

基于深度强化学习的文本相似语义计算模型

陈观林^{1,2}, 侍晓龙^{1,2}, 周 梁³, 翁文勇¹

(1. 浙大城市学院 计算机与计算科学学院 浙江 杭州 310015; 2. 浙江大学
计算机学院 浙江 杭州 310027; 3. 杭州市大数据管理服务中心 浙江 杭州 310020)

摘要: 语义相似计算是自然语言处理领域一个常见问题, 现有的基于深度学习的语义相似计算模型大多数是通过卷积网络或者长短时记忆模型来提取语义特征, 但是这种语义特征的提取方式存在语义信息丢失的问题。提出两点改进传统深度学习模型在提取语义特征时的语义丢失现象。首先是改进注意力相互加权模型。基于相互加权方式做出改进, 使用多个加权重矩阵加权语义, 同时提出新的正则项计算方法。其次在语义相似计算模型中引入强化学习的方法对文本进行自动分组处理, 在语义相似计算领域最常用的 Siamese Network 模型上使用强化学习算法, 改善长短时记忆模型在提取句子的语义时所面临的语义丢失现象。通过实验验证, 改进的方法处理中文句子有不错的效果。

关键词: 深度学习; 语义相似计算; 强化学习

中图分类号: TP319

文献标志码: A

文章编号: 1671-6841(2020)03-0001-08

DOI: 10.13705/j.issn.1671-6841.2020038

0 引言

语义相似计算是机器进行语义理解的一种手段^[1-2], 通过相似类比从而让机器能间接地理解问题。语义相似计算应用十分广泛, 比如百度问答系统、新闻推荐领域等。

相比于传统的基于统计的语义相似计算模型, 基于机器学习和深度学习的语义相似计算模型能更加细致地表达句子的语义特征和结构特征, 比如词向量^[3]、长短时记忆网络(long short-term memory, LSTM)模型^[4]。这些语义特征模型存在一些深度学习模型的通病, 比如长短时记忆模型虽然是专门用来处理时序模型的, 这种模型可以较好地表达句子的结构信息, 但是它在处理长句子过程中会因为反向传播算法而带来梯度消失问题, 从而丧失很多语义信息, 尤其是句子靠前部分的词语信息, 因为梯度很难传到靠近前面词语的位置。

深度学习的快速发展促进了词义、语义理解方面很大的进展, 比如 Word2vec^[5]、LSTM 等对语义特征提取有很好的表达方式。词向量模型的出现很好解决了词义级别上的表达, 通过收集大量语料库可以很好训练出词与词之间的关系表达, 词义关系通常体现在其词向量的空间距离上的关系。LSTM 模型主要是循环神经网络的改进模型, 用来解决具有时序相关性的问题。通过使用 LSTM 模型提取句子语义信息, 可以很好地表达句子结构上的信息, 然后映射到固定向量特征中, 通过计算向量的距离来表示相似程度, 这是比较主流的处理方式。卷积神经网络(convolutional neural networks, CNN)在图像处理领域取得了巨大成功, 很多研究人员把 CNN 当作语义特征抽取的一种方式, 将句子的词向量拼接成句子矩阵, 使用 CNN 网络卷积转化为语义矩阵, 通过池化等方法从语义矩阵抽象出语义向量, 计算向量的余弦距离或者欧氏距离等。注意力模型对图像和自然语言处理的深度学习模型有重要的影响, 注意力模型可以用于 LSTM 等模型当中, 缓解模型的语义丢失现象。

深度学习的语义计算模型一般是基于 Siamese^[6]模型, Siamese 网络本质上也是解决降维问题, 将句子的

收稿日期: 2020-02-17

基金项目: 国家自然科学基金项目(71303215); 浙江省科技计划项目(2020C03091, 2017C31110)。

作者简介: 陈观林(1978—), 男, 浙江台州人, 教授, 主要从事大数据及人工智能研究, E-mail: chenguanlin@zucc.edu.cn; 通信作者: 周梁(1982—), 男, 浙江杭州人, 高级工程师, 主要从事大数据研究, E-mail: zl@xxzx.hz.gov.cn。

