TransOMCS: From Linguistic Graphs to Commonsense Knowledge

Hongming Zhang[1], **Daniel Khashabi**[2], **Yangqiu Song**[1], **Dan Roth**[3]
[1]The Hong Kong University of Science and Technology
[2]Allen Institute for AI
[3]University of Pennsylvania
hzhangal@cse.ust.hk, danielk@allenai.org, yqsong@cse.ust.hk, danroth@seas.upenn.edu

Abstract

Commonsense knowledge acquisition is a key problem for artificial intelligence. Conventional methods of acquiring commonsense knowledge generally require laborious and costly human annotations, which are not feasible on a large scale. In this paper, we explore a practical way of mining commonsense knowledge from linguistic graphs, with the goal of transferring cheap knowledge obtained with linguistic patterns into expensive commonsense knowledge. The result is a conversion of ASER [Zhang *et al.*, 2020], a large-scale selectional preference knowledge resource, into TransOMCS, of the same representation as *ConceptNet* [Liu and Singh, 2004] but two orders of magnitude larger. Experimental results demonstrate the transferability of linguistic knowledge to commonsense knowledge and the effectiveness of the proposed approach in terms of quantity, novelty, and quality. TransOMCS is publicly available.[1]

1 Introduction

Commonsense knowledge is defined as the knowledge that people share but often omit when communicating with each other. In their seminal work, [Liu and Singh, 2004] defined commonsense knowledge as the knowledge "used in a technical sense to refer to the millions of basic facts and understandings possessed by most people." After around 20 years of development, ConceptNet 5.5 [Speer *et al.*, 2017], built based on the original ConceptNet [Liu and Singh, 2004], contains 21 million edges connecting over 8 million nodes. However, most of the knowledge assertions in ConceptNet 5.5 are still facts about entities integrated from other knowledge sources. The core of ConceptNet, which is inherited from the Open Mind CommonSense (OMCS) project [Liu and Singh, 2004], only contains 600K pieces of high-quality commonsense knowledge in the format of tuples, e.g., ('song', *UsedFor*, 'sing'). The gap between the small scale of existing commonsense knowledge resources and the broad demand of downstream applications motivates us to explore richer approaches to acquire commonsense knowledge from raw text, which is cheaper and more feasible.

Throughout the history of AI, many works were developed to extract various kinds of knowledge from raw texts with human-designed linguistic patterns. For example, OpenIE [Etzioni *et al.*, 2008] aims at identifying open relations between different entities (e.g., 'Paris'-*CapitalOf*-'France') and Hearst patterns [Hearst, 1992] are used to extract hyponyms (e.g., 'apple'-*IsA*-'fruit'). These patterns are often of high-precision. However, they typically suffer from brittleness and low coverage. To overcome the limitations of pattern-based methods, supervised commonsense knowledge acquisition methods were proposed. They either model the commonsense knowledge acquisition problem as a knowledge graph completion task to predict relations between concepts [Li *et al.*, 2016] or model it as a generation task by leveraging pre-trained language models to generate tail concepts [Bosselut *et al.*, 2019]. However, these approaches often are supervised with expensive annotations and are restricted to the distribution of the training data.

In this paper, we propose to discover new commonsense knowledge from linguistic graphs whose nodes are words and edges are linguistic relations, which is motivated by the observations [Resnik, 1997; Zhang *et al.*, 2019] that selectional preferences over linguistic relations can reflect humans' commonsense about their word choice in various contexts. Here, we use the linguistic graphs from ASER [Zhang *et al.*, 2020], which is extracted from raw text with dependency parser and explicit discourse connectives and provides 27 millions of eventualities extracted using dependency patterns and 10 millions of discourse relations as its core. Then, we develop an algorithm for discovering patterns from the overlap of existing commonsense and linguistic knowledge bases and use a commonsense knowledge ranking model to select the highest-quality extracted knowledge. As a result, we can build TransOMCS, a new commonsense knowledge graph.

In summary, our contributions are: (1) We formally define the task of mining commonsense knowledge from linguistic graphs and propose an approach to address it; (2) We construct a large-scale commonsense resource TransOMCS, with size two orders of magnitude larger than OMCS; (3) We conduct both intrinsic and extrinsic experiments to show the value of TransOMCS.

