Commonsense Knowledge Base Completion and Generation

Itsumi Saito **Kyosuke Nishida** **Hisako Asano** **Junji Tomita**
NTT Media Intelligence Laboratories
{saito.itsumi, nishida.kyosuke}@lab.ntt.co.jp,
{asano.hisako, tomita.junji}@lab.ntt.co.jp

Abstract

This study focuses on acquisition of commonsense knowledge. A previous study proposed a commonsense knowledge base completion (CKB completion) method that predicts a confidence score of triplet-style knowledge for improving the coverage of CKBs. To improve the accuracy of CKB completion and expand the size of CKBs, we formulate a new commonsense knowledge base generation task (CKB generation) and propose a joint learning method that incorporates both CKB completion and CKB generation. Experimental results show that the joint learning method improved completion accuracy and the generation model created reasonable knowledge. Our generation model could also be used to augment data and improve the accuracy of completion.

1 Introduction

Knowledge bases (KBs) are a kind of information network, and they have been applied to many natural language processing tasks such as question answering (Yang and Mitchell, 2017; Long et al., 2017) and dialog tasks (Young et al., 2018). In this paper, we focus on commonsense knowledge bases (CKBs). Commonsense knowledge is also referred to as background knowledge and is used in natural language application tasks that require reasoning based on implicit knowledge. For example, machine comprehension tasks that need commonsense reasoning have been proposed very recently (Lin et al., 2017; Ostermann et al., 2018). In particular, Wang et al. (2018) used commonsense knowledge provided by ConceptNet (Speer et al., 2017) to efficiently resolve ambiguities and infer implicit information.

Information in CKB is represented in RDF-style triples $\langle t_1, r, t_2 \rangle$, where t_1 and t_2 are arbitrary words or phrases, and $r \in R$ is a relation between t_1 and t_2. For example, $\langle \text{go to restaurant}, \mathit{subevent}, \text{order food} \rangle$ means "order food" happens as a subevent of "go to restaurant". Although researchers have developed techniques for acquiring CKB from raw text with patterns (Angeli and Manning, 2013), it has been pointed out that some sorts of knowledge are rarely expressed explicitly in textual corpora (Gordon and Van Durme, 2013). Therefore, researchers have developed curated CKB resources by manual annotation (Speer et al., 2017). While manually created knowledge has high precision, these resources suffer from lack of coverage.

Knowledge base completion methods are used to improve the coverage of existing general-purpose KBs, such as Freebase (Bollacker et al., 2008; Bordes et al., 2013; Lin et al., 2015). For example, given a node pair $\langle \text{Athens}, \text{Greece} \rangle$, a completion method predicts the missing relation "IsLocatedIn". Such KBs consist of well-connected entities; thus, the completion methods are mainly used to find missing links of the existing nodes. On the other hand, CKBs are very sparse because their nodes contain arbitrary phrases and it is difficult to define all phrases in advance. Therefore, it is important to consider CKB completion that can robustly take arbitrary phrases as input queries, even if they are not contained in the CKBs, to improve the coverage.

Li et al. (2016b) proposed an on-the-fly CKB completion model to improve the coverage of CKBs. They defined the CKB completion task as a binary classification distinguishing true knowledge from false knowledge for arbitrary triples. They proposed a simple neural network model that can embed arbitrary phrases on-the-fly and achieved reasonable accuracy for ConceptNet. Here, in order to acquire new knowledge by using a CKB completion model, we have to prepare triplet candidates as input for the completion

Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018), pages 141–150
Brussels, Belgium, October 31 - November 1, 2018. ©2018 Association for Computational Linguistics

model, because the model can only verify whether the triple is true or not. Li et al. (2016b) extracted such triplet candidates from the raw text of Wikipedia and also randomly selected from the phrase and relation set of ConceptNet. Extracts from raw text likely contain unseen phrases, i.e., ones which do not exist in the CKB, and these phrases are useful for expanding the node size of the CKB; however, they reported that the quality of triples acquired from Wikipedia were significantly lower than that of combination triples from ConceptNet, because patterns extracted from Wikipedia by using linguistic patterns are noisier than those from ConceptNet. For acquiring new knowledge with high quality, there are still problems with expanding new nodes and with the accuracy of CKB completion.

In this study, we focus on problems of increasing the node size of CKBs and increasing the connectivity of CKBs. We introduce a new commonsense knowledge base generation (CKB generation) task for generating new nodes. We also devise a model that jointly learns the completion and generation tasks. The generation task can generate an arbitrary phrase t_2 from an input query and relation pair $\langle t_1, r\rangle$. The joint learning of the two tasks improves the completion task and triples generated by the generation model can be used as additional training data for the completion model.

