code released by baseline work as detailed in Appendix A.

4 Analysis

Here we study the following key questions for effectively using metadata shaping: **Section 4.1** What are the roles of different varieties of metadata? **Section 4.2** What are the effects of metadata shaping on slices concerning tail versus popular entities?

4.1 Framework: Role of Metadata Types

Metadata Effects *Class-discriminative metadata correlates with reduced model uncertainty. High quality metadata, as found in Wikidata, results in improved classification performance.*

To investigate the effects of metadata on model uncertainty, we compute the entropy of $\hat{p}_\phi$ softmax scores over the output classes as a measure of uncertainty, and compute the average across test set examples. Lower uncertainty is correlated with improved classification F1 (See Figure 2 (Left)).

We compute pmi scores for inserted metadata tokens as a measure of class-discriminativeness. We rank individual tokens k by pmi(y, k) (for task classes y), computed over the training dataset. On FewRel, for test examples containing a top-20 pmi word for the gold class, the accuracy is 27.6% higher when compared to the slice with no top-20 pmi words for the class. Notably, 74.1% more examples contain a top-20 pmi word for their class when pmi is computed on shaped data vs. unshaped training data.

Metadata Selection *Simple information theoretic heuristics are effective for selecting metadata, despite the complexity of the underlying contextual embeddings.*

We apply Algorithm 1, which ranks metadata tags by their provided information gain, to select metadata tags for the tasks. Given x_i with a set $\boldsymbol{M(x_i)}$ of metadata tags, our goal is to select n to use for shaping. We compare four selection approaches: using the highest ("High Rank") and lowest ("Low Rank") ranked tokens by Algorithm 1, random metadata from $\boldsymbol{M(x_i)}$ ("Random"), and the most popular metadata tokens across the union of $\boldsymbol{M(x_i)}, \forall x_i \in \boldsymbol{D_{train}}$ ("Popular"), selecting the same number of metadata tags per example for each baseline. We observe that "High Rank" consistently gives the best performance, evaluated over three seeds, and note that even "Random" yields decent performance vs. the BERT-baseline, indicating the simplicity of the method (Table 2).

Considering the distribution of selected category tokens under each scheme, the KL-divergence between the categories selected by "Low Rank" vs. "Popular" is 0.2 (FewRel), 4.6 (OpenEntity), while the KL-divergence between "High Rank" vs. "Popular" is 2.8 (FewRel), 2.4 (OpenEntity). Popular tokens are not simply the best candidates; instead, Algorithm 1 selects discriminative metadata.

For OpenEntity, metadata are relatively sparse, so categories appear less frequently in general and it is reasonable that coarse grained types have more overlap with "High Rank". For e.g., "business" is in the top-10 most frequent types under "High Rank", while "non-profit" (occurs in 2 train examples) is in the top-10 most frequent types for "Low Rank". Metadata tokens overall occur more frequently in FewRel (See Table 3), so finer-grained types are also quite discriminative. The most frequent category under "Low Rank" is "occupation" (occurs in 2.4k train examples), but the top-10 categories under "High Rank" are finer-grained, containing "director" and "politician" (each occurs in $>$ 300 train examples).

Benchmark	Strategy	Test F1
FewRel	BERT-base	84.9
	Random	87.2 $\pm$0.8
	Popular	87.9 $\pm$0.1
	Low Rank	87.8 $\pm$0.4
	High Rank	**88.9** $\pm$0.6
OpenEntity	BERT-base	73.2
	Random	74.3 $\pm$0.7
	Popular	74.5 $\pm$0.4
	Low Rank	74.1 $\pm$0.4
	High Rank	**74.8** $\pm$0.1
TACRED	BERT-base	72.0
	Random	73.8 $\pm$1.6
	Popular	73.6 $\pm$0.9
	Low Rank	73.3 $\pm$1.0
	High Rank	**74.7** $\pm$0.5

Table 2: Average and standard deviation over 3 random seeds. Each method selects up to n metadata tokens per entity. For FewRel, TACRED, $n = 3$ per subject, object. For OpenEntity $n = 2$ per main entity as 33% of OpenEntity train examples have ≥ 2 categories available (80.7% have ≥ 3 categories on FewRel). Note we use larger n for the main results in Table 1.

