Commonsense Knowledge Salience Evaluation with a Benchmark Dataset in E-commerce

Yincen Qu[1], Ningyu Zhang[2], Hui Chen[1], Zelin Dai[1], Zezhong Xu[2], Chengming Wang[1], Xiaoyu Wang[1], Qiang Chen[1], Huajun Chen[2]

[1]Alibaba Group, [2]Zhejiang University & AZFT Joint Lab for Knowledge Engine

Introduction

In e-commerce, the salience of commonsense knowledge (CSK) is beneficial for widespread applications such as product search and recommendation. For example, when users search for 'running' in e-commerce, they would like to find products highly related to running, such as 'running shoes' rather than 'shoes'. Nevertheless, many existing CSK collections rank statements solely by confidence scores, and there is no information about which ones are salient from a human perspective.

In this work, we define the task of supervised salience evaluation, where given a CSK triple, the model is required to learn whether the triple is salient or not. In addition to formulating the new task, we also release a new Benchmark dataset of Salience Evaluation in E-commerce (BSEE) and hope to promote related research on commonsense knowledge salience evaluation. We conduct experiments in the dataset with several representative baseline models. The experimental results show that salience evaluation is a hard task where models perform poorly on our evaluation set.

Dataset

Task Design

We say that (s, p, o) is salient if, typically, most humans thinking of s would associate it with p, o. This somewhat informal definition is based on (and assumes) common human selectional preference as well as commonsense knowledge.

Since the definition of salience is marginally vague and hard to annotate, we propose to harness the concepts of sufficiency and necessity to discern salience. The relationship between the salience S_i of instance i and the necessary factor Nec_i and the sufficient factor Suf_i is expressed by the following equation:

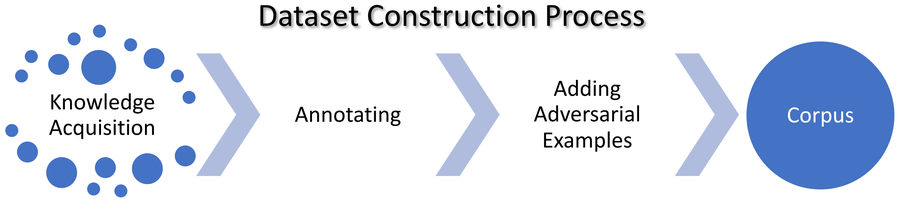
$$\mathrm{S}_i = \lambda \mathrm{Suf}_i + (1-\lambda)\mathrm{Nec}_i$$

where $\lambda \in [0,1]$is the parameter that weight the necessity and sufficiency.

$(\mathrm{running}_s, \mathrm{requires}_p, \mathrm{shoes}_o)$ encodes more sufficiency, since in most situations running requires shoes.

$(\mathrm{running}_s, \mathrm{requires}_p, \mathrm{weight\ running\ vest}_o)$ encodes more necessity, as running is almost the only cause for requiring weight running vest.

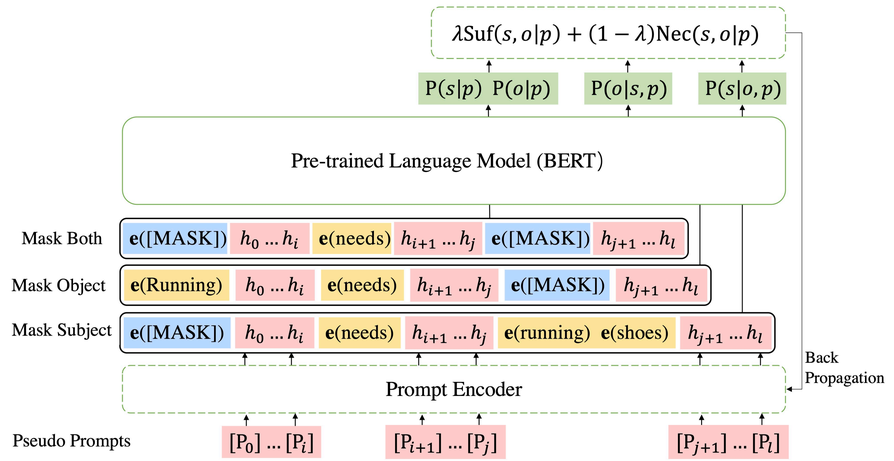
Dataset Construction Process



Annotation Cases

Subject	Predicate	Object	Sufficiency	Necessity	Salience
Bride	Capable Of	Wedding	Often True	Often True	Salient
Student	Capable Of	Drinking Water	Often true	Rarely True	Not Salient
Running	Requires	Wireless Mouse	Rarely True	Rarely True	Not Salient
Running	Requires	Running Shoes	Often True	Often True	Salient
Running	Requires	Weight Running Vest	Occasionally True	Often True	Not Salient

