基于知识表示的实体关系抽取软件

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

海量非结构化文本蕴含了大量有价值的结构化信息，是相关领域科学研究和实际应用中知识的重要来源，从非结构化文本中自动抽取结构化信息则是获取这些知识的必要途径。实体关系抽取作为信息抽取、自然语言理解、信息检索等领域的核心任务和重要环节,能够从文本中自动抽取实体对间的语义关系，这对于结构化信息的自动抽取具有重要意义。

传统的实体关系抽取方法主要分为三类：基于规则的方法；基于特征的方法；基于深度学习的方法。基于规则和基于特征的方法需要人工定义或抽取大量手工规则或特征，费时费力，且规则与特征往往仅适用于单一数据集，可扩展性较差。基于深度学习的方法将特征映射到低维稠密的向量空间中，使得模型可以自己学习数据的特征，在一系列关系抽取任务中表现出了极高的性能。然而，目前大部分深度学习方法仅仅利用了文本信息，大量知识库中包含的结构化知识，如形如（头实体，关系，尾实体）的三元组知识，未被充分利用。这些结构化的知识可以作为文本信息的补充，对于实体关系抽取具有一定的帮助作用。

本软件提出了一种基于知识表示的实体关系抽取方法，将非结构化的文本信息与结构化的知识结合，主要包括三部分内容：（1）文本信息编码。将实体对的上下文映射到低维语义空间，并利用神经网络对文本进行编码，获得文本表示；（2）知识表示学习。将知识库中的三元组信息利用知识表示学习，获得其相应的知识表示（实体表示与关系表示）；（3）融合知识表示与文本表示的关系分类。分别基于两个实体表示，利用门控机制，控制上下文特征的传播；并进一步基于关系表示，利用共享注意力机制，计算上下文特征的权重，获得加权上下文特征表示，用于关系分类。从而实现了基于知识表示的关系抽取，解决了因缺乏知识指导难以获得与实体对及实体关系相关的重要上下文信息的问题，为阅读理解、对话系统、信息检索等相应下游任务提供支持。

1.1　软件名称

中文：基于知识表示的实体关系抽取软件　简称 KR-ERES1.0
英文：Knowledge Representation-based Entity Relation Extraction 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：融合大规模结构化知识库与非结构化实体关系标注文本，训练获得基于知识表示的实体关系抽取模型，用于实体关系抽取。

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： 1145 行源代码

1.4 创作目的

将知识库中结构化知识引入实体关系抽取任务，融合非结构化的实体上下文文本信息，解决缺乏知识指导、难以获得与实体对及实体关系相关的重要上下文信息的问题，从而实现高性能的实体关系抽取。

1.5 主要功能

(1) 输入模块

输入模块主要包括三部分：①模型训练、测试数据以及知识库：这一部分需要用户根据其自身需求，自行准备。包括非结构化实体关系标注文本，即指定的实体对及其对应的上下文文本（训练语料）；待标注文本（验证、测试语料）；以及结构化知识库三元组集合（知识库）。数据详细格式在 **3.1 用户手册-输入模块**中进行描述；②初始化向量：此部分为可选项，用户可以选择采用自己训练的词向量、知识表示向量作为初始化参数，软件根据用户输入自动为模型中相应的向量进行初始化赋值，若用户未提供此项输入，则默认为随机初始化。③模型训练超参数：此部分为模型所有的超参数设置，包含迭代次数、批处理大小、卷积神经网络窗口大小、词向量维度、隐层维度等，用户也可根据需求可自行设定。

(2) 数据预处理模块

此模块主要包括两部分：①样例抽取：这一部分将用户给定的文档级别的数据进行预处理，构建句内、跨句两种样例作为模型的输入，具体的方法是：若指定两个实体出现在同一个句子中，则抽取它们共同所在的句子作为句内样例，若两个实体出现在不同的句子中，则抽取它们各自所在的句子作为跨句样例；若用户输入为句子级别的数据，则无需进行该步操作，直接将其作为句内样例使用。②数据封装：这一部分是将抽取后的样例进行封装保存，训练、测试样例以列表形式存储，预训练词向量与知识表示以 numpy 矩阵的形式存储，便于模型的训练与测试，所有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

此模块定义了基于知识表示的实体关系抽取模型（Knowledge Representation-based Entity Relation Extraction Model, KR-ERE），其中包括了基于实体表示的门卷积网络与基于关系表示的注意力机制，用户可以根据需求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基于实体表示的实体关系抽取模型（KR-ERE），并进行训练，在每次训练完毕后，会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。用户可以在上述参数定义模块中修改训练参数。

