融合知识库和文本的知识表示学习软件

使用说明书

编写人：

周惠巍，郎成垒，刘喆，雷弼尊，李雪菲，徐奕斌

大连理工大学

目　录

第一部分 软件简介 3

第二部分 安装说明 6

第三部分 使用手册 7

第一部分　软件简介

随着知识的不断增长，人们构建了大量的知识库用于存储不同领域、不同类型的知识。知识库中的知识一般以三元组（头实体，关系，尾实体）的形式表示，这些知识三元组对于知识抽取、知识问答等下游任务具有重要的意义。然而 one-hot 形式表示的知识三元组难以描述实体与关系间复杂的语义关系，存在数据稀疏问题。同时，许多非结构化文本中也存在大量描述实体对关系的信息，这些信息可以辅助知识的学习。如何将知识库中的结构化信息与文本中的非结构化信息进行融合，学习实体表示和关系表示，是一个亟待解决的问题。本软件实现了融合知识库和文本的知识表示学习方法，学习知识库中的关系和实体表示，同时学习文本中的关系和实体表示，并将两者进行对齐，统一到同一特征空间，缓解了知识库数据稀疏的问题，为下游知识抽取、知识推理与知识问答等任务提供支持。

本软件提出的融合知识库和文本的知识表示学习方法，主要包括（1）知识三元组的表示学习，（2）知识库与文本对齐两部分。其中，（1）知识三元组的表示学习继承了翻译模型的思想，将关系视为头实体到尾实体的翻译，并将三元组中的头实体、关系和尾实体映射成低维的向量表示，在低维向量空间中计算它们之间的语义联系；（2）知识库与文本对齐通过将描述三元组的文本经过卷积神经网络等编码器进行编码后，获得实体、关系表示，再与三元组中的实体、关系表示进行对齐，从而将知识表示与文本表示统一在同一空间。最终将两部分的优化目标按照一定比例进行加和，进行联合优化，

1.1　软件名称

中文:融合文本的知识表示学习软件　简称 KB-TextRL1.0
英文: Knowledge Base-Text Combined Knowledge Representation Learning Software

1.2　软件适用行业与用途

适用行业：自然语言处理相关行业。
用途：融合知识库中的知识三元组（头实体，关系，尾实体）、以及描述头实、尾体具有该种关系的文本，学习获得实体和关系的低维向量表示，即知识表示，为知识抽取、知识推理、知识扩充提供依据。

1.3　软件开发平台

软件运行平台：Linux 操作系统 （Ubuntu16.04 及以上）
编程语言：Python3.6
版本号：　1.0

程序量： 1337 行源代码

1.4 创作目的

(1) 学习获得实体和关系的低维向量表示，即知识表示，解决 one-hot 形式表示的知识三元组存在的数据稀疏问题。

(2) 将知识库中的三元组结构信息，与文本中的描述该三元组的文本信息，融合到同一特征空间，获得实体和关系的低维向量表示，为下游相关任务提供支持。

1.5 主要功能

(1)输入模块

首先，输入包含三部分，第一部分为用户提供的待训练知识三元组及其对应的描述文本；第二部分为可选项，用户可以提供自定义的预训练词向量、实体向量、关系向量作为模型中词、实体、关系表示的初始化向量，也可以提供预训练的模型参数与编码器参数，用户若未提供此项输入，则默认全部随机初始化；第三部分为模型参数设置，用户可以根据自身需求，自行选择编码器，定义知识表示向量维度、训练轮数、学习率等模型超参数。

(2) 数据预处理模块

此模块主要将用户提供的输入数据进行封装保存，以便训练模型再次使用，输入数据将被封装成 pkl 文件，存放在用户指定的目录。

(3) 模型定义模块

此模块定义了用于训练知识表示的 TransText 模型，其中包括两种文本编码器（lstm 与 cnn 编码器），用户可以根据需求自行选择合适的编码器对文本进行编码，或在上述输入模块中修改模型参数。

(4) 模型训练模块

此模块将封装后的数据导入融合文本的知识表示学习模型，对模型进行训练，并在每次训练完毕后导入评价模块，对模型性能进行评价，并保存评价结果最好的模型及其参数。用户可以在上述输入模块中修改训练参数。

(5) 评价模块

在 i 轮迭代后（i 为用户自定义参数），此模块将对模型进行评价，评价方法为三元组补全任务，本软件中即为预测三元组中的“关系”标签，并默认根据平均排名（MR）衡量模型性能的好坏，返回评价指标。

1.6 技术特点

(1) 低维向量空间学习知识表示

通过将知识三元组映射为低维稠密的向量表示，在低维向量空间中学习实体表示和关系表示，可以提高计算效率，并在一定程度上解决数据稀疏的问题。

(2) 联合学习知识库与文本中的三元组信息

知识库中的三元组结构信息以结构化的形式出现，文本中的三元组结构信息则以非结构化的形式出现。本软件采用实体对齐与关系对齐策略，将知识库中结构化的三元组信息与文本中非结构化的三元组信息进行联合学习，实现多元异质信息融合。知识库信息和文本信息相互补充，提升了知识表示的学习质量。同时将两者映射到统一的语义空间中，可以广泛地应用于不同类别的下游任务中。

