Modelling Commonsense Properties using Pre-Trained Bi-Encoders

Amit Gajbhiye[*], **Luis Espinosa-Anke**[*◇], **Steven Schockaert**[*]
[*]CardiffNLP, Cardiff University, United Kingdom
[◇]AMPLYFI, United Kingdom
{gajbhiyea, espinosa-ankel, schockaerts1}@cardiff.ac.uk

Abstract

Grasping the commonsense properties of everyday concepts is an important prerequisite to language understanding. While contextualised language models are reportedly capable of predicting such commonsense properties with human-level accuracy, we argue that such results have been inflated because of the high similarity between training and test concepts. This means that models which capture concept similarity can perform well, even if they do not capture any knowledge of the commonsense properties themselves. In settings where there is no overlap between the properties that are considered during training and testing, we find that the empirical performance of standard language models drops dramatically. To address this, we study the possibility of fine-tuning language models to explicitly model concepts and their properties. In particular, we train separate concept and property encoders on two types of readily available data: extracted hyponym-hypernym pairs and generic sentences. Our experimental[1] results show that the resulting encoders allow us to predict commonsense properties with much higher accuracy than is possible by directly fine-tuning language models. We also present experimental results for the related task of unsupervised hypernym discovery.

1 Introduction

Pre-trained language models (Devlin et al., 2019) have been found to capture a surprisingly rich amount of knowledge about the world (Petroni et al., 2019). Focusing on commonsense knowledge, Forbes et al. (2019) used BERT to predict whether a given concept (e.g. *teddy bear*) satisfies a given commonsense property (e.g. *is dangerous*). To this end, they convert the input into a simple sentence (e.g. "*A teddy bear is dangerous*") and treat the task as a standard sentence classification task. Remarkably, they found this approach to surpass human performance. Shwartz and Choi (2020) moreover found that language models can, to some extent, capture commonsense properties that are rarely expressed in text, thus mitigating the issue of reporting bias that has traditionally plagued initiatives for learning commonsense knowledge from text (Gordon and Durme, 2013).

Despite these encouraging signs, however, modelling commonsense properties remains highly challenging. A key concern is that language models are typically fine-tuned on a training set that contains the same properties as those in the test set. For instance, the test data from Forbes et al. (2019) includes the question whether *peach* has the property *eaten in summer*, while the training data asserts that *apple*, *banana*, *orange* and *pear* all have this property. To do well on this task, the model does not actually need to capture the knowledge that peaches are eaten in summer; it is sufficient to capture that *peach* is similar to *apple*, *banana*, *orange* and *pear*. For this reason, we propose new training-test splits, which ensure that the properties occurring in the test data do not occur in the training data. Our experiments show that the ability of language models to predict commonsense properties drops dramatically in this setting.

Our aim is to develop a strategy for modelling the commonsense properties of concepts. Given the limitations that arise when language models are used directly, a natural approach is to pre-train a language model on some kind of auxiliary data. Unfortunately, resources encoding the commonsense properties of concepts tend to be prohibitively noisy. To illustrate this point, Table 1 lists the properties of some everyday concepts according to three well-known resources: ConceptNet (Speer et al., 2017), which is a large commonsense knowledge graph, COMET-2020[2] (Hwang et al., 2021), which

[1] Code and datasets are available at https://github.com/amitgajbhiye/biencoder_concept_property

[2] We used the demo at https://mosaickg.apps.allenai.org/model_comet2020_entities.

	ConceptNet	COMET-2020	Ascent++
banana	yellow, good to eat	one of the main ingredients, eaten as a snack, one of many fruits, found in garden, black	rich, ripe, yellow, green, brown, sweet, great, black, useful, safe, delicious, healthy, nutricious, ...
lion	a feline	found in jungle, one of many animals, one of many species, two legs, very large	free, extinct, hungry, close, unique, active, nocturnal, old, dangerous, great, happy, right, ...
airplane	good for quickly travelling long distances	flying, air travel, flying machine, very small, flight	heavy, new, important, white, safe, unique, full, larger, clean, slow, low, unstable, electric, ...

Table 1: Properties of some example concepts, according to three commonsense knowledge resources.

predicts triples using a generative language model that was trained on several commonsense knowledge graphs, and Ascent++ (Nguyen et al., 2021), which is a commonsense knowledge base that was extracted from web text. Given the noisy nature of such resources, we rely on a database with hypernyms instead. The underlying intuition is that hypernyms can be extracted from text relatively easily, while fine-grained hypernyms often implicitly describe commonsense properties. For instance, Microsoft Concept Graph (Ji et al., 2019) lists *potassium rich food* as a hypernym of *banana* and *large and dangerous carnivore* as a hypernym of *lion*. We also experiment with GenericsKB (Bhakthavatsalam et al., 2020), a large collection of generic sentences (e.g. "*Coffee contains minerals and antioxidants which help prevent diabetes*"), to obtain concept-property pairs for pre-training. Given such pre-training data, we then train a concept encoder Φ_{con} and a property encoder Φ_{prop} such that $\sigma(\Phi_{\text{con}}(c) \cdot \Phi_{\text{prop}}(p))$ indicates the probability that concept c has property p.

In summary, our main contributions are as follows: (i) we propose a new evaluation setting which is more realistic than the standard benchmarks for predicting commonsense properties; (ii) we analyse the potential of hypernymy datasets and generic sentences to act as pre-training data; and (iii) we develop a simple but effective bi-encoder architecture for modelling commonsense properties.

2 Related Work

Several authors have analysed the extent to which language models such as BERT capture commonsense knowledge. As already mentioned, Forbes et al. (2019) evaluated the ability of BERT to predict commonsense properties from the McRae dataset (McRae et al., 2005), which we also use in our experiments. The same dataset was used by Weir et al. (2020) to analyse whether BERT-based language models could generate concept names from their associated properties; e.g. given the input "*A ⟨mask⟩ has fur, is big, and has claws*", the model is expected to predict that ⟨*mask*⟩ corresponds to the word *bear*. Conversely, Apidianaki and Garí Soler (2021) considered the problem of generating adjectival properties from prompts such as "mittens are generally ⟨*mask*⟩". Note that the latter two works evaluated pre-trained models directly, without fine-tuning, whereas the experiments Forbes et al. (2019) involved fine-tuning the language model on a task-specific training set first. When the main motivation is to probe the abilities of language models, avoiding fine-tuning has the advantage that any observed abilities reflect what is captured by the pre-trained language model itself, rather than learned during the fine-tuning phase. However, Li et al. (2021) argue that the extent to which pre-trained language models capture commonsense knowledge is limited, suggesting that some form of fine-tuning is essential in practice. Interestingly, this remains the case for large language models. For instance, their model had 7 billion parameters, while West et al. (2021) report that the predictions from GPT-3 (Brown et al., 2020) had to be filtered by a so-called critic model when distilling a commonsense knowledge graph.