[1]https://github.com/HKUST-KnowComp/TransOMCS

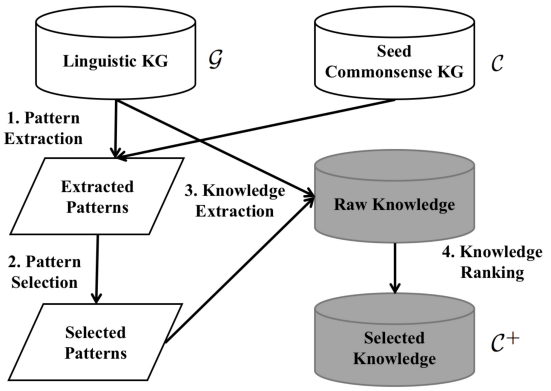

Figure 1: The overall framework.

2 Problem Definition

We start by defining the task of mining commonsense knowledge from linguistic graphs. Given a seed commonsense knowledge set $\mathcal{C}$ (which contains m tuples) and a linguistic graph set $\mathcal{G}$ (which contains n linguistic graphs G) with $m \ll n$. Each commonsense fact is in a tuple format $(h, r, t) \in \mathcal{C}$, where $r \in \mathcal{R}$, the set of human-defined commonsense relations (e.g., 'UsedFor', 'CapableOf', 'AtLocation', 'MotivatedByGoal'), and h and t are arbitrary phrases. Our objective is to infer a new commonsense knowledge set $\mathcal{C}^+$ (with m^+ pieces of commonsense knowledge) from $\mathcal{G}$ with the help of $\mathcal{C}$ such that $m^+ \gg m$.

3 Method

3.1 Overview

The proposed framework is shown in Figure 1. In the beginning, for each seed commonsense tuple $(h, r, t) \in \mathcal{C}$, we match and select supporting linguistic graphs that contain all the terms in h and t, and then extract the linguistic patterns for each commonsense relation with the matched commonsense tuple and linguistic graph pairs. Next, we use a pattern filtering module to select the highest quality patterns. Finally, we train a discriminative model to evaluate the quality of the extracted commonsense knowledge. The details are as follows.

3.2 Knowledge Resources

We first introduce the details about the selected commonsense and linguistic knowledge resources. For the seed commonsense knowledge, we use the English subset of ConceptNet 5.5 [Speer *et al.*, 2017]. Following conventional approaches [Saito *et al.*, 2018; Bosselut *et al.*, 2019], we consider only the relations covered by the original OMCS project [Liu and Singh, 2004], except those with vague meanings (i.e., 'RelatedTo') or those well-acquired by other knowledge resources (i.e., 'IsA'[2]). In total, 36,954 words, 149,908 concepts, and 207,407 tuples are contained in the selected dataset as $\mathcal{C}$. For the linguistic knowledge resource,

[2]The extraction of 'IsA' relations belongs to the task of hyponym detection and such knowledge has been well preserved by knowledge resources like Probase [Wu *et al.*, 2012].

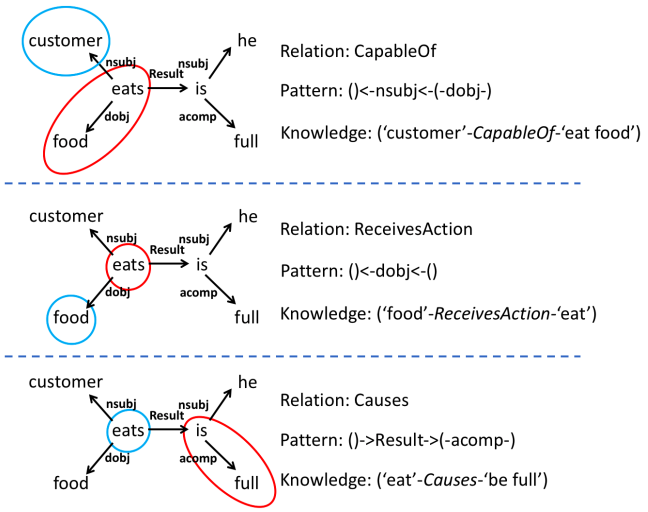

Figure 2: Example of linguistic graphs and extracted patterns for different commonsense relations, which are extracted with the matching of words in seed commonsense tuples and the graphs. Given a linguistic graph as the input, these patterns can be applied to extract OMCS-like commonsense knowledge. Extracted head and tail concepts are indicated with blue and red circles respectively.

we use the core subset of ASER [Zhang *et al.*, 2020] with 37.9 million linguistic graphs[3] to form $\mathcal{G}$.

