Benchmarking Commonsense Knowledge Base Population with an Effective Evaluation Dataset

Tianqing Fang1,*, Weiqi Wang1,*, Sehyun Choi1, Shibo Hao2, Hongming Zhang1, Yangqiu Song1, & Bin He3

1Department of Computer Science and Engineering, HKUST
2School of Electronics Engineering and Computer Science, Peking University
3Huawei Noah’s Ark Lab

tfangaa@cse.ust.hk, {wwangbw, schoiaj}@connect.ust.hk, haoshibo@pku.edu.cn, {hzhangal, yqsong}@cse.ust.hk, hebin.nlp@huawei.com

Abstract

Reasoning over commonsense knowledge bases (CSKBs) whose elements are in the form of free-text is an important yet hard task in NLP. While CSKB completion only fills the missing links within the domain of the CSKB, CSKB population is alternatively proposed with the goal of reasoning unseen assertions from external resources. In this task, CSKBs are grounded to a large-scale eventuality (activity, state, and event) graph to discriminate whether novel triples from the eventuality graph are plausible or not. However, existing evaluations on the population task are either not accurate (automatic evaluation with randomly sampled negative examples) or of small scale (human annotation). In this paper, we benchmark the CSKB population task with a new large-scale dataset by first aligning four popular CSKBs, and then presenting a high-quality human-annotated evaluation set to probe neural models’ commonsense reasoning ability. We also propose a novel inductive commonsense reasoning model that reasons over graphs. Experimental results show that generalizing commonsense reasoning on unseen assertions is inherently a hard task. Models achieving high accuracy during training perform poorly on the evaluation set, with a large gap between human performance. Codes and data are available at https://github.com/HKUST-KnowComp/CSKB-Population.

1 Introduction

Commonsense reasoning is one of the core problems in the field of artificial intelligence. Throughout the development in computational commonsense, commonsense knowledge bases (CSKB) (Speer et al., 2017; Sap et al., 2019) are constructed to enhance models’ reasoning ability. As human-annotated CSKBs are far from complete due to the scale of crowd-sourcing, reasoning tasks such as CSKB completion (Li et al., 2016; Malaviya et al., 2020; Moghimifar et al., 2021) and population (Fang et al., 2021) are proposed to enrich the missing facts. The CSKB completion task is defined based on the setting of predicting missing links within the CSKB. On the other hand, the population task grounds commonsense knowledge in CSKBs to large-scale automatically extracted candidates, and requires models to determine whether a candidate triple, \((head, relation, tail)\), is plausible or not, based on the information from both the CSKB and the large number of candidates which essentially form a large-scale graph structure. An illustration of the difference between completion and population is shown in Figure 1.

There are two advantages of the population task. First, the population can not only add links but also nodes to an existing CSKB, while completion can only add links. The populated CSKB can also help reduce the selection bias problem (Heckman, 1979) from which most machine learning models would
suffer, and will benefit a lot of downstream applications such as commonsense generation (Bosse-lut et al., 2019). Second, commonsense knowledge is usually implicit knowledge that requires multiple-hop reasoning, while current CSKBs are lacking such complex graph structures. For example, in ATOMIC (Sap et al., 2019), a human-annotated if-then commonsense knowledge base among daily events and (mental) states, the average hops between matched heads and tails in ASER, an automatically extracted knowledge base among activities, states, and events based on discourse relationships, is 2.4 (Zhang et al., 2021). Evidence in Section 4.5 (Table 3) also shows similar results for other CSKBs. However, reasoning solely on existing CSKBs can be viewed as a simple triple classification task without considering complex graph structure (as shown in Table 3, the graphs in CSKBs are much sparser). The population task, which provides a richer graph structure, can explicitly leverage the large-scale corpus to perform commonsense reasoning over multiple hops on the graph.

However, there are two major limitations for the evaluation of the CSKB population task. First, automatic evaluation metrics, which are based on distinguishing ground truth annotations from automatically sampled negative examples (either a random head or a random tail), are not accurate enough. Instead of directly treating the random samples as negative, solid human annotations are needed to provide hard labels for commonsense triples. Second, the human evaluation in the original paper of CSKB population (Fang et al., 2021) cannot be generally used for benchmarking. They first populate the CSKB and then asked human annotators to annotate a small subset to check whether the populated results are accurate or not. A better benchmark should be based on random samples from all candidates and the scale should be large enough to cover diverse events and states.

To effectively and accurately evaluate CSKB population, in this paper, we benchmark CSKB population by firstly proposing a comprehensive dataset aligning four popular CSKBs and a large-scale automatically extracted knowledge graph, and then providing a large-scale human-annotated evaluation set. Four event-centered CSKBs that cover daily events, namely ConceptNet (Speer et al., 2017) (the event-related relations are selected), ATOMIC (Sap et al., 2019), ATOMIC20 (Hwang et al., 2020), and GLUCOSE (Mostafazadeh et al., 2020), are used to constitute the commonsense relations. We align the CSKBs together into the same format and ground them to a large-scale eventuality (including activity, state, and event) knowledge graph, ASER (Zhang et al., 2020, 2021). Then, instead of annotating every possible node pair in the graph, which takes an infeasible \( O(|V|^2) \) amount of annotation, we sample a large subset of candidate edges grounded in ASER to annotate. In total, 31.7K high-quality triples are annotated as the development set and test set.

To evaluate the commonsense reasoning ability of machine learning models based on our benchmark data, we first propose some models that learn to perform CSKB population inductively over the knowledge graph. Then we conduct extensive evaluations and analysis of the results to demonstrate that CSKB population is a hard task where models perform poorly on our evaluation set far below human performance.