语义映射到一个低维的向量空间。LSTM Siamese Network 网络^[7]是 Siamese 网络的一个实例框架,以单个字符作为输入单元,用 LSTM 网络来代替 Siamese 网络的函数,通过 LSTM 来提取特征得到一个固定长度向量,然后通过全连接层后抽取两个样本的最终特征作为计算距离的向量,采用余弦距离表示语义相似程度。Siamese LSTM 模型^[8]最大的亮点是使用了曼哈顿距离来衡量句子语义相似程度。最近在斯坦福数据集中表现较好的是 DRCN^[9]模型,该模型将前层的特征拼接到下一层,从而可以长时间保留前层的信息。

Information distilled LSTM(ID-LSTM) ^[10]模型将强化学习算法和 LSTM 结合用于文本分类任务,通过训练一个蒸馏网络将句子中非重要词蒸馏出去。为了判断蒸馏网络模型好坏,引入强化学习的策略梯度算法, LSTM 抽取后的语义特征输入到分类网络当中进行文本分类,并且使用分类网络输出的结果作为回报值来更新蒸馏网络模型。

本文基于 Siamese Network 模型,加入强化学习的方法,通过一系列句子词语蒸馏的方法,将句子中不重要的词语蒸馏出去,从而可以改善 LSTM 进行语义提取过程中对重要的词语学习不到的问题,实验结果表明该方法对中文句子有不错的效果。

1 DDPG 算法

深度确定性策略梯度算法(deep deterministic policy gradient, DDPG) ^[11]是一种强化学习算法,使用策略梯度方法^[12-13]来更新神经网络的参数。DDPG 算法有策略网络 Actor 和估值网络 Critic 两个部分组成: Actor 是动作执行者,输入的是环境特征,输出的是动作; Critic 输入的是环境特征以及策略网络输出的动作,而输出的是评判该 Actor 最终能获得总回报值的期望。同时为了解决策略更新过程中神经网络收敛不稳定情况,DDPG 算法在更新梯度的时候使用了软更新的策略,在训练过程中定义在线网络和目标网络,并不是直接将在线网络复制给目标,而是以微小的更新量更新给目标网络,这样可以使训练过程更加稳定。DDPG 网络算法步骤如下。

- 1) 初始化 Actor 和 Critic 的在线网络和目标网络。
- 2) 将在线网络参数拷贝给目标网络。
- 3) 循环以下步骤:
 - ① 在线的 Actor 会根据传入的环境 s_t 执行动作 a_t ;
 - ② 执行动作以后获得一个回报值 r_t , 并且更新新的状态 s_{t+1} ;
 - ③ 将这个过程中的 s_t, a_t, r_t, s_{t+1} 存储到一个缓冲区中;
 - ④ 互动多个过程后,从缓冲区采样然后训练 Critic 网络,采用传统 Back Propagation 的更新方式;
 - ⑤ 采用策略更新的方式更新 Actor;
 - ⑥ 采用软更新的方式将在线的参数更新到目标网络当中。

2 算法设计

2.1 模型总体结构

本文将强化学习算法和 Siamese LSTM 模型进行融合,训练出具有一定自动蒸馏句子能力的语义相似计算模型,模型的整体结构如图 1 所示。

模型的整体结构是一个强化学习的模型,类似于 DDPG 算法模型,有两个组成部分,最外围的策略网络是句子蒸馏网络模块,该网络使用多层深度神经网络模型,可以看成是一个 Actor 网络。内层的整体架构是一个 Siamese LSTM 模型,可以看成是一个 Critic 网络,Multi-LSTM 使用了两层的 LSTM 模型叠加,并且在第二层 LSTM 模型的隐藏层输出加入 Attention 模型来加权语义,最终将提取的语义向量用曼哈顿距离来表示语义相似度。模型的训练模式也和 DDPG 算法相似,Actor 网络部分和 Critic 网络部分分别训练,内部的环境模型通过反向传播算法(back propagation, BP) ^[14]来更新,外部的策略网络根据环境的损失值使用策略梯度来更新。