Our contributions are summarized as follows:

- We define a new task, called commonsense knowledge base generation, and propose a method for joint learning of knowledge base completion and knowledge base generation.

- Experimental results demonstrate that our method achieved state-of-the-art CKB completion results on both ConceptNet and Japanese commonsense knowledge datasets.

- Experimental results show that our CKB generation can generate reasonable knowledge and augmented data generated by the model can improve CKB completion.

2 Task Definition

Our study focuses on two tasks, CKB completion and CKB generation. We describe the settings of these tasks below.

Problem 1 (CKB completion)**.** *Given a triple* $\langle t_1, r, t_2\rangle$*,* CKB completion *provides a confidence score that distinguishes true triples from false ones.* t_1 *and* t_2 *are arbitrary phrases.* r *is a relation in a set* R*.*

Problem 2 (CKB generation)**.** *Given a pair of* $t_1(t_2)$ *and* $r \in R$*,* CKB generation *generates* $t_2(t_1)$*, which has a relationship* r *with* $t_1(t_2)$*.* t_1 *and* t_2 *are arbitrary phrases.*

3 Proposed Method

The proposed method is illustrated in Figure 1. Our method consists of two models. It performs both the CKB completion task and CKB generation task. Two models share the parameters of a phrase encoder, word embeddings, and relation embeddings. We describe these models in detail in Sections 3.1 and 3.2.

3.1 CKB Completion Model

The basic structure of our CKB completion model is similar to that of Li et al. (2016b). The main difference between ours and theirs is that our method learns the CKB completion and generation tasks jointly. The completion model only considers the binary classification task, and therefore, it can be easily overfitted when there are not enough training data. By incorporating the generation model, the shared layers are trained for both binary classification and phrase generation. This is expected to be a good constraint to prevent overfitting.

Previous model Li et al. (2016b) defined a CKB completion model that estimates a confidence score of an arbitrary triple $\langle t_1, r, t_2\rangle$. They used a simple neural network model to formulate $\mathrm{score}(t_1, r, t_2) \in \mathbb{R}$.

$$\mathrm{score}(t_1, r, t_2) = W_2 g(W_1 v_{in} + b_1) + b_2 \qquad (1)$$

where $v_{in} = \mathrm{concat}(v_{12}, v_r) \in \mathbb{R}^{d_v + d_r}$. $v_{12} \in \mathbb{R}^{d_v}$ is a phrase representation of concatenating t_1 and t_2. $v_r \in \mathbb{R}^{d_r}$ is a relation embedding for r. g is a nonlinear activation function. Note that we use ReLU for g.

Our model Our CKB completion model is based on Li et al.'s (2016b). However, the shared structure and the formulation of the phrase representations v_{12} are different. Li et al. (2016b) used the average of the word embeddings (called DNN AVG) and max pooling of LSTM (called DNN LSTM) for calculating v_{12}. On the other hand, we

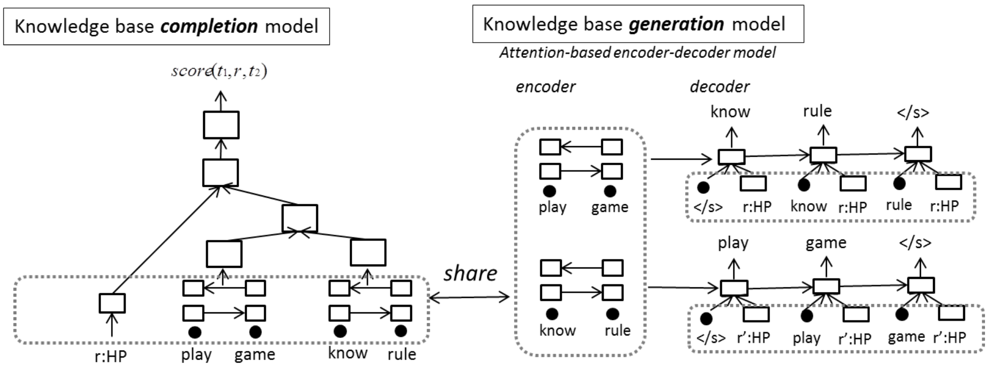

Figure 1: Architecture of proposed method. The CKB completion model estimates the score of $\langle t_1 =$ "play game"$, r =$ "HasPrerequisite (HP)"$, t_2 =$ "know rule"$\rangle$, and the CKB generation model generates t_2 from $\langle t_1, r\rangle$ and t_1 from $\langle t_2, r'\rangle$. r':HP denotes the reverse direction of "HasPrerequisite".

