Commonsense Knowledge Mining from Pretrained Models

Joshua Feldman*, **Joe Davison***, **Alexander M. Rush**
School of Engineering and Applied Sciences
Harvard University
{joshua_feldman@g, jddavison@g, srush@seas}.harvard.edu

Abstract

Inferring commonsense knowledge is a key challenge in natural language processing, but due to the sparsity of training data, previous work has shown that supervised methods for commonsense knowledge mining underperform when evaluated on novel data. In this work, we develop a method for generating commonsense knowledge using a large, pre-trained bidirectional language model. By transforming relational triples into masked sentences, we can use this model to rank a triple's validity by the estimated pointwise mutual information between the two entities. Since we do not update the weights of the bidirectional model, our approach is not biased by the coverage of any one commonsense knowledge base. Though this method performs worse on a test set than models explicitly trained on a corresponding training set, it outperforms these methods when mining commonsense knowledge from new sources, suggesting that unsupervised techniques may generalize better than current supervised approaches.

1 Introduction

Commonsense knowledge consists of facts about the world which are assumed to be widely known. For this reason, commonsense knowledge is rarely stated explicitly in natural language, making it challenging to infer this information without an enormous amount of data (Gordon and Van Durme, 2013). Some have even argued that machine learning models cannot learn common sense implicitly (Davis and Marcus, 2015).

One method for mollifying this issue is directly augmenting models with commonsense knowledge bases (Young et al., 2018), which typically contain high-quality information but with low coverage. These knowledge bases are represented as a graph, with nodes consisting of conceptual entities (i.e. `dog`, `running away`, `excited`, etc.) and the pre-defined edges representing the nature of the relations between concepts (`IsA`, `UsedFor`, `CapableOf`, etc.). Commonsense knowledge base completion (CKBC) is a machine learning task motivated by the need to improve the coverage of these resources. In this formulation of the problem, one is supplied with a list of candidate entity-relation-entity triples, and the task is to distinguish which of the triples express valid commonsense knowledge and which are fictitious (Li et al., 2016).

Several approaches have been proposed for training models for commonsense knowledge base completion (Li et al., 2016; Jastrzebski et al., 2018). Each of these approaches uses some sort of supervised training on a particular knowledge base, evaluating the model's performance on a held-out test set from the same database. These works use relations from ConceptNet, a crowd-sourced database of structured commonsense knowledge, to train and validate their models (Liu and Singh, 2004). However, it has been shown that these methods generalize poorly to novel data (Li et al., 2016; Jastrzebski et al., 2018). Jastrzebski et al. (2018) demonstrated that much of the data in the ConceptNet test set were simply rephrased relations from the training set, and that this train-test set leakage led to artificially inflated test performance metrics. This problem of train-test leakage is typical in knowledge base completion tasks (Toutanova et al., 2015; Dettmers et al., 2018).

Instead of training a predictive model on any specific database, we attempt to utilize the world knowledge of large language models to identify commonsense facts directly. By constructing a candidate piece of knowledge as a sentence, we can use a language model to approximate the likelihood of this text as a proxy for its truthfulness.

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing
and the 9th International Joint Conference on Natural Language Processing*, pages 1173–1178,
Hong Kong, China, November 3–7, 2019. ©2019 Association for Computational Linguistics

In particular, we use a masked language model to estimate point-wise mutual information between entities in a possible relation, an approach that differs significantly from fine-tuning approaches used for other language modeling tasks. Since the weights of the model are fixed, our approach is not biased by the coverage of any one dataset. As we might expect, our method underperforms when compared to previous benchmarks on the ConceptNet common sense triples dataset (Li et al., 2016), but demonstrates a superior ability to generalize when mining novel commonsense knowledge from Wikipedia.