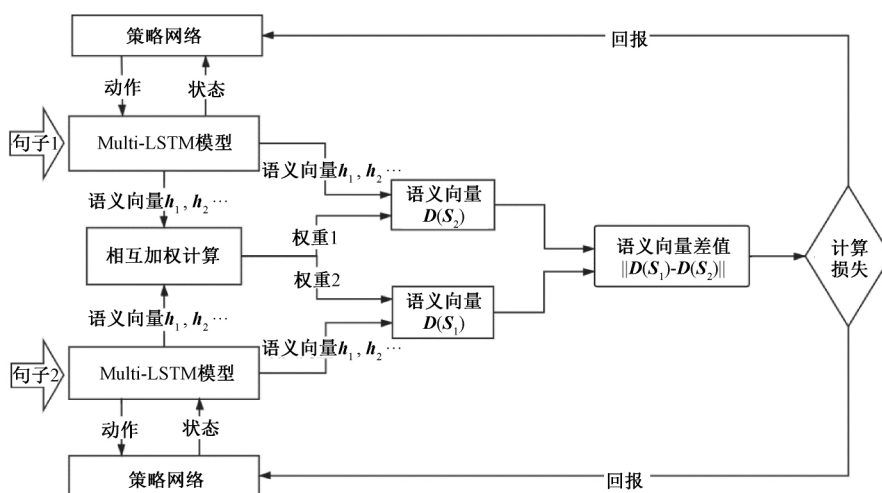


图1 模型整体结构

Figure 1 The overall architecture of the model

2.2 句子蒸馏网络模型

句子蒸馏网络是由策略网络和 Multi-LSTM 网络组成的结构,Multi-LSTM 是环境模型的一个组成部分,用于句子的语义抽取,是环境和 Actor 交互的唯一接口。Policy Network 模型就是 Actor 网络,模型的详细表述如图 2 所示。

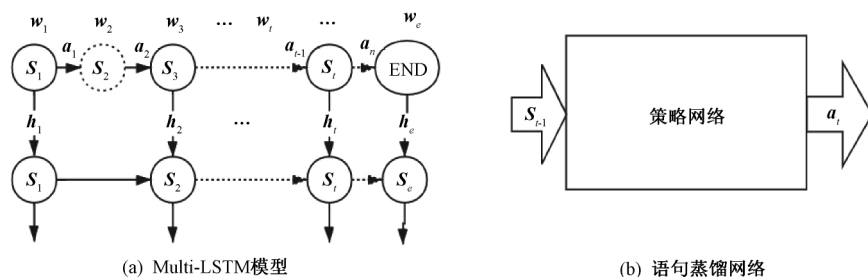


图2 句子蒸馏网络模型

Figure 2 The network model of sentence distillation

图 2(a) 是 Multi-LSTM 模型,包含两层的 LSTM 网络,图 2(b) 是 Policy Network 模型,是一个两层的神经网络模型,其中 \$\{w_1, w_2, \dots, w_t, \dots, w_e\}\$ 表示每个时刻输入句子的词向量, \$\{S_1, S_2, \dots, S_t, \dots, S_e\}\$ 表示每个时刻细胞的状态,初始时刻将 LSTM 细胞初始化为 0, \$\{h_1, h_2, \dots, h_t, \dots, h_e\}\$ 表示每个时刻 LSTM 的隐藏层输出, \$\{a_1, a_2, a_{t-1}, a_n\}\$ 表示每个时刻 Policy Network 的输出动作值,当词向量输入到 LSTM 模型当中的时候,都会先将 LSTM 细胞当前的状态和隐藏层的输出以及词向量合并成状态 \$S_t\$, \$S_t = S_{t-1} \oplus h_{t-1} \oplus w_t\$。

将第一层 LSTM 细胞当前状态和隐藏层输出以及词向量合并成状态 \$S_t\$ 后,将状态传入句子蒸馏网络判断当前传入词是否应该被保留,如果判断保留就将词向量传入第一层 LSTM 模型当中进行计算,如果判断不保留则跳过当前词。同时将采用两层的 LSTM 模型来进行语义特征提取,通过句子蒸馏网络可以将一个长句子中的非关键词去掉,从而保留句子的核心词语,使得语义相似判断效果更好。