formulate the phrase embedding by using attention pooling of LSTM and a bilinear function.

$$\boldsymbol{h}_j^i = \mathrm{BiLSTM}(x_j^i, \boldsymbol{h}_{j-1}^i)(i = 1, 2) \tag{2}$$

$$v_i = \sum_{j=1}^{J} \frac{\exp(e_j)}{\sum_{k=1}^{J} \exp(e_k)} \boldsymbol{h}_j^i \tag{3}$$

$$e_k = u^\top \tanh(W \boldsymbol{h}_k^i) \tag{4}$$

$$v_{12} = \mathrm{Bilinear}(v_1, v_2) \tag{5}$$

$$v_{in} = \mathrm{concat}(v_{12}, v_r) \tag{6}$$

where J is the word length of phrase t_i, u is a linear transformation vector for calculating the attention vector, x_j^i and $\boldsymbol{h}_j^i$ are the j th word embedding and hidden state of the LSTM for phrase t_i, and v_r is the relation embedding. Note that we calculated v_{12} for DNN AVG and DNN LSTM by concatenating v_1 and v_2. We used batch normalization (Ioffe and Szegedy, 2015) for v_{in} before passing through the next layer.

3.2 CKB Generation Model

We use an attentional encoder-decoder model to generate phrase knowledge. Here, we expected that the quality of the phrase representation would be increased by sharing the BiLSTM and embeddings between the CKB completion and CKB generation models.

For constructing the encoder-decoder model, we use relation information in addition to word sequences. Let $\boldsymbol{X} = (x_1, x_2, ..., x_J)$ be the input word sequences and $\boldsymbol{Y} = (y_1, y_2, ..., y_T)$ be the output word sequences. The conditional generation probability of $\boldsymbol{Y}$ is as follows:

$$p(\boldsymbol{Y}|\boldsymbol{X}, \theta) = \prod_{t=1}^{T} p(y_t|y_{<t}, \boldsymbol{c}_t, r) \tag{7}$$

$$p(y_t|y_{<t}, \boldsymbol{c}_t, r) = g(y_{t-1}, \boldsymbol{s}_t, \boldsymbol{c}_t, r) \tag{8}$$

$$\boldsymbol{s}_t = \mathrm{LSTM}(\mathrm{concat}(v_{y_{t-1}}, v_r), \boldsymbol{s}_{t-1}) \tag{9}$$

where θ is a set of model parameters, $\boldsymbol{s}_t$ is a hidden state of the decoder, and $\boldsymbol{c}_t$ is a context vector of input sequences that is weighted by the attention probability and calculated as

$$\boldsymbol{h}_j = \mathrm{BiLSTM}(x_j, \boldsymbol{h}_{j-1}) \tag{10}$$

$$\boldsymbol{c}_t = \sum_{j=1}^{J} \frac{\exp(e_t)}{\sum_{k=1}^{J} \exp(e_k)} \boldsymbol{h}_j \tag{11}$$

$$e_k = v^\top \tanh(W_a \boldsymbol{s}_t + W_e \boldsymbol{h}_k) \tag{12}$$

Here, the BiLSTM, which is the encoder of the CKB generation model, is shared with that of the CKB completion model described in equation (2). As shown in equation (9), we use relation embedding $\boldsymbol{v}_r$ as additional input information. There are several related studies on incorporating additional label information in a decoder (Li et al., 2016a). Although the previous work used additional labels mainly for representing individuality or style information, we use this idea to represent relation information. We also use the technique of tying word vectors and word classifiers (Inan et al., 2016). The encoder BiLSTM is a single-layer bidirectional LSTM, and the decoder LSTM is a single-layer LSTM.

We use a triple $\langle t_1, r, t_2\rangle$ for training the encoder-decoder model. We train our models to be dual directional. In the forward direction, the model predicts t_2 with the input $\langle t_1, r\rangle$, and in the backward direction, it predicts t_1 with the input $\langle t_2, r\rangle$. Here, since the relation r has a direction, we introduce a new relation r' for each r to train dual-directional CKB generation in one model. In the reverse direction, we replace the relation label r with r'; namely, the output is t_1, and the input is $\langle t_2, r'\rangle$. Therefore, in our CKB generation model, the vocabulary size of the relation is twice that of the original relation set.