Related Work Schwartz et al. (2017) and Trinh and Le (2018) demonstrate a similar approach to using language models for tasks requiring commonsense, such as the Story Cloze Task and the Winograd Schema Challenge, respectively (Mostafazadeh et al., 2016; Levesque et al., 2012). Bosselut et al. (2019) and Trinh and Le (2019) use unidirectional language models for CKBC, but their approach requires a supervised training step. Our approach differs in that we intentionally avoid training on any particular database, relying instead on the language model's general world knowledge. Additionally, we use a bidirectional masked model which provides a more flexible framework for likelihood estimation and allows us to estimate point-wise mutual information. Although it is beyond the scope of this paper, it would be interesting to adapt the methods presented here for the related task of generating new commonsense knowledge (Saito et al., 2018).

2 Method

Given a commonsense head-relation-tail triple $x = (\mathbf{h}, r, \mathbf{t})$, we are interested in determining the validity of that tuple as a representation of a commonsense fact. Specifically, we would like to determine a numeric score $y \in \mathbb{R}$ reflecting our confidence that a given tuple represents true knowledge.

We assume that heads and tails are arbitrary-length sequences of words in a vocabulary $\mathcal{V}$ so that $\mathbf{h} = \{h_1, h_2, \ldots, h_n\}$ and $\mathbf{t} = \{t_1, t_2, \ldots, t_m\}$. We further assume that we have a known set of possible relations $\mathcal{R}$ so that $r \in \mathcal{R}$.

The goal is to determine a function f that maps relational triples to validity scores. We propose decomposing $f(x) = \sigma(\tau(x))$ into two subcomponents: a sentence generation function τ which maps a triple to a single sentence, and a scoring model σ which then determines a validity score y.

Our approach relies on two types of pretrained language models. Standard unidirectional models are typically represented as autoregressive probabilities:

$$p(w_1, w_2, \ldots, w_m) = \prod_i^m p(w_i | w_1, \ldots, w_{i-1})$$

Masked bidirectional models such as BERT, proposed by Devlin et al. (2018), instead model in both directions, training word representations conditioned both on future and past words. The masking allows any number of words in the sequence to be hidden. This setup provides an intuitive framework to evaluate the probability of any word in a sequence conditioned on the rest of the sequence,

$$p(w_i | w'_{1:i-1}, w'_{i+1:m})$$

where $w' \in \mathcal{V} \cup \{\kappa\}$ and κ is a special token indicating a masked word.

2.1 Generating Sentences from Triples

We first consider methods for turning a triple such as `(ferret, AtLocation, pet store)` into a sentence such as "the ferret is in the pet store". Our approach is to generate a set of candidate sentences via hand-crafted templates and select the best proposal according to a language model.

For each relation $r \in \mathcal{R}$, we hand-craft a set of sentence templates. For example, one template in our experiments for the relation `AtLocation` is, "you are likely to find `HEAD` in `TAIL`". For the above example, this would yield the sentence, "You are likely to find ferret in pet store".

Because these sentences are not always grammatically correct, such as in the above example, we apply a simple set of transformations. These consist of inserting articles before nouns, converting verbs into gerunds, and pluralizing nouns which follow numbers. See the supplementary materials for details and Table 1 for an example. We then enumerate a set of alternative sentences $\mathcal{S} = \{S_1, \ldots, S_j\}$ resulting from each template and from all combinations of transformations. This yields a set of candidate sentences for each data point. We then select the candidate sentence with the highest log-likelihood according to a pre-trained unidirectional language model P_{coh}.

$$S^* = \arg\max_{S \in \mathcal{S}} [\log P_{\text{coh}}(S)]$$

Candidate Sentence S_i	$\log p(S_i)$
"musician can playing musical instrument"	-5.7
"musician can be play musical instrument"	-4.9
"musician often play musical instrument"	-5.5
"a musician can play a musical instrument"	$\mathbf{-2.9}$

Table 1: Example of generating candidate sentences. Several enumerated sentences for the triple `(musician, CapableOf, play musical instrument)`. The sentence with the highest log-likelihood according to a pretrained language model is selected.

We refer to this method of generating a sentence from a triple as COHERENCY RANKING. Coherency Ranking operates under the assumption that natural, grammatical sentences will have a higher likelihood than ungrammatical or unnatural sentences. See an example subset of sentence candidates and their corresponding scores in Table 1. From a qualitative evaluation of the selected sentences, we find that this approach produces sentences of significantly higher quality than those generated by deterministic rules alone. We also perform an ablation study in our experiments demonstrating the effect of each component on CKBC performance.