2.3 协同注意力加权模型

协同注意力是一种相互加权的机制,是一种软注意力的加权方式。LSTM 的第二层输出每个时刻的语义信息。将这些语义信息进行相互加权,其中两个多层的 LSTM 模型的权重可以根据输入的文本类型来决定是否共享参数。模型的结构如图 3 所示。第二层的 \$\{H_1, H_2, \dots, H_t, \dots, H_e\}\$ 输出状态传入加权模型当中进行加权。协同注意力模型的内部加权方式有很多种,比如可以直接将两个多层 LSTM 模型的最高层生成的每个时刻语义矩阵直接进行相乘,然后经过 Softmax 函数来生成权重信息,也可以用额外的语义矩阵进行加权,如图 4 所示的加权方式,其中 \$H_1\$ 和 \$H_2\$ 表示 LSTM 的第二层隐藏层输出拼接成的语义矩阵, \$W_s\$ 是一个 \$L \times L\$ 加权矩阵, \$H_1\$ 是一个 \$L \times n\$ 形状的矩阵, \$H_2\$ 是一个 \$L \times m\$ 形状的矩阵,进行矩阵操作运算:

$$W_1 = \text{softmax}(\text{sum}(H_1^T W_s H_2)), W_2 = \text{softmax}(\text{sum}(H_1^T W_s H_2)^T)。$$

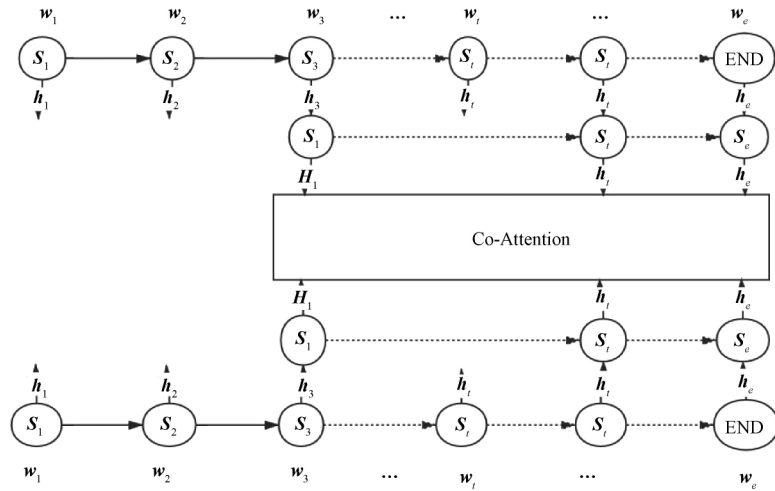


图 3 Co-Attention 加权模型

Figure 3 The Co-Attention weighted model

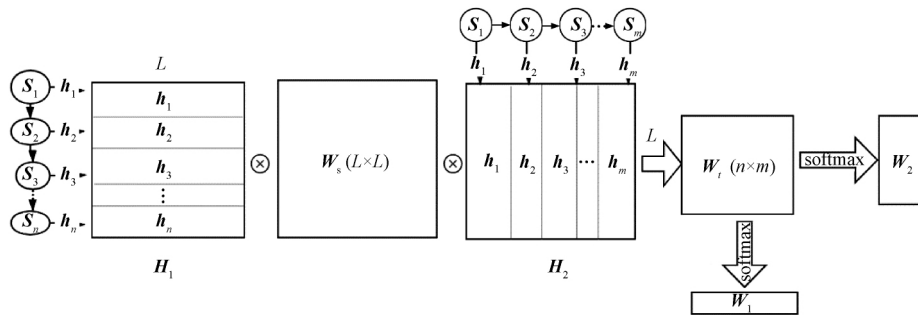


图 4 Co-attention 加权方式

Figure 4 The Co-attention weighted method

经过运算可以获得一个 $n \times m$ 的权重矩阵,然后将每行的参数同样相加,并将每列的参数相加,经过 Softmax 函数进行归一化后,可以分别获得 H_1 语义矩阵和 H_2 语义矩阵的每个时刻分别对应的语义权重向量,最后将各自的权重向量和语义矩阵相乘可以获取最后句子的语义向量。