The strategy taken by COMET (Bosselut et al., 2019) is to fine-tune a GPT model (Radford et al.) on triples from commonsense knowledge graphs. Being based on an autoregressive language model, COMET can be used to predict concepts that take the form of short phrases, which is often needed when reasoning about events (e.g. to express motivations or effects). However, as illustrated in Table 1, COMET is less suitable for modelling the commonsense properties of concepts. Other approaches have focused on improving the commonsense reasoning abilities of general purpose language models. For instance, Zhou et al. (2021) introduce a self-supervised pre-training tasks to encourage language models to better capture the commonsense relations between everyday concepts.

A final line of related work concerns the modelling of hypernymy. Several authors have proposed specialised embedding models for this task (Dasgupta et al., 2021; Le et al., 2019). Most relevant to our work, Takeoka et al. (2021) fine-tune a BERT-based language model to predict the validity of a concept–hypernym pair. Inspired by the effectiveness of Hearst patterns (Hearst, 1992), they use prompts of the form "[HYPERNYM] such as [CONCEPT]" (and similar for other Hearst patterns). The extent to which the pre-trained BERT model captures hypnernymy has also been studied. For instance, Hanna and Mareček (2021) use prompts where the prediction of the $\langle mask \rangle$ token can be interpreted as the prediction of a hypernym, to avoid the need for fine-tuning the model.

3 Methodology

Let a set of concept–property pairs $\mathcal{K}$ be given, where $(c, p) \in \mathcal{K}$ means that concept c is asserted to have the property p. We write $\mathcal{C}$ and $\mathcal{P}$ for the sets of concepts and properties in $\mathcal{K}$, i.e. $\mathcal{C} = \{c \mid (c, p) \in \mathcal{K}\}$ and $\mathcal{P} = \{p \mid (c, p) \in \mathcal{K}\}$. We use the term "property" in a broad sense, covering both semantic attributes, as in the pair (*banana*, *sweet*), and hypernyms, as in the pair (*banana*,*fruit*). This is motivated by the fact that hypernyms often encode knowledge about semantic attributes, as in the pair (*banana*, *sweet fruit*). In particular, our hypothesis is that, by treating hypernyms and semantic attributes in a unified way, we can pre-train a model on readily available hypernym datasets and use it to predict semantic attributes.

We want to train a model that can predict for a given pair (c, p) whether c has property p. Two general strategies can be pursued when developing such models. The first strategy is to use a so-called cross-encoder, which amounts to fine-tuning a single language model to predict whether a given input (c, p) represents a valid pair or not. The second strategy is to use a so-called bi-encoder, which amounts to the idea that c and p are separately encoded, with the resulting vectors then being used to predict whether (c, p) is a valid pair. In this paper, we pursue the latter strategy. This is primarily motivated by the fact that the concept and property encoders enable a wider range of applications. A cross-encoder can only be used to predict whether a given pair (c, p) is valid or not. In contrast, a bi-encoder model can also be used to efficiently find the properties p of a given concept c. Moreover, the resulting concept and property embeddings may themselves be useful as static representations of word meaning, e.g. as label embeddings for zero-shot or few-shot learning (Socher et al., 2013; Ma et al.; Xing et al., 2019; Li et al., 2020; Yan et al., 2021). Finally, bi-encoders can be trained more efficiently than cross-encoders.

Datasets To train our model, we need a large set of concept–property pairs $\mathcal{K}$. Unfortunately, high-quality knowledge of this kind is not readily available. Part of the underlying issue is that properties of concepts are rarely explicitly stated in text, which is why directly using information extraction techniques is not straightforward. However, initiatives for extracting hypernyms from text have been much more successful, starting with the seminal work of Hearst (1992). A key observation is that fine-grained hypernyms often express commonsense properties, typically as a mechanism for refining hypernyms that would otherwise be too broad. For instance, Microsoft Concept Graph (Ji et al., 2019) lists *vitamin C rich food* as a hypernym of *strawberry*, as a refinement of the more general hypernym *food*. By pre-training our model on concept–hypernym pairs, we may thus expect it to learn about commonsense properties as well. To directly test this hypothesis, we use a set of such concept–hypernym pairs as our pre-training set $\mathcal{K}$. Specifically, we collect the 100K concept–hypernym pairs from Microsoft Concept Graph[3] with the highest confidence score[4] We will refer to this dataset as MSCG.

As a second strategy, we attempt to convert the MSCG dataset into a set of concept–property pairs. To this end, we look for pairs (c, h_1) and (c, h_2) where h_2 is a suffix of h_1. Specifically, if $h_1 = mh_2$ and m is an adjectival phrase, then we assume that m describes a property of c. For instance, MSCG contains the pairs (*strawberry*, *vitamin C rich food*) and (*strawberry*, *food*). Based on this, we would include the pair (*strawberry*, *vitamin C rich*) in $\mathcal{K}$. Clearly this is a heuristic strategy, which

[3] https://concept.research.microsoft.com/Home/Download

[4] Specifically, we used those pairs maximising the *Relations* frequency.

may produce non-sensical or misleading pairs. For instance, according to MSCG, strawberry is a *low-sugar berry*, but this does not entail that strawberry has the property *low-sugar* in general. However, we may expect most of the pairs that are generated with this strategy to be meaningful. A total of 8186 concept–property pairs were obtained in this way. We refer to the resulting dataset as PREFIX.