We summarize the contributions of the paper as follow: (1) We provide a novel benchmark for CSKB population over new assertions that cover four human-annotated CSKBs, with a large-scale human-annotated evaluation set. (2) We propose a novel inductive commonsense reasoning model that incorporates both semantics and graph structure. (3) We conduct extensive experiments and evaluations on how different models, commonsense resources for training, and graph structures may influence the commonsense reasoning results.

2 Related Works

2.1 Commonsense Knowledge Bases

Since the proposal of Cyc (Lenat, 1995) and ConceptNet (Liu and Singh, 2004; Speer et al., 2017), a growing number of large-scale human-annotated CSKBs are developed (Sap et al., 2019; Bisk et al., 2020; Sakaguchi et al., 2020; Mostafazadeh et al., 2020; Forbes et al., 2020; Lourie et al., 2020; Hwang et al., 2020; Ilievski et al., 2020). While ConceptNet mainly depicts the commonsense relations between entities and only small portion of events, recent important CSKBs have been more devoted to event-centric commonsense knowledge. For example, ATOMIC (Sap et al., 2019) defines 9 social interaction relations and ~880K triples are annotated. ATOMIC20 (Hwang et al., 2020) further unifies the relations with ConceptNet, together with several new relations, to form a larger CSKB con-
taining 16 event-related relations. Another CSKB is GLUCOSE (Mostafazadeh et al., 2020), which extracts sentences from ROC Stories and defines 10 commonsense dimensions to explores the causes and effects given the base event. In this paper, we select ConceptNet, ATOMIC, ATOMIC2020, and GLUCOSE to align them together because they are all event-centric and relatively more normalized compared to other CSKBs like SocialChemistry101 (Forbes et al., 2020).

### 2.2 Knowledge Base Completion and Population

Knowledge Base (KB) completion is well studied using knowledge base embedding learned from triples (Bordes et al., 2013; Yang et al., 2015; Sun et al., 2019) and graph neural networks with a scoring function decoder (Shang et al., 2019). Pretrained language models are also applied on such completion task (Yao et al., 2019; Wang et al., 2020b) where information of knowledge triples is translated into the input to BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019). Knowledge base population (Ji and Grishman, 2011) typically includes entity linking (Shen et al., 2014) and slot filling (Surdeanu and Ji, 2014) for conventional KBs, and many relation extraction approaches have been proposed (Roth and Yih, 2002; Chan and Roth, 2010; Mintz et al., 2009; Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Lin et al., 2016; Zeng et al., 2017). Universal schema and matrix factorization can also be used to learn latent features of databases and perform population (Riedel et al., 2013; Verga et al., 2016; Toutanova et al., 2015; McCallum et al., 2017).

Besides completion tasks on conventional entity-centric KBs like Freebase (Bollacker et al., 2008), completion tasks on CSKBs are also studied on ConceptNet and ATOMIC. Bi-linear models are used to conduct triple classification on ConceptNet (Li et al., 2016; Saito et al., 2018). Besides, knowledge base embedding models plus BERT-based graph densifier (Malaviya et al., 2020; Wang et al., 2020a) are used to perform link prediction. For CSKB population, BERT plus GraphSAGE (Hamilton et al., 2017) is designed to learn a reasoning model on unseen assertions (Fang et al., 2021).

Commonsense knowledge generation, such as COMET (Bosselut et al., 2019) and LAMA (Petroni et al., 2019), is essentially a CSKB population problem. However, it requires the known heads and relations to acquire more tails so it does not fit our evaluation. Recently, various prompts are proposed to change the predicate lexicalization (Jiang et al., 2020; Shin et al., 2020; Zhong et al., 2021) but still how to obtain more legitimate heads for probing remains unclear. Our work can benefit them by obtaining more training examples, mining more commonsense prompts, as well as getting more potential heads for the generation.

### 3 Task Definition

Denote the source CSKB about events as $C = \{(h, r, t)|h \in H, r \in R, t \in T\}$, where $H$, $R$, and $T$ are the set of the commonsense heads, relations, and tails. Suppose we have another much larger eventuality (including activity, state, and event) knowledge graph extracted from texts, denoted as $G = (V, E)$, where $V$ is the set of all vertices and $E$ is the set of edges. $G'$ is the graph acquired by aligning $C$ and $G$ into the same format. The goal of CSKB population is to learn a scoring function given a candidate triple $(h, r, t)$, where plausible commonsense triples should be scored higher. The training of CSKB population can inherit the setting of triple classification, where ground truth examples are from the CSKB $C$ and negative triples are randomly sampled. In the evaluation phase, the model is required to score the triples from $G$ that are not included in $C$ and be compared with human-annotated labels.

### 4 Dataset Preparation

#### 4.1 Selection of CSKBs

As we aim at exploring commonsense relations among general events, we summarize several criteria for selecting CSKBs. First, the CSKB should be well symbolically structured to be generalizable. While the nodes in CSKB can inevitably be free-text to represent more diverse semantics,
Table 2: Overlaps between eventuality graphs and commonsense knowledge graphs. We report the proportion of (h, r, t) triples where both the head and tail can be found in the eventuality graph.

| # Triples | ATOMIC (No clause) | ATOMIC\textsubscript{20} (4 relations) | ConceptNet (Event-centered) | GLUCOSE | # Eventuality |
|-----------|---------------------|---------------------------------|-----------------------------|---------|--------------|
| Knowlywood | 449,056 | 124,935 | 10,159 | 117,828 | - |
| ASER     | 2.63%  | 2.87%   | 16.50% | 2.96%  | 929,546 |
|          | 61.95% | 38.50%  | 44.94% | 84.57% | 52,940,258 |

4.2 Alignment of CSKBs

To effectively align the four CSKBs, we propose best-effort rules to align the formats for both nodes and edges. First, for the nodes in each CSKB, we normalize the person-centric subjects and objects as “PersonX”, “PersonY”, and “PersonZ”, etc, according to the order of their occurrence, and the object-centric subjects and objects as “SomethingA” and “SomethingB”. Second, to reduce the semantic overlaps of different relations, we aggregate all commonsense relations to the relations defined in ATOMIC\textsubscript{20}, as it is comprehensive enough to cover the relations in other resources like GLUCOSE, with some simple alignment in Table 1.

