Commonsense Knowledge Graph Reasoning by
Selection or Generation? Why?

Cunxiang Wang$^{1,2}$, Jinhang Wu$^3$, Luxin Liu$^3$ and Yue Zhang$^1$
$^1$School of Engineering, Westlake University, China
$^2$Zhejiang University, China, $^3$Alibaba Inc, China
{wangcunxiang, zhangyue}@westlake.edu.cn
{jinhang.wjh, xique.llx}@alibaba-inc.com

Abstract

Commonsense knowledge graph reasoning (CKGR) is the task of predicting a missing entity given one existing and the relation in a commonsense knowledge graph (CKG). Existing methods can be classified into two categories generation method and selection method. Each method has its own advantage. We theoretically and empirically compare the two methods, finding the selection method is more suitable than the generation method in CKGR. Given the observation, we further combine the structure of neural Text Encoder and Knowledge Graph Embedding models to solve the selection method’s two problems, achieving competitive results. We provide a basic framework and baseline model for subsequent CKGR tasks by selection methods.

1 Introduction

Common sense has received increasing attention in the natural language processing community (Sap et al., 2018; Bosselut et al., 2019). It has been found that current AI systems lack commonsense knowledge bengio2019from. The Commonsense Knowledge Graph (CKG) (Speer and Havasi, 2012; Sap et al., 2018) can be a promising way to introduce structured and explainable commonsense knowledge into AI/NLP systems (Davis and Marcus, 2015; Lin et al., 2019; Wang et al., 2019). Similar to general KGs, such as FreeBase (Bollacker et al., 2008), CKGs consist of tuples structure as (source entity, relation, target entity). However, compared to general KGs, CKGs have two significant differences. First, CKGs emphasize on commonsense knowledge. Second, entities in CKGs are semantically-rich short text rather than noun entities.

CKGs are data-sparse and costly to construct. So, automated methods can be useful to expand the existing CKGs, which we call commonsense knowledge graph reasoning (CKGR). The task is to find one or several correct target entities as output given a source entity and a relation as input. For example, given “go to zoo” as source entity and “Causes” as relation, we can infer that “see animal” is a suitable target entity. Shown in the Figure 1, target entities can either be obtained by selecting from existing entities or machine generation.

Broadly speaking, CKGR methods can be classified into two categories – the generation method (Bosselut et al., 2019) and the selection method (Bordes et al., 2013). In particular, given one entity and the relation, generation methods synthesize the missing entity as a text sequence. In contrast, selection methods score a list of candidate entities for ranking. The two types of methods have been relatively separately investigated, and there has been little comparison between them.

We find that generation methods have two main disadvantages. The first is the new semantic entities problem. In particular, the most important advantage claimed for generation methods is that it can generate new entities that do not appear in the dataset. However, when we analyze the generated results of COMET (Bosselut et al., 2019), we find that there is no new semantic entities. Most generated entities can be directly found in training data, and the remaining generating entities also have corresponding training entities with similar meanings.
The second is the human evaluation problem. It can be difficult to automatically evaluate the generated results in commonsense areas (Edunov et al., 2019), and it is costly and relatively unreproducible.

The selection method was the dominant method for general relational KGs (Socher et al., 2013). Knowledge Graph Embedding (KGE) models, such as TransE (Bordes et al., 2013) and TuckER (Balazevic et al., 2019), are the most representative selection method models. Recently, Malaviya et al. (2020) used the selection method on CKGs. Compared to the generation method, the selection method has several benefits. First, it is relatively easy to evaluate, and we can easily control whether a given tuple structure as \((\text{source entity}, \text{relation}, \text{target entity})\) is correct or not. Second, it can avoid unexpected outputs, such as incorrect or offensive entities, which is beneficial in most industrial applications.

One issue of selection methods is that they cannot directly deal with the rich textual information in CKGs. In addition, if an entity of a test tuple is unseen in the training set, it does not have a trained embedding, which results in the OOV situation where the tuple cannot be scored. The unseen-entities problem is the crucial difference between CKG reasoning and traditional knowledge graph completion (Bordes et al., 2013). All source entities in the test set of the CKG ATOMIC (Sap et al., 2018) are unseen. Malaviya et al. (2020) circumvented the unseen problem by re-splitting the test set of ATOMIC, avoiding test entities that are not existent in the training data. However, in practical application scenarios, we cannot expect that all input entities to already exist in the database.