Finally, going beyond concept-hypernym pairs, we derive a dataset from GenericsKB (Bhakthavatsalam et al., 2020), which contains generic sentences such as "*Bananas are an important food staple in the tropics*". Due to the regular structure of such sentences, we can easily convert them into concept–property pairs; e.g. the aforementioned sentence would become (*banana*, *important food staple in the tropics*). We collect a set of 100K such pairs, by processing the sentences with the highest confidence (i.e. the ones which are most likely to be generic sentences) whose length is at most 7. The reason why we focus on shorter sentences is because they are more likely to capture salient information. We refer to this dataset as GKB.

Training Objective Given the pairs in $\mathcal{K}$, we pretrain two encoders, Φ_{con} and Φ_{prop}, using binary cross-entropy. In particular, the loss function for a given mini-batch is defined as follows:

$$\mathcal{L} = -\sum_{(c,p)\in\mathcal{K}_{\text{batch}}} \log\sigma\big(\Phi_{\text{con}}(c)\cdot\Phi_{\text{prop}}(p)\big) - \sum_{(c,p)\in\mathcal{N}_{\text{batch}}} \log\big(1-\sigma\big(\Phi_{\text{con}}(c)\cdot\Phi_{\text{prop}}(p)\big)\big)$$

Here $\mathcal{K}_{\text{batch}}$ represents the subset of $\mathcal{K}$ that is in the current mini-batch. For efficiency reasons, we sample these mini-batches as follows. First, we sample a subset $\mathcal{C}_{\text{batch}}$ of K concepts from $\mathcal{C}$. Then, for each concept c in $\mathcal{C}_{\text{batch}}$ we sample one property $p \in \mathcal{P}$ such that $(c, p) \in \mathcal{K}$. Let $\mathcal{P}_{\text{batch}}$ be the set of properties that are thus obtained. The sets of positive examples $\mathcal{K}_{\text{batch}}$ and negative examples $\mathcal{N}_{\text{batch}}$ are then defined as follows:

$$\mathcal{K}_{\text{batch}} = (\mathcal{C}_{\text{batch}} \times \mathcal{P}_{\text{batch}}) \cap \mathcal{K}$$
$$\mathcal{N}_{\text{batch}} = (\mathcal{C}_{\text{batch}} \times \mathcal{P}_{\text{batch}}) \setminus \mathcal{K}$$

In other words, the positive examples are the pairs from $\mathcal{K}$ that involve a concept from $\mathcal{C}_{\text{batch}}$ and a property from $\mathcal{P}_{\text{batch}}$. The negative examples are all the other pairs that we can form by taking a concept from $\mathcal{C}_{\text{batch}}$ and a property from $\mathcal{P}_{\text{batch}}$. This in-batch negative sampling strategy ensures that after encoding $|\mathcal{C}_{\text{batch}}|$ concepts and $|\mathcal{P}_{\text{batch}}|$ properties, we can take $|\mathcal{C}_{\text{batch}}| \times |\mathcal{P}_{\text{batch}}|$ training examples into account. Given that the encoders Φ_{con} and Φ_{prop} correspond to fine-tuned language models, and the encoding steps are thus time-consuming, in-batch negative sampling enables a significant speed-up compared to naive strategies in which positive and negative examples are sampled independently. Similar strategies are commonly used in neural information retrieval (Gillick et al., 2019).

Concept and Property Encoders The encoders Φ_{con} and Φ_{prop} correspond to fine-tuned encoder-only language models, such as BERT (Devlin et al., 2019). An important design decision is how the input to these language models is presented. For the concept encoder, the input corresponds to a string of the form "[prefix] c [suffix]", which is usually referred to as the prompt. How this prompt is chosen often has a substantial impact on the performance of a model. For instance, language models have been reported to under-perform if the input is too short (Bouraoui et al., 2020; Jiang et al., 2020). Given the importance of the choice of prompt, several strategies for automatically learning a suitable prompt have been proposed (Shin et al., 2020; Liu et al., 2021). In practice, however, carefully chosen manually designed prompts often outperform such automatically learned prompts (Ushio et al., 2021; Logan et al., 2021). For this reason, we have manually generated a number of templates and evaluated their performance on a held-out portion of the MSCG dataset. Based on this analysis[5], we use the following prompt:

$$\langle \textit{cls} \rangle \text{ [CONCEPT] means } \langle \textit{mask} \rangle \langle \textit{sep} \rangle$$

where $\langle \textit{cls} \rangle$, $\langle \textit{mask} \rangle$ and $\langle \textit{sep} \rangle$ are special tokens from the BERT vocabulary, while [CONCEPT] represents the concept to be modelled. The embedding of the concept is taken to be the contextualised vector of the $\langle \textit{mask} \rangle$ token, i.e. the representation of this token in the final layer of the language model. We use the same prompt for concepts and properties. However, note that concepts and properties are encoded using different encoders. Intuitively, we think of $\Phi_{\text{con}}(c)$ as a representation of a prototypical instance of concept c, while we view $\Phi_{\text{prop}}(p)$ as a representation of the property p itself. This is why, even when $p = c$, we would expect $\Phi_{\text{con}}(c)$ and $\Phi_{\text{prop}}(c)$ to be different. Under this

[5] Details can be found in Appendix A.

view, $\sigma(\Phi_{\text{con}}(c) \cdot \Phi_{\text{prop}}(p))$ captures the probability that a prototypical instance of c would have the property p. In other words, by using different encoders for concepts and properties, we can capture the default nature of the pairs in $\mathcal{K}$ in a natural way.

4 Experiments

In our experiments, we primarily focus on commonsense property classification, i.e. predicting whether some concept has a given property. We also demonstrate the usefulness of the concept and property encoders on the task of hypernym discovery. Finally, we also present a qualitative analysis.

Training Details We pre-train the concept and property encoders on the datasets introduced in Section 3. We also consider variants in which these datasets are combined; e.g. we write MSCG+PREFIX for the dataset combining the pairs from MSCG and PREFIX. To pre-train our model, we construct separate validation data in the same way. In particular, for MSCG, we select the validation split by taking the next 10K most confident pairs from Microsoft Concept Graph (i.e. after removing the pairs from the MSCG dataset itself), and similar for the other datasets. We train the model for 100 epochs, using early stopping with a patience of 20. We use the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate of $2e{-}6$ and set the batch size to 8. We use BERT-base-uncased as our default language model (Devlin et al., 2019), although we have also experimented with BERT-large-uncased, RoBERTa-base and RoBERTa-large (Liu et al., 2019).