ConceptNet. We select Causes and HasSubEvent from ConceptNet to constitute the event-related relations. As heads and tails in ConceptNet don’t contain subjects, we add a “PersonX” in front of the original heads and tails to make them complete eventualities.

ATOMIC\textsubscript{20}. In ATOMIC and ATOMIC\textsubscript{20}, heads are structured events with “PersonX” as subjects, while tails are human-written free-text where subjects tend to be missing. We add “PersonX” for the tails without subjects under agent-driven relations, the relations that aim to investigate causes or effects on “PersonX” himself, and add “PersonY” for the tails missing subjects under theme-driven relations, the relations that investigate commonsense causes or effects on other people like “PersonY”.

GLUCOSE. For GLUCOSE, we leverage the parsed and structured version in this study. We replace the personal pronouns “SomeoneA” and “SomeoneB” with “PersonX” and “PersonY” respectively. For other object-centric placeholders like “Something”, we keep them unchanged. The relations in GLUCOSE are then converted to ATOMIC relations according to the conversion rule in the original paper (Mostafazadeh et al., 2020). Moreover, gWant, gReact, and gEffect are the new relations for the triples in GLUCOSE where the subjects are object-centric. The prefix “g” stands for general, to be distinguished from “x” (for PersonX) and “o” (for PersonY).

4.3 Selection of the Eventuality KG

Taking scale and the diversity of relationships in the KG into account, we select two automatically extracted eventuality knowledge graphs as candidates for the population task, Knowlywood (Tandon et al., 2015) and ASER (Zhang et al., 2020).
They both have complex graph structures that are suitable for multiple-hop reasoning. We first check how much commonsense knowledge is included in those eventuality graphs to see if it’s possible to ground a large proportion of commonsense knowledge triples on the graphs. Best-effort alignment rules are designed to align the formats of CSKBs and eventuality KGs. For Knowlywood, as the patterns are mostly simple verb-object pairs, we leverage the v-o pairs directly and add a subject in front of the pairs. For ASER, we aggregate the raw personal pronouns like he and she to normalized “PersonX”. As ASER adopts more complicated patterns of defining eventualities, a more detailed pre-process of the alignment between ASER and CSKBs will be illustrated in Section 4.4. We report the proportion of triples in every CSKB whose head and tail can both be matched to the eventuality graph in Table 2. ASER covers a significantly larger proportion of head-tail pairs in the four CSKBs than Knowlywood. The reason behind is that on the one hand ASER is of much larger scale, and on the other hand ASER contains eventualities with more complicated structures like s-v-o-p-o (s for subject, v for verb, o for object, and p for preposition), compared with the fact that Knowlywood mostly covers s-v pairs only. In the end, we select ASER as the eventuality graph for population.

### 4.4 Pre-process of the Eventuality Graph

We introduce the normalization process of ASER, which converts its knowledge among everyday eventualities into normalized form to be aligned with the CSKBs as discussed in Section 4.2. Each eventuality in ASER has a subject. We consider singular personal pronouns, i.e., “I”, “you”, “he”, “she”, “someone”, “guy”, “man”, “woman”, “somebody”, and replace the concrete personal pronouns in ASER with normalized formats such as “PersonX” and “PersonY”. Specifically, for an original ASER edge where both the head and tail share the same person-centric subject, we replace the subject with “PersonX” and the subsequent personal pronouns in the two eventualities with “PersonY” and “PersonZ” according to the order of the occurrence if exists. For the two neighboring eventualities where the subjects are different person-centric pronouns, we replace one with “PersonX” and the other with “PersonY”. In addition, to preserve the complex graph structure in ASER, for all the converted edges, we duplicate them by replacing the “PersonX” in it with “PersonY”, and “PersonY” with “PersonX”, to preserve the sub-structure in ASER as much as possible. An illustration of the converting process is shown in Figure 2. The normalized version of ASER is denoted as ASER_{norm}.

### 4.5 The Aligned Graph $G^c$

With the pre-process in Section 4.2 and 4.4, we can successfully align the CSKBs and ASER together in the same format. To demonstrate ASER’s
coverage on the knowledge in CSKBs, we present the proportion of heads, tails, and edges that can be found in the ASER norm via exact string match in Table 3. For edges, we report the proportion of edges where the corresponding heads and tails can be connected by a path in ASER. We also report the average shortest path length in ASER for those matched edges from the CSKB in the #hops column, showing that ASER can entail such commonsense knowledge within several hops of path reasoning, which builds the foundation of commonsense reasoning on ASER. In addition, the average degree in $G^c$ and $C$ for heads and tails from each CSKB is also presented in the table. The total number of triples for each relation in the CSKBs is presented in Table 4. There are 18 commonsense relations in total for CSKBs and 15 relations in ASER. More detailed descriptions and examples of the unification are presented in the Appendix (Table 11, 12, and 14).