Task Agnostic Metadata Effects *Using metadata correlates with reduced task-specific LM uncertainty. We observe shaping also correlates with reduced LM uncertainty in a task-agnostic way.*

We perform additional masked language modeling (MLM) over the shaped task training data using an off-the-shelf BERT-MLM model to learn model $\hat{p}_\theta$. We minimize the following loss function and evaluate the model perplexity on the task test data:

$$\mathcal{L}_{\text{mlm}} = \mathbb{E}_{s\sim D, m\sim M, i\sim I}\big[-\log(\hat{p}_\theta(s_{m_i}|s_m/i))\big]. \quad (5)$$

where I is the masked token distribution and s_{m_i} is the masked token at position i in the shaped sequence s_m.[4] Through minimizing the MLM loss,

[4]We use the Hugging Face implementation for masking

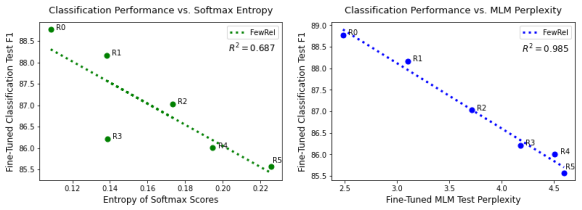

Figure 2: Test F1 for $\hat{p}_\phi$ (no additional pretraining) vs. average entropy of $\hat{p}_\phi$ softmax scores (Top) and vs. perplexity of a model $\hat{p}_\theta$ (w/ pretraining) (Bottom). $\hat{p}_\phi$ and $\hat{p}_\theta$ use the same shaped training data. Each point is a different metadata shaping scheme (median over 3 Random Seeds): for *R0* all inserted tokens are true tokens associated with the entity in the KB. For *RX*, X true metadata tokens are replaced by random (noise) tokens from the full vocabulary. For each point, the total number of metadata tokens is constant per example.

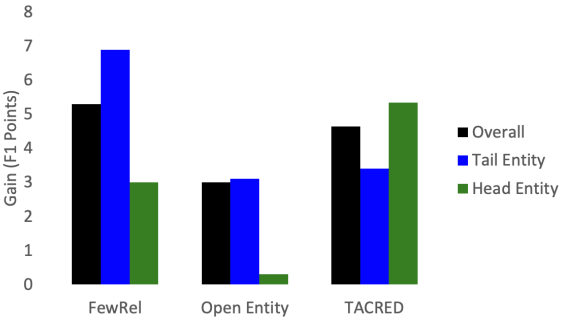

Figure 3: The gain from training the BERT-base LM with metadata shaped data over training with unshaped data, split by the popularity of the entity span in the test example.

$\hat{p}_\theta$ learns direct dependencies between tokens in the data (Zhang and Hashimoto, 2021). In Figure 2 (Right), we observe a correlation between reduced perplexity for $\hat{p}_\theta$, and higher downstream performance for $\hat{p}_\phi$ across multiple tasks, both using the same training data. Overall, shaping increases the likelihood of the data, and we observe a correlation between the intrinsic perplexity metric and the extrinsic downstream metrics as a result of the same shaping scheme. Table 4 (Appendix B) reports the same correlations for all benchmarks.

Metadata Noise *We hypothesize that noisier metadata can provide implicit regularization. Noise arises from varied word choice, word order, and blank noising.*

Feature noising (Wang et al., 2013) is effective to prevent overfitting and while regularization is typically applied directly to model parameters, Xie et al. (2017); Dao et al. (2019) regularize through the data. We hypothesize that using metadata with diverse word choice and order (e.g., entity descriptions) and blank noising (e.g., by masking metadata tokens), can help reduce overfitting, and we provide initial empirical results in Appendix B.

