Commonsense Knowledge Completion

Peifeng Wang
University of Southern California
peifengw@usc.edu

Abstract

Knowledge graphs, especially commonsense knowledge graphs, often face the issue of sparsity, which prevents them from serving well the downstream tasks. Knowledge completion aims to address this issue by populating the knowledge graphs with newly predicted facts. Most of the works focus on encyclopedic knowledge graphs, in which the entities and relations space are well defined. Only a few attention has been drawn to commonsense knowledge completion previously. The recent progress in language models again raises a surge interest in mining commonsense knowledge from these large-capacity models. In this note, we introduce several challenges in commonsense knowledge completion and how the traditional and recent methods (with language models) tackle these challenges.

1 Background

One particular type of knowledge which help machine better understand natural language is referred to as commonsense knowledge or background knowledge. This kind of knowledge is rarely stated explicitly in textual corpora and people try to infer this knowledge from raw text by patterns or manual annotation. Although with high precision, these methods suffer from low coverage. Therefore, researchers have been developing techniques to automatically increase the coverage by inferring missing facts. These techniques are categorized as (commonsense) knowledge completion or generation. In particular, there are two kinds of knowledge completion tasks.

1. **Triplet Classification**. Given a triplet fact (s, r, o), the goal is to develop a parametric model which provides a confidence score for evaluating the fact.
2. **Link Prediciton**. Given a incomplete triplet fact $(s, r, ?)$ or $(?, r, o)$, the goal is to predict the missing entity.

A lot of efforts have been put into encyclopedic knowledge graphs like Freebase and Wordnet. These knowledge bases have a well-defined space for entities, meaning that their entities constitute a fix set. This closed set greatly facilitate the knowledge completion since we could leverage the graph structure for better knowledge representation. However, the entities from commonsense knowledge like ConceptNet are usually arbitrary phrases. The non-canonicalized, free-form text of the entities leads to a discrepancy that we might query information about an specific entity using different words from those contained in the knowledge graphs. So it poses a requirement for the knowledge completion models for commonsense knowledge to be able to handle queries without the precise linguist form as in the original knowledge graphs.

2 Previous Method

Li et al. [2016] is one of the earliest works which propose models for defining a score function. The score function could provide a confidence score for a triplet with arbitrary entities. Two types of score function are considered as follows.

2.1 Score Functions

The first score function is based on the Bilinear Model [Yang et al., 2015] which proves to work well on traditional knowledge completion:

$$f_{\text{Bilinear}}(s, r, o) = \mathbf{s}^\top \mathbf{M_r} \mathbf{o}, \tag{1}$$

where $\mathbf{s}, \mathbf{o} \in \mathcal{R}^k$ are the representations of subject and object, $\mathbf{M_r} \in \mathcal{R}^{k \times k}$ is the parameter matrix for relation r.

Another score function is based on neural networks:

$$\begin{aligned} \mathbf{u} &= \sigma(\mathbf{W^1 v_{in}} + \mathbf{b^1}) \\ f_{NN}(s, r, o) &= \mathbf{W^2 u} + \mathbf{b^2}, \end{aligned} \tag{2}$$

where $\mathbf{v_{in}}$ is the representation of the whole triplet. The model is then trained to give high scores for positive triplets and low scores for negative triplets with closed world assumption.

2.2 Entity Representation

To encode the free textual entities into embedding space, the paper considers two approaches. One is averaging the word embeddings of the entity mention. Another is taking the pooling of the output from a bidirectional LSTM. For Eq. 1, the LSTM is seperately used for the subject and object. In Eq. 2, the LSTM is fed with the concatenation of subject and object and then the output is concatenated with a relation embedding $\mathbf{v_r}$ to create the NN input vector $\mathbf{v_{in}}$.