4 Training

Loss Function We use the following loss function for training: $L(\theta) = L_{\mathrm{c}} + \lambda L_{\mathrm{g}}$, where θ is the set of model parameters, L_{c} is the loss function of our CKB completion model, and L_{g} is the loss function of our CKB generation model. We use binary cross entropy for L_{c}.

$$L_{\mathrm{c}}(\tau, l) = -\frac{1}{N}\sum_{n=1}^{N}\{l\log\sigma(score(\tau)) + (1-l)\log(1-\sigma(score(\tau)))\}, \tag{13}$$

where τ indicates the triple $\langle t_1, r, t_2\rangle$, l is a binary variable that is 1 if the triple is a positive example (true triple) and 0 if the triple is a negative example (false triple), which we will explain in the next subsection. σ is a sigmoid function. We formulate the loss function for the encoder-decoder (CKB generation) model by using the cross entropy:

$$L_{\mathrm{g}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T^{(n)}}\log p(y_t^{(n)}|y_{<t}^{(n)}, c_t^{(n)}, r^{(n)}), \tag{14}$$

where N is the sample size, $T^{(n)}$ is the number of words in the output phrase, c_t is the context vector of the input sequence, and r is the relation label.

Negative sampling We generate negative examples automatically for training the CKB completion model by using random sampling. Specifically, we create three negative examples $\tau_{neg1} = \langle t_1^{neg}, r, t_2\rangle$, $\tau_{neg2} = \langle t_1, r^{neg}, t_2\rangle$, and $\tau_{neg3} = \langle t_1, r, t_2^{neg}\rangle$ for the positive triple τ by replacing each component. Here, t_1^{neg} and t_2^{neg} are sampled in mini-batches, while r^{neg} is sampled in all relation sets.

Generating augmentation data using CKB generation model For training the CKB completion and generation model, we need a large amount of data that covers a wide range of commonsense knowledge. Since our CKB generation model can make new triples, we use it to make the augmentation data. We use the original training data as seed data and generate new triples on the basis of it. More specifically, given a training triple $\langle t_1, r, t_2\rangle$, we generate a new t_2^{gen} with the input $\langle t_1, r\rangle$ and new t_1^{gen} with the input $\langle t_2, r'\rangle$. This idea is inspired by a technique for improving NMT models (Sennrich et al., 2016). To filter out unreliable candidates, we use the CKB completion score as a threshold. We refer to the generated augmentation data as "auggen" in the experiment section.

	ConceptNet	Ja-KB
train	100,000	192,714
validation1	1,200	13,778
validation2	1,200	-
test	2,400	13,778
size of relation	34	7
size of vocabulary	21,471	18,119
average word length	2.02	3.96

Table 1: Summary of data

5 Experimental Setup

5.1 Data

For the experiments with English, we used the ConceptNet 100K data released by Li et al. (2016b)[1]. The original ConceptNet is a large-scale and multi-lingual CKB. However, the evaluation set, which was created from a subset of the whole ConceptNet, consists of data only in English and contains many short phrases including single words. In order to evaluate the robustness of CKB completion models in terms of the language and long phrases, we created a new open-domain Japanese commonsense knowledge dataset, Ja-KB. The statistics of these data are listed in Table 1. There are more relation labels in ConceptNet than in Ja-KB, because we limited the relation types, which often contain nouns and verbs, when creating the Ja-KB data. The relation set of Ja-KB is Causes, MotivatedBy, Subevent, HasPrerequisite, ObstructedBy, Antonym, and Synonym. The average length of phrases in Ja-KB is longer than in ConceptNet because of the data creation process. The details of our dataset are described below:

To create the Ja-KB data, we used crowdsourcing like in Open Mind Common Sense (OMCS) (Singh et al., 2002). Since data annotated by

[1] http://ttic.uchicago.edu/ kgimpel/commonsense.html

crowd workers is usually noisy, we performed a two-step data collection process to eliminate noisy data. In the data creating step, a crowd worker created triples $\langle t_1, r, t_2 \rangle$ from the provided keywords. The keywords consisted of combinations of nouns and verbs that frequently appeared in Web texts.

Each crowd worker created an arbitrary phrase t_1 (or t_2) by using the provided keywords and then selected a relation $r \in R$ and created a corresponding phrase t_2 (or t_1). In the evaluation step, three workers chose a suitable $r \in R$ when they were given $\langle t_1, t_2 \rangle$, which were created by another worker. Since a worker does not know which relation r the creator selected in the creation step, we can measure the reliability of the created knowledge from the overlap of the selected relations. We used triples for which three or more workers selected the same relation label r. In our preliminary study, we found that the accuracy of CKB completion is lower when using low-reliability data.