4.1 Commonsense Property Classification

Datasets For commonsense property classification, we use the extended version of the McRae dataset (McRae et al., 2005) that was introduced by Forbes et al. (2019). This dataset involves a set $\mathcal{C}$ of 514 concepts and a set $\mathcal{P}$ of 50 properties. For each concept c and property p, the dataset specifies whether c has property p. The set $\mathcal{C}$ is split into a training set $\mathcal{C}_{\text{train}}$ and a test set $\mathcal{C}_{\text{test}}$[6]. During training, the models have access to the ground truth of every pair in $\mathcal{C}_{\text{train}} \times \mathcal{P}$. The models are then tested on all pairs in $\mathcal{C}_{\text{test}} \times \mathcal{P}$. We report the results in terms of the F1 score of the positive label.

As argued in the introduction, by training and testing on the same set of properties, we may not be able to faithfully test a model's ability to predict commonsense properties. For this reason, we consider an alternative setting where the set of properties is instead split into a training set $\mathcal{P}_{\text{train}}$ and a test set $\mathcal{P}_{\text{test}}$. During training, the model then gets access to the ground truth for the pairs in $\mathcal{C} \times \mathcal{P}_{\text{train}}$, while the model is evaluated on the pairs in $\mathcal{C} \times \mathcal{P}_{\text{test}}$. We use 5-fold cross-validation for this setting. Our hypothesis is that this setting will be more difficult, as it would be harder to find properties in the training data that are similar to those from the test set. However, there are nonetheless some similarities between these properties. We therefore also consider a setting in which both the concepts and properties are split into train and test fragments. The model is then trained on the pairs in $\mathcal{C}_{\text{train}} \times \mathcal{P}_{\text{train}}$ and evaluated on the pairs in $\mathcal{C}_{\text{test}} \times \mathcal{P}_{\text{test}}$. We again use a form of cross-validation. In particular, we split $\mathcal{C}$ into three folds: $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$. We similarly split $\mathcal{P}$ into three folds: $\mathcal{P}_1$, $\mathcal{P}_2$ and $\mathcal{P}_3$. In the first iteration, we train on the pairs in $(\mathcal{C}_1 \cup \mathcal{C}_2) \times (\mathcal{P}_1 \cup \mathcal{P}_2)$ and test on the pairs in $\mathcal{C}_3 \times \mathcal{P}_3$. This process is repeated nine times (as we have three ways to choose the concept test split and three ways to choose the property test split).

We have also used the CSLB Concept Property Norms[7], as a second benchmark for commonsense property classification. This dataset covers 638 concepts and 3350 properties. Similar as for McRae, the dataset indicates which concepts have which properties, but there are no standard splits. Moreover, the dataset does not explicitly contain negative examples. For this reason, for each positive example (c, p), we introduce 20 negative examples by replacing p with another property p' (such that (c, p') is not a positive example). This strategy is imperfect, as there will inevitably be some false negatives, but it should still allow us to compare the relative performance of different models. Mirroring the settings from the McRae dataset, we consider a concept-based training-test split (*Con*), a property-based split (*Prop*), and a setting where both concepts and properties are split into training and test sets (*Con+Prop*). For consistency, we use the same cross-validation strategies as for the McRae dataset. In particular, for *Con* we use a fixed split (with 90% of the concepts used for training and 10% for testing). For *Prop*, we use 5-fold cross-validation, while for *Con+Prop* we used the

[6] The split is available at https://github.com/mbforbes/physical-commonsense.

[7] https://cslb.psychol.cam.ac.uk/propnorms

Language Model	Pre-training dataset	McRae			CSLB		
		Con	Prop	C+P	Con	Prop	C+P
Random baseline		26.0	26.5	26.0	8.6	8.4	8.6
Always true		30.3	30.0	30.0	9.1	9.1	9.1
BERT-large sentence classifier (Forbes et al., 2019)		74	-	-	-	-	-
Human performance (Forbes et al., 2019)		67	-	-	-	-	-
BERT-base	No pre-training	77.7	30.7	25.2	51.8	34.1	22.4
BERT-base	MSCG	79.9	46.6	41.6	54.0	36.8	28.9
BERT-base	PREFIX	78.3	44.8	41.0	52.2	33.2	24.3
BERT-base	GKB	79.3	**50.7**	**46.0**	52.1	37.2	30.2
BERT-base	MSCG+PREFIX	80.2	47.8	43.2	53.6	37.3	29.7
BERT-base	MSCG+GKB	80.4	50.3	43.6	54.8	37.1	28.9
BERT-base	MSCG+PREFIX+GKB	79.8	49.6	44.5	54.5	39.1	32.6
BERT-large	No pre-training	75.3	36.6	25.5	54.3	36.4	17.7
RoBERTa-base	No pre-training	41.0	9.4	0.0	38.1	28.7	9.6
RoBERTa-large	No pre-training	73.7	26.9	9.4	55.3	37.8	24.8
BERT-large	MSCG+PREFIX+GKB	**80.5**	49.3	45.5	57.7	41.8	36.4
RoBERTa-base	MSCG+PREFIX+GKB	75.6	42.4	38.1	49.9	36.4	24.3
RoBERTa-large	MSCG+PREFIX+GKB	80.1	46.5	42.5	**59.0**	**42.5**	**36.0**

Table 2: Results in terms of F1 score (percentage) for commonsense property prediction.

3 × 3 fold cross-validation strategy.

Results The results for commonsense property classification are summarised in Table 2. We include the following baselines. First, the *Random* baseline predicts the positive label with 50% chance. Similarly, *Always true* predicts the positive label in all cases. Next, for the concept-split of the McRae dataset, we compare with the method from Forbes et al. (2019), where each pair (c, p) was converted into a natural language sentence. For instance, (*apple*, *is electrical*) is converted to the sentence "*An apple requires electricity*", which is then fed to a BERT classifier. Due to its manual nature, this method cannot be applied to new properties. We also include the estimate of human performance that was reported by Forbes et al. (2019). Finally, we consider a variant of our model which is directly trained on the McRae and CSLB training data, without the pre-training step.