3 Language Model Methods

Although we have been arguing that commonsense knowledge is seldom expressed explicitly in natural language, the recent success of large language models on several NLP tasks has raised researchers' interest in investigating whether these large-capacity models encode some commonsense knowledge from huge corpora. One feasibility is that after pre-training on corpora, the language models manage to extract some implicitly stated knowledge. Here, we introduce two main categories of related works, the generative models and the discriminative ones.

3.1 Generative Models

3.1.1 COMET

Instead of representing knowledge via symbolic facts or continuous embeddings, COMmonsEnse Transformer (COMET) [Bosselut et al., 2019] proposes to model the commonsense knowledge neurally. COMET uses existing triplet facts as seed to adapt the representation of a language model (GPT) to knowledge generation. The benefits come with two folds. One is that both structured and unstructured knowledge are greatly leveraged and fused. Another is that language models are particularly suitable for generating commonsense knowledge facts in the sense that they are often loosely structured open-text descriptions.

Method. COMET uses GPT as the backbone, which is a pre-trained multi-layer transformer decoder. It recursively generate a output distribution of the target tokens based on the left context. In order to adapt GPT to triplets, COMET firstly converts each fact (s, r, o) to its natural language form:

$$\mathbf{X} = \{X^s, X^r, X^o\}, \tag{3}$$

where X^s is the token sequence that makes up the subject mention and similarly for X^r and X^o. Then COMET is trained o maximize the conditional log-likelihood of predicting the phrase object tokens, X^o:

$$\mathcal{L} = -\sum_{t=|s|+|r|}^{|s|+|r|+|o|} \log P(x_t | x_{<t}). \tag{4}$$

Experimental Results. COMET leverages ATOMIC and ConceptNet as the two knowledge seed sets. The promising results demonstrate COMET's ability to generate novel knowledge of high quality.

Model	PPL	Score	N/T *sro*	N/T *o*	Human
LSTM - *s*	-	60.83	**86.25**	7.83	63.86
CKBG (Saito et al., 2018)	-	57.17	**86.25**	**8.67**	53.95
COMET (- pretrain)	8.05	89.25	36.17	6.00	83.49
COMET - RELTOK	4.39	95.17	56.42	2.62	**92.11**
COMET	**4.32**	**95.25**	59.25	3.75	91.69

Table 1: COMET's generation on ConceptNet.

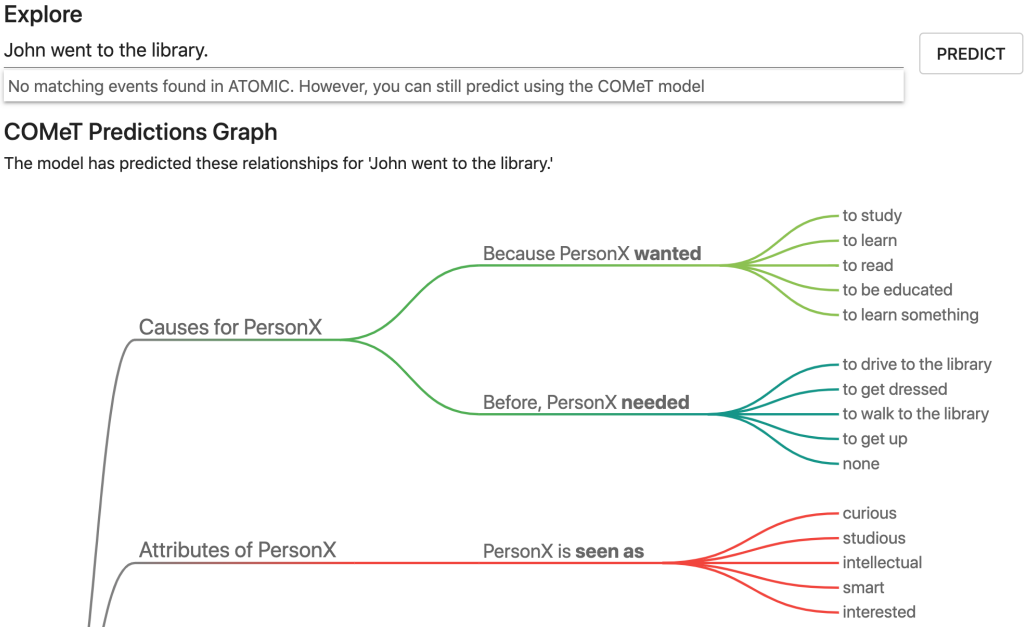

Figure 1: Prediction from COMET trained on ATOMIC. More results could be found at paper's demo `https://mosaickg.apps.allenai.org/`.