4.2 Evaluation: Tail and Head Slices

Section 3 shows the overall gain from shaping. We now consider fine-grained slices of examples containing head vs. tail entities and observe gains are 4.4x larger on the tail slice on average (Figure 3). [5]

Subpopulations *Metadata are helpful on the tail as they establish subpopulations.*

We hypothesize that if a pattern is learned for an entity-subpopulation occurring in the training data, the model may perform better on rare entities that also participate in the subpopulation, but were not individually observed during training. On FewRel, we take the top-20 TF-IDF words associated with each category signal during training as linguistic cues captured by the model for the category subpopulation, consistent with Goel et al. (2021). For example, "government" is in the top-20 TF-IDF words for the "politician" entity category. At test time, we select the slice of examples containing any of these words for any of the categories inserted in the example. The performance is 9.0/3.5 F1 points higher on examples with unseen subject/object entities with vs. without a top-20 TF-IDF word for a subject/object category.

Metadata Effects on Popular Entities *For popular entities the LM can learn entity-specific patterns well, and be mislead by subpopulation-level patterns corresponding to metadata.*

Although we observe overall improvements, here we examine the effect of metadata on the popular entity slice within our conceptual framework.

Let p be a popular pattern (i.e., entity mention) in the training data, and let m be a metadata token associated with p. Intuitively, the LM can learn entity-specific patterns from occurrences of p, but coarser grained subpopulation-level patterns corresponding to m. If m and p are class-discriminative for different sets of classes, then m can mislead the LM. To evaluate this, consider subject and object entity spans $p \in P$ seen ≥ 1 time during training. For test examples let $\mathcal{Y}_p$ be the set of classes y for which there is a $p \in P$ in the ex-

and fine-tuning the BERT-base MLM (Wolf et al., 2020).

[5] A consideration for TACRED is that 42% of these head spans are stopwords (e.g., pronouns) or numbers; just 7% are for FewRel. This is based on unseen object spans for FewRel and TACRED, as $> 90\%$ of subject spans are unseen.

ample with $\text{pmi}(y, p) > 0$, and define $\mathcal{Y}_m$ as the classes y for which there is a metadata token m with $\text{pmi}(y, m) > 0$ in the example. The examples where $\mathcal{Y}_p \neq \emptyset$, $\mathcal{Y}_m \neq \emptyset$, and $\mathcal{Y}_p$ contains the true class, but $\mathcal{Y}_m$ does not, represents the slice where metadata can mislead the model. On this slice of FewRel, the gain from the shaped model is 2.3 F1 points less than the gain on the slice of all examples with $\mathcal{Y}_p \neq \emptyset$ and $\mathcal{Y}_m \neq \emptyset$, supporting our intuition.

An example entity-specific vs. subpopulation-level tension in FewRel is: $p =$ "Thames River" is class-discriminative for $y =$"located in or next to body of water", but its $m =$"river" is class-discriminative for $y =$"mouth of the watercourse".

5 Related Work

Incorporating Knowledge in LMs Discussed in Section 3.2, significant prior work incorporates knowledge by changing the base LM architecture or loss function. Peters et al. (2019); Alt et al. (2020) also use NER, POS Wikpedia, or Wordnet metadata, but do not conceptually explain the benefit or selection process. Orr et al. (2020) demonstrates that category metadata improves tail performance for NED. We do not modify the base LM.

Prior work inserts metadata for entities in the data itself. Particularly relevant, Joshi et al. (2020); Logeswaran et al. (2019); Raiman and Raiman (2018) each uses a single form of metadata (either descriptions or types) for a single task-type (either QA or NED) demonstrating empirical benefits. Metadata shaping combines different varieties of metadata and applies generally to classification tasks, and we provide conceptual grounding.

Feature Selection This work is inspired by techniques in feature selection based on information gain (Guyon and Elisseeff, 2003). In contrast to traditional feature schemas (Levin, 1993; Marcus et al., 1993; Mintz et al., 2009), metadata shaping annotations are expressed in natural language to flexibly include diverse structured and unstructured metadata. We show the classic methods (Berger et al., 1996) are informative even when using complicated contextual embeddings. In our setting of entity-rich tasks, we also draw a connection from maximum entropy to explain how metadata can reduce generalization error.