### 4.6 Evaluation Set Preparation

For the ground truth commonsense triples from the CSKBs, we split them into train, development, and test set with the proportion 8:1:1. Negative examples are sampled by selecting a random head and a random tail from the aligned $G^c$ such that the ratio of negative and ground truth triples is 1:1. To form a diverse evaluation set, we sample 20K triples from the original automatically constructed test set (denoted as “Original Test Set”), 20K from the edges in ASER where heads come from CSKBs and tails are from ASER (denoted as “CSKB head + ASER tail”), and 20K triples in ASER where both heads and tails come from ASER (denoted as “ASER edges”). The detailed methods of selecting candidate triples for annotation is listed in the Appendix B.2. The distribution of different relations in this evaluation set is the same as in the original test set. The sampled evaluation set is then annotated to acquire ground labels.

### 5 Human Annotation

#### 5.1 Setups

The human annotation is carried out on Amazon Mechanical Turk. Workers are provided with sentences in the form of natural language translated from knowledge triples (e.g., for $\text{xReact}$, an $(h, r, t)$ triple is translated to “If $h$, then, PersonX feels $t$”). Additionally, following Hwang et al. (2020), annotators are asked to rate each triple in a four-point Likert scale: Always/Often, Sometimes/Likely, Farfetched/Never, and Invalid. Triples receiving the former two labels will be treated as Plausible or otherwise Implausible. Each HIT (task) includes 10 triples with the same relation type, and each sentence is labeled by 5 workers. We take the majority vote among 5 votes as the final result for each triple. To avoid ambiguity and control the quality, we finalize the dataset by selecting triples where workers reach an agreement on at least 4 votes.

#### 5.2 Quality Control

For strict quality control, we carry out two rounds of qualification tests to select workers and provide a special training round. First, workers satisfying the following requirements are invited to participate in our qualification tests: 1) at least 1K HITs approved, and 2) at least 95% approval rate. Second, a qualification question set including both straightforward and tricky questions is created by experts, who are authors of this paper and have a clear understanding of this task. 760 triples sampled from the original dataset are annotated by the experts. Each worker needs to answer a HIT containing 10 questions from the qualification set and their answers are compared with the expert annotation. Annotators who correctly answer at least 8 out of 10 questions are selected in the second round. 671 workers participated in the qualification test, among which 141 (21.01%) workers are selected as our main round annotators. To further enhance

| Relation | ATOMIC($G^c$) | ConceptNet | GLUCOSE |
|----------|---------------|-------------|---------|
| oEffect  | 21,497        | 0           | 7,595   |
| xEffect  | 61,021        | 0           | 30,596  |
| gEffect  | 0             | 0           | 8,577   |
| oWant    | 35,477        | 0           | 1,766   |
| xWant    | 83,776        | 0           | 11,439  |
| gWant    | 0             | 0           | 5,138   |
| oReact   | 21,110        | 0           | 3,077   |
| xReact   | 50,535        | 0           | 13,203  |
| gReact   | 0             | 0           | 2,683   |
| xAttr    | 89,337        | 0           | 7,664   |
| xNeed    | 61,487        | 0           | 0       |
| xIntent  | 29,034        | 0           | 8,292   |
| isBefore | 18,798        | 0           | 0       |
| isAfter  | 18,600        | 0           | 0       |
| HinderedBy | 87,580     | 0           | 0       |
| xReason  | 189           | 0           | 0       |
| Causes   | 0             | 42          | 26,746  |
| HasSubEvent | 0            | 9,934      | 0       |

Total 578,252 10,165 126,776

Table 4: Relation distribution statistics for different CSKBs. Due to the filter in Section 4.1, the statistics are different from the original papers.
Table 5: Statistics of the annotated evaluation set. # triples indicates the number of triples in the dataset, % Plausible indicates the proportion of plausible triples after majority voting, and % Novel Nodes is the proportion of nodes that do not appear in the training CSKBs. We also report the scale of the un-annotated training set (including random negative examples) for reference.

| Relation | #Eval. | #Train |
|----------|--------|--------|
| xWant    | 2,605  | 152,634|
| oWant    | 999    | 59,688 |
| gWant    | 207    | 8,093  |
| xEffect  | 2,757  | 144,799|
| oEffect  | 667    | 46,555 |
| gEffect  | 287    | 13,529 |
| xReact   | 2,999  | 100,853|
| oReact   | 921    | 38,581 |
| gReact   | 164    | 4,169  |
| xAttr    | 2,561  | 152,949|
| xIntent  | 1,152  | 59,138 |
| xNeed    | 1,100,362|
| Causes   | 1,152  | 59,688 |
| xReason  | 16     | 320    |
| isBefore | 879    | 27,784 |
| isAfter  | 1,152  | 27,414 |
| HinderedBy | 4821 | 127,320|
| HasSubEvent | 459 | 16,410 |

Table 6: Number of triples of each relation in the Eval. (dev+test) and Train set.

6 Experiments

In this section, we introduce the baselines and our proposed model KG-BERT\textsuperscript{SAGE} for the CSKB population task, as well as the experimental setups.

6.1 Model

The objective of a population model is to determine the plausibility of an \((h, r, t)\) triple, where nodes can frequently be out of the domain of the training set. In this sense, transductive methods based on knowledge base embeddings (Malaviya et al., 2020) are not studied here. We present several ways of encoding triples in an inductive manner.

BERT. The embeddings of \(h, r, t\) are encoded as the embeddings of the [CLS] tokens after feeding them separately as sentences to BERT. For example, the relation xReact is encoded as the BERT embedding of “[CLS] xReact [SEP]”. The embeddings are then concatenated as the final representation of the triple, \([s_h, s_r, s_t]\).

BERT\textsuperscript{SAGE}. The idea of BERT\textsuperscript{SAGE} (Fang et al., 2021) is to leverage the neighbor information of nodes through a graph neural network layer for their final embedding. For \(h\), denote its BERT embedding as \(s_h\), then the final embedding of \(h\) is \(e_h = [s_h, \sum_{v \in \mathcal{N}(h)} s_{v}/|\mathcal{N}(h)|]\), where \(\mathcal{N}(h)\) is the neighbor function that returns the neighbors of \(h\) from \(\mathcal{G}\). The final representation of the triple is then \([e_h, s_r, e_t]\).