3.3 Pattern Extraction

Given a matched pair of a commonsense tuple $(h, r, t) \in \mathcal{C}$ and a linguistic graph $G \in \mathcal{G}$, the goal of the pattern extraction module is to find a pattern over linguistic relations such that given r, we can accurately extract all the words in h and t from G. Here, we formally define each pattern P as follows:

Definition 1. *Each pattern P contains three components: head structure p_h, tail structure p_t, and internal structure p_i. p_h and p_t are the smallest linguistic sub-graph in G that can cover all the words in h and t, respectively. p_i is shortest path from p_h to p_t in G.*

First, we extract p_h and p_t from G to cover all the words in h and t. We take the head pattern as an example. For each word in h, we first find its position in G. To avoid any ambiguity, if we find more than one match in G, we discard the current pair and record no pattern. Then, with the position of the first word in h as the start node, we conduct a breadth-first search (BFS) algorithm over G to find a sub-structure of G that covers all and only the words in h. If the BFS algorithm finds such a sub-structure, we treat it as p_h, otherwise, we discard this example and return no pattern. We extract the tail pattern p_t with G and t in a similar way. After extracting p_h and p_t, we then collect the internal structure p_i, which is the shortest path from p_h to p_t over G. To do so, we collapse all the nodes and edges in p_h and p_t into single 'head' and 'tail' nodes, respectively. Then we use 'head' as the starting node to conduct a new BFS to find the shortest path from

[3]As both the internal structure of eventualities and external relations between eventualities could be converted to commonsense knowledge, we treat all the eventualities and eventuality pairs in ASER as the linguistic graphs.

node 'head' to 'tail'. We aggregate p_h, p_i, and p_t together to generate the overall pattern P. Examples of patterns are shown in Figure 2. For each commonsense relation r, we first collect all the commonsense tuples of relation r in $\mathcal{C}$ to form the subset $\mathcal{C}^r$. Then for each $(h, r, t) \in \mathcal{C}^r$, we go over the whole $\mathcal{G}$ to extract matched patterns with the aforementioned algorithm. The time complexity of the overall algorithm is $O(|\mathcal{C}| \cdot |\mathcal{G}| \cdot N^2)$, where $|\mathcal{C}|$ is the size of $\mathcal{C}$, $|\mathcal{G}|$ is the size of $\mathcal{G}$ and N is the maximum number of the nodes in G.

3.4 Pattern Selection and Knowledge Extraction

We evaluate the plausibility of pattern P regarding commonsense relation r as follows:

$$P(P|r) = \frac{F(P|r)}{\sum_{P' \in \mathcal{P}^r} F(P'|r)}, \tag{1}$$

where $F(P|r)$ is the scoring function we use to determine the quality of P regarding r and $\mathcal{P}^r$ is the set of patterns extracted for r. We design $F(P|r)$ as follows:

$$F(P|r) = C(P|r) \cdot L(P) \cdot U(P|r), \tag{2}$$

where $C(P|r)$ indicates counts of observing P regarding r, $L(P)$ indicates the length of P to encourage complex patterns, and $U(P|r) = \frac{C(P|r)/\sqrt{|\mathcal{C}^r|}}{\sum_{r' \in \mathcal{R}} C(P|r')/\sqrt{|\mathcal{C}^{r'}|}}$ is the uniqueness score of P regarding r. We select the patterns with the plausibility score higher than 0.05. On average, 2.8 high confident patterns are extracted for each relation. After extracting the matched patterns, we then apply the extracted patterns to the whole linguistic graph set $\mathcal{G}$ to extract commonsense knowledge. For each $G \in \mathcal{G}$ and each relation r, we go through all the extracted patterns $P \in \mathcal{P}^r$. If we can find a matched P in G, we will then extract the corresponding words of p_h and p_t in G as the head words and tail words, respectively. The extracted head and tail word pairs are then considered as a candidate knowledge, related via r.

3.5 Commonsense Knowledge Ranking

To minimize the influence of the pattern noise, we propose a knowledge ranking module to rank all extracted knowledge based on the confidence. To do so, we first use human annotators to label a tiny subset of extracted knowledge and then use it to train a classifier. We assign a confidence score to each extracted knowledge via the learned classifier.

Dataset Preparation

For each commonsense relation type, we randomly select 1,000 tuples[4] to annotate. For each tuple, five annotators from Amazon Mechanical Turk are asked to decide if they think the tuple is a plausible commonsense statement. If at least four annotators vote for plausibility, we label that tuple as a positive example. Otherwise, we label it as a negative example. Additionally, we include all the matched examples from OMCS as positive examples. In total, we obtain 25,923 annotations for 20 commonsense relations. Among these annotations, 18,221 are positive examples and 7,702 are negative examples. On average, each tuple has 12.7 supporting linguistic graphs. We randomly select 10% of the dataset as the test set and the rest as the training set.[5]