Prompting Prompting can serve similar goals, but often requires human-picked prompt tokens (Keskar et al., 2019; Aghajanyan et al., 2021) or task-specific templates (Han et al., 2021; Chen et al., 2021), while metadata shaping provides a flexible baseline across metadata-types and task-types. Prompting typically aims to better elicit *implicit* knowledge from the base LM (Liu et al., 2021), while metadata shaping focuses on *explicitly* incorporating retrieved signals not found in the original task. Shaping is applied at train and test time and does not introduce new parameters, as required by methods which use learned prompts.

Data Augmentation One approach to tackle the tail is to generate additional examples for tail entities (Wei and Zou, 2019; Xie et al., 2020; Dai and Adel, 2020). However, this can be sample inefficient since augmentations do not explicitly signal that different entities are in the same subpopulation (Horn and Perona, 2017), so the model would need view each entity individually in different contexts. Metadata shaping and prompting (Scao and Rush, 2021) may be viewed as implicit augmentation.

6 Conclusion

We propose metadata shaping to improve tail performance. The method is a simple and general baseline that is competitive with SoTA approaches for entity-rich tasks. We empirically show that the method improves tail performance and formalize why metadata can reduce generalization error.

Benefits from shaping include the ability to use simpler, off-the-shelf LMs, tackle the tail efficiently by learning subpopulation-level (rather than only entity-specific) patterns, and provide a conceptual understanding by relying on the rich set of tools for reasoning about data distributions (in contrast to the limited set of tools for reasoning about model architectural changes). In limitations, pretrained entity embeddings may encode facts beyond the scope of the provided metadata tokens, and this work did not consider including metadata for non-subject and non-object entities that appear in the examples. Both shaping and the baseline knowledge-aware LMs (e.g., Zhang et al. (2019)) rely on accurate external knowledge — Wikidata contains high-quality, human-curated entity metadata and in future work, we want to compare the approaches using noisy and incomplete KBs (West et al., 2014).

While this work focused on entity-rich tasks, metadata shaping is not limited to this setting and we hope this work motivates further research on addressing the tail challenge through the data alongside the model.

References

Armen Aghajanyan, Dmytro Okhonko, Mike Lewis, Mandar Joshi, Hu Xu, Gargi Ghosh, and Luke Zettlemoyer. 2021. Htlm: Hyper-text pre-training and prompting of language models. In *arXiv:2107.06955v1*.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. Tacred revisited: A thorough evaluation of the tacred relation extraction task. In *ACL*.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. In *Computational Linguistics*.

Michael S Bernstein, Jaime Teevan, Susan Dumais, Daniel Liebling, and Eric Horvitz. 2012. Direct answers for search queries in the long tail. In *SIGCHI*.

Olivier Bodenreider. 2004. The unified medical language system (umls): integrating biomedical terminology. In *Nucleic Acids Research*.

Vincent Chen, Sen Wu, Alex Ratner, J. Weng, and Christopher Ré. 2019. Slice-based learning: A programming model for residual learning in critical data slices. In *Advances in neural information processing systems*, pages 9392–9402.

Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2021. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *arXiv:2104.07650v5*.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *arXiv:1807.04905v1*.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. In *Journal of Machine Learning Research*.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *CVPR*.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. *Proceedings of the 28th International Conference on Computational Linguistics*.

Tri Dao, Albert Gu, Alexander Ratner, Chris De Sa Virginia Smith, and Christopher Ré. 2019. A kernel theory of modern data augmentation. *Proceedings of the 36th International Conference on Machine Learning (PMLR)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Vitaly Feldman. 2021. Does learning require memorization? a short tale about a long tail. In *arXiv:1906.05271v4*.