KG-BERT. KG-BERT\textsuperscript{(a)} (Yao et al., 2019) encodes a triple by concatenating the elements in \((h, r, t)\) into a single sentence and encode it with BERT. Specifically, the input is the string concatenation of [CLS], \(h\), [SEP], \(r\), [SEP], \(t\), and [SEP].

KG-BERT\textsuperscript{SAGE}. As KG-BERT doesn’t take into account graph structures directly, we propose to add an additional graph SAmpling and AGgregation layer (Hamilton et al., 2017) to better learn the graph structures. Specifically, denoting the embedding of the \((h, r, t)\) triple by KG-BERT as KG-BERT\textsuperscript{(h, r, t)}, the model of KG-BERT\textsuperscript{SAGE} is the concatenation of KG-BERT\textsuperscript{(h, r, t)}, \(\sum_{(r', v) \in \mathcal{N}(h)}\) KG-BERT\textsuperscript{(h, r', v)}/|\(\mathcal{N}(h)\)|, and \(\sum_{(r', v) \in \mathcal{N}(t)}\) KG-BERT\textsuperscript{(v, r', t)}/|\(\mathcal{N}(t)\)|. Here, \(\mathcal{N}(h)\) returns the neighboring edges of node \(h\).

More details about the models and experimental details are listed in the Appendix Section C.

6.2 Setup

We train the population model using a triple classification task, where ground truth triples come
from the original CSKB, and the negative examples are randomly sampled from the aligned graph \(G^r\). The model needs to discriminate whether an \((h, r, t)\) triple in the human-annotated evaluation set is plausible or not. For evaluation, we use the AUC score as the evaluation metric, as this com-
mmonsense reasoning task is essentially a ranking task that is expected to rank plausible assertions higher than those farfetched assertions.

We use BERT\textsubscript{base} from the Transformer\textsuperscript{1} library, and use learning rate \(5 \times 10^{-5}\) and batch size 32 for all models. The statistics of each relation is shown in Table 6. We select the best models individually for each relation based on the corresponding development set. Besides AUC scores for each relation, we also report the AUC score for all relations by the weighted sum of the break-down scores, weighted by the proportion of test examples of the relation. This is reasonable as AUC essentially represents the probability that a positive example will be ranked higher than a negative example.

### 6.3 Main Results

The main experimental results are shown in Table 7. KG-BERT\textsc{Sage} performs the best among all, as it both encodes an \((h, r, t)\) as a whole and takes full advantage of neighboring information in the graph. Moreover, all models are significantly lower than human performance with a relatively large margin.

\[ \text{Table 7: Experimental results on CSKB population. We report the AUC (×100) here for each relation. The improvement under “all” is statistically significant using Randomization Test (Cohen, 1995), with } p < 0.05. \]

| Relation | \(\text{BERT}\) | \(\text{BERTSAGE}\) | \(\text{KG-BERT}\) | \(\text{KG-BERT\textsc{Sage}}\) |
|----------|----------------|-----------------|----------------|---------------------|
| \(G^r\)  | 64.8 68.6 62.0 | 69.8 68.6 68.0 | 63.2 69.8 69.0 | 66.0 68.8 68.2 |
| \(G^c\)  | 54.0 69.8 64.0 | 57.3 61.0 57.0 | 53.3 64.9 64.6 | 56.4 64.9 64.6 |
| \(G^p\)  | 70.8 70.8 62.3 | 70.8 70.8 62.3 | 73.8 73.8 63.8 | 75.8 75.8 63.8 |
| \(G^t\)  | 64.8 64.8 64.8 | 64.8 64.8 64.8 | 64.8 64.8 64.8 | 64.8 64.8 64.8 |
| \(G^h\)  | 58.8 68.6 58.8 | 58.8 68.6 58.8 | 58.8 68.6 58.8 | 58.8 68.6 58.8 |
| \(G^v\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^w\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^a\)  | 63.2 63.2 63.2 | 63.2 63.2 63.2 | 63.2 63.2 63.2 | 63.2 63.2 63.2 |
| \(G^b\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^d\)  | 64.8 64.8 64.8 | 64.8 64.8 64.8 | 64.8 64.8 64.8 | 64.8 64.8 64.8 |
| \(G^e\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^f\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^g\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^h\)  | 63.2 63.2 63.2 | 63.2 63.2 63.2 | 63.2 63.2 63.2 | 63.2 63.2 63.2 |
| \(G^i\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^j\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^k\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^l\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^m\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^n\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^o\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^p\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^q\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^r\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^s\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^t\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^u\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^v\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^w\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^x\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |
| \(G^y\)  | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 | 61.0 61.0 61.0 |
| \(G^z\)  | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 | 62.0 62.0 62.0 |

\[ \text{Table 8: Effects of different training sets.} \]

[1]https://transformer.huggingface.co/
Table 9: AUC scores grouped by the types of the evaluation sets defined in 4.6. The latter two groups are harder for neural models to distinguish.

| Model          | Original Test Set | CSKB head + ASER tail | ASER edges |
|----------------|-------------------|------------------------|------------|
| BERT           | 65.0              | 47.9                   | 44.6       |
| BERTSAGE       | 67.2              | 49.4                   | 46.2       |
| KG-BERT        | 77.8              | 55.2                   | 50.3       |
| KG-BERTSAGE    | 78.2              | 57.5                   | 52.3       |

Table 10: Examples of error predictions made by KG-BERTSAGE, where the head and tail are semantically related while not conformed to the designated commonsense relation.