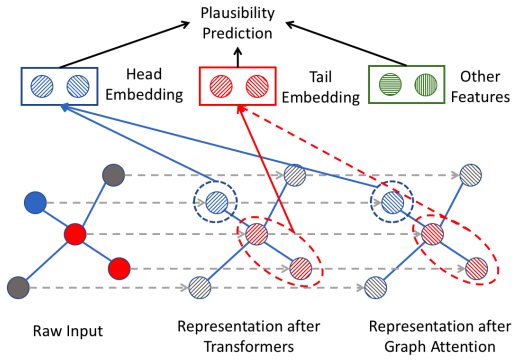

Figure 3: Demonstration of the model. The blue and red colors denote the head words and tail words, respectively. The gray and green colors and indicate the other words and features, respectively.

Task Definition

The goal of this module is to assign confidence scores to all extracted knowledge so that we can rank all the knowledge based on their quality. As for each extracted tuple, we may observe multiple supporting linguistic graphs. We model it as a multi-instance learning problem. Formally, we define the overall annotated knowledge set as $\mathcal{K}$. For each $k \in \mathcal{K}$, denote its supporting linguistic graph set as $\mathcal{G}^k$. We use $F(k|\mathcal{G}^k)$ to denote the plausibility scoring function of k given $\mathcal{G}^k$, and it can be defined as follows:

$$F(k|\mathcal{G}^k) = \frac{1}{|\mathcal{G}^k|} \cdot \sum_{g \in \mathcal{G}^k} f(k|g), \tag{3}$$

where $|\mathcal{G}^k|$ is the number of graphs in $\mathcal{G}^k$ and $f(k|g)$ is the plausibility score of k given g.

Model Details

The proposed model, as shown in Figure 3, contains three components: transformer, graph attention and the final plausibility prediction.

Transformer: We use the transformer module as the basic layer of our model. Formally, assuming g contains n words $w_1, w_2, ..., w_n$, we denote the representation of all words after the transformer module[6] as $\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_n$. In our model, we adopt the architecture of BERT [Devlin *et al.*, 2019] and their pre-trained parameters (BERT-base) as the initialization.

Graph Attention: Different from conventional text classification tasks, linguistic relations play a critical role in our task. Thus in our model, we adopt the graph attention module [Velickovic *et al.*, 2018] to encode the information of

[4]For relation 'DefinedAs' and 'LocatedNear', as our model only extracted 26 and 7 tuples respectively, we annotate all of them and exclude them when we compute the overall accuracy in Section 4.

[5]As the annotated dataset is slightly imbalanced, when we randomly select the test set, we make sure the number of positive and negative examples are equal.

[6]For technical details of the Transformer network, you may refer to the original paper [Vaswani *et al.*, 2017].

these linguistic relations. For each w in g, we denote its representation as $\hat{\mathbf{e}}$, which is defined as follows:

$$\hat{\mathbf{e}} = \sum_{\mathbf{e}' \in N(\mathbf{e})} a_{\mathbf{e},\mathbf{e}'} \cdot \mathbf{e}', \tag{4}$$

where $N(e)$ is the representation set of words that are connected to w and $a_{\mathbf{e},\mathbf{e}'}$ is the attention weight of $\mathbf{e}'$ regarding $\mathbf{e}$. Here, we define the attention weight as:

$$a_{\mathbf{e},\mathbf{e}'} = \frac{e^{\text{NN}_a([\mathbf{e},\mathbf{e}'])}}{\sum_{\bar{\mathbf{e}} \in N(\mathbf{e})} e^{\text{NN}_a([\mathbf{e},\bar{\mathbf{e}}])}}, \tag{5}$$

where $[.]$ indicates vector concatenation and NN_a is the dense neural network we use to predict the attention weight before the softmax module. After the graph attention module, the representation of words are then denoted as $\hat{\mathbf{e}_1}, \hat{\mathbf{e}_2}, ..., \hat{\mathbf{e}_n}$.

Plausibility Prediction: In the last part of our model, we first concatenate $\mathbf{e}$ and $\hat{\mathbf{e}}$ together for all the words and then create head embedding $\mathbf{o}_{head}$ and tail embedding $\mathbf{o}_{tail}$ using mean pooling over $[\mathbf{e}, \hat{\mathbf{e}}]$ of all words appear in the head or tail respectively. Besides embedding features, two important features, graph frequency $\mathbf{o}_{fre}$ (how many times this graph appears) and graph type $\mathbf{o}_{type}$ (whether this graph is node or edge), are also considered for the final plausibility prediction. We define plausibility prediction function $f(k|g)$ as:

$$f(k|g) = \text{NN}_p([\mathbf{o}_{head}, \mathbf{o}_{tail}, \mathbf{o}_{fre}, \mathbf{o}_{type}]), \tag{6}$$

where NN_p is a fully connected layer we use to predict the plausibility. We use the cross-entropy as the loss function and stochastic gradient descent as the optimization method.