(3) 联合优化三种损失

本软件训练的目标函数包括三部分损失：知识表示的三元组损失（合页损失）、实体对齐损失以及关系对齐损失（均方差损失），三部分损失可以按照用户指定的不同比例进行累加，然后进行训练，损失详细定义如下。

①. 知识库中三元组的合页损失（Hinge Loss）：

形如(h,r,t)的知识三元组，其中h表示头实体，r表示关系，t表示尾实体，其对应的知识表示为$(\mathbf{h},\mathbf{r},\mathbf{t})$。TransText 模型借鉴翻译模型的思想，将关系视为头实体到尾实体的翻译，即希望满足$\mathbf{h}+\mathbf{r}\approx\mathbf{t}$。采用最大间隔方法，定义合页损失函数如下：

$$L_{triple}=\sum_{(h,r,t)\in S}\ \sum_{(h',r,t)or(h,r,t')\in S'}\max(0,\gamma+\|\mathbf{h}+\mathbf{r}-\mathbf{t}\|-\|\mathbf{h}'+\mathbf{r}-\mathbf{t}'\|)$$

其中，S表示所有合法三元组的集合，S'表示错误三元组集合，γ为合法三元组与错误三元组之间的距离。错误三元组通过将合法三元组的头实体h或尾实体t随机替换为另一错误实体h'或t'获得。

②. 文本描述的实体表示与知识库的实体表示间对齐的均方差损失（MSE Loss）：

给定一个三元组(h,r,t)的句子集合$B=\{s_1,s_2,...,s_m\}$，其中任意一个句子$s_i=\{w_1,w_2,...,w_n\}$包含n个词，将句子经过编码器得到$\mathbf{S}_i=[\mathbf{s}_1,\mathbf{s}_2,...,\mathbf{s}_n]$，其中$\mathbf{s}_j$是编码后的词表示，我们将头实体包含的所有单词的词表示进行平均，得到文本编码的头实体表示$\mathbf{h}_s=\frac{1}{n_h}\sum_{s_i\in B}\sum_{w_k\in word(h)}\mathbf{s}_k$，同理得到文本编码的尾实体表示$\mathbf{t}_s=\frac{1}{n_t}\sum_{s_i\in B}\sum_{w_k\in word(t)}\mathbf{s}_k$，则实体对

齐损失定义如下：

$$L_{entity} = \sum_{(h,r,t)\in S} (\| \mathbf{h}_s - \mathbf{h} \|^2 + \| \mathbf{t}_s - \mathbf{t} \|^2)$$

③. 文本描述的关系表示与知识库的关系表示间对齐的均方差损失（MSE Loss）：

将上述$\mathbf{S}_i = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_n]$，经过最大池化操作，获得句子表示$\mathbf{c}_i$，再将句子集合中所有句子表示拼接，经过平均池化操作，获得的句子集表示$\mathbf{r}_s = \frac{1}{m}\sum_i \mathbf{c}_i$作为文本编码的关系表示，则关系对齐损失定义如下：

$$L_{relation} = \sum_{(h,r,t)\in S} \| \mathbf{r}_s - \mathbf{r} \|^2$$

将三部分损失加权求和，训练联合同一特征空间下的知识表示。

$$L_{total} = L_{triple} + \alpha(L_{entity} + L_{relation})$$

其中α是对齐损失的权重（惩罚因子）。

第二部分　安装说明

2.1　软件运行环境

软件运行的硬件环境的最低配置： CPU 主频 2.0 G Hz，内存容量 4 G，硬盘剩余容量 4 G，屏幕分辨率 800×600。

为提高计算速度，改善用户体验，硬件建议配置：CPU 主频 2.4 G Hz 以上，内存容量 4G，若须使用 cuda 加速，则需保证显存容量 2G 以上，硬盘剩余容量 10 G 以上，屏幕分辨率 1024×768。

软件运行的软件环境：Linux 操作系统（Ubuntu16.04 或更高版本）。

python3.6

torch1.0.0

numpy1.15.4

2.2　安装过程说明

本软件是绿色软件，无须安装。

2.3 备注

未解技术问题，可联系

E-mail: zhouhuiwei@dlut.edu.cn

第三部分 使用手册

用户在使用此软件前需首先提供训练数据，训练数据的格式与要求将在 3.1 输入模块中进行详细说明，然后进行数据预处理：

```
$ python ./GenerateData.py
```

以上将获得封装后的训练/验证数据，默认存储在./source/目录下，接下来训练模型：

```
$ python ./Train.py
```

模型训练将在每 i 轮训练结束后（i 为用户自定义参数）自动对验证数据进行评价，并保存评价结果最优的模型与相应的知识表示。

下面将分别介绍用户需涉及的各模块文件及变量的具体含义及使用方法。

3.1 输入模块

输入模块用于提供模型训练的测试数据与参数设置，主要包含两部分：用户提供的待训练三元组及对应文本数据；模型参数设置，如编码器结构、训练轮数、学习率、向量维度等。

三元组训练数据共包包含六个文件（3 个必选，3 个可选），其中包括训练数据共三个文件，知识表示与词向量的初始表示共三个文件。文件默认存储在./data/Input/目录下。