| Head          | Relation | Tail                   | Label | Pred. |
|---------------|----------|------------------------|-------|-------|
| PersonX go to nurse | **xEff** | PersonX use to get headache | 0     | 1     |
| PersonX have a quiz     | **Cause** | PersonX have pen       | 0     | 1     |
| PersonX be strong         | **Want**  | PersonX like PersonX   | 0     | 1     |
| PersonX feel a pain       | **Intent**| PersonX finger have be chop off | 0     | 1     |

Moreover, by taking a brief inspection of the test set, we found that errors occur when encountering triples that are not logically sound but semantically related. Some examples are presented in Table 10. For the triple (PersonX go to nurse, xEffect, PersonX use to get headache), the head event and tail event are highly related. However, the fact that someone gets a headache should be the reason instead of the result of going to the nurse. More similar errors are presented in the rest of the table. These failures may be because when using BERT-based models the training may not be well performed for the logical relations or discourse but still recognizing the semantic relatedness patterns.

8 Conclusion

In this paper, we benchmark the CSKB population task by proposing a dataset by aligning four popular CSKBs and an eventuality graph ASER, and provide a high-quality human-annotated evaluation set to test models’ reasoning ability. We also propose KG-BERTSAGE to both incorporate the semantic of knowledge triples and the subgraph structure to conduct reasoning, which achieves the best performance among other counterparts. Experimental results also show that the task of reasoning unseen triples outside of the domain of CSKB is a hard task where current models are far away from human performance, which brings challenges to the community for future research.

Acknowledgement

The authors of this paper were supported by the NSFC Fund (U20B2053) from the NSFC of China, the RIF (R6020-19 and R6021-20) and the GRF (16211520) from RGC of Hong Kong, the MHKJFS (MHP/001/19) from ITC of Hong Kong, with special thanks to the Gift Fund from Huawei Noah’s Ark Lab.
References

Yonatan Bisk, Rowan Zellers, Ronan LeBras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: reasoning about physical commonsense in natural language. In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020, pages 7432–7439. AAAI Press.

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, pages 1247–1250.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In Neural Information Processing Systems (NIPS), pages 1–9.

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

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In Proceedings of the 23rd International Conference on Computational Linguistics, pages 152–160.

Paul R Cohen. 1995. Empirical methods for artificial intelligence, volume 139. MIT press Cambridge, MA.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.

Tianqing Fang, Hongming Zhang, Weiqi Wang, Yangqiu Song, and Bin He. 2021. Discos: Bridging the gap between discourse knowledge and commonsense knowledge. In Proceedings of the Web Conference 2021, pages 2648–2659.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. Psychological bulletin, 76(5):378.

Maxwell Forbes, Jena D. Hwang, Vered Shwartz, Maarten Sap, and Yejin Choi. 2020. Social chemistry 101: Learning to reason about social and moral norms. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 653–670. Association for Computational Linguistics.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, pages 1025–1035.

James J Heckman. 1979. Sample selection bias as a specification error. Econometrica: Journal of the econometric society, pages 153–161.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pages 541–550.

Jena D Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2020. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. arXiv preprint arXiv:2010.05953.

Filip Ilievski, Pedro Szekely, and Bin Zhang. 2020. Cskg: The commonsense knowledge graph. arXiv preprint arXiv:2012.11490.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, pages 1148–1158.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. Trans. Assoc. Comput. Linguistics, 8:423–438.

Douglas B Lenat. 1995. Cyc: A large-scale investment in knowledge infrastructure. Communications of the ACM, 38(11):33–38.

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

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 Association for Computational Linguistics (Volume 1: Long Papers), pages 2124–2133.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.
Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. Multi-instance multi-label learning for relation extraction. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pages 455–465.

Niket Tandon, Gerard De Melo, Abir De, and Gerhard Weikum. 2015. Knowlywood: Mining activity knowledge from hollywood narratives. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, pages 223–232.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In Proceedings of the 2015 conference on empirical methods in natural language processing, pages 1499–1509.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 886–896.

Bin Wang, Guangtao Wang, Jing Huang, Jiaxuan You, Jure Leskovec, and C-C Jay Kuo. 2020a. Inductive learning on commonsense knowledge graph completion. arXiv preprint arXiv:2009.09263.

Bo Wang, Tao Shen, Guodong Long, Tianyi Zhou, and Yi Chang. 2020b. Semantic triple encoder for fast open-set link prediction. arXiv preprint arXiv:2004.14781.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In 3rd International Conference on Learning Representations, ICLR 2015. San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kgbert: Bert for knowledge graph completion. arXiv preprint arXiv:1909.03193.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2017. Incorporating relation paths in neural relation extraction. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1768–1777.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020. Aser: A large-scale eventuality knowledge graph. In Proceedings of The Web Conference 2020, pages 201–211.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: learning vs. learning to recall. arXiv preprint arXiv: 2104.05240.
A  Additional Details of Commonsense Relations

During human annotation, we translate the symbolic knowledge triples into human language for annotators to better understand the questions. A \((h, r, t)\) triple where \(h\), \(r\), and \(t\) are the head, relation, and tail, is translated to \(if \ h, \ then \ [\text{Description}]\), \(t\). Here, the description placeholder \([\text{Description}]\) comes from rules in Table 11, which is modified from Hwang et al. (2020). These descriptions can also be regarded as definitions of those commonsense relations.

Moreover, the definitions of the discourse relations in ASER are presented in Table 12. We also present the statistics of relation distribution for ASER\textsubscript{norm}. in Table 13.