From Table 1, the paper shows the following two observations. (1) COMET with a pre-trained GPT had a clear improvement over the randomly initialized model COMET (-pretrain). (2) Converting symbolic relations to natural language is also helpful in better adapting GPT to generate knowledge compared with COMET-RELTOK baseline. Some impressive results on ATOMIC dataset are also illustrated in Figure 1.

3.1.2 Language Models as Knowledge Bases?

Rather than adapting language models with knowledge seeds, researchers are also investigating whether commonsense knowledge could be extracted from language models for free.

The paper Petroni et al. [2019] proposes to use language models as the interface for querying knowledge. The motivation is as follows. Extracting relational data from text or other modalities to populate the knowledge bases requires complex NLP pipelines involving entity extraction, coreference resolution, entity linking and relation extraction components that often need supervised data and fixed schemas. Errors can easily propagate and accumulate throughout the pipeline. So their proposed solution is to query neural language models for relational data by asking them to fill in masked tokens in sequences like "You are likely to find a overflow in a [Mask]". Some of the results on ConceptNet queries could be found in Figure 2.

Several resulting benefits include (1) requiring no schema engineering, (2) no need for human annotations, and (3) supporting an open set of queries.

One major limit of this method is that they only consider single token entities generation, since the mask token could exist in any position in the query and they want to save the trouble of multi-token

ConceptNet	AtLocation	You are likely to find a overflow in a ____.	drain	sewer [-3.1] , canal [-3.2] , toilet [-3.3] , stream [-3.6] , **drain [-3.6]**
	CapableOf	Ravens can ____.	fly	**fly [-1.5]** , fight [-1.8] , kill [-2.2] , die [-3.2] , hunt [-3.4]
	CausesDesire	Joke would make you want to ____.	laugh	cry [-1.7] , die [-1.7] , **laugh [-2.0]** , vomit [-2.6] , scream [-2.6]
	Causes	Sometimes virus causes ____.	infection	disease [-1.2] , cancer [-2.0] , **infection [-2.6]** , plague [-3.3] , fever [-3.4]
	HasA	Birds have ____.	feathers	wings [-1.8] , nests [-3.1] , **feathers [-3.2]** , died [-3.7] , eggs [-3.9]
	HasPrerequisite	Typing requires ____.	speed	patience [-3.5] , precision [-3.6] , registration [-3.8] , accuracy [-4.0] , **speed [-4.1]**
	HasProperty	Time is ____.	finite	short [-1.7] , passing [-1.8] , precious [-2.9] , irrelevant [-3.2] , gone [-4.0]
	MotivatedByGoal	You would celebrate because you are ____.	alive	happy [-2.4] , human [-3.3] , **alive [-3.3]** , young [-3.6] , free [-3.9]
	ReceivesAction	Skills can be ____.	taught	acquired [-2.5] , useful [-2.5] , learned [-2.8] , combined [-3.9] , varied [-3.9]
	UsedFor	A pond is for ____.	fish	swimming [-1.3] , fishing [-1.4] , bathing [-2.0] , **fish [-2.8]** , recreation [-3.1]

Figure 2: Examples of generation from BERT-large.

decoding. Another is that for constructing queries for ConceptNet, they have to find sentences that contain both the subject and the object from the Open Mind Common Sense (OMCS) corpus. How to create natural language query for commonsense knowledge automatically for probing language models is not investigated.