The next set of results compare the performance of the different pre-training datasets. For these results, all models were initialised with BERT-base. We can clearly see that the pre-trained bi-encoder models outperform the variant without pre-training in nearly all settings (with the results for PREFIX on the CSLB property-split the only exception). This confirms our hypothesis that Microsoft Concept Graph and GenericsKB capture useful information about commonsense properties. Comparing the different corpora, PREFIX achieves the weakest results, which can be explained by the much smaller size of this dataset. However, combining PREFIX+MSCG outperforms MSCG in all but one case. Furthermore, as expected, the property-split (*Prop*) is considerably harder than the standard concept-split (*Con*), with the *C+P* setting being even harder. In fact, for the latter setting, the BERT-base model without pre-training cannot outperform the random classifier for McRae. Note that for CSLB, outperforming the random classifier is easier, given that more training data is available for that dataset. Crucially, while the best baselines only slightly underperform the pre-trained models for the concept-split, much larger differences are seen for the other splits. Overall, these findings confirm our hypothesis that predicting commonsense properties remains a highly challenging problem.

Finally, the table also shows results for some other language models. While the large models generally outperform their base counterparts, the differences are relatively small, and the improvements are not consistent. This finding is in accordance with the conclusion from Li et al. (2021) that even large language models are limited in the amount of commonsense knowledge they capture, and in particular that finding the right pre-training task is crucial. The RoBERTa results without pre-training are particularly disappointing, with learning failing completely in some cases. Even with the pre-training datasets, BERT-base outperforms RoBERTa base, and BERT-large outperforms RoBERTa-large.

	Con	**Prop**	**C+P**
Skip-gram ($k = 1$)	70.8	25.0	17.5
Skip-gram ($k = 3$)	53.4	9.5	5.7
GloVe ($k = 1$)	68.8	20.3	21.7
GloVe ($k = 3$)	51.4	6.8	4.9
BERT-base ($k = 1$)	**72.0**	**28.2**	**27.0**
BERT-base ($k = 3$)	55.6	14.6	19.1

Table 3: Evaluation of a nearest neighbour strategy for the McRae dataset (F1 score percentage).

		MAP	MRR	P@5
General	APSyn	1.7	3.7	1.7
	balAPInc	1.7	3.9	1.7
	SLQS	0.7	1.7	0.7
	Apollo	2.7	6.1	2.8
	Ours	**3.8**	**7.0**	**3.1**
Music	APSyn	1.1	2.6	1.3
	balAPInc	1.4	3.6	1.6
	SLQS	0.6	1.3	0.7
	ADAPT	1.9	**5.3**	1.9
	Ours	**2.3**	5.1	**2.6**
Medical	APSyn	0.7	1.4	0.7
	balAPInc	0.9	2.1	1.1
	SLQS	0.3	0.7	0.3
	ADAPT	**8.1**	**20.6**	**8.3**
	Ours	4.0	9.0	3.9

Table 4: Result of the hypernym discovery experiment.

Analysis As we have argued, models can perform well on the *Con* setting by simply transferring knowledge about similar concepts from the training data. This is analysed in more detail in Table 3, which shows the performance of a nearest neighbour classifier. To classify a test pair (c, p) we find the k concepts $c_1, ..., c_k$ from the training split that are most similar to c in terms of cosine similarity. Then we predict the positive label for (c, p) if the majority of $(c_1, p), ..., (c_k, p)$ are assigned the positive label. We test this approach for $k = 1$ and $k = 3$, using embeddings from GloVe (Pennington et al., 2014) and Skip-gram[8] (Mikolov et al., 2013), and using the embeddings predicted by our BERT-base encoder pre-trained on MSCG+PREFIX+GKB. For the *Prop* setting, we similarly predict the label of (c, p) based on the labels of the training pairs $(c, p_1), ..., (c, p_k)$, with $p_1, ..., p_k$ the k properties from the training data that are most similar to p. Finally, for *C+P*, we predict the majority label among the training pairs (c_i, p_j) with $i, j \in \{1, ..., k\}$, where $c_1, ..., c_k$ are the training concepts most similar to c and $p_1, ..., p_k$ are the training concepts most similar to p. The results in Table 3 clearly support our hypothesis about the concept-split. In particular, the nearest neighbour classifier is highly effective for the concept-split (for $k = 1$), outperforming the estimate of human performance from Forbes et al. (2019) for all embedding types, and approaching the performance of the language models without our pre-training task. In contrast, for the *Prop* and *C+P* settings, the nearest neighbour classifier performs, at best, similarly to the random classifier.

4.2 Hypernym Discovery

Given an input word (e.g. *cat*), the aim of the *hypernym discovery* task is to retrieve a set of valid hypernyms (e.g. *animal*, *mammal*, *feline*, etc.). We use this task to analyse the quality of the pre-trained concept and property encoders when used without any fine-tuning on task-specific training data. We use the data from the SemEval 2018 Hypernym Discovery task (Camacho-Collados et al., 2018), focusing on the concept-only split (i.e. without considering named entities). There are three variants of this task: an open-domain setting (referred to as *general*) and two domain-specific settings, focusing on the *music* and *medical* domains. Each variant is associated with a large vocabulary of candidate terms, consisting of 218,753 terms for *general*, 69,118 terms for *music* and 93,888 terms for *medicine*. To solve this task, each word from the vocabulary is encoded using Φ_{prop}. We then use maximum inner product search to efficiently find those words w from the vocabulary that maximise $\Phi_{\mathsf{con}}(t) \cdot \Phi_{\mathsf{prop}}(w)$ for a given target word t. From the retrieved list of words, we remove those that contain the term t itself and those that end with an adjective. For this experiment, we use BERT-large encoders pre-trained on MSCG+PREFIX+GKB. We compare our method with the following baselines for this task: APSyn (Santus et al., 2016), balAPInc (Kotlerman et al., 2010), SLQS (Santus et al., 2014), ADAPT (Maldonado and Klubička, 2018) and Apollo (Onofrei et al., 2018). We report the published results from the SemEval task Camacho-Collados et al. (2018) (where ADAPT only participated in the *general* setting and Apollo only participated in the *music* and *medical* settings). The latter systems achieved the best per-

[8] We used the 300 dimensional Skip-gram vectors trained on Google News and GloVe vectors trained on Common Crawl, available from `https://radimrehurek.com/gensim/models/word2vec.html`.

formance among the unsupervised methods[9]. Following Camacho-Collados et al. (2018), we report Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), and Precision at 5 (P@5), in percentage terms. Table 4 shows that our method outperforms all baselines for *General*, performs similar to ADAPT for *Music* and worse than ADAPT for *Medical*. This is remarkable, given that our method was not designed or tuned for this task. The underperformance on *Medical* can be explained by the lack of training examples from this domain in the pre-training data. As can be observed, the results for all models are low. An error analysis, presented below, revealed that this is largely due to the fact that many correct hypernyms are not included in the ground truth.