Karan Goel, Nazneen Rajani, Jesse Vig, Samson Tan, Jason Wu, Stephan Zheng, Caiming Xiong, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the nlp evaluation landscape. *arXiv preprint arXiv:2101.04840*.

Isabelle Guyon and Andre Elisseeff. 2003. An introduction to variable and feature selection. In *Journal of Machine Learning Research*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. In *arXiv:2105.11259v3*.

Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2018. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In *EMNLP*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. Industrial-strength natural language processing in python.

Grant Van Horn and Pietro Perona. 2017. The devil is in the tails: Fine-grained classification in the wild. In *arXiv:1709.01450*.

Mandar Joshi, Kenton Lee, Yi Luan, and Kristina Toutanova. 2020. Contextualized representations using textual encyclopedic knowledge. In *ArXiv*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv:1909.05858v2*.

J. Lafferty, A. McCallum, , and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Beth Levin. 1993. *English Verb Classes and Alternations*. The University of Chicago Press.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. In *arXiv:2107.13586*.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of AAAI*.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *arXiv:1906.07348v1*.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. *Building a Large Annotated Corpus of English: The Penn Treebank*.

George A Miller. 1995. Wordnet: a lexical database for english. In *Communications of the ACM*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

Laurel Orr, Megan Leszczynski, Simran Arora, Sen Wu, Neel Guha, Xiao Ling, and Chris Ré. 2020. Bootleg: Chasing the tail with self-supervised named entity disambiguation. In *Arxiv*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (PMLR)*.

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. *arXiv preprint arXiv:1909.04164*.

Nina Poerner, Ulli Waltinger, and Hinrich Schutze. 2020. E-bert: Efficient-yet-effective entity embeddings for bert. *arXiv:1911.03681v2*.

Jonathan Raiman and Olivier Raiman. 2018. Deeptype: multilingual entity linking by neural type system evolution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv:1910.01108v4*.

Teven Le Scao and Alexander M. Rush. 2021. How many data points is a prompt worth? In *NAACL HLT*.

Yusheng Su, Xu Han, Zhengyan Zhang, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. Cokebert: Contextual knowledge selection and embedding towards enhanced pre-trained language models. In *arXiv:2009.13964*.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2020. Long range arena: A benchmark for efficient transformers. In *arXiv:2011.04006v1*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei1, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020a. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv:2002.01808v5 2020*.

Sida I Wang, Mengqiu Wang, Stefan Wager, Percy Liang, and Christopher D Manning. 2013. Feature noising for log-linear structured prediction. *Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2020b. Kepler: A unified model for knowledge embedding and pre-trained language representation. *arXiv:1911.06136v3*.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *EMNLP 2019*.

Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *International World Wide Web Conference Committee (IW3C2)*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training. *NeurIPS 2020*.

Zhang Xie, Sida I. Wang, Jiwei Li, Daniel Levy, Aiming Nie, Dan Jurafsky, and Andrew Y. Ng. 2017. Data noising as smoothing in neural network language models. *ICLR 2017*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained Encyclopedia: Weakly Supervised Knowledge-Pretrained Language Model. In *International Conference on Learning Representations (ICLR) 2020*.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: Deep contextualized entity representations with entity-aware self-attention. *EMNLP 2020*.

Tianyi Zhang and Tatsunori Hashimoto. 2021. On the inductive bias of masked language modeling: From statistical to syntactic dependencies. *arXiv:2104.05694*.

Yue Zhang and Joakim Nivre. 2011. Transition-based dependency parsing with rich non-local features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position aware attention and supervised data improve slot filling. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. 2014. Capturing long-tail distributions of object subcategories. In *CVPR*.

A Appendix

A.1 Dataset Details

Benchmarks The datasets can be downloaded here: https://github.com/thunlp/ERNIE.

Metadata We tag original dataset examples with a state-of-the-art pretrained entity-linking model from (Orr et al., 2020),[6] which was trained on an October 2020 Wikipedia dump with train, validation, test splits of 51M, 4.9M, and 4.9M sentences. FewRel includes entity annotations. The types we use as category metadata for all tasks are those appearing at least 100 times in Wikidata for entities this Wikipedia training data used bh Orr et al. (2020). Descriptions are sourced from Wikidata descriptions and the first 50 words of the entity Wikipedia page. Table 3 reports the availability of metadata for examples across the benchmark tasks.