3.2 Discriminative Models

It is also feasible to apply language models to conduct the triplet classification task. Afterall, language models could provide a probability in some way for a given input. Thus, these discriminative models usually fall into the following paradigm. Firstly, they simply convert the triplet facts into natural language with minor adaption with regard to some specific language model being used. Then feed the converted facts to the language models to get a plausibility on whether they hold.

3.2.1 Commonsense Knowledge Mining from Pretrained Models

Another work trying to extract commonsense knowledge from fixed language models is Davison et al. [2019] except that they aim at solving triplet classification. The motivation of this paper is also different in that they claim those methods trained on ConceptNet generalize poorly to novel data. 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 articially inflated test performance metrics.

To determine a function $f(x)$ that maps a triplet fact x to its confidence score, the paper proposes to decompose $f(x) = \sigma(\tau(x))$ into two sub-components: a sentence generation function τ maps a triplet to a natural language sentence and a score function σ provides a confidence score.

The sentence generation function τ is based on the combination of rules and language model. The paper handcrafts a set of sentence templates $\mathbf{S}$ for each relation r. Then they select the one with highest log-likelihood when applied on a specific triplet according to a unidirectional language model P_{LM}:

$$\tau(x) = \underset{S \in \mathbf{S}}{\operatorname{argmax}}[\log P_{LM}(S)] \tag{5}$$

As for the score function σ, the paper proposes to use the estimated point-wise mutual information (PMI) of the subject and object conditioned on the relation:

$$\text{PMI}(o, s|r) = \log p(o|s, r) - \log p(o|r), \tag{6}$$

where each of the probability is estimated by a masked bidirectional language model. For $p(o|s, r)$, they mask out all the object tokens and then greedily approximate substitute back the token with highest probability. Likewise, for $p(o|r)$, they mask out the subject and object tokens and sequentially unmask the object tokens only.

The experimental results in Table 2 show that although inferior to previous supervised methods, this unsupervised method generalize better to new source of knowledge where there is no corresponding training data. It validates that without fine-tuning the language model, the proposed method is not biased towards any commonsense knowledge base.

3.2.2 Exploiting Structural and Semantic Context for Commonsense Knowledge Base Completion

The last work Malaviya et al. [2020] on discriminative model provides a unique insight on why free form text for representing entities poses a challenge for commonsense knowledge completion.

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	$\mathbf{3.00} \pm 0.12$
Supervised		
DNN	**89.2**	2.50
FACTORIZED	89.0	2.61
PROTOTYPICAL	79.4	2.55

Table 2: F1 scores on triplet classification for ConceptNet and Wikipedia knowledge.

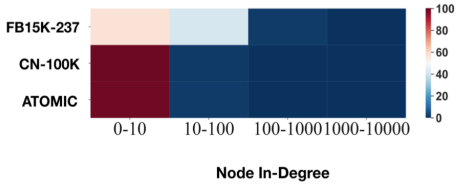

Figure 3: Degree distribution for different KBs.

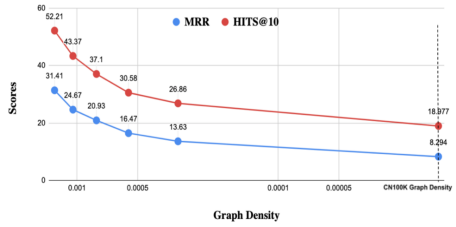

Figure 4: Degrading performance of ConvTransE model on FB15K-237 dataset with decreasing density.

Consider the nodes "prevent tooth decay" and "tooth decay" which are conceptually related, but not equivalent. The conceptual diversity and expressiveness of the graph lead to the number of nodes with orders of magnitude larger, and thus the sparsity issue is substantially severer (as shown in Figure 3.

As a result, most of the existing knowledge completion methods which implicitly assume densely connected graphs would suffer greatly from the sparsity. The claim is validated by the results in Figure 4 which shows that a high performing model degrades quickly as the the graph density is reduced.