Given the above observation, we combine the architecture of neural text encoders, such as CNN and LSTM, with the standard KGE models, such as TransE (Bordes et al., 2013) and TuckER (Balazevic et al., 2019). With text sequence node embedding, all KGE models can be simply applied to the CKGR task. Experiments on two representative datasets show that our model can achieve competitive results in the CKGR task, showing its effectiveness in helping a selection method solve rich textual information problem and unseen-entities problem. To our knowledge, we are the first to build a selection method for CKGR that allows unseen entities in the test data. ¹

### 2 Datasets-CKGs

We focus on two predominant CKGs, namely ConceptNet (ConceptNet-100K) (Li et al., 2016) and ATOMIC (Sap et al., 2018). The detailed statistics of the two datasets are listed in Appendix Table 1.

**ConceptNet-100K** is a sub graph obtained from the Open Mind Common Sense (OMCS) entries in the ConceptNet5 dataset (Speer and Havasi, 2012) by Li et al. (2016). It contains 100,000 tuples as the training set, 1,200 as the dev set, and 1,200 as the test set. A tuple in ConceptNet-100K is “go to zoo” “causes” “see animal”. The percentage of Unseen Source Entities in the test set is 2.8%.

**ATOMIC** is a daily event CKG. It contains 709,996 tuples as the training set, 79,600 as the dev set, and 87,481 as the test set. The source entities are typically described as “PersonX does something (on/with/.../ PersonY)”, for example, “PersonX puts his trust in PersonY”. ATOMIC contains 9 types of relations, including “xAttribute”, “xIntent”, “xReact”, “xEffect”, “xNeed”, “xWant” and “oReact”, “oWant”, “oEffect”, here “x” represents “PersonX” and “o” represents “PersonY”. The test source entities are all unseen at training set.

### 3 Analysis of the Generation Method

There are two representative generation methods for CKGR, including CKB Generation (Saito et al., 2018) and COMET (Bosselut et al., 2019). We focus on COMET, which is a “seq2seq” model that uses a Transformer language model architecture (GPT) (Radford, 2018) to generate target entities.

Original experimental results of COMET on ConceptNet and ATOMIC from the authors (Bosselut et al., 2019), include the human evaluation on 1,200 instances from ConceptNet-100k and 900 instances from ATOMIC. We conduct analysis on the same 1,200 and 900 instances. The results are shown in Table 1. Most generated entities have already appeared in the training set (96.42% for ConceptNet-100K; 96.44% for ATOMIC). We manually compared and found that even for the remaining entities (3.58% for ConceptNet-100K; 3.56% for ATOMIC), there also exist similar or better entities in the training set. For example, the input source entity is “read newspaper”. The input relation is “motivated by goal”. COMET’s output is “know about current event”, which does not appear in the training data, but there is a similar entity “learn about current event” in the training data.

We list all new output target entities by COMET,

¹We will release our data and source code at [URL] (removed for submission).
Dataset | In the Training set | Not in the Training set
---|---|---
**ConceptNet-100k** | **1157 (96.42%)** | **43 (3.58%)**
**ATOMIC** | **868 (96.44%)** | **32 (3.56%)**

(a) How many generated entities are in or not in the training set.

Dataset | Exists similar/better Entity in Training set | No similar/better Entity in the Training set
---|---|---
**ConceptNet-100k** | **43 (3.58%)** | **0 (0.00%)**
**ATOMIC** | **32 (3.56%)** | **0 (0.00%)**

(b) Analysis of the generated entities which do not directly appear in the training set.

Table 1: Analysis on the generated entities by COMET. We chose the “greedy decoding” results, which are reported having the highest human evaluation scores (Bosselut et al., 2019).

which do not directly appear in the training data and their similar/better counterparts from training data in the Appendix.

### 4 Integration Text Encoding to a Selection Model

Following previous works on KGE (Bordes et al., 2013; Balazevic et al., 2019), the CKGR task can be described as follows. For any test tuple \((s \ r \ t)\), the input is “s r t”, the output is suitable “t”s from existing entities. To measure the model performance, for each test tuple \((s \ r \ t)\), every existing entity \(e_k\) in the CKG will be seen as a candidate target entity and makes a candidate tuple \((s \ r \ e_k)\).