(5) 输出模块

在每轮迭代后，此模块将对模型进行测试，即用模型对用户提供的验证集与测试集进行预测，并输出预测标签，输出文件将存放在用户指定文件目录下。

1.6 技术特点

本软件利用门卷积网络与共享注意力机制来融合非结构化的文本信息与结构化知识，实现基于知识表示的实体关系抽取。首先对文本信息采用卷积神经网络进行编码，获得上下文特征；并分别基于两个实体表示，利用门控机制控制上下文特征的传播，保留与实体信息相关的上下文特征，门卷积网络具体操作定义如下：

$$c_i^k = \tanh(\mathbf{W}_c * x_i + b_c) \odot \text{relu}(\mathbf{W}_e * x + \mathbf{V}_e \cdot e^k + b_e),$$

其中，$\mathbf{W}_c, b_c, \mathbf{W}_e, \mathbf{V}_e, b_e$ 为门卷积网络的训练参数，$\odot$ 为对应元素相乘，x_i 为输入文本的词向量表示，e^k 为输入的实体表示，$k \in \{1,2\}$ 表示头实体、尾实体中的一个，c_i^k 为经过门卷积网络后的上下文特征，门卷积网络可以在融合文本信息与实体信息的同时，保证计算效率，使得有效的文本特征可以被高效地获取。

为了进一步获得与实体关系相关的上下文，本软件基于关系表示，利用共享注意力机制，计算经过门卷积网络后的上下文特征的权重，获得加权上下文特征表示，用于关系分类。首先利用上下文卷积特征与关系表示计算每个上下文词的权重 α_i^k：

$$\alpha_i^k = \frac{\exp(\tanh(\mathbf{W}_a c_i^k + b_a) \cdot r)}{\sum_j \exp(\tanh(\mathbf{W}_a c_j^k + b_a) \cdot r)}$$

其中，$\mathbf{W}_a, b_a$ 为注意力机制的训练参数，c_i^k 为上下文卷积特征，r 为从知识库中抽取并训练的关系表示，α_i^k 为计算获得的每个词的权重，$k \in \{1,2\}$ 表示头实体、尾实体中的一个。接着，利用权重 α_i^k 对上下文卷积特征进行加权求和，获得最终的上下文表示 m^k：

$$m^k = \sum_i \alpha_i^k c_i^k$$

基于关系表示的注意力机制可以进一步融合文本信息与关系信息，有效地将更多的权重

分配给与知识库中抽取的关系相关的上下文。这为模型最终的关系分类提供了更丰富、更准确的信息。一定程度上缓解了由缺乏知识指导导致的，难以从上下文中提取与实体对及实体关系相关信息的问题。

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，本软件须使用 cuda 进行加速，需保证显存容量 1G 以上，硬盘剩余容量 1G 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

用户在使用此软件时需首先提供训练数据，用户可以根据任务需求提供文档级别的数据或直接提供训练、验证、测试样例。针对文档级别的数据（数据格式在 **3.1 输入模块**中进行说明），本软件需要先进行样例抽取，按照 **1.5 软件简介-主要功能**中所述规则抽取实体对所在的句子作为样例：

$ python ./extract_instance.py

获得样例后，本软件将进一步对样例进行预处理与封装，样例的格式与要求将在 3.1 输入模块中进行详细说明：

$ python ./process_data.py

以上操作将获得封装后的数据，默认存储在./data/process/目录下，接下来训练模型：

$ python ./main.py

模型将在每轮训练结束后，自动对验证集与测试集进行评价，并保存在验证集中评价结果最优的模型与相应的模型预测结果。

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所需数据与模型的超参数设置，主要包含三部分：用户提供的实体关系抽取数据与相应知识库；用户提供的预训练词向量与预训练知识表示向量（可选项）；模型参数设置，如训练轮数、学习率、批尺寸、向量维度等。

用户可以提供文档级别的数据或直接提供训练、验证、测试句子，若提供的为文档级别的数据，则需先进行样例抽取，构建训练、验证、测试样例。文档级别的数据共包含 3 个文件（训练集、验证集、测试集），默认存储在./data/original/目录下。在进行样例抽取时，将根据两个实体是否在同一个句子中获得句内样例和跨句样例，两部分样例均包含训练集、验证集、测试集 3 个文件，共 6 个文件，默认存储在./data/process/intra/和./data/process/inter/目录下。若用户直接提供训练、验证、测试样例，则只需提供./data/process/intra/目录下的训练集、验证集、测试集 3 个文件即可。