To represent a entity, the paper proposes to infuse information from both language models and local graph structure. As shown in the encoder part of Figure 5, the representation of an entity is the combination of the outputs from the BERT and RGCN. The BERT is applied on the entity tokens and learns to transfer knowledge from language to the knowledge graph. The RGCN is applied on the neighborhood of the entity which encodes the local graph structure.

As argued by the paper, the sparsity of commonsense knowledge graph makes it challenging to perform information propagation across an entity's neighborhood. The paper addresses this issue by adding a new synthetic *sim* relation to each pair of similar entities. The similarity is defined as the cosine similarity between the entities' embeddings given by the fine-tuned BERT. Then all pairs of entities with similarity above a preset threshold are connected by the *sim* relation. The results from Table 3 show that the graph densification help improve the knowledge completion.

4 Conclusion and Discussion

The methods discussed above try to leverage information from either existing structured knowledge or unstructured text encoded in language models. They tackle the sparsity issuse brought by the free text entities with several strategies. Still, it's not clear whether and how language models capture the commonsense knowledge. Another question is that whether encoding commonsense knowledge in a triplet form is the best choice.

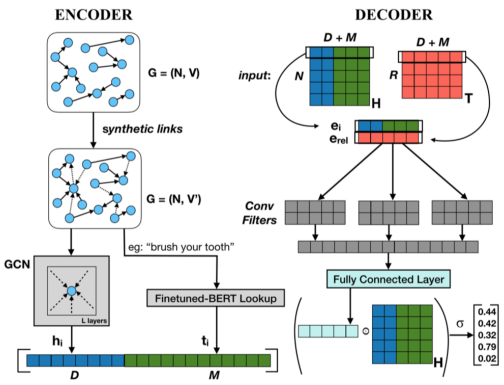

Figure 5: Model Architecture

	CN-100K				ATOMIC			
	MRR	HITS@1	@3	@10	MRR	HITS@1	@3	@10
DISTMULT	8.97	4.51	9.76	17.44	12.39	9.24	15.18	18.30
COMPLEX	11.40	7.42	12.45	19.01	**14.24**	**13.27**	14.13	15.96
CONVE	20.88	13.97	22.91	34.02	10.07	8.24	10.29	13.37
CONVTRANSE	18.68	7.87	23.87	38.95	12.94	12.92	12.95	12.98
COMET-NORMALIZED	6.07	0.08	2.92	21.17	3.36*	0.00*	2.15*	15.75*
COMET-TOTAL	6.21	0.00	0.00	24.00	4.91*	0.00*	2.40*	21.60*
BERT + CONVTRANSE	49.56	38.12	55.5	71.54	12.33	10.21	12.78	16.20
GCN + CONVTRANSE	29.80	21.25	33.04	47.50	13.12	10.70	13.74	17.68
SIM + GCN + CONVTRANSE	30.03	21.33	33.46	46.75	13.88	11.50	**14.44**	**18.38**
GCN + BERT + CONVTRANSE	50.38	38.79	56.46	72.96	10.8	9.04	11.21	14.10
SIM + GCN + BERT + CONVTRANSE	**51.11**	**39.42**	**59.58**	**73.59**	10.33	8.41	10.79	13.86

Table 3: Knowledge Completion Results on ConceptNet and ATOMIC.

References

A. Bosselut, H. Rashkin, M. Sap, C. Malaviya, A. Çelikyilmaz, and Y. Choi. Comet: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

J. Davison, J. Feldman, and A. M. Rush. Commonsense knowledge mining from pretrained models. 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.

X. Li, A. Taheri, L. Tu, and K. Gimpel. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016.

C. Malaviya, C. Bhagavatula, A. Bosselut, and Y. Choi. Exploiting structural and semantic context for commonsense knowledge base completion. In *AAAI*, 2020.

F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, and S. Riedel. Language models as knowledge bases? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. In *ICLR*, 2015.