Model Performance

We train the classifier for different relations separately as some relations are not exclusive with each other (e.g., 'RecievesAction' and 'UsedFor') and test our model on our collected data. We compare our model with random guess and directly using BERT, which is identical to our model excluding the graph attention part, in Table 1. Experimental results show that both the Transformer and the graph attention modules make significant contributions to the prediction quality of the extracted knowledge.

	Random	BERT	Proposed model
Accuracy	50%	70.91%	73.23%

Table 1: Performance of different plausibility prediction models.

4 Intrinsic Evaluation

We compare our approach with recently developed commonsense knowledge acquisition methods COMET [Bosselut *et al.*, 2019] and LAMA [Petroni *et al.*, 2019].

4.1 Evaluation Metrics

Quantity: We first measure *quantity* of the algorithms. Specifically, we are interested in two metrics: the number of acquired commonsense knowledge tuples and the number of unique words.

Novelty: For measuring *novelty* of the outputs, we follow [Bosselut *et al.*, 2019] and report the proportion of novel tuples and concepts, denoted as Novel_t and Novel_c, respectively. Here 'novel', similar to the definition in [Bosselut *et al.*, 2019], means that one cannot find the whole tuple or concept in the training/development data via string match. For COMET and LAMA, as the head concept is given as input and thus only the tail concept is generated by the models, we only select the tail concepts to calculate the concept novelty.

Quality: We use Amazon Mechanical Turk to evaluate the *quality* of the acquired commonsense tuples. For each model, we randomly select 100 tuples from the overall set to test the quality. For each tuple, five workers are invited to annotate whether this tuple is plausible or not. If at least four annotators label the tuple as plausible, we then consider it as a plausible tuple. Additionally, to investigate the quality of tuples with novel concepts, for each model, we also report the performance of all sampled novel tuples. As we use the accuracy (# valid tuples/ # all tuples) to evaluate the quality, we denote the overall quality of tuples with novel concepts and the whole tuple set as ACC_n and ACC_o, respectively.

4.2 Baseline Methods

1. **COMET:** Proposed by [Bosselut *et al.*, 2019], COMET leverages the pre-trained language model GPT [Radford *et al.*, 2018] to learn from annotated resources to generate commonsense knowledge.
2. **LAMA:** Proposed by [Petroni *et al.*, 2019], LAMA leverages BERT [Devlin *et al.*, 2019] to acquire commonsense knowledge. Unlike COMET, LAMA is unsupervised.

We denote our method as **TransOMCS**, indicating that we transfer knowledge from linguistic patterns to OMCS-style commonsense knowledge.

4.3 Implementation Details

For both COMET and LAMA, we include two variants in our experiments. The first one follows COMET's paper and uses concept/relation pairs among the most confident subset of OMCS as input. Due to the small size of this subset (only 1.2K positive examples), even if we use the 10-beam search decoding, we can only generate 12K tuples. To overcome this limitation and test whether these models can be used for generating large-scale commonsense knowledge, we consider a different evaluation setting where we randomly select 24K concepts from the concepts extracted by our approach and then randomly pair the selected concepts with commonsense relations as the input. We refer to the first and modified settings with $_{Original}$ and $_{Extended}$ subscripts, respectively.

4.4 Result Analysis

The summary of model evaluations is listed in Table 2. Compared with all baseline methods in their original settings, TransOMCS outperforms them in quantity. Even the smallest subset (top 1%) of TransOMCS outperforms their largest generation strategy (10-beam search) by ten times. TransOMCS also outperforms COMET in terms of novelty, especially the percentage of novel concepts. The reason behind this is that COMET is a pure machine-learning approach and it learns