We randomly selected the test and validation data among the data for which all workers chose the same label. The remaining data were used as training data. For the training data, we added the same number of triples as the evaluator selected same label for considering data reliability. For example, if three evaluators selected the same label for a triple, we added the three triples. For the test and validation data, we randomly sampled negative examples, as described in Section 4, whose size was the same as the number of positive examples according to (Li et al., 2016b). The details are described in the Supplementary Material.

5.2 Model Configurations

We set the dimensions of the hidden layer of the shared BiLSTM to 200, the word and relation embeddings to 200, and the intermediate hidden layer of the completion model to 1000. We set the batch size to 100, dropout rate to 0.2, and weight decay to 0.00001. For optimization, we used SGD and set the initial learning rate to 1.0. We set the reduction of the learning rate to 0.5 and adjusted the learning rate. We set λ of the loss function to 1.0. fastText (Bojanowski et al., 2016) and Wikipedia text were used to train the initial word embeddings. When generating the augmentation data, we set the threshold score of CKB completion to 0.95 for the ConceptNet data and 0.8 for the Ja-KB data. The additional data amounted to about 200,000 triples.

5.3 Baseline Method

CKB completion As baselines, we used the DNN AVG and DNN LSTM models (Li et al., 2016b) that were described in Section 3.1. To assess the effectiveness of joint learning, we compared our CKB completion model only (proposed w/o CKB generation) and the joint model (proposed w/ CKB generation). Moreover, we evaluated the effectiveness of simply adding augmentation data, as described in Section 4 to the training data (+auggen). We used the accuracy of binary classification as the evaluation measure. The threshold was determined by using the validation1 data to maximize the accuracy of binary classification for each method, as in (Li et al., 2016b).

CKB generation We used a simple attentional encoder-decoder model that does not use relation information as a baseline (base). We compared the proposed model with and without joint learning (proposed and proposed w/o CKBC). We also evaluated the effectiveness of simply adding augmentation data as described in Section 4 to the training data (+auggen).

6 Results

6.1 CKB completion

Does joint learning method improve the accuracy of CKB completion? Table 2 shows the accuracy of the CKB completion model. The bottom two lines show the best performances reported in (Li et al., 2016b). The results indicate that our method improved the accuracy of CKB completion compared with the previous method. Our method achieved 0.945 accuracy on the validation2 data. This result is close to human accuracy (about 0.95). By comparing the results of the single model (proposed w/o CKB generation) and joint model (proposed w/ CKB generation), we can see that the joint model improved the accuracy for both ConceptNet and Ja-KB. This indicates that the loss function of CKB generation works as a good constraint for the CKB completion model.

Does data augmentation from CKB generation improve the accuracy of CKB completion? Table 2 shows that augmentation data slightly improved the accuracy of both the ConceptNet test data and Ja-KB test data.

	ConceptNet		Ja-KB
method	valid2	test	test
base (DNN AVG)	0.923	0.929	0.904
base (DNN LSTM)	0.927	0.936	0.901
proposed w/o CKBG	0.927	0.932	0.907
proposed w/ CKBG	**0.945**	0.947	0.910
proposed w/ CKBG (+auggen)	0.944	**0.954**	**0.912**
Li et al (Li et al., 2016b)	0.913	0.920	-
human (Li et al., 2016b)	0.950	-	-

Table 2: Results of CKB completion. CKBG denotes CKB generation.

	ConceptNet	Ja-KB
base(DNN AVG)	0.66	0.58
proposed	0.74	0.62
proposed (+auggen)	0.72	0.61

Table 3: Accuracy of binary classification for manually annotated triples

Human evaluation for assessing the quality of CKB completion Since negative examples were randomly selected from the whole test set in the experiments described above (Table 2), it was easy to distinguish some of them as positive and negative examples. To evaluate the ability of CKB completion in a more difficult setting, we eliminated obviously-false triples and performed manual annotation with the remaining triples. Then we conducted a binary classification experiment with these annotated triples. The details are described below:

First, we prepared triple candidates by using the ConceptNet and Ja-KB datasets. We replaced one of the phrases of the existing triple with a similar phrase, where the similarity was calculated by using the average of the word embeddings. We made 100 replacement triples per triple. Next, we scored the prepared triples by using our CKB completion model and randomly sampled 500 triples whose CKB completion scores were larger than a threshold. Then, ten annotators gave subjective evaluation scores to all 500 triples. In this evaluation, the annotators rated the degree of agreement with each statement (triple) on 0-4 rating scale (0 = strongly disagree, 4 = strongly agree), where the annotator interpreted each triple as a statement by using the relation explanation. For example, $\langle dog, HasA, tail \rangle$ means "a dog has a tail". Finally, we sampled the top 100 triples which had small variance from the 500 annotated data and labeled those having average scores of 3 or over with 1 (positive examples; 57% and 55% of the top 100 triples of CN and Ja-KB, respectively) and those having average scores lower than 3 with 0 (negative examples; 43% and 45%).