We calculate every candidate tuple’s confidence score and compute the rank \(rk_t\) of the tuple with the correct “t”. Following Balazevic et al. (2019), we use \(rk_t\) to measure metrics Mean Rank (MR), Mean Reciprocal Rank (MRR), and HITS@10/3/1. Mean Rank is the mean of \(rk_t\), Mean Reciprocal Rank is the mean of the inverse of \(rk_t\). Hits@k measures the percentage of \(rk_t\) within the top k. Lower MR, MRR and higher Hits@10/3/1 indicate the better results and better performance. Since there might also be other correct tuples with the same “s r” but different “t”s in training, dev or testing sets, we remove them when computing \(rk_t\).

#### 4.1 Model

We aim to extend standard KGE models for handling text and unseen entities. The model consists of two modules: Text Encoder module and the standard KGE module. We call the model as Text Encoder Enhanced (Encoder Model Name + KGE Model Name), such as TEE (CNN + TuckER). For each tuple \((s \ r \ t)\), Text Encoder Module is used to encode the embeddings of each word of the source entity, relation, and target entity into three embedding vectors. Then the three embedding vectors are used in a KGE Module to calculate the confidence score of the tuple.

In the experiments, we choose two relatively simple models (i.e., CNN and BiLSTM) as the text encoder models. For the KGE Module, we use one representative KGE model TransE (Bordes et al., 2013) and one state-of-art KGE model TuckER (Balazevic et al., 2019), which serve as the baseline models in the experiments. In general, other text encoder models and KGE models can also be used.

Since the TuckER model in the original paper (Balazevic et al., 2019) train entire graphs at each training batch and the CKGs contain a much larger amount of entities than normal KGs, it leads to intolerable memory cost when we apply the training method directly on CKGs. We revise the training function by taking sampling methods used by Bordes et al. (2013); Lin et al. (2015), using one positive entity and one random negative entity per training case. We call the TuckER Model with the original training method as original TuckER and ours as revised TuckER. The training objective of the original TuckER is

\[
L = \frac{1}{n_e} \sum_{i=1}^{n_e} (\gamma^{(i)} \log(p^{(i)}) + (1 - \gamma^{(i)}) \log(1 - p^{(i)}))
\]

where \(n_e\) is all entities, \(p \in \mathbb{R}^{n_e}\) is the vector of predicted probabilities and \(\gamma \in \mathbb{R}^{n_e}\) is the binary label vector. For our revised TuckER, the training objective is the same, but \(n_e\) denotes the current training entities instead, \(p\) becomes \(\in \mathbb{R}^{2}\), and \(\gamma\) becomes \(\in \mathbb{R}^{2}\). To make fair in the comparison, the revised TuckER, instead of the original one, is used as the baseline model.
| Model                  | ATOMIC     | ConceptNet-100k |
|------------------------|------------|-----------------|
|                        | MR  | MRR  | Hits@10/3/1 | MR  | MRR  | Hits@10/3/1 |
| TransE (Bordes et al., 2013) | 39,140 | 0.000 | 0.0/0.0/0.0 | 12,311 | 0.025 | 5.8/2.1/0.7 |
| TuckER (Balazevic et al., 2019) | 152,457 | 0.000 | 0.0/0.0/0.0 | 3,401 | 0.285 | 50.5/35.3/20.1 |
| TEE (CNN+TransE)       | 31,161 | 0.004 | 3.6/1.3/0.4 | 4,754 | 0.013 | 16.7/7.9/1.3 |
| TEE (BiLSTM+TransE)   | 18,005 | 0.009 | 6.5/2.5/0.9 | 6,271 | 0.011 | 15.4/6.7/1.1 |
| TEE (CNN+TuckER)      | 3,342  | 0.181 | 23.5/19.9/14.5 | 1,355 | 0.553 | 74.5/62.7/45.3 |
| TEE (BiLSTM+TuckER)   | 1,562  | 0.461 | 55.7/47.4/41.4 | 1,702 | 0.113 | 24.0/12.4/5.9 |

Table 2: Experimental results of TuckER and our models on ATOMIC and ConceptNet-100k

5 Results

The results of the baseline models and our models are shown in Table 2.