Model	# Vocab	# Tuple	Novel$_t$	Novel$_c$	ACC$_n$	ACC$_o$
COMET$_{Original}$ (Greedy decoding)	715	1,200	33.96%	5.27%	58%	90%
COMET$_{Original}$ (Beam search - 10 beams)	2,232	12,000	64.95%	27.15%	35%	44%
COMET$_{Extended}$ (Greedy decoding)	3,912	24,000	**99.98%**	55.56%	34%	47%
COMET$_{Extended}$ (Beam search - 10 beams)	8,108	240,000	**99.98%**	78.59%	23%	27%
LAMA$_{Original}$ (Top 1)	328	1,200	-	-	-	49%
LAMA$_{Original}$ (Top 10)	1,649	12,000	-	-	-	20%
LAMA$_{Extended}$ (Top 1)	1,443	24,000	-	-	-	29%
LAMA$_{Extended}$ (Top 10)	5,465	240,000	-	-	-	10%
TransOMCS$_{Original}$ (no ranking)	33,238	533,449	99.53%	89.20%	72%	74%
TransOMCS (Top 1%)	37,517	184,816	95.71%	75.65%	**86%**	87%
TransOMCS (Top 10%)	56,411	1,848,160	99.55%	92.17%	69%	74%
TransOMCS (Top 30%)	68,438	5,544,482	99.83%	95.22%	67%	69%
TransOMCS (Top 50%)	83,823	9,240,803	99.89%	96.32%	60%	62%
TransOMCS (no ranking)	**100,659**	**18,481,607**	99.94%	**98.30%**	54%	56%
OMCS in ConceptNet 5.0	36,954	207,427	-	-	-	**92%**

Table 2: Main Results. For our proposed method, we present both the full TransOMCS and several subsets based on the plausibility ranking scores. TransOMCS$_{Original}$ means the subset, whose head/relation appear in the test set as used by COMET$_{Original}$ and LAMA$_{Original}$. As LAMA is unsupervised, the Novelty metric is not applicable. For our model, for the fair comparison, we exclude all annotated knowledge.

	Annotation	Scale	Novelty	Quality
OMCS	full	small	high	high
COMET	supervision	large	low	depends
LAMA	no	large	high	low
TransOMCS	supervision	large	high	high

Table 3: Comparison of different commonsense resources or acquisition methods. The quality of COMET depends on the scale of its generated knowledge (high quality on the test set, but low quality on the large scale generation.) COMET and TransOMCS require a small number of annotations as supervision.

to generate the tail concept in the training set. It is possible that the stronger their models are, the more likely they overfit the training data, the fewer novel concepts are generated. As for the quality, when the training data is similar to the test data, COMET provides the best quality. For example, in the original setting, the greedy decoding strategy achieves 90% overall accuracy. As a comparison, the quality of LAMA is less satisfying, which is mainly because LAMA is fully unsupervised. Besides that, in the extended setting, the quality of both models drops, which is mainly because the randomly generated pairs could include more rare words. Compared with them, TransOMCS (top 1%) provides the comparable quality with COMET, but with a larger scale. When the quantity is comparable, TransOMCS outperforms both of them in terms of quality. We summarize the comparisons in Table 3.

4.5 Case Study

We show case studies in Figure 4 to further analyze different acquisition methods.

COMET: COMET is the only one that can generate long concepts. At the same time, it also suffers from generating meaningless words. For example, two of the generated concepts end with 'be', which is confusing and meaningless. This is probably because COMET only generates lemmas rather than normal words. Besides that, COMET could overfit the training data, even though the ten outputs are not exactly the same, four of them mean the same thing ('kill others').

LAMA: The most significant advantage of LAMA is that it is unsupervised. However, it has two major drawbacks: (1) it can only generate one-token concepts, which is far away from enough for commonsense knowledge; (2) the quality of LAMA is not as good as the other two methods.

TransOMCS: Compared with COMET, TransOMCS can generate more novel commonsense knowledge. For example, our model knows that 'human' is capable of having cells and creating life. Besides that, unlike LAMA, TransOMCS can generate multi-token concepts. At the same time, our approach also has two limitations: (1) it cannot extract long concepts, which are difficult to find an exact pattern match; (2) as the extraction process strictly follows the pattern matching, it could extract semantic incomplete knowledge. For example, 'human' is capable of 'have'. The original linguistic graph should be "human have something", but as the pattern is '()<-nsubj<-()', the object is missing.

5 Extrinsic Evaluation

We conduct experiments on two downstream tasks: *commonsense reading-comprehension* [Ostermann *et al.*, 2018] and *dialogue generation* [Li *et al.*, 2017]. We select [Wang *et al.*, 2018] and [Luong *et al.*, 2015; Zhang *et al.*, 2020] as our base models for the comprehension and the generation task, respectively. For the fair comparison, we keep all the model architecture and parameters the same across different trials. We report the results with the common metric of each dataset: *accuracy* on the comprehension task and the *BLEU* score [Papineni *et al.*, 2002] on the dialog generation task.