单个的加权矩阵往往会加权语义矩阵中的某一个方面,我们为了获取语句更丰富的语义信息,使用多个加权矩阵来对 LSTM 每个时刻输出的语义进行加权,生成多个权重向量。为了避免多个权重矩阵最终生成的权重矩阵相同从而失去丰富性,在最终的函数当中会定义一个正则项,正则项的推导如下。

假设我们定义了 n 个加权矩阵 $W_{s1}, W_{s2}, \dots, W_{sn}$,用这 n 个加权矩阵来对语义矩阵 H_1 和 H_2 进行语义加权,按照相互加权的矩阵运算公式 $W_t = \text{softmax}(\text{sum}(H_1^T W_s H_2))$,最终我们可以获得 n 个权重矩阵 $W_{t1}, W_{t2}, \dots, W_{tn}$,为了使得这 n 个加权矩阵能够加权语义矩阵不同方面的语义特征,我们希望这 n 个权重矩阵能够尽可能不相同,因此我们对这 n 个权重矩阵进行加和求平均值,记作 \bar{W}_t ,为了保证任意两个权重矩阵之间的距离尽可能大,从而引用降维线性判别分析算法(linear discriminant analysis, LDA)中的类间散度思想来对这个问题进行求解,将 \bar{W}_t 看作是一个中心的权重矩阵,则我们的原问题可以化为任意一个权重矩阵距离中心权重矩阵的距离尽可能的大,因此可以定义正则项 $L = \sum_{i=1}^n W_{ti} - \bar{W}_t$,通过最大化这个正则项,也就是最小化倒数保证每个权重矩阵尽可能的不相同,从而可以保证每个权重矩阵都能够抽取不同层次的语义信息。

2.4 损失函数设计

首先采用曼哈顿距离来衡量两个语义向量之间的距离,公式为 $d = \exp(-\|h_a - h_b\|)$, $d \in [0, 1]$,其中

h_a 和 h_b 表示最终的语义向量。同时本文将采用对比损失函数作为最终损失函数,对比损失函数的公式为 $l = \frac{1}{2N} \sum_{n=1}^N y d^2 + (1 - y) \max(\text{margin} - d, 0)^2$, 其中: y 代表输入到模型的样本标签语义是否相似,取值为 $\{0, 1\}$; margin 表示一个设置的阈值。对比损失函数可以很好表达成对样本的匹配程度,当样本标签为 1 时,也就是样本相似时候,对比损失函数只有加号前面的一项,当原本相似的样本在抽象成语义向量的空间距离比较大的时候,损失函数会变大。当样本标签为 0 时,对比损失函数只有加号的后面一项,当样本不相似,而抽象出来的距离很小则损失函数变大。

2.5 模型训练设计

算法整体的流程和 DDPG 算法流程类似,有在线和目标网络两个部分,依据策略更新的方式更新网络。模型在训练过程中不容易收敛,会因为参数大小或者参数的递减策略设置不正确等原因使整个模型达不到理想的效果。为了使训练过程更加稳定,我们将使用高策略的更新方式来更新网络。高策略更新方式定义在线和目标两组网络,两组网络的结构完全相同,但是更新时间不同。在每个批量训练之前要将目标网络的参数赋值给在线网络,然后在线网络作为训练网络来参与整体的训练,并且在一个批量当中进行每个样本的更新,当一个批量训练完毕后,要将在在线网络当中的参数更新到目标网络上,更新方式使用的是软方式,即设置一个参数 β ,则最终更新到目标网络上的参数为 $\text{Target}_\theta = (1 - \beta) \text{Target}_\theta + \beta \text{Online}_\theta$ 。

最后在下个批量训练开始之前,将目标网络的参数再次赋值给在线网络进行下一轮的训练,同时本文将采用曼哈顿距离来表示句子之间相似性,损失函数采用对比损失值作为训练。

在训练过程中还用到一些训练技巧,比如在强化学习训练过程中,当要学习的环境很复杂的情况下,开始训练收敛过程会很慢,或者很可能不会收敛,因此会有一个预训练的过程。预训练过程可以认为是模型学习环境的一个初始的合理参数的设置过程。对于句子蒸馏网络的预训练部分,由于本文后续实验采用数据的中文词汇特殊性(句子的前几个词非常重要),一般将句子的前几个词组保留,后面的词组以一定的概率随机蒸馏出去。