Table 3 indicates the binary classification accuracy for the 100 sampled triples. While the proposed method improved accuracy, the accuracy of +auggen was slightly lower than it. This indicates that we have to select the augmentation data and the thresholds more carefully to improve the accuracy of difficult examples. Moreover, the overall score is lower than the result of Table 2. This indicates there is room for improving the CKB completion accuracy for difficult examples. To distinguish more difficult examples and improve the accuracy of knowledge acquisition, we have to develop a better negative sampling strategy for training.

6.2 CKB generation

It is difficult to evaluate the quality of the CKB generation model directly, since there are many correct phrase candidates in addition to phrases that appear in the test data. For that reason, we evaluated our CKB generation model from different viewpoints.

Can our CKB generation model generate reasonable phrases? To see whether the top-n phrases generated from each query in the test set included the reference phrase that corresponds to the query, we calculated the recall of the reference phrases as follows:

$$recall = N_{match}/N_{reference}, \tag{15}$$

where N_{match} is the number of generated phrases that exactly match the reference phrases. Figure 2 shows the recall of the reference phrases for each CKB generation model. The results shown in the figure are averages over the test queries. Compared with the baseline system, our CKB generation model achieved higher recalls on both ConceptNet and Ja-KB. This indicates that considering relation information worked well.

The effectiveness of using augmentation data is also illustrated in Figure 2. For the Ja-KB data, recall improved as a result of adding augmentation data. Since the phrase length of the node in ConceptNet is shorter than in Ja-KB, it is easier to cover reference phrases for ConceptNet.

Can our CKB generation model generate new phrases? To evaluate the effectiveness of our

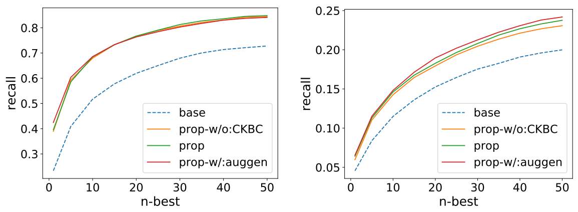

Figure 2: Recall of reference phrase (left: ConceptNet, right: Ja-KB)

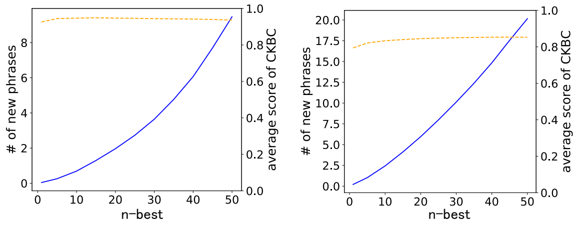

Figure 3: Average number of new phrases generated by CKBG (blue lines) and average score of CKBC of each triple (orange dashed lines). Left: ConceptNet, Right: Ja-KB.

generated triple $\langle t_1, r, t_2 \rangle$ input $\langle t_1, r \rangle$, output t_2	score-g	score-c
play game , HP , learn rule *	-3.57	0.985
play game , HP , have game **	-3.87	0.955
play game , HP , find someone to play **	-4.20	0.984
play game , HP , find friend *	-4.23	0.978
play game , HP , skill	-4.24	0.988
play game , UsedFor , entertainment	-2.21	0.950
play game , UsedFor , fun	-2.29	0.934
play game , UsedFor , have fun *	-2.64	0.920
play game , UsedFor , enjoyment	-3.13	0.976
play game , UsedFor , recreation *	-3.38	0.971

Table 4: Examples of phrases created using CKB generation model. The relation label "HP" represents HasPrerequisite. t_2 is the generated phrase, and the input is $\langle t_1, r \rangle$. * represents that the generated triple is new, and ** represents that the generated t_2 is new.

generation model at increasing the node size of a CKB, we determined whether our model could generate new phrases that are not included in the existing CKB. Figure 3 shows that the average number of such new phrases in the n-best outputs of our model that were generated from a query pair of a phrase and a relation in the test set of ConceptNet and Ja-KB. We can see from the figure that our model could make triples that contain new phrases by generating multiple phrases from a query pair. The figure also plots the average CKB completion score of each generated triple that contains new phrases; the results confirm that the generated triples had a high CKB completion score.