参数设置文件中存储了包括输入、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。文件名 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
TrainingSet.txt	文档级别训练数据（训练集），数据格式如下（下同）： 文档 ID|t|文档标题 文档 ID|a|文档内容 文档 ID \t 实体起始位置 \t 实体结束位置 \t 实体提及名称 \t 实体类型 \t 实体 ID … 文档 ID \t 关系类型 \t 头实体 ID \t 尾实体 ID …
DevelopmentSet.txt	文档级别验证数据（验证集）

TestSet.txt	文档级别测试数据（测试集）
TrainingSet.instance	训练样例（句内/跨句），数据格式如下（下同）： 关系类型 \t 文章 ID_头实体 ID_头实体起始位置_头实体结束位置_尾实体 ID_尾实体起始位置_尾实体结束位置 \t 实体对所在上下文（句内样例为两个实体共同所在的句子，跨句样例为两个实体分别所在的句子的拼接）
DevelopmentSet.instance	验证样例（句内/跨句）
TestSet.instance	测试样例（句内/跨句）
data.pkl	封装后的所有数据
words.vocab	语料中涉及的单词与其对应编号的级联
miss.vocab	语料中涉及的单词，未在给定词向量中找到对应向量的所有单词的词表
entity2id.txt	所有实体与其对应 ID 的级联
relation2id.txt	所有关系与其对应 ID 的级联
train.txt	知识库中的知识三元组，数据格式如下： 头实体 \t 关系 \t 尾实体 ID
word2vec.vec	可选，预训练词向量，若不存在则将随机初始化
entity2vec.bern	可选，预训练实体向量（表示），若不存在则将随机初始化
relation2vec.bern	可选，预训练关系向量（表示），若不存在则将随机初始化
pretrain_model.param	可选，预训练模型参数，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超参数，其成员变量信息如下表所示：

成员变量名称	使用说明
ori_train_path	文档级别训练数据（训练集）的文件路径
ori_dev_path	文档级别验证数据（验证集）的文件路径
ori_test_path	文档级别测试数据（测试集）的文件路径
intra_path	句内样例的文件路径
inter_path	跨句样例的文件路径
train_ins_path	训练样例（训练集）的文件路径
dev_ins_path	验证样例（验证集）的文件路径
test_ins_path	测试样例（测试集）的文件路径
clean_data	经过封装后的数据路径，为pkl文件

word_vec_path	预训练词向量的文件路径
entity_index_path	所有实体与其对应ID的级联文件路径
relation_index_path	所有关系与其对应ID的级联文件路径
triple_path	知识库三元组的文件路径
entity_vec_path	预训练实体向量（表示）的文件路径
relation_vec_path	预训练关系向量（表示）的文件路径
result_path	所有输出文件的路径
intra	句内样例结果输出文件的子路径
inter	跨句样例结果输出文件的子路径
document	文档级别结果的输出文件的子路径
pretrain_model_path	预训练模型参数的文件路径
model_save_path	训练获得的模型的输出路径
word_vec_dim	词向量维度
kg_vec_dim	知识表示向量（实体向量、关系向量）维度
convolution_dim	卷积通道数（隐层维度）
kernel_size	卷积核大小，是一个列表，列表中的元素为不同卷积核的大小
intra_lr	句内样例学习率
inter_lr	跨句样例学习率
class_number	关系类别数
epoch_number	训练轮数
batch_size	批尺寸
is_document	是否输入为文档级别的数据

3.2 输出模块

输出模块主要用于存放训练获得的实体关系抽取模型参数、预测结果等。训练获得的模型默认存放在./result/model/目录下，模型预测结果存放在./result/目录下，其中句内样例的结果存放在/intra/子目录下，其中跨句样例的结果存放在/inter/子目录下，文档级别的结果（句内样例与跨句样例合并后）存放在./result/merge/目录下。各输出文件具体内容说明如下：

输出文件	说明
vresult_{i}.txt	验证集的预测结果（句内/句间），{i}表示第 i 轮迭代后（下同）
vprob_{i}.txt	验证集样例预测为各类别的概率文件（句内/跨句）
tresult_{i}.txt	测试集的预测结果（句内/跨句）
tprob_{i}.txt	测试集样例预测为各类别的概率文件（句内/跨句）
tmerge_result_{j}_{k}.txt	测试集的预测结果（文档级别），j 和 k 分别表示：

	验证集句内样例在第 j 轮迭代后取得了最优性能，验证集跨句样例在第 k 轮迭代后取得了最优性能
KR_ERE_model.param	训练获得的 KR-ERE 模型