**ATOMIC.** Since no source entities of the testset in ATOMIC have trained embeddings by standard KGE models, the test tuples cannot be scored by the baseline models. As a result, the results of baselines on ATOMIC are equal to random guesses. TEE (CNN+TransE), TEE (BiLSTM+TransE), and TEE (CNN+TuckER), TEE (BiLSTM+TuckER) have significantly better results over the baselines, which showing the ability of Text Encoder Enhanced model in handling unseen entities. We find that 86% of the words in the ATOMIC test entities appear in the training entities, which can explain why the TEE model can solve the unseen-entities problem on ATOMIC.

**ConceptNet-100k.** Both TEE (CNN+TransE) and TEE (BiLSTM+TransE) perform better in Mean Rank and Hits@10/3/1, but worse in MRR. TEE (CNN+TuckER) has better performance in all metrics compared with the baseline TuckER model. The results show that even though most test source entities are seen at training data, adding text information can also benefit the results. However, TEE (BiLSTM + TuckER) has a negative performance in most metrics on ConceptNet-100k compared with TuckER, which indicates that if the entities are too short and the baseline model already performs well, using BiLSTM to encode them may cause overfitting.

TEE (BiLSTM + TuckER) performs better than TEE (CNN + TuckER) on ATOMIC but worse on ConceptNet-100k. This is because the entities of ATOMIC have longer text than entities of ConceptNet-100k, which is 4.41 to 1.72. BiLSTM may be stronger than CNN in encoding longer text. However, when encoding short text with one or two words, BiLSTM underperforms CNN.

6 Related Work

**Standard KGE Models.** A range of models have investigated on how to represent a knowledge graph, including TransE (Bordes et al., 2013), NTN (Socher et al., 2013), TuckER (Balazevic et al., 2019), etc. Standard KGE models cannot work in ATOMIC because the source entities in the testing set do not appear in the training set. External Text Enhanced KGE Models, such as DKRL (Xie et al., 2016) and Jointly (Xu et al., 2017), are also related to our work. They use descriptions of entities crawled from Wikipedia to enhance the entity representation, but do not encode the entity text information. In contrast, we concentrate on the entity text information yet do not import external descriptions of entities.

**Malaviya et al. (2020)** is a concurrent paper that considers link prediction task on ConceptNet-100k and ATOMIC. However, there are some key differences. First, the goal is different. While they try to apply off-shelf KGE models for CKGs completion, testing their performance, our goal is to do CKGR. Second, it does not solve the unseen-entities problem because unseen entities cannot get proper embedding in their testing period. Their ATOMIC results are reported using a different partition from existing work, which does not contain unseen entities. In addition, our model can train over the full graph in a reasonable time/memory cost and do not need “subgraph sampling”, which theirs need.

7 Conclusion

We compared the advantages of two types of methods for commonsense knowledge graph reasoning, namely, the generation method and the selection method. The generation method requires large human labor to evaluate, and we empirically find that it does not actually generate new semantic entities. The selection method faces rich textual information problem and unseen-entities problem. We solve the issues by combining the structure of neural Text Encoder and Knowledge Graph Embedding models. Experiments show that our model can outperform standard KGE baselines. To our knowledge, we are the first to build a selection method for CKGR that allows unseen entities in the test data.
References

Ivana Balazevic, Carl Allen, and Timothy Hospedales. 2019. TuckER: Tensor factorization for knowledge graph completion. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5188–5197, Hong Kong, China. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In SIGMOD 2008, SIGMOD ’08, pages 1247–1250, New York, NY, USA. ACM.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 26, pages 2787–2795. Curran Associates, Inc.

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, Florence, Italy. Association for Computational Linguistics.

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

Sergey Edunov, Myle Ott, Marc’Aurelio Ranzato, and Michael Auli. 2019. On the evaluation of machine translation systems trained with back-translation. ArXiv, abs/1908.05204.

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, Berlin, Germany. Association for Computational Linguistics.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Yankai Lin, Zhiyuan Liu, Xuan Zhu, Xuan Zhu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In 29th AAAI, pages 2181–2187.

Chaitanya Malaviya, Chandra Bhagavatula, Antoine Bosselut, and Yejin Choi. 2020. Exploiting structural and semantic context for commonsense knowledge base completion.

Alec Radford. 2018. Improving language understanding by generative pre-training.

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

Maarten Sap, Ronan LeBras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. 2018. ATOMIC: an atlas of machine commonsense for if-then reasoning. CoRR, abs/1811.00146.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, NIPS 26, pages 926–934, Curran Associates, Inc.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12), pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4020–4026, Florence, Italy. Association for Computational Linguistics.