B  Additional Details of Pre-processing

B.1  Examples of Format Unification

Table 14 demonstrates several examples for unifying the formats of different resources. In ConceptNet and Knowlywood, the nodes are mostly verb or verb-object phrases, and we add a subject “PersonX” in front of each node. For ATOMIC, the main modification part is the tails, where subjects tend to be missing. We treat agent-driven (relations investigating causes and effects on PersonX) and theme-driven (relations investigating causes and effects on PersonY) differently, and add PersonX or PersonY in front of the tails whose subjects are missing. For ASER, rules are used to discriminate PersonX and PersonY in a certain edge. Two examples for ASER and ATOMIC demonstrating the differences between PersonX and PersonY are provided in the table. For GLUCOSE, we simply replace SomeoneA with PersonX and SomeoneB with PersonY accordingly. Moreover, all the words are lemmatized using Stanford CoreNLP parser\(^2\) to normalized forms.

B.2  Selecting Candidate Triples from ASER

The evaluation set comes from three parts:

1. **Original Test Set:** The edges that are randomly sampled from the original automatically constructed test set, as illustrated in Section 4.6.

2. **CSKB head + ASER tail:** The edges are sampled from the edges in ASER where the heads come from the nodes in CSKBs and tails

| Relation     | Descriptions                              |
|--------------|-------------------------------------------|
| oEffect      | then, PersonY will                         |
| xEffect      | then, PersonX will                         |
| gEffect      | then, other people or things will          |
| oWant        | then, PersonY wants to                     |
| xWant        | then, PersonX wants to                     |
| gWant        | then, other people or things want to       |
| oReact       | then, PersonY feels                        |
| xReact       | then, PersonX feels                        |
| gReact       | then, other people or things feel          |
| xAttr        | PersonX is seen as                         |
| xHead        | but before, PersonX needed                 |
| xIntent      | because PersonX wanted                     |
| isBefore     | happens before                             |
| isAfter      | happens after                              |
| HinderedBy   | can be hindered by                         |
| xReason      | because                                    |
| Causes       | causes                                     |
| HasSubEvent  | includes the event/action                  |

Table 11: Descriptions of different commonsense relations, which are translation rules from knowledge triples \((h, r, t)\) to human language, \(\text{"if } h, \ then \ [\text{Description}], \ t\) (Hwang et al., 2020).

| Relation  | Descriptions                           |
|-----------|----------------------------------------|
| Precedence| \(h\) happens before \(t\)             |
| Succession| \(h\) happens after \(h\)              |
| Synchronous| \(h\) happens the same time as \(t\)  |
| Reason    | \(h\) happens because \(t\)            |
| Result    | \(h\) result in \(t\)                  |
| Condition | Only when \(h\) happens, \(h\) can happen |
| Contrast  | \(h\) and \(t\) share significant difference regarding some property |
| Concession| \(h\) and \(t\) are alternative situations of each other. |
| Alternative| \(h\) and \(t\) both happen             |
| Conjunction| \(h\) and \(t\) both happen             |
| Restatement| \(h\) restates \(t\)                 |
| Instantiation| \(t\) is a more detailed description of \(h\) |
| ChosenAlternative| \(h\) and \(t\) are alternative situations of each other, but the subject prefers \(h\) |
| Exception  | \(t\) is an exception of \(h\)           |
| CoOccurrence| \(h\) and \(t\) co-occur at the same sentence |

Table 12: Descriptions of discourse relations in ASER (Zhang et al., 2021).

from ASER. This corresponds to the settings in COMET (Bosselut et al., 2019) and DISCOS (Fang et al., 2021).

3. **ASER edges:** The edges are sampled from the whole ASER graph.

Instead of randomly sampling negative examples which may be easy to distinguish, we sample some candidate edges from ASER with some simple rules to fit the chronological order and syntactical patterns for each commonsense relation, thus providing a harder evaluation set for machines to concentrate more on commonsense. The discourse relations defined in ASER at Table 12 inherently represent some chronological order, which can be matched to each commonsense relation based on some alignment rules.

First, for each commonsense relation, we sample the edges in ASER with the same basic chronolog-
As BERT adopts sub-word encoding, the relations, despite being complicated symbols, can be split into several meaningful components for BERT to encode. For example, xReact will be split into “x” and “react”, which can demonstrates both the semantics of “x” (the relation is based on PersonX) and “react” (the reaction of the head event).

For KG-BERT, we encode a (h, r, t) triple by feeding the concatenation of the three elements into BERT. Specifically, “[CLS] w_1^h w_2^h · · · w_l^h [SEP] r [SEP] w_1^t w_2^t · · · w_m^t [SEP]” is fed into BERT and we regard the embedding of [CLS] as the final representation of the triple.

Denote the embedding of a (h, r, t) triple acquired by KG-BERT as KG-BERT(h, r, t). The function \( N(v) \) is defined as returning the incoming neighbor-relation pairs, which is \( \{(r, u)|(u, r, v) \in G\} \) (\( G \) is ASER in our case.) \( N(v) \) is defined as the function that returns the set \( \{(r, u)|(r, v, u) \in G\} \), which are neighboring edges. The model KG-BERT\_SAGE then encodes a (h, r, t) triple as:

\[
[\text{KG-BERT}(h, r, t), \\
\sum_{(r', v) \in N(h)} \text{KG-BERT}(h, r', v)/|N(h)|, \\
\sum_{(r', v) \in N(t)} \text{KG-BERT}(v, r', t)/|N(t)|]
\]

Moreover, as the average number of degrees for nodes in ASER is quite high, we follow the idea

| Relation          | number of edges |
|-------------------|-----------------|
| Precedence        | 4,957,481       |
| Succession        | 1,783,154       |
| Synchronous       | 8,317,572       |
| Reason            | 5,888,968       |
| Result            | 5,562,565       |
| Condition         | 8,109,020       |
| Contrast          | 23,208,195      |
| Concession        | 1,189,167       |
| Alternative       | 1,508,729       |
| Conjunction       | 37,802,734      |
| Restatement       | 159,667         |
| Instantiation     | 51,502          |
| ChosenAlternative | 33,840          |
| Exception         | 91,286          |
| Cq_Ccurrence      | 124,330,714     |