The overall result is shown in Table 4. For the reading comprehension task, adding the top 1% of TransOMCS contributes 0.37 overall accuracy, compared to 0.21 contribution of OMCS. Meanwhile, the contributions of COMET and LAMA are minor for this task. For the dialogue generation task, TransOMCS shows remarkable improvement in the quality of generated responses. At the same time, adding other knowledge resources to OMCS does not provide any meaningful improvements to the performance. The reason

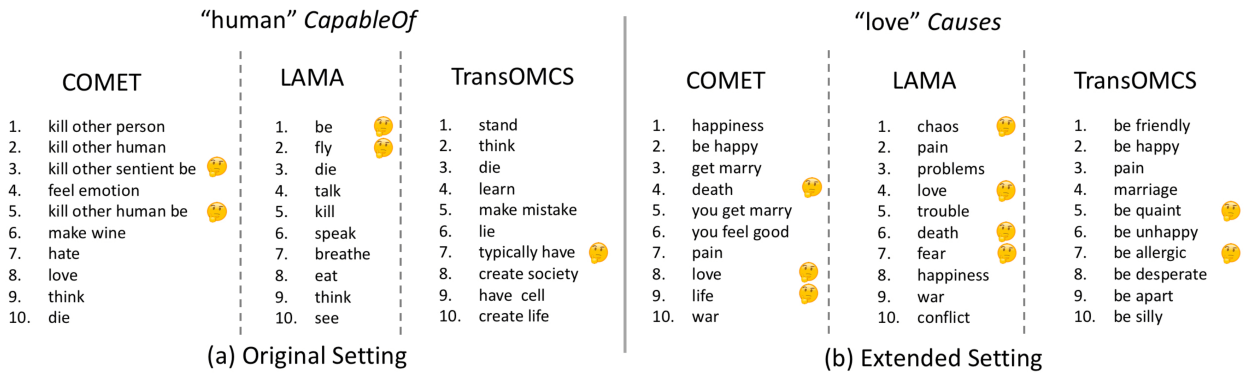

Figure 4: Case study of three knowledge acquisition methods. Two head-relation pairs are selected from the original and extended settings respectively. We indicate low quality tuples with confusing face emojis. For each model, we select the top ten most confident results.

Commonsense Knowledge Resource	Reading Comprehension		Dialog Generation	
	Accuracy (%)	Δ (%)	BLEU	Δ
Base model (no external knowledge resource)	82.90	-	0.54	-
+OMCS	83.11	+0.21	0.72	+0.18
+COMET$_{Original}$ (Greedy decoding)	83.12	+0.22	0.61	+0.07
+COMET$_{Extended}$ (Beam search - 10 beams)	83.03	+0.13	0.68	+0.14
+LAMA$_{Original}$ (Top 1)	83.13	+0.23	0.56	+0.02
+LAMA$_{Extended}$ (Top 10)	83.17	+0.27	0.57	+0.03
+TransOMCS (Top 1%)	**83.27**	**+0.37**	**1.85**	**+1.31**

Table 4: Experimental results on downstream tasks. For COMET and LAMA, we report the performance of their most accurate and largest setting. For TransOMCS, as current models cannot handle large-scale data, we only report the performance of the most confident 1%. All the numbers are computed based on the average of four different random seeds rather than the best seed as reported in the original paper.

behind this could be that COMET and LAMA provide limited high quality novel commonsense knowledge. For example, the original OMCS on average contributes 1.46 supporting tuples[7] and TransOMCS contribute another 3.36 supporting tuples. As a comparison, COMET$_{Original}$, COMET$_{Extended}$, LAMA$_{Original}$, and LAMA$_{Extended}$ only provide 0.01, 0.07, 0.49, 0.01 additional tuples respectively. This experiment result shows that TransOMCS has more novel knowledge.

6 Related Work

Commonsense knowledge covers a variety of knowledge types like knowledge about typical location or causes of events in OMCS [Liu and Singh, 2004], events causes, effects, and temporal properties in ATOMIC [Sap *et al.*, 2019] and [Zhou *et al.*, 2020], and physical attributes of objects [Elazar *et al.*, 2019]. As an important knowledge resource for AI systems, commonsense knowledge has been shown helpful in many downstream tasks such as question answering [Lin *et al.*, 2019] and reading comprehension [Wang *et al.*, 2018]. Conventional commonsense resources (e.g., OMCS and ATOMIC) are often constructed via crowdsourcing aimed to provide high-quality results; although such expensive processes restrict their scale. Recently, several attempts [Li *et al.*, 2016; Davison *et al.*, 2019] have been made to enrich the existing commonsense knowledge by learning to predict new relations between concepts. However, these approaches cannot generate new nodes (concepts). To address this problem, several models were proposed to generate commonsense tuples in either supervised [Bosselut *et al.*, 2019] or unsupervised [Petroni *et al.*, 2019] fashions. Different from them, this work shows that linguistic patterns can be extracted for commonsense relations and consequently be used for producing valuable commonsense knowledge.