Ruobing Xie, Zhiyuan Liu, Jia Jia, Huanbo Luan, and Maosong Sun. 2016. Representation learning of knowledge graphs with entity descriptions. In 30th AAAI, AAAI’16, pages 2659–2665. AAAI Press.

Jiacheng Xu, Xipeng Qiu, Kan Chen, and Xuanjing Huang. 2017. Knowledge graph representation with jointly structural and textual encoding. In IJCAI-17, pages 1318–1324.
| Dataset         | Tuple Number | Entity Number | Relation Number | Word Number | Average Length | Unseen Tuple Proportion |
|-----------------|--------------|---------------|-----------------|-------------|----------------|--------------------------|
| ConceptNet-100K | 10000/1200/1200 | 78279         | 34              | 21498       | 1.72           | 2.8%                     |
| ATOMIC          | 709996/79600/87481 | 304903       | 9               | 29415       | 4.41           | 100%                     |

Table 3: Detailed data of the two datasets. The first/second/third column is the tuple/entity/relation number of the dataset; The forth column is the word number of entities and relations; The next column is the average length of entities’ texts. The last column is the proportion of unseen test tuples in all test tuples.
| Input Source Entity | Input Relation | Output Target Entity (Generated by COMET) | Similar/Better Entity in Training Set |
|---------------------|----------------|--------------------------------------------|---------------------------------------|
| go to get haircut    | causes         | hair to be cut                              | hair be become short                  |
| man                 | desires        | be with woman                               | have sex with girl                    |
| telephone           | is a           | call                                        | communication tool                    |
| food                | is used for    | keep you alive                              | stay alive                            |
| read newspaper      | motivated by goal | know about current event                  | learn about current event             |
| get up early        | has a prerequisite | be up early                  | wake up early                          |
| mechanic            | is capable of  | repair car                                  | repair car body                       |
| mother              | is capable of  | love her child                              | mother her child                      |
| pipe                | is capable of  | hold tobacco                                | hold tobacco in pipe                  |
| water               | is capable of  | be freeze water                             | be freeze                              |
| needle              | is used for    | inject drug                                 | inject substance into                 |
| have asthma attack  | has a prerequisite | have asthma                  | someone body                           |
| car                 | receives action | park in garage                             | person with asthma                    |
| submarine           | is a           | dangerous occupation                        | put car in garage                     |
| awake               | requires       | you feel groggy                            | be dangerous occupation               |
| have haircut         | causes         | hair to grow                                | feel groggy                           |
| music stand         | is used for    | hold your instrument                       | hair become short                     |
| brake               | is capable of  | stop car                                    | hold music                            |
| get job             | requires       | you get pay                                 | slow vehicle                          |
| triangle            | has a          | four side                                   | get pay                               |
| lake                | is a           | large lake                                  | 3 side                                |
| urinate             | motivated by goal | you be urinate                          | large body of water                   |
| pillow case         | is used for    | keep pillow                                 | eliminate excess fluid                |
| cat                 | is a           | small than cat                              | from your body                        |
| there               | requires       | no way to predict future                   | keep pillow clean                      |
| eat breakfast in bed| is capable of  | spill coffee on sheet                      | carnivors                             |
| screwdriver         | is a           | fit screw                                   | no proof of god                       |
| high rise           | is capable of  | call skyscraper                            | spill coffee on bedcover              |
| magician            | has a prerequisite | surprise them               | tighten screw                          |
| surprise someone    | receives action | train to jump over obstacle               | skyscraper                            |
| horse               | has a          | many side                                   | perform trick                         |
| chess board          | is a           | form of rain                                | not tell your plan to them            |
| rain                | requires       | talk about work                            | be train ot jump over obstacle        |
| chat with friend     | requires       | tickle someone                             | 4 side                                |
| tickle              | has a prerequisite | clean your wound                       | weather phenomenon                    |
| take bus            | is capable of  | get bus pass                                | have something to talk about          |
| nurse               | is capable of  | clean wound                                 | find someone you can tickle           |
| terrorist           | has a prerequisite | attack other country                   | buy bus pass                          |
| play game           | is capable of  | have game                                   | clean wound                           |
|                     | has a prerequisite | have game                                   | plan attack                           |