Error Analysis Table 5 shows some of the predictions of our model for the *General* setting of the hypernym discovery task. The first set of results shows examples where many of the predicted hypernyms are intuitively correct. However, only few of these hypernyms are covered by the ground truth; ground truth predictions are shown in bold. This illustrates the rather noisy nature of the dataset, and serves as an explanation for the low overall F1 score of the different unsupervised models. The second set of results in Table 5 covers cases where most of the predictions are incorrect. In some cases, e.g. for *children*, the model predicts semantic properties rather than hypernyms, which shows that simply filtering predictions that end with an adjective is not always sufficient. The case of *broiler chicken* shows that the model sometimes predicts terms that are semantically related, but which are clearly not hypernyms (nor semantic attributes). As a variant of this observation, the case of *sigma* shows that the model sometimes tends to predict co-hyponyms.

4.3 Qualitative Analysis

As a qualitative analysis, we use our pre-trained models to predict which properties are associated with a given concept. We consider the set of all properties that appear at least 10 times in an extended version of the PREFIX+GKB dataset[10], leading to a set of 5223 candidate properties. We again use maximum inner product search to efficiently identify the properties whose embeddings are closest to the concept embedding $\phi_{\textsf{con}}(c)$. Table 6 shows the seven nearest properties for a number of selected concepts, where we used BERT-base pretrained on MSCG+PREFIX+GKB. Specifically, the table first revisits the examples from Table 1. Subsequently, the table lists physical concepts, for which we expected predicting properties to be easier, and abstract concepts, for which we expected the task to be harder. Finally, we included adjectives to explore whether our model can be used for learning property entailment.

The results contain a mixture of hypernyms and semantic attributes, which is a reflection of how the model was trained. For physical concepts, the results are generally meaningful, with a few exceptions. For instance, *military vehicle* is incorrectly listed as a hypernym of *airplane*. Regarding the abstract concepts, the top predictions are mostly meaningful, but we can also see terms that are semantically related but are neither hypernyms nor semantic attributes; e.g. we see *parties* as a property of *celebration*. Finally, for the adjectives, we see several instances where the entailment direction is reversed, for instance when *dessert* is mentioned as a property of *sugary*.

5 Conclusions

We studied the problem of modelling the commonsense properties of concepts. We argued that the standard evaluation setting does not faithfully assess the extent to which models capture knowledge about commonsense properties, and proposed two new evaluation settings. These new settings were found to be highly challenging for language models, with performance being close to random. We furthermore found that pre-training a bi-encoder model on hypernymy data or generic sentences can lead to substantial performance gains. However, there remains a lot of room for further improvements, which will likely require novel insights.

Acknowledgments

This work has been supported by EPSRC grant EP/V025961/1. We acknowledge the support of the Supercomputing Wales project, which is part-funded by the European Regional Development Fund (ERDF) via Welsh Government.

[9] The hypernym discovery datasets are strongly biased in which hypernyms were preferred by the annotators. Such biases can only be learned from the task-specific training data, which is why we do not compare with supervised methods.

[10] This extended dataset involves 500K pairs from Microsoft Concept Graph and 500K sentences from GenericsKB; analysis about this extended dataset is provided in Appendix B.

Hyponym	Top-5 Predicted Hypernyms
liberty	principle, notion, ideal, universal value, humanitas
longbow	hunting weapon, **weapon**, bow and arrow, wieldy, choptank
wine	**drink**, **beverage**, liquidity, alcoholic beverage, drinking alcohol
manslaughter	culpable homicide, murder charge, offence, justifiable homicide, first-degree murder
shopping	chore, specific activity, everyday, simple interest, pursuit
running	aerobic, cardio, endurance training, aerobic exercise, sport
computer industry	sector, sunrise industry, growth industry, field of operation, game industry
learner	understander, student, realizer, know-all, nonjoinder
snow	weather condition, weather, cold weather, bad weather, wet-weather
bounty hunter	vigilante, hired gun, bandit, bondman, trail boss
metre	**unit of length**, unit of measure, measuring unit, quantity unit, derived unit
hero	protagonist, archetype, archetypic, personage, literaty character
website	resource, e-resource, information source, **medium**, source
violin	**string instrument**, **musical instrument**, second fiddle, **bowed instrument**, **stringed instrument**
arms	head and shoulders, legs, straighten, stiffen, bare bones
cooking ingredient	composition, culinary, adjunct, importune, condiment
children	learn, memorize, make fun, come to life, lose track
broiler chicken	chicken cordon bleu, chicken stock, hot chicken, kung pao chicken, chicken broth
observation	qualitative, empirical research, qualitative analysis, data collection, qualitative research
sigma	lambda, upsilon, fraternity, epsilon, alpha and omega
apartment	tenantless, adjacent, low-rent, homeplace, residential building
wetsuit	drysuit, nonsuit, life-jacket, diving equipment, diving suit
yesterday	thisday, tomorrow, timea, timeless, evermore
taxi	off-license, car rental, bus service, bike rental, cab fare

Table 5: Error analysis for hypernym discovery on the general dataset. Correctly predicted hypernyms are shown in bold.