3 数据与实验分析

3.1 实验数据

本论文的实验数据使用的是网络爬取的数据,包括一整套汽车名称数据、汽车的配件信息以及售后信息等,用户同样会提供他们收集的汽车信息的数据库,我们要将这两个数据库的信息进行整合,使得相同型号的汽车信息能被整合到一起。但是用户提供的汽车名称和本文数据库中的汽车名称不完全相同,如表 1 所示,该表是部分本文标注好的数据,右边是本文数据的命名标准,左边是用户的数据库。可能会有型号上的描述不一样,通过语义相似计算的方法将用户提供的名称和数据库中的车辆名称做一个相似性匹配,从而确定是我们数据库当中的哪个型号的车,然后将所有的数据进行整合。实验数据有 6 万多对已经标注的配对的数据,训练和测试数据集比例为 5 : 1。在生成训练数据过程中要 1 : 2 随机生成负样本,对于每一对标注的语句对,随机从样本数据中选择非配对的句子作为负样本。

表 1 数据命名标准

Table 1 The standard of naming data

用户数据库汽车描述信息	本文数据库汽车描述信息
东风大力神 EQ4186L 半挂牵引汽车	东风 EQ4186L 半挂牵引汽车
东风天龙 DFL5250GJBA 混凝土搅拌运输车	东风 DFL5250GJBAX1 混凝土搅拌运输车
进口奔驰 BENZ SLK200K 1.8 AT(04-11)	奔驰 BENZ SLK200K 跑车 奔驰 SLK 2010 款 SLK 200K
进口宝马 BMW E46 323Ci 2.5 AT	宝马 BMW 323Ci 轿车
一汽大众速腾 FV7206 2.0 MT(200604-201203) 速腾 2.0L MT 舒适型	速腾 FV7206E 轿车 速腾 2.0L 手动挡 舒适型
东风雪铁龙 世嘉 DC7205DB 2.0L MT 两厢 夺冠版(200904) 世嘉 2009 款 两厢 2.0L 手动 夺冠版	东风雪铁龙 DC7205DB 轿车 世嘉 2009 款 两厢 2.0L 手动 夺冠版

3.2 词向量训练

本文使用 gensim 工具来训练 Word2vec 的词向量, gensim 是一个 python 库, 能实现很多常用自然语言处理的算法, 比如 latent semantic analysis(LSA)、latent dirichlet allocation(LDA) 等, 首先使用数据库当中的所有汽车作为语料库来训练词向量, 总量大概有 1 000 多万条数据。对数据库当中的词进行一些特殊符号及格式除杂后, 使用 jieba 分词工具对汽车描述名称进行分词, 然后使用 gensim 工具进行中文的词向量训练。

3.3 多加权模型实验与结果分析

3.3.1 模型的训练 本次实验是验证多加权协同注意力模型有效性, 该模型主要在 Siamese LSTM 上实现, 加权方式会定义额外的加权矩阵进行语义的加权, 具体结构参考 2.3 节。首先是 3 个 Co-attention 模型, 3 个模型的不同点是加权矩阵的数量不同。训练的两个模型的参数如表 2 所示。

两个模型都使用 Siamese LSTM 架构和两层的 LSTM 来计算输入句子的语义信息。输入句子长度大于 30 的句子将被采用截断处理。输入句子的长度小于 30 的句子使用特殊字符填充。每次输入一个 Batch 进行训练。句子之间的相似度采用曼哈顿距离来表示, 损失函数使用对比损失函数来计算。最后使用 Adam 优化算法^[15]来调整参数。

3.3.2 实验结果分析 本次实验主要是验证多加权的注意力模型效果。试验结果表明: 当加权矩阵从 1 个矩阵变成 3 个矩阵, 并且使用一定正则项来训练以后, 准确率有明显提升, 从 0.85 提升到 0.87; 但是从 3 个加权矩阵上升到 10 个加权矩阵的时候, 实验效果提升的幅度明显小很多。由此我们可以看出, 多加权的注意力模型相比于单个加权模型来说有一定的提升效果, 但是当加权矩阵过多的时候, 提升就很小, 同时也会增加很多计算量。