Benchmark	Train	Valid	Test
TACRED	68124	22631	15509
Category	54k/46k	16k/15k	9k/10k
Description	50k/43k	15k/14k	8k/9k
FewRel	8k	16k	16k
Category	8k/8k	16k/15k	16k/15k
Description	7k/8k	15k/16k	15k/16k
OpenEntity	1998	1998	1998
Category	674	674	647
Description	655	672	649

Table 3: We show the benchmark split sizes (row 1), and the # of examples tagged with category and description metadata (rows 2 and 3). We give numbers for the subject and object entity-span on relation extraction and the main entity-span for entity-typing. The tasks have represent a range of proportions of shaped examples (e.g., essentially all FewRel examples have metadata, while metadata is sparsely available for OpenEntity).

A.2 Training Details

We use the pretrained BERT-base-uncased model for each task to encode the input text. We take the hidden layer representation corresponding to the [CLS] token and use a linear classification layer for prediction. All models are trained on 1 Tesla P100 GPU (1.5 min/epoch for OpenEntity, 7.5 min/epoch for FewRel, 28 min/epoch for TACRED). For all tasks, we select the best learning rate from {1e-6, 2e-6, 1e-5, 2e-5, 1e-4} and use the scoring implementations released by Zhang et al. (2019).

Entity Typing Hyperparameters include 2e-5 learning rate, no regularization parameter and 256 max. sequence length, batch size of 16 and no gradient accumulation or warmup. We report the test score for the epoch with the best validation score within 20 epochs.

Relation Extraction Hyperparameters include 2e-5 learning rate and no regularization parameter. For FewRel, we use batch size of 16, 512 maximum sequence length, and no gradient accumulation or warmup. For TACRED, we use a batch size 48, 256 maximum sequence length, and no gradient accumulation or warmup. We report the test score for the epoch with the best validation score within 15 epochs (FewRel) and 8 epochs (TACRED).

A.3 Metadata Implementation Details

We report the test score at the epoch with the highest validation score. For the results in Table 1, we evaluated the number of metadata tokens to insert, whether place the tokens directly following or at the end of the example, and whether to use blank noising on the metadata tokens. Metadata tokens are ranked by Algorithm 1.

We use up to 20 metadata categories per subject and object on FewRel, up to 25 metadata categories per subject on OpenEntity, and up to 5 metadata categories per subject and object on TACRED. Note that categories (e.g., "United States federal executive department") can include multiple tokens, selecting these maximum values by grid search. For FewRel and OpenEntity, we insert metadata tokens directly after the corresponding entity mention, and

[6] https://github.com/HazyResearch/bootleg

for TACRED, we inserted all metadata at the end of the example. For OpenEntity we randomly mask 10% of metadata tokens at training time as implicit regularization, and for relation extraction, we use no blank noising. The impact of position and blank noising are further discussed in Appendix B.3.

A.4 Baseline Implementations

We produce numbers for key baselines which do not report for the benchmarks we consider, using provided code.[7] [8]

- We produce numbers for KnowBERT-Wiki on TACRED-Revisited using a learning rate of $3e-5$, $\beta_2 = 0.98$, and choosing the best score for epochs $\in 1$, **2**, 3, 4 and the remaining provided configurations.
- We produce numbers for ERNIE on TACRED-Revisited using the provided training script and configurations they use for the original TACRED task.

B Additional Experiments

B.1 Task Agnostic Metadata Effects

In Table 4 we report the same experiment conducted in Section 4.1, for all benchmark tasks considered in this work. Each point represents the median test score over 3 random seeds.

B.2 Metadata Noise

Noisier metadata appear to provide implicit regularization. Noise arises from varied word choice and order, as found in entity descriptions, or blank noising (i.e. random token deletion).