Concept	Predicted properties
banana	food, fruit, fresh, plant, edible, tropical, commercially important
lion	animal, mammal, wildcat, carnivore, species, very territorial, mammalian
airplane	vehicle, aircraft, stationary, application, object, military vehicle, automotive
straw	material, combustible, porous, stuff, fibrous, located in wood, has sections
ice	cold, has temperature, has surfaces, located in freezers, has density, authorization, albums
yacht	boat, vehicle, vessel, recreational, ship, expensive, aircraft
coffee	beverages, drinks, beverage, drink, liquid, liquids, located in supermarkets
steel	material, non-ferrous, non ferrous, rigid, product, industrial, heavy
fire	causes burns, creates heat, produces heat, causes damage, can have effects, generates heat, produce crops
beer	beverage, drink, alcoholic, liquor, liquid, beverages, drinks
democracy	principle, idea, democratic, ideology, concept, morality, value, moral
disappointment	negative, feeling, emotion, emotional, feelings, positive, depression
promotion	marketing, achievement, activity, corporate, factor, acts, activities
celebration	event, festivity, occasion, social events, parties, events, activities
forgiveness	moral, value, love, virtue, emotion, benign, principle
lawyer	professional, adult, allied, profession, consultant, closely related, expert
stressful	situation, factor, emotional, difficult, unexpected, uncomfortable, traumatic
poisonous	poison, harmless, harmful, dangerous, toxin, aggressive, sharp
sugary	dessert, taste, food, delicious, chocolate, frozen dessert, candy
rewarding	activities, clocks, happiness, treatments, approval, actions, human activities
modern	style, genre, contemporary, fashion, broad, musical style, english
alcoholic	alcoholic, liquor, drink, beverage, mixed, alcohol, addictive, aggressive

Table 6: Qualitative analysis, showing the top neighbours of the embeddings of selected concepts.

References

Marianna Apidianaki and Aina Garí Soler. 2021. ALL dolphins are intelligent and SOME are friendly: Probing BERT for nouns' semantic properties and their prototypicality. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 79–94.

Sumithra Bhakthavatsalam, Chloe Anastasiades, and Peter Clark. 2020. Genericskb: A knowledge base of generic statements. *CoRR*, abs/2005.00660.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings* *of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. 2020. Inducing relational knowledge from bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7456–7463.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 Task 9: Hypernym Discovery. In *Proceedings of SemEval*, New Orleans, LA, United States.

Shib Sankar Dasgupta, Michael Boratko, Shriya Atmakuri, Xiang Lorraine Li, Dhruvesh Patel, and Andrew McCallum. 2021. Word2box: Learning word representation using box embeddings. *CoRR*, abs/2106.14361.

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.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? In *Proceedings of the 41th Annual Meeting of the Cognitive Science Society*, pages 1753–1759.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 528–537, Hong Kong, China. Association for Computational Linguistics.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction, AKBC@CIKM 13, San Francisco, California, USA, October 27-28, 2013*, pages 25–30. ACM.

Michael Hanna and David Mareček. 2021. Analyzing BERT's knowledge of hypernymy via prompting. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 275–282, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 Volume 2: The 14th International Conference on Computational Linguistics*.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 6384–6392. AAAI Press.

Lei Ji, Yujing Wang, Botian Shi, Dawei Zhang, Zhongyuan Wang, and Jun Yan. 2019. Microsoft concept graph: Mining semantic concepts for short text understanding. *Data Intell.*, 1(3):238–270.

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.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Matthew Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3231–3241, Florence, Italy. Association for Computational Linguistics.

Aoxue Li, Weiran Huang, Xu Lan, Jiashi Feng, Zhenguo Li, and Liwei Wang. 2020. Boosting few-shot learning with adaptive margin loss. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12573–12581.

Xiang Lorraine Li, Adhiguna Kuncoro, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2021. A systematic investigation of commonsense understanding in large language models. *CoRR*, abs/2111.00607.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. GPT understands, too. *arXiv preprint arXiv:2103.10385*.

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

Robert L. Logan, Ivana Balazevic, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. 2021. Cutting down on prompts and parameters: Simple few-shot learning with language models. *CoRR*, abs/2106.13353.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Yukun Ma, Erik Cambria, and Sa Gao. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180.

Alfredo Maldonado and Filip Klubička. 2018. ADAPT at SemEval-2018 task 9: Skip-gram word embeddings for unsupervised hypernym discovery in specialised corpora. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 924–927, New Orleans, Louisiana. Association for Computational Linguistics.

Ken McRae et al. 2005. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37:547–559.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Tuan-Phong Nguyen, Simon Razniewski, Julien Romero, and Gerhard Weikum. 2021. Refined commonsense knowledge from large-scale web contents. *CoRR*, abs/2112.04596.

Mihaela Onofrei, Ionuț Hulub, Diana Trandabăț, and Daniela Gîfu. 2018. Apollo at SemEval-2018 task 9: Detecting hypernymy relations using syntactic dependencies. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 898–902, New Orleans, Louisiana. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2463–2473. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Enrico Santus, Alessandro Lenci, Tin-Shing Chiu, Qin Lu, and Chu-Ren Huang. 2016. Unsupervised measure of word similarity: how to outperform co-occurrence and vector cosine in vsms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte Im Walde. 2014. Chasing hypernyms in vector spaces with entropy. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 38–42.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6863–6870, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, pages 935–943.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 4444–4451. AAAI Press.

Kunihiro Takeoka, Kosuke Akimoto, and Masafumi Oyamada. 2021. Low-resource taxonomy enrichment with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2747–2758, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Asahi Ushio, Jose Camacho-Collados, and Steven Schockaert. 2021. Distilling relation embeddings from pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9044–9062, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nathaniel Weir, Adam Poliak, and Benjamin Van Durme. 2020. Probing neural language models for human tacit assumptions. In *Proceedings of the 42th*

Annual Meeting of the Cognitive Science Society - Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020, virtual, July 29 - August 1, 2020.

Peter West, Chandra Bhagavatula, Jack Hessel, Jena D. Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2021. Symbolic knowledge distillation: from general language models to commonsense models. *CoRR*, abs/2110.07178.

Chen Xing, Negar Rostamzadeh, Boris N. Oreshkin, and Pedro O. Pinheiro. 2019. Adaptive cross-modal few-shot learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems*, pages 4848–4858.