Table 13: Statistics of relations in ASER\_norm.
Table 14: Examples of format unification of CSKBs and eventuality graphs.

| Resource | Original Format | Aligned Format |
|----------|-----------------|----------------|
| ConceptNet | get exercise HasSubEvent ride bicycle | PersonX get exercise PersonX ride bicycle |
| ATOMIC | PersonX gets exercise xReact tired | PersonX get exercise PersonX be tired |
| GLUCOSE | SomeoneA gets exercise Dim 1 (xEffect) SomeoneA gets tired | PersonX get exercise PersonX be tired |
| Knowlywood | get exercise NextActivity take shower | PersonX get exercise PersonX take shower |
| ASER | he gets exercise Result he is tired | PersonX get exercise PersonX be tired |
| he visits her at work Precedence she is happy | PersonX visit PersonY at work PersonY is happy |

Table 15: Rules of selecting candidate triples. For a certain commonsense relation \( r_{cs} \) in the first column, \((head, r_{ASER}, tail)\) in ASER, where \( r_{ASER} \) belongs to the corresponding cell in the second column, can be selected as a candidate \((head, r_{cs}, tail)\) for annotation.

| Commonsense Relations | ASER Relations | Patterns |
|-----------------------|----------------|----------|
| Effect, Want          | Result, Precedence, Condition, Succession, Reason, Concession, Alternative, Synchronous, Restatement | - |
| isBefore, Causes      | Result, Precedence, Condition, Succession, Reason, Concession, Alternative, Synchronous, Restatement | s-be-a/o, s-v-be-a/o, s-v, spass-v |
| React                 | Result, Precedence, Condition, Succession, Reason, Concession, Alternative, Synchronous, Restatement | - |
| xIntent, xNeed, isAfter | Condition, Succession, Reason, Result, Precedence | - |
| xAttr                 | Synchronous, Reason, Result, Condition, Conjunction, Restatement | s-be-a/o, s-v-a, s-v-be-a/o, s-v, spass-v |
| HinderedBy            | Concession, Alternative, Synchronous, Conjunction | - |
| HasSubEvent           | Synchronous, Conjunction | - |

Table 16: Experimental results using two different neighboring functions.

| Model                  | Average AUC |
|------------------------|-------------|
| KG-BERTSAGE (Dir)      | 66.2        |
| KG-BERTSAGE (Undir)    | 67.2        |

KG-BERTSAGE is shown in Table 16. By incorporating bi-directional information of each vertex, the performance of CSKB population can be largely improved.

in GraphSAGE (Hamilton et al., 2017) to conduct uniform sampling on the neighbor set. 4 neighbors are randomly sampled during training.

C.2 Neighboring Function \( \mathcal{N} \)

The edges in ASER are directed. We try two kinds of neighboring functions:

\[
\mathcal{N}(v) = \{(r, u)| (v, r, u) \in \mathcal{G}\} \tag{1}
\]

\[
\mathcal{N}(v) = \{(r, u)| (v, r, u) \in \mathcal{G} \text{ or } (u, r, v) \in \mathcal{G}\} \tag{2}
\]

Equation (1) is the function that returns the outgoing edges of vertex \( v \). Equation (2) is the function that returns the bi-directional edges of vertex \( v \). The overall results using the two mechanisms of
| Head                                           | Relation | Tail                                           | Label | Source               |
|------------------------------------------------|----------|------------------------------------------------|-------|----------------------|
| PersonX give PersonY ride                      | xNeed    | PersonX need to wear proper clothes            | Plau. | Triples in CSKBs     |
| PersonX be wait for taxi                       | isAfter  | PersonX hail a taxi                           | Plau. | (Original Test Set)  |
| PersonX be diagnose with something             | Causes   | PersonX be sad                                | Plau. | Randomly sampled examples |
| PersonX feel something                         | xEffect  | PersonX figure                                | Implau.|                        |
| PersonX be patient with ignorance              | HinderedBy | PersonY have the right vocabulary            | Implau.|                        |
| PersonY plan PersonY meaning                  | HasSubEvent | PersonY open it mechanically                | Implau.|                        |
| PersonX spill coffee                           | oEffect  | PersonY have to server                        | Plau. | CSKB head + ASER tail |
| PersonX care for PersonY                       | xNeed    | PersonX want to stay together                 | Plau. |                        |
| PersonX save money                             | HasSubEvent | PeopleX can not afford something             | Plau. |                        |
| PersonX decide to order a pizza                | xReact   | PersonX have just move                        | Implau.|                        |
| it be almost christmas                         | gReact   | PersonX be panic                              | Implau.|                        |
| arm be break                                   | isBefore | PersonY ask                                  | Implau.|                        |
| PersonX go early in morning                    | xEffect  | PersonX do not have to deal with crowd        | Plau. | ASER edges           |
| PersonX have time to think it over PersonX     | xReact   | PersonX be glad                               | Plau. |                        |
| PersonX have a good work-life balance          | xIntent  | PersonX be happy                              | Plau. |                        |
| PersonX weight it by value                     | oWant    | PersonY bet                                  | Implau.|                        |
| PersonX be hang out on reddit                  | oReact   | PersonY can not imagine                       | Implau.|                        |
| PersonX can get PersonY out shell              | xIntent  | PersonX just start poach PersonY              | Implau.|                        |

Table 17: Examples of the human-annotated populated triples.