Here we provide initial empirical results.

Blank noising (Xie et al., 2017) by randomly masking 10% of inserted metadata tokens during training leads to a consistent boost on OpenEntity: 0.1 ("High Rank"), 0.5 ("Popular"), 0.5 ("Low Rank") F1 points higher than the respective scores from Table 2 over the same 3 random seeds. We observe no consistent benefit from masking on FewRel. Since metadata are sparsely available for OpenEntity examples, we hypothesize that blank noising of the category tokens can prevent over-reliance on the signal. Future work could investigate advanced masking strategies, for example masking discriminative words in the training data.

[7] https://github.com/allenai/kb

[8] https://github.com/thunlp/ERNIE

Benchmark	R^2
FewRel	0.985
TACRED	0.782*
OpenEntity	0.956

Table 4: Correlation (R^2) between test F1 of $\hat{p}_\phi$ (no additional pretraining) vs. perplexity of the independent model $\hat{p}_\theta$ (w/ additional pretraining) for three tasks, using the procedure described in Figure 2. *Without one outlier corresponding to shaping with all random tokens ($R^2 = 0.02$ with this point).

Descriptions use varied word choice and order vs. category metadata.[9] To study whether shaping with description versus category tokens lead the model to rely more on metadata tokens, we consider two shaping schemes that use 10 metadata tokens: 10 category tokens and 5 category, 5 description, where the categories are randomly selected. We observe both give the ∼same score on FewRel, 89.8 F1 and 89.5 F1, and use models trained with these two schemes to evaluate on test data where 10% of metadata tokens per example are randomly removed. Performance drops by 1.4 F1 for the former and 1.0 F1 for the latter.

B.3 Implementation Choices

We also analyze the degree of sensitivity of metadata shaping to how the metadata are inserted in examples (e.g., special tokens, the number of metadata tokens, and position).

Boundary Tokens *Designating the boundary between original tokens in the example and inserted metadata tokens improves model performance.*

Inserting boundary tokens (e.g., "**#**") in the example, at the start and end of a span of inserted metadata, consistently provides a boost across the tasks. Comparing performance with metadata and boundary tokens to performance with metadata and no boundary tokens, we observe a 0.7 F1 (FewRel), 1.4 F1 (OpenEntity) boost in our main results. We use boundary tokens for all results in this work.

Task Structure Tokens such as "[START_SUBJECT]" and "[END_SUBJECT]", designate the relevant entities in the examples. With no other shaping, inserting these tokens provides a 26.3 (FewRel), 24.7 (OpenEntity) F1

[9] Over FewRel training data: on average a word in the set of descriptions appears 8 times vs. 18 times for words in the set of categories, and the description set contains 3.3x the number of unique words vs. set of categories.

point boost vs. training the BERT model without task structure tokens. These tokens are already commonly used.

Token Insertion *We observe low sensitivity to increasing the context length and to token placement (i.e., inserting metadata directly-following the entity-span vs at the end of the sentence).*

We evaluate performance a the maximum number of inserted tokens per entity, n, increases. [10] We insert metadata tokens in a random order (to control for the effect of different metadata having different levels of class-discriminativeness) and observe that for FewRel, $n \in \{1, 5, 10, 20\}$ gives $\{85.4, 86.4, 87.6, 88.5\}$ test F1. On OpenEntity, $n \in \{1, 5, 10, 20, 40\}$ gives $\{74.9, 75.7, 74.8, 74.5, 75.8\}$ test F1. Overall performance changes gracefully with n and we observe low sensitivity to longer contexts.

The benefit of inserting metadata directly-following the entity span vs at the end of the example differed across tasks (e.g., for TACRED, placement at the end performs better, for the other tasks, placement directly-following performs better), though the observed difference was small. In Section 4, tokens are inserted directly-following the relevant entity span for all tasks.

[10] Per subject and object entity for FewRel, and per main entity for OpenEntity. I.e., $n = 10$ for FewRel yields a maximum of 20 total inserted tokens for the example.