参数设置模块包括输入、输出文件路径、预训练资源路径等路径参数，批尺寸、训练轮数、向量维度等模型超参数。此模块包含的文件为 config.py。

各子文件具体内容及使用说明如下：

3.1.1 输入文件说明

文件名称	使用说明
entity2id.txt	训练/测试数据中所有实体与其对应 ID 的级联。
relation2id.txt	训练/测试数据中所有关系与其对应 ID 的级联。
train.txt	训练三元组与相关文本数据，数据格式如下：

	头实体 \t 关系 \t 尾实体 \t 文本句子数 头实体开头位置 \t 头实体结尾位置 \t 尾实体开头位置 \t 尾实体结尾位置 \t 句子 1 头实体开头位置 \t 头实体结尾位置 \t 尾实体开头位置 \t 尾实体结尾位置 \t 句子 2 …
pretrain_w2v.txt	可选，预训练词向量，若不存在则将随机初始化。
pretrain_e2v.txt	可选，预训练实体向量，若不存在则将随机初始化。
pretrain_r2v.txt	可选，预训练关系向量，若不存在则将随机初始化。

3.1.2 config.py 文件说明

config.py 中定义了 Config 类，用于存储训练与测试过程中所涉及的所有参数，其成员变量信息如下表所示：

成员变量名称	使用说明
inputPath	所有输入文件的路径。
entity2idFile	所有实体与其对应ID的级联文件路径。
relation2idFile	所有关系与其对应ID的级联文件路径。
triplesFile	所有三元组及其对应文本的文件路径。
pretrainW2VFile	预训练词向量的文件路径。
pretrainE2VFile	预训练实体向量的文件路径。
pretrainR2VFile	预训练关系向量的文件路径。
pretrainEncoder	文本编码器初始参数的文件路径。
pretrainModel	整个模型初始参数的文件路径。
outputPath	所有输出文件的路径。
saveW2VFile	训练获得的词向量的输出路径。
saveE2VFile	训练获得的实体向量（表示）的输出路径。
saveR2VFile	训练获得的关系向量（表示）的输出路径。
trainDataPath	封装后的训练数据，为pkl文件。
evalDataPath	封装后的验证数据（由训练数据分割得到），为pkl文件。
parameterPath	封装后的模型初始化参数，为pkl文件。
modelPath	模型的输出路径。
batchSize	批尺寸。
shuffle	是否打乱训练数据。
numWorkers	导入数据的进程数。
dropLast	是否丢弃最后一批不完整的数据。
repProba	替换头实体的概率。

WORD_EMB_DIM	词向量维度。
KG_EMB_DIM	实体向量维度。
splitRate	分割验证集的比例。
evaluate	是否进行验证。
TransText	TransText模型参数，是一个字典，包含“EmbeddingDim”：输入向量维度，“KgDim”：知识表示向量维度，“Margin”：正负例间的最大间隔（即γ），“Alpha”：对齐损失惩罚系数，“L”：采用L1或L2损失。
lstm	lstm编码器参数，是一个字典，包含“hiddenDim”：隐层维度，“biDirection”:是否双向。
cnn	lstm编码器参数，是一个字典，包含“hiddenDim”：隐层维度，“biDirection”:是否双向。
maxSentenceLen	文本句子最大长度。
maxSentenceNum	每个三元组允许的最多文本句子个数。
usegpu	是否使用cuda加速。
modelName	选择模型名称，目前只支持TransText模型。
encoderName	选择文本编码器的名称，目前支持cnn和lstm模型。
optimizer	优化器选择。
weightDecay	损失中L2正则项系数。
evalMethod	验证集评价指标，目前仅支持“MR”，即平均排名。
simMeasure	向量相似度计算方法，目前仅支持L2距离。
modelSaveType	模型存储格式，可选“param”：仅存储模型参数，“full”：存储模型参数和结构。
epochs	训练轮数。
evalEpoch	每迭代几轮进行一次测试。
learningRate	学习率。
lrdecay	学习率衰减比率。
lrdecayEpoch	学习率衰减轮数。
loadEncoder	是否加载预训练编码器。

3.2 输出模块

输出模块主要用于存放训练获得知识表示，词向量与模型参数等。训练获得的模型默认存放在./source/model/目录下，其余输出文件默认存放在./data/Output/目录下。各输出文件具体内容说明如下：

输出文件	说明

word2vec.txt	经过训练的词向量。
entity2vec.txt	训练获得的实体表示向量。
relation2vec.txt	训练获得的关系表示向量。
lstm.param	训练获得的 lstm 编码器的参数。
cnn.param	训练获得的 cnn 编码器的参数。
TransText_ent100_rel100.param	训练获得的 TransText 模型参数（不包含模型结构，知识表示向量 100 维）
TransText_ent100_rel100.full	训练获得的 TransText 模型参数（包含模型结构，知识表示向量 100 维）