Kun Yan, Zied Bouraoui, Ping Wang, Shoaib Jameel, and Steven Schockaert. 2021. Aligning visual prototypes with BERT embeddings for few-shot learning. In *Proceedings of the International Conference on Multimedia Retrieval*, pages 367–375.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, and Xiang Ren. 2021. Pre-training text-to-text transformers for concept-centric common sense. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

A Prompt Analysis

Previous work has found that the prompt which is used can have a material impact on the performance of BERT-based encoders (Bouraoui et al., 2020; Jiang et al., 2020; Shin et al., 2020; Liu et al., 2021; Ushio et al., 2021; Logan et al., 2021). To analyse the impact of the prompt in our setting, and make a suitable choice, we experimented with a number of different, manually chosen prompts. For these experiments, we used the most confident 11,000 concept-property pairs of the MSCG dataset for training, and the next 1200 concept-property pairs for tuning. The batch size is set to 8. We used the AdamW optimizer and learning rate $2e{-}6$, using early stopping with a patience of 20. The results in Table 7 are reported in terms of the F1 score (percentage) of the positive label. For the first two results in the table, a different prompt was used for the concept and property encoders. The property prompts corresponding to these two configurations are (not shown in the table):

- ⟨*cls*⟩ Property: [CONCEPT] ⟨*sep*⟩
- ⟨*cls*⟩ Yesterday, I saw a thing which is [PROPERTY] ⟨*sep*⟩

For the first six configurations in the table, we use the average of the embeddings of all tokens, in the final layer of the BERT-base model, as the embedding of the concept and property. For the remaining seven configurations, we use the embedding of the ⟨*mask*⟩ token in the final layer instead. The results show that many of the prompts lead to a relatively similar performance, as long as the prompt is sensible. The example with the nine mask tokens (Prompt 5) show that without a semantically informative prompt the performance drops somewhat. A similar observation can be made for the prompt about the spaceship (Prompt 10). Earlier work has suggested that longer prompts tend to perform better. To some extent this is confirmed by our results. For instance, Prompt 12 outperforms the similar but shorter Prompts 9 and 11, although Prompt 13, which is an even longer variant, performs worse. Moreover, we can see that some of the shortest prompts nonetheless perform well. All things being equal, having a shorter prompt is desirable, as it means we can use larger batch sizes and faster training. For this reason, we have decided, based on these results, to use Prompt 7, whose performance is close to that of the best-performing prompt, despite also being one of the shortest ones.

B Size of the Pre-Training Corpus

The MSCG corpus was obtained by taking the 100K pairs from Microsoft Concept Graph (Ji et al., 2019) with the highest confidence. Similarly, GKB was obtained by taking the 100K sentences with the highest confidence in GenericsKB (Bhakthavatsalam et al., 2020). This choice represents a trade-off: choosing more pairs would increase the overall amount of training data, which could improve the performance of the encoders. However, this would also mean including less reliable pairs, which might have a negative effect. In particular, both Microsoft Concept Graph and GenericsKB have been extracted from text corpora. In both cases, it can be clearly observed that the pairs/sentences with the lowest confidence are often rather noisy. To analyse this trade-off, Table 8 shows the results of an experiment where we used the top 500K pairs in MSCG and the 500K most confidence sentences in GenericsKB. Similarly, PREFIX was derived from the larger MSCG dataset for these experiments. The results show a small improvement for MSCG. However, for the *Prop* and *C+P* settings, the GKB results are actually worse for the 500K setting. These results suggest that the optimal setting might use more than 100K pairs from

	Prompt	F1
1.	⟨*cls*⟩ Concept: [CONCEPT] ⟨*sep*⟩	85.6
2.	⟨*cls*⟩ Yesterday, I saw another [CONCEPT] ⟨*sep*⟩	86.1
3.	⟨*cls*⟩ The notion we are modelling is [CONCEPT] ⟨*sep*⟩	86.7
4.	⟨*cls*⟩ The notion we are modelling: [CONCEPT] ⟨*sep*⟩	87.3
5.	⟨*cls*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ [CONCEPT] ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*mask*⟩ ⟨*sep*⟩	84.8
6.	⟨*cls*⟩ The notion we are modelling is called CONCEPT ⟨*sep*⟩	86.0
7.	⟨*cls*⟩ CONCEPT means ⟨*mask*⟩ ⟨*sep*⟩	87.1
8.	⟨*cls*⟩ CONCEPT ⟨*sep*⟩ ⟨*mask*⟩ ⟨*sep*⟩	86.6
9.	⟨*cls*⟩ The notion we are modelling is CONCEPT ⟨*sep*⟩ ⟨*mask*⟩ ⟨*sep*⟩	86.8
10.	⟨*cls*⟩ The spaceship we are modelling is CONCEPT ⟨*sep*⟩ ⟨*mask*⟩ ⟨*sep*⟩	85.8
11.	⟨*cls*⟩ We are modelling CONCEPT ⟨*sep*⟩ ⟨*mask*⟩ ⟨*sep*⟩	86.4
12.	⟨*cls*⟩ The notion we are modelling this morning is CONCEPT ⟨*sep*⟩ ⟨*mask*⟩ ⟨*sep*⟩	87.0
13.	⟨*cls*⟩ As I have mentioned earlier, the notion we are modelling this morning is CONCEPT ⟨*sep*⟩ ⟨*mask*⟩ ⟨*sep*⟩	86.3

Table 7: Performance of different prompts on a held-out portion of the MSCG dataset, in terms of F1-score percentage. BERT-base was used as the language model in these experiments.

		Con	Prop	C+P
500K	MSCG	80.1	48.6	42.8
	PREFIX	78.1	45.0	41.7
	GKB	80.6	48.8	43.7
	MSCG+PREFIX	79.8	49.1	43.4
	MSCG+GKB	**80.7**	48.8	41.5
	MSCG+PREFIX+GKB	80.3	47.5	41.0
100K	MSCG	79.9	46.6	41.6
	PREFIX	78.3	44.8	41.0
	GKB	79.3	**50.7**	**46.0**
	MSCG+PREFIX	80.2	47.8	43.2
	MSCG+GKB	80.4	50.3	43.6
	MSCG+PREFIX+GKB	79.8	49.6	44.5

Table 8: Evaluation on the McRae dataset of a variant in which 500K pairs from Microsoft Concept Graph and GenericsKB were used. Results are reported in terms of F1 score percentage. BERT-base was used as the language model in these experiments.

Microsoft Concept Graph, but fewer than 500K sentences from GenericsKB. However, the results also show that any performance gains arising from optimising the selection of the pre-training data are likely to be small.