Model



The normalized necessity and sufficiency of (s, p, o) are defined as:

$$\mathrm{Nec}(s,o|p) = \frac{\log \mathrm{P}(s|p,o) - \alpha \log \mathrm{P}(s|p)}{-\log \mathrm{P}(s|p,o) - \alpha \log \mathrm{P}(o|p)}$$

$$\mathrm{Suf}(s,o|p) = \frac{\log \mathrm{P}(o|s,p) - \alpha \log \mathrm{P}(o|p)}{-\log \mathrm{P}(o|s,p) - \alpha \log \mathrm{P}(s|p)}$$

$$\mathrm{S}_i = \lambda \mathrm{Suf}_i + (1-\lambda)\mathrm{Nec}_i$$

The final loss is MSE loss:

$$L = \frac{1}{n}\sum_{i=1}^{n}(S_i' - S_i)^2$$

Masked language model (MLM) is utilized to compute the probability of masked word.

To better capture the underlying knowledge in MLM, we organize the masked words, triple, and prompts into different templates. So the input of MLM is as follows:

$$\mathrm{T}_1 = s[\mathrm{P}_{0:i}]p[\mathrm{P}_{i+1:j}][\mathrm{MASK}][\mathrm{P}_{j+1:l}]$$

$$\mathrm{T}_2 = [\mathrm{MASK}][\mathrm{P}_{0:i}]p[\mathrm{P}_{i+1:j}]o[\mathrm{P}_{j+1:l}]$$

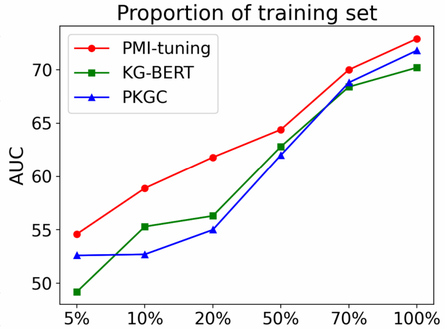
$$\mathrm{T}_3 = [\mathrm{MASK}][\mathrm{P}_{0:i}]p[\mathrm{P}_{i+1:j}][\mathrm{MASK}][\mathrm{P}_{j+1:l}]$$

Experiments

Main Results

Models	Random Split			Concept Split		
	F1	Acc.	AUC	F1	Acc.	AUC
BERTSAGE (Fang et al., 2021c)	73.1	74.2	75.7	54.5	60.1	67.1
StAR (Wang et al., 2021)	79.4	85.2	89.7	57.1	61.4	69.2
KG-BERT (Yao et al., 2019)	95.4	97.2	98.5	59.7	63.0	70.2
GenKGC (Xie et al., 2022)	**96.4**	**97.7**	**99.4**	60.3	60.2	71.2
PKGC (Lv et al., 2022)	89.7	93.0	96.5	61.2	62.9	71.8
PMI-tuning (Simplified)	90.1	92.3	96.2	**62.6**	**63.3**	**72.9**
PMI-tuning (Original)	87.4	91.1	94.8	**63.4**	**64.1**	**74.3**

Analysis



λ	F1	Acc.	AUC
0	62.0	60.3	70.1
0.3	62.2	61.2	72.1
0.5	**62.6**	**63.3**	**72.9**
0.7	61.5	62.4	72.8
1	61.1	61.0	72.7

Encoder	F1	Acc.	AUC
RoFormer	60.1	59.8	70.5
RoBERTa	62.5	60.6	72.3
BERT-wwm	**63.0**	**62.1**	**73.5**
MacBERT	62.0	**62.1**	73.2