Table 4: Detailed data of ConceptNet-100k related to (b) of Table 2. The first column is the Source Entity; the second is the Relation; the third is Target Entity generated by COMET; the last column is the corresponding similar/better entity in the training data to the generated entity. (Some (source entity + relation)s repeats, we combine them together, so there are less than 43 lines.)
| Input Source Entity                                      | Input Relation | Output Target Entity | Similar/Better Entity in Training Set |
|---------------------------------------------------------|----------------|----------------------|---------------------------------------|
| PersonX likes the movie                                | xEffect        | buys the ticket       | He buys the tickets.                  |
| PersonX compares ___ with another                       | xIntent        | to compare            | to compare the two                    |
| PersonX walks in front of PersonY                       | xIntent        | to be in front        | to be in front of personY             |
| PersonX transmits in PersonY form                      | oWant          | to get personX form back | to get it back                        |
| PersonX reports every ___                               | xNeed          | to have a report      | to have a report of the facts         |
| PersonX compares ___ with another                       | xNeed          | to have a different opinion | has different opinions                |
| PersonX gives it PersonXs name                          | xNeed          | to have a name        | to have name                          |
| PersonX puts PersonYs head down                        | xNeed          | to have a head        | to grab PersonYs head                 |
| PersonX takes down a peg                               | xNeed          | to have a form        | to get a form                         |
| PersonX transmits in PersonY form                       | xNeed          | to have a form        | get a form                            |
| PersonX tells my mother                                | xNeed          | to call my mother     | to call my mom                        |
| PersonX grabs a towel                                   | xNeed          | to find a towel       | to find a towel to dry off.           |
| PersonX takes the offer                                 | xNeed          | to have a proposal    | to make a proposal                    |
| PersonX takes the offer                                 | xNeed          | to have a proposal    | to make a proposal                    |
| PersonX finds PersonXs daughter                        | xNeed          | to search for their daughter | to go look for her                    |
| PersonX creates PersonY structure                       | xNeed          | to know how to construct | to learn how to construct it          |
| PersonX visits PersonYs cousin                          | xNeed          | to drive to personYs cousins house | to drive to his place                 |
| PersonX wants to learn to swim                          | xNeed          | to have a swimming suit | get a swimming suit                   |
| PersonX keeps ___ with the growth                       | xWant          | to keep the growth going | to increase the growth to next level |
| PersonX compares ___ with another                       | xWant          | to compare other things | to compare the two                   |
| PersonX bosses PersonY around                           | xWant          | to greet personYs wife | to greet PersonYs wife                |
| PersonX bosses PersonY around                           | xWant          | to get personYs wife  | to get PersonY to do                  |
| PersonX visits PersonYs cousin                          | xNeed          | to drive to persons cousins house | to talk to his place                  |
| PersonX wants to learn to swim                          | xNeed          | to have a swimming suit | get a swimming suit                   |
| PersonX keeps ___ with the growth                       | xWant          | to keep the growth going | to increase the growth to next level |
| PersonX compares ___ with another                       | xWant          | to compare other things | to compare the two                   |
| PersonX bosses PersonY around                           | xWant          | to greet personYs wife | to greet PersonYs wife                |
| PersonX bosses PersonY around                           | xWant          | to get personYs wife  | to get PersonY to do                  |
| PersonX pulls PersonYs hand away                        | xWant          | to let go of personYs hand | to let go of PersonYs hand            |
| PersonX transmits in PersonY form                       | xWant          | to get personYs attention | to get PersonYs attention              |
| PersonX throws food                                     | xWant          | to throw more food     | to throw food away                    |
| PersonX tells my mother                                 | xWant          | to tell my mother      | to talk to my mom                     |
| PersonX never been to one                               | xWant          | to go to the one       | to go to it                           |
| PersonX sees ___ online                                 | xWant          | to read the reviews    | to read reviews                       |
| PersonX takes ___ up on the offer                       | xWant          | to make a good offer   | To make X a good offer                |
| PersonX finds PersonXs daughter                         | xWant          | to hug daughter        | to hug their daughter                 |
| PersonX visits PersonYs cousin                          | xWant          | to talk to personYs cousin | to chat with the cousin               |

Table 5: Detailed data of ATOMIC related to (b) of Table 2. The first column is the Source Entity; the second is the Relation; the third is Target Entity generated by COMET; the last column is the corresponding similar/better entity in the training data to the generated entity.