2.2 Scoring Generated Triples

Assuming we have generated a proper sentence from a relational triple, we now need a way to score its validity with a pretrained model that considers the relationship between the relation entities. We therefore propose using the estimated point-wise mutual information (PMI) of the head $\mathbf{h}$ and tail $\mathbf{t}$ of a triple conditioned on the relation r, defined as,

$$\text{PMI}(\mathbf{t}, \mathbf{h}|r) = \log p(\mathbf{t}|\mathbf{h}, r) - \log p(\mathbf{t}|r)$$

We can estimate these scores by using a masked bidirectional language model, P_{cmp}. In the case where the tail is a single word, the model allows us to evaluate the conditional likelihood of a single triple component $p(\mathbf{t}|\mathbf{h}, r)$ by computing $P_{\text{cmp}}(w_i = \mathbf{t} \; | w_{1:i-1}, w_{i+1:m})$ for the tail word.

In practice, the tail might be realized as a j-word phrase. To handle this complexity, we use a greedy approximation of its probability. We first mask all of the tail words and compute the probability of each. We then find the word with highest probability p_k, substitute it back in, and repeat j times. Finally, we calculate the total conditional likelihood of the tail by the product of these terms, $p(\mathbf{t}|\mathbf{h}, r) = \prod_{k=1}^{j} p_k$.

The marginal $p(\mathbf{t}|r)$ is computed similarly, but in this case we mask the head throughout. For example, to compute the marginal tail probability for the sentence, "You are likely to find a ferret in the pet store" we mask both the head and the tail and then sequentially unmask the tail words only: "You are likely to find a κ_{h1} in the κ_{t1} κ_{t2}". If κ_{t2} = "store" has a higher probability than κ_{t1} = "pet", we unmask "store" and compute "You are likely to find a κ_{h1} in the κ_{t1} store". The marginal likelihood $p(\mathbf{t}|r)$ is then the product of the two probabilities.

The final score combines the marginal and conditional likelihoods by employing a weighted form of the point-wise mutual information,

$$\text{PMI}_\lambda(\mathbf{t}, \mathbf{h}|r) = \lambda \log p(\mathbf{t}|\mathbf{h}, r) - \log p(\mathbf{t}|r)$$

where λ is treated as a hyperparameter. Although exact PMI is symmetrical, the approximate model itself is not. We therefore average $\text{PMI}_\lambda(\mathbf{t}, \mathbf{h}|r)$ and $\text{PMI}_\lambda(\mathbf{h}, \mathbf{t}|r)$ to reduce the variance of our estimates, computing the masked head values rather than the tail values in the latter.

3 Experiments

To evaluate the Coherency Ranking approach we measure whether it can distinguish between valid and invalid triples. For our masked model, we use BERT-large (Devlin et al., 2018). For sentence ranking, we use the GPT-2 117M LM (Radford et al., 2019). The relation templates and grammar transformation rules which we use can be found in the supplementary materials.

We compare the proposed method to several baselines. Following Trinh and Le (2018), we evaluate a simple CONCATENATION method for generating sentences, splitting the relation r into separate words and concatenating it with the head and tail. For the triple `(ferret, AtLocation, pet store)`, the Concatenation approach would yield, "ferret at location pet store".

We also evaluate CKBC performance when we construct sentences by applying a single hand-crafted template. Since each triple is mapped to a sentence with a single template without any grammatical transformations, we refer to this as the TEMPLATE method. Using the Template approach, `(ferret, AtLocation, pet store)`

Model	Task 1	Task 2
Unsupervised		
CONCATENATION	68.8	2.95 ± 0.11
TEMPLATE	72.2	2.98 ± 0.11
TEMPL.+GRAMMAR	74.4	2.56 ± 0.13
COHERENCY RANK	78.8	**3.00** ± 0.12
Supervised		
DNN	**89.2**	2.50
FACTORIZED	89.0	2.61
PROTOTYPICAL	79.4	2.55

Table 2: Main results for Task 1: Commonsense knowledge base completion (test F1 score) and Task 2: Wikipedia mining (quality scores out of 4). Results are included from the sentence generation methods of simple concatenation, hand-crafted templates, templates plus grammatical transformations, and coherency ranking. DNN, Factorized, and Prototypical models are described in Jastrzebski et al. (2018).

would become "You are likely to find ferret in pet store" using the template "you are likely to find HEAD in TAIL".

