A Supplemental Material

A.1 KnowBert training details

We generally followed the fine-tuning procedure in Devlin et al. (2019), using Adam (Kingma and Ba, 2015) with weight decay 0.1 and learning rate schedule that linearly increases then decreases. Similar to ULMFiT (Howard and Ruder, 2018), we found it beneficial to vary the learning rate in different layers when fine tuning with this simple schedule: the randomly initialized newly added layers in the KAR surrounding each KB had the largest learning rate α; all BERT layers below had learning rate 0.25α; and BERT layers above had 0.5α. KnowBert-Wiki was fine-tuned for a total of 750K gradient updates, KnowBert-WordNet for 500K updates, and KnowBert-W+W for 500K updates (after pretraining KnowBert-Wiki). Fine-tuning was done on a single Titan RTX GPU with batch size of 32 using gradient accumulation for long sequences, except for the final 250K steps of KnowBert-W+W which was trained with batch size of 64 using 4 GPUs. At the end of training, masked LM perplexity continued to decrease, suggesting that further training would improve the results.

Due to computational expense, we performed very little hyperparameter tuning during the pretraining stage, and none with the full KnowBert-W+W model. All tuning was performed with partial training runs by monitoring masked LM perplexity in the early portion of training and terminating the under performing configurations. For each KnowBert-WordNet and KnowBert-Wiki we ran experiments with two learning rates (maximum learning rate of 1e-4 and 2e-4) and chose the best value after a few hundred thousand gradient updates (2e-4 for KnowBert-WordNet and 1e-4 for KnowBert-Wiki).

When training KnowBert-WordNet and KnowBert-Wiki we chose random batches from the unlabeled corpus and annotated entity linking datasets with a ratio of 4:1 (80% unlabeled, 20% labeled). For KnowBert-W+W we used a 85%/7.5%/7.5% sampling ratio.

A.2 Task fine-tuning details

This section details the fine tuning procedures and hyperparameters for each of the end tasks. All optimization was performed with the Adam optimizer with linear warmup of learning rate over the first 10% of gradient updates to a maximum value, then linear decay over the remainder of training. Gradients were clipped if their norm exceeded 1.0, and weight decay on all non-bias parameters was set to 0.01. Grid search was used for hyperparameter tuning (maximum values bolded below), using five random restarts for each hyperparameter setting for all datasets except TACRED (which used a single seed). Early stopping was performed on the development set. Batch size was 32 in all cases.

TACRED This dataset provides annotations for 106K sentences with typed subject and object spans and relationship labels across 41 different classes (plus the no-relation label). The hyperparameter search space was:

- learning rate: [**3e-5**, 5e-5]
- number epochs: [1, 2, **3**, 4, 5]
- β_2: [**0.98**, 0.999]

Maximum development micro F_1 is 71.7%.

SemEval 2010 Task 8 This dataset provides annotations for 10K sentences with untyped subject and object spans and relationship labels across 18 different classes (plus the no-relation label). As the task does not define a standard development split, we randomly sampled 500 of the 8000 training examples for development. The hyperparameter search space was:

- learning rate: [**3e-5**, 5e-5]
- number epochs: [1, 2, **3**, 4, 5, 6, 7, 8, 9, 10]

with $\beta_2 = 0.98$. We used the provided `semeval2010_task8_scorer-v1.2.pl` script to compute F_1. The maximum development F_1 averaged across the random restarts was 89.1 $\pm$ 0.77 (maximum value was 90.5 across the seeds).

WiC WiC is a binary classification task with 7.5K annotated sentence pairs. Due to the small size of the dataset, we found it helpful to use model averaging to reduce the variance in the development accuracy across random restarts. The hyperparameter search space was:

- learning rate: [**1e-5**, 2e-5, 3e-5, 5e-5]
- number epochs: [2, 3, 4, **5**]
- $beta_2$: [0.98, **0.999**]

	BERT		KnowBert		
	base	large	Wiki	Wordnet	W+W
companyFoundedBy	0.08	0.08	0.23	0.21	0.28
movieDirectedBy	0.05	0.04	0.08	0.09	0.10
movieStars	0.06	0.05	0.09	0.09	0.11
personCityOfBirth	0.18	0.20	0.33	0.35	0.40
personCountryOfBirth	0.16	0.17	0.33	0.36	0.47
personCountryOfDeath	0.18	0.18	0.30	0.30	0.44
personEducatedAt	0.11	0.10	0.37	0.29	0.43
personEmployer	0.03	0.02	0.20	0.13	0.21
personFather	0.15	0.14	0.42	0.40	0.48
personMemberOfBand	0.11	0.11	0.19	0.18	0.23
personMemberOfSportsTeam	0.01	0.03	0.05	0.08	0.13
personMother	0.11	0.11	0.28	0.24	0.32
personOccupation	0.14	0.24	0.24	0.24	0.30
personSpouse	0.05	0.06	0.12	0.08	0.18
songPerformedBy	0.07	0.05	0.10	0.10	0.13
videoGamePlatform	0.16	0.22	0.27	0.20	0.34
writtenTextAuthor	0.06	0.05	0.15	0.16	0.18
Total	0.09	0.11	0.26	0.22	0.31

Table 1: Full results on the Wikidata probing task including all relations.

- weight averaging decay: [no averaging, **0.95**, 0.99]

The maximum development accuracy was 72.6.

Entity typing As described in Section 4.3, we evaluated on a subset of data corresponding to entities classified by nine different general classes: person, location, object, organization, place, entity, object, time, and event. β_2 was set to 0.98. The hyperparameter search space was:

- learning rate: [2e-5, **3e-5**]
- number epochs: [2, 3, 4, 5, 6, 7, 8, 9, **10**, 11, 12, 13, 14]
- $beta_2$: [0.98, **0.999**]
- weight averaging decay: [**no averaging**, 0.99]

The maximum development F_1 was 75.5 $\pm$ 0.38 averaged across five seeds, with maximum value of 76.0.

A.3 Wikidata probing results

Table 1 shows the results for the Wikidata probing task for all relationships.

References

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

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of ACL 2018*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.