Generated examples Table 4 lists examples of phrases created by the generation model; score-g indicates the logarithmic probability of the generation model, and score-c indicates the score of the completion model. The upper row lists the top-five results with the input $\langle t_1, r \rangle$=(play game, HasPrerequisite). The lower row lists the top-five results with the input $\langle t_1, r \rangle$=(play game, UsedFor). These results indicate that our CKB generation model can generate reasonable candidates including new triples that reflected relation information. More examples are shown in the Supplementary Material.

How high is the quality of knowledge acquired with our CKB generation? We performed subjective evaluations of the quality of the triples generated with our model. First, we generated two types of query pairs: ones generated from ConceptNet (CN_gen) and ones generated from Wikipedia (Wiki_gen). In CN_gen, we used all phrase and relation pairs $\langle t, r \rangle$ appearing in the test data. In Wiki_gen, we used triples extracted by using the POS tag sequence pattern for each relation according to Li et al. (2016b) and scored each triple with CKB completion scores. Then, we used $\langle t, r \rangle$ pairs of 10000 triples that had higher scores than a threshold as the input query pairs.

Next, we generated a phrase t_{gen} from $\langle t, r \rangle$ and made new triples $\langle t, r, t_{gen} \rangle$ with our CKB generation model. We sorted the generated triples according to the CKB completion score and selected the top-100 new triples for CN_gen and Wiki_gen. The annotators assigned a (semantic) quality score and grammatical score to each triple. We used a 0-4 degree agreement score (described in 6.1) for evaluating triple quality and a 0-2 score (0. Doesn't make sense. 1. There are some grammatical errors. 2. There are no grammatical errors.) for the evaluation of grammatical quality. We recruited ten annotators who were native speakers of each language.

We show the results in Table 5. The quality score of each triple of CN_gen was quite high. The quality score of Wiki_gen was lower than that of CN_gen. Since Wikipedia has lots of specific information, it is difficult to extract an input query that is useful for making commonsense knowledge. This tendency is similar to the results reported in Li et al. (Li et al., 2016b). The grammatical score was high for both CN_gen and Wiki_gen.

	ConceptNet		Ja-KB	
method	semantic	grammar	semantic	grammar
CN_gen	3.452	1.651	3.466	1.996
Wiki_gen	2.685	1.749	2.415	1.849

Table 5: Subjective evaluation of CKB generation model

This indicates that our CKB generation model can generate phrases that have almost no grammatical errors for high confidence triples for top ranked triples.

7 Related Work

Knowledge base completion for entity-relation triples There are many studies that embed graph structures such as TransE, TransR, HolE, and STransE (Bordes et al., 2013; Lin et al., 2015; Nickel et al., 2016; Nguyen et al., 2016). Their methods aim to learn low-dimensional representations for entities and relationships by using topological features. Although these methods are widely used, they rely on the connectivity of the existing KB and are only suitable for predicting relationships between existing, well-connected entities (Shi and Weninger, 2018). Therefore, it is difficult to get good representations for new nodes that have no connections with existing nodes.

Several studies have added text information to the graph embeddings (Zhong et al., 2015; Wang and Li, 2016; Xiao et al., 2017). These studies aim to incorporate richer information in the graph embedding. They combine a graph embedding model and a text embedding model into one. The text information they use is the description or definition statement of each node. For example, they would use the description "Barack Obama is the 44th and current President of United States" for the node "Barack Obama" and make better quality embeddings. Although these methods effectively incorporate text information, they assume that the descriptions of entities can be easily acquired. For example, they use the originally aligned descriptions (e.g., DBpedia, Freebase) or descriptions acquired by using a simple entity linking method. Moreover, the methods use topological information, and they are not designed for on-the-fly knowledge base completion.

Knowledge base completion for commonsense triples In commonsense knowledge base completion, the nodes of the KB consist of arbitrary phrases (word sequences), and there are a huge number of unique nodes. In such case, the KB graph becomes very sparse, and consequently, there is almost no merit to considering the topological features of the KBs. Moreover, on-the-fly KBC is needed because we have to handle new nodes as input. It is thus more important to formulate phrase and relation embeddings that can robustly represent arbitrary phrases. There are a few studies on CKB completion models. In particular, Li et al. (2016b) and Socher et al. (2013) proposed a simple KBC model for CKB. The formulations of CKB completion in the two studies are the same, and we evaluated Li et al. (2016b)'s method as a baseline.