Next, we extend the Template method by applying deterministic grammatical transformations, which we refer to as the TEMPLATE + GRAMMAR approach. Like the full approach, these transformations involve adding articles before nouns, converting verbs into gerunds, and pluralizing nouns following numbers. The Template + Grammar approach differs from Coherency Ranking in that all transformations are applied to every sentence instead of applying combinations of transformations and templates, which are then ranked by a language model. Returning to our example, the Template + Grammar method produces "You are likely to find a ferret in a pet store". While this sentence is grammatical, applying this method to `(star, AtLocation, outer space)` yields "You are likely to find a star in an outer space", which is incorrect.

We compare our results to the supervised models from the work of Jastrzebski et al. (2018) and the best performing model from Li et al. (2016). Jastrzebski et al. (2018) introduce FACTORIZED and PROTOTYPICAL models. The Factorized model embeds the head, relation, and tail in a vector space and then produces a score by taking a linear combination of the inner products between each pair of embeddings. The Prototypical model is similar, but does not include the inner product between head and tail. Li et al. (2016) evaluate a deep neural network (DNN) for CKBC. They concatenate embeddings for the head, relation, and tail, which they then feed through a multilayer perceptron with one hidden layer. All three models are trained on 100,000 ConceptNet triples.

Task 1: Commonsense Knowledge Base Completion Our experimental setup follows Li et al. (2016), evaluating our model with their test set (n = 2400) containing an equal number of valid and invalid triples. The valid triples are from the crowd-sourced Open Mind Common Sense (OMCS) entries in the ConceptNet 5 dataset (Speer and Havasi, 2012). Invalid triples are generated by replacing an element of a valid tuple with another randomly selected element.

We use our scoring method to classify each tuple as valid or invalid. To this end, we use our method to assign a score to each tuple and then group the resulting scores into two clusters. Instances in the cluster with the higher mean PMI are labeled as valid, and the remainder are labeled as invalid. We use expectation-maximization with a mixture of Gaussians to cluster. We also tune the PMI weight via grid search over 90 points from $\lambda \in [0.5, 5.]$, using the Akaike information criterion of the Gaussian mixture model for evaluation (Akaike, 1974).

Table 2 shows the full results. Our unsupervised approach achieves a test set F1 score of 78.8, comparable to the 79.4 F1 score found by the supervised prototypical approach. The Factorized and DNN models significantly outperformed our approach with F1 scores of 89.2 and 89.0, respectively. Our grid search found an optimal λ value of 1.65 for the Concatenation sentence generation model and 1.55 for the Coherency Ranking model. The Template and Template + Grammar methods found lambda values of 1.20 and 0.95, respectively.

Task 2: Mining Wikipedia To assess the model's ability to generalize to unseen data, we evaluate our unsupervised model in comparison to previous supervised methods on the task of mining commonsense knowledge from Wikipedia. In their evaluations, Li et al. (2016) curate a set of 1.7M triples across 10 relations by applying part-of-speech patterns to Wikipedia articles. We sample 300 triples from each relation. We apply our method to evaluate these 3000 triples. Using the approach described by Speer and Havasi (2012), and followed by Li et al. (2016) and Jastrzebski et al. (2018), two human annotators manually rate the 100 triples with the highest predicted score on a 0 to 4 scale: 0 (Doesn't make sense), 1 (Not true), 2 (Opinion/Don't know), 3 (Sometimes true), and 4 (Generally true). We tuned λ by measuring the quality of the 100 triples with the highest predicted score across $\lambda \in \{1, 2, \ldots, 9, 10\}$.