[7]Here by supporting tuple, we mean that the head and tail concept appear in the post and response respectively.

7 Conclusion

In this paper, we exhibit the transferability from linguistic knowledge (dependency and discourse knowledge) to commonsense knowledge (i.e., the OMCS-style knowledge) by showing that a large amount of high-quality commonsense knowledge can be extracted from linguistic graphs. We formally define the task of mining commonsense knowledge from linguistic graphs and present TransOMCS, a commonsense knowledge resource extracted from linguistic graphs into the format of the OMCS subset of ConceptNet, but two orders of magnitude larger than the original OMCS. While TransOMCS is noisier than OMCS, it can still make significant contributions to downstream tasks due to its larger coverage, as evident by the extrinsic experiments.

Acknowledgements

This paper was supported by the Early Career Scheme (ECS, No. 26206717) from the Research Grants Council in HK and the Tencent AI Lab Rhino-Bird Focused Research Program, with partial support from DARPA grant FA8750-19-2-1004.

References

[Bosselut *et al.*, 2019] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: commonsense transformers for automatic knowledge graph construction. In *Proceedings of ACL 2019*, pages 4762–4779, 2019.

[Davison *et al.*, 2019] Joe Davison, Joshua Feldman, and Alexander M. Rush. Commonsense knowledge mining from pretrained models. In *Proceedings of EMNLP-IJCNLP 2019*, pages 1173–1178, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186, 2019.

[Elazar *et al.*, 2019] Yanai Elazar, Abhijit Mahabal, Deepak Ramachandran, Tania Bedrax-Weiss, and Dan Roth. How large are lions? inducing distributions over quantitative attributes. In *Proceedings of ACL 2019*, pages 3973–3983, 2019.

[Etzioni *et al.*, 2008] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12):68–74, 2008.

[Hearst, 1992] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of COLING 1992*, 1992.

[Li *et al.*, 2016] Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. Commonsense knowledge base completion. In *Proceedings of ACL 2016*, pages 1445–1455, 2016.

[Li *et al.*, 2017] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of IJCNLP 2017*, pages 986–995, 2017.

[Lin *et al.*, 2019] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of EMNLP-IJCNLP*, pages 2829–2839, 2019.

[Liu and Singh, 2004] Hugo Liu and Push Singh. Conceptnet: a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, 2004.

[Luong *et al.*, 2015] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP 2015*, pages 1412–1421, 2015.

[Ostermann *et al.*, 2018] Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. Semeval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of SemEval 2018*, pages 747–757, 2018.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2012*, pages 311–318, 2002.

[Petroni *et al.*, 2019] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. Language models as knowledge bases? In *Proceedings of EMNLP-IJCNLP 2019*, pages 2463–2473, 2019.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[Resnik, 1997] Philip Resnik. Selectional preference and sense disambiguation. *Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.

[Saito *et al.*, 2018] Itsumi Saito, Kyosuke Nishida, Hisako Asano, and Junji Tomita. Commonsense knowledge base completion and generation. In *Proceedings of CoNLL 2018*, pages 141–150, 2018.

[Sap *et al.*, 2019] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. Atomic: an atlas of machine commonsense for if-then reasoning. In *Proceedings of AAAI 2019*, pages 3027–3035, 2019.

[Speer *et al.*, 2017] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of AAAI 2017*, pages 4444–4451, 2017.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, **L**ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of NIPS 2017*, pages 5998–6008, 2017.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of ICLR 2018*, 2018.

[Wang *et al.*, 2018] Liang Wang, Meng Sun, Wei Zhao, Kewei Shen, and Jingming Liu. Yuanfudao at semeval-2018 task 11: Three-way attention and relational knowledge for commonsense machine comprehension. *arXiv preprint arXiv:1803.00191*, 2018.

[Wu *et al.*, 2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of SIGMOD 2012*, pages 481–492. ACM, 2012.

[Zhang *et al.*, 2019] Hongming Zhang, Hantian Ding, and Yangqiu Song. SP-10K: A large-scale evaluation set for selectional preference acquisition. In *Proceedings of ACL 2019*, pages 722–731, 2019.

[Zhang *et al.*, 2020] Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. ASER: A large-scale eventuality knowledge graph. In *Proceedings of WWW 2020*, pages 201–211, 2020.

[Zhou *et al.*, 2020] Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. Temporal common sense acquisition with minimal supervision. In *Proceedings of ACL 2020*, 2020.