Open Information Extraction Open Information Extraction (OpenIE) aims to extract triple knowledge from raw text. It finds triples that have specific predefined relations by using lexical and syntactic patterns (Mintz et al., 2009; Fader et al., 2011). Several neural-network-based relation extraction methods have been proposed (Lin et al., 2016; Zhang et al., 2017). These models construct classifiers to estimate the relation between two arbitrary entities. OpenIE models are trained with sentence-level annotation data or distant supervision, while our model is trained with triples in a knowledge base. Since openIE can extract new triples from raw text, it can be used to make augmentation data for the CKB completion model.

Knowledge generation There are several studies on the knowledge generation task that use neural network models. For example, Hu et al. (2017) proposed an event prediction model that uses a sequence-to-sequence model. Prakash et al. (2016) and Li et al. (2017) proposed a paraphrase generation model. These studies targeted only specific relationships and did not explicitly incorporate relations into the generation model. Our CKB generation model explicitly incorporates relation information into the decoder and can model multiple relationships in one model.

8 Conclusion

We proposed a new CKB generation task and joint learning method of CKB completion and generation. Experimental results with two commonsense datasets demonstrated that our model has two strengths: it improves the coverage of the knowledge bases. While conventional completion tasks are limited to verifying given triples, our generative model can create new knowledge including new phrases that are not in the knowledge bases. Second, our completion model can improve the verification accuracy. Two characteristics of our completion model contribute to this improvement: (i) the model shares the hidden layers, word embedding, and relation embedding with the generation model to acquire good phrase and relation representations, and (ii) it can be trained with the augmentation data created by the generation model.

In this study, we did not utilize raw text information such as from Wikipedia during training except for pre-trained word embeddings. We would like to extend our method so that it can incorporate raw text information. Moreover, we would like to develop a method that effectively utilizes this commonsense knowledge for other NLP tasks that need commonsense reasoning.

References

Gabor Angeli and Christopher Manning. 2013. Philosophers are mortal: Inferring the truth of unseen facts. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 133–142.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250. ACM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on NIPS*, pages 2787–2795.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on EMNLP*, pages 1535–1545.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, AKBC '13, pages 25–30. ACM.

Linmei Hu, Juanzi Li, Liqiang Nie, Xiaoli Li, and Chao Shao. 2017. What happens next? future subevent prediction using contextual hierarchical LSTM. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3450–3456.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2016. Tying word vectors and word classifiers: A loss framework for language modeling. *CoRR*, abs/1611.01462.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 994–1003.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016b. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 1445–1455.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017. Paraphrase generation with deep reinforcement learning. *CoRR*, abs/1711.00279.

Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with heterogeneous knowledge for commonsense machine comprehension. In *Proceedings of the 2017 Conference on EMNLP*, pages 2032–2043.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2181–2187.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the ACL*.

Teng Long, Emmanuel Bengio, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2017. World knowledge for reading comprehension: Rare entity prediction with hierarchical lstms using external descriptions. In *Proceedings of the 2017 Conference on EMNLP*, pages 825–834.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th IJCNLP*, pages 1003–1011.

Dat Quoc Nguyen, Kairit Sirts, Lizhen Qu, and Mark Johnson. 2016. Stranse: a novel embedding model of entities and relationships in knowledge bases. In *Proceedings of the 2016 Conference of the NAACL: HLT*, pages 460–466.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1955–1961.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018. Mcscript: A novel dataset for assessing machine comprehension using script knowledge. *CoRR*, abs/1803.05223.

aaditya prakash, Sadid A. Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. In *Proceedings of COLING 2016*, pages 2923–2934.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the ACL*, pages 86–96.

Baoxu Shi and Tim Weninger. 2018. Open-world knowledge graph completion. *AAAI Conference on Artificial Intelligence*.

Push Singh, Thomas Lin, Erik T. Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1223–1237.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the 26th International Conference on NIPS*, pages 926–934.

Robert 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*, pages 4444–4451.

Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. 2018. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT*.

Zhigang Wang and Juanzi Li. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1293–1299.

Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu. 2017. SSP: semantic space projection for knowledge graph embedding with text descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3104–3110.

Bishan Yang and Tom Mitchell. 2017. Leveraging knowledge bases in lstms for improving machine reading. In *Proceedings of the 55th Annual Meeting of the ACL*, pages 1436–1446.

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

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on EMNLP*, pages 1730–1740.

Huaping Zhong, Jianwen Zhang, Zhen Wang, Hai Wan, and Zheng Chen. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of the 2015 Conference on EMNLP*, pages 267–272.