The top 100 triples selected by our model were assigned a mean rating of 3.00 ($\lambda = 4$) with a standard error of 0.11 under the Coherency Ranking approach, well exceeding the performance of current supervised methods (Table 2). Standard errors were calculated using 1000 bootstrap samples of the top 100 triples. The ratings assigned by the two human annotators had a 0.50 Pearson correlation and 0.23 kappa inter-annotator agreement. Rater disagreements occur most frequently when triples are ambiguous or difficult to interpret. Notably, if we bucket the five scores into just two categories of *true* and *false*, this disagreement rate drops by 50%. To give a sense of the types of commonsense knowledge our models struggle to capture, we report the top 100 most confident predictions that receive an average score below 3 in the supplementary material. Notably, some of the top 100 triples our model identified were indeed true, but would not be reasonably considered common sense (e.g. `(vector bundle, HasProperty, manifold)`). This suggests that our approach may be applicable to mining knowledge beyond common sense.

Analysis: Sentence Generation In order to measure the impact of sentence generation on our model, we select a sample of 100 sentences and group the results by a) whether the sentence contained a grammatical error, and b) whether the sentence misrepresented the meaning of the triple. For example, the triple `(golf, HasProperty, good)` yields the sentence "golf is a good", which is grammatically correct but conveys the wrong meaning. On both Wikipedia mining and CKBC, we find that misrepresenting meaning has an adverse impact on model performance. In CKBC, we also find that grammar has a high impact on the resulting F1 scores (Table 3). Future work could therefore focus on designing templates that more reliably encode a relation's true meaning.

Task 1	**N (/100)**	**F1 Score**
Grammatical	75	79.1
Ungrammatical	25	66.7
Correct Meaning	91	77.6
Wrong Meaning	9	66.7
Task 2	-	**Quality**
Grammatical	83	3.01
Ungrammatical	17	2.88
Correct Meaning	88	3.22
Wrong Meaning	12	1.18

Table 3: Test results examining the effect of sentence meaning and grammaticality on task performance. Scores are shown for a sample of 100 triples split by whether the generated sentence is grammatical and whether it conveys the correct meaning of the triple.

4 Conclusion

We introduce a robust unsupervised method for commonsense knowledge base completion using the world knowledge of pre-trained language models. We develop a method for expressing knowledge triples as sentences. Using a bidirectional masked language model on these sentences, we can then estimate the weighted point-wise mutual information of a triple as a proxy for its validity. Though our approach performs worse on a held-out test set developed by Li et al. (2016), it does so without any previous exposure to the ConceptNet database, ensuring that this performance is not biased. In the future, we hope to explore whether this approach can be extended to mining facts that are not commonsense and to generating new commonsense knowledge outside of any given database of candidate triples. We also see potential benefit in the development of a more expansive set of evaluation methods for commonsense knowledge mining, which would strengthen the validity of our conclusions.

Acknowledgments

This work was supported by NSF research award 1845664.

References

Hirotugu Akaike. 1974. A new look at the statistical model identification. In *Selected Papers of Hirotugu Akaike*, pages 215–222. Springer.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. COMET: commonsense transformers for automatic knowledge graph construction. *CoRR*, abs/1906.05317.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58(9):92–103.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 25–30. ACM.

Stanisław Jastrzebski, Dzmitry Bahdanau, Seyedarian Hosseini, Michael Noukhovitch, Yoshua Bengio, and Jackie Chi Kit Cheung. 2018. Commonsense mining as knowledge base completion? a study on the impact of novelty. *arXiv preprint arXiv:1804.09259*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1445–1455.

Hugo Liu and Push Singh. 2004. Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv:1604.01696*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1:8.

Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. 2018. Commonsense knowledge base completion and generation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 141–150, Brussels, Belgium. Association for Computational Linguistics.

Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the ROC story cloze task. *CoRR*, abs/1702.01841.

Robert Speer and Catherine Havasi. 2012. Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning. *CoRR*, abs/1806.02847.

Trieu H. Trinh and Quoc V. Le. 2019. Do language models have common sense?

Tom Young, Erik Cambria, Iti Chaturvedi, Hao Zhou, Subham Biswas, and Minlie Huang. 2018. Augmenting end-to-end dialogue systems with commonsense knowledge. In *Thirty-Second AAAI Conference on Artificial Intelligence*.