表 2 Co-attention 模型参数

Table 2 The parameters of the Co-attention model

参数名称	参数含义	参数数值
Batch_size	每次训练获取的语句对数	20 对
Seq_length	模型输入句子的最大长度	30
Embedding_dim	输入词向量的维数	200
Dropout_keep_prob	Dropout 概率	1.0
Hidden_units	LSTM 隐藏层维度	100
Att_matrix_num	加权矩阵的数量	1/3/10 个
Margin	对比损失函数参数	1.0
Lr	初始学习率	0.01
Optimizer	优化函数	Adam

3.4 基于句子蒸馏语义计算模型实验分析

本次实验选取测试数据中用户数据库命名的数据作为原始数据, 设置阈值为 0.5, 对于每个用户的汽车名称, 和我们数据库的汽车名称作相似性计算, 大于 0.5 的作为相同车名备选项, 然后把相似值排序后选出最相似的一个作为最终的相似车型名称。本次实验使用 Siamese LSTM 模型作为对比模型, 最终试验结果在测试数据集下, 我们提出的基于强化学习的模型相较于 Siamese LSTM 模型准确率从 91% 上升到 95.7%。

由于训练数据不够充分, 所以训练出来的蒸馏模型对于部分句子蒸馏出的信息比较多, 比如必要的汽车名称或者一些车型的具体型号被蒸馏出去, 导致相同车型的车名称计算相似性时值会很小, 从而被丢弃, 所以召回率会有一些损失, 从 100% 下降到 96%。但是蒸馏模型能够蒸馏出去一些非必要的词语, 比如像“商务型”等修饰词, 可以使得语义抽取更加简练, 能够突出重点信息。而非蒸馏的模型, 可能会受汽车名称中过多的无效或者是错误的修饰性词语的干扰, 比如我们用户数据库有一个汽车为“骏捷 SY7182UZ 1.8T AT (200603) 骏捷 1.8T AT 舒适型轿车”, 我们的车型库的数据里有“中华 SY7182UZ 轿车 骏捷 1.8T 自动挡 尊贵型”和“中华 SY7182US 轿车 骏捷 1.8T 手动挡 舒适型”, 在 Siamese LSTM 模型计算下一个会比前一个相似性还高, 这样的情况在样本数据中大量存在。但是判断是不是同一辆车一般只需要车名和紧接着车名后面的车型号就能唯一识别出来, 因此蒸馏模型可以一定程度地将后面修饰的蒸馏出去, 因此会增加模型判断的准确率。

表3为句子蒸馏前后的效果,其中第1列是原来的汽车名称,第2列是经过分词工具分词后去除特殊符号后的模型输入数据,第3列的是蒸馏后的效果。虽然训练数据不足,但是可以看出句子蒸馏模型会对句子中一些不太重要的修饰性词语蒸馏出去,基本上会保留主要的汽车名称以及必要的汽车统一型号名称,这些信息是识别汽车唯一性的充要条件。但也会导致比如排量、年份等信息丢失,这与具体的训练数据不足有关。

表3 句子蒸馏效果

Table 3 The sentence distillation results

原语句	分词后句子	蒸馏后句子
上汽通用雪佛兰 乐骋 SGM7160AT 1.6L AT 豪华型(200801) 乐骋 2008 款 1.6L SX 自动	上汽 通用 雪佛兰 乐骋 SGM7160AT 1.6 L AT 豪华型 200801 乐骋 2008 款 1.6 L SX 自动	上汽通用雪佛兰乐骋 SGM7160AT 乐骋 2008 款自动
一汽大众速腾 FV7206 2.0 MT (200604 - 201203) 速腾 2.0L MT 舒适型	一汽大众 速腾 FV7206 2.0 MT 200604 - 201203 速腾 2.0 L MT 舒适型	一汽大众速腾 FV7206 速腾 MT
进口宝马 BMW E46 323Ci 2.5 AT	进口 宝马 BMW E46 323Ci 2.5 AT	进口宝马 BMW
进口奔驰 BENZ SLK200K 1.8 AT(04-11)	进口 奔驰 BENZ SLK200K 1.8 AT 04 - 11	进口奔驰 BENZ
广汽本田奥德赛 HG6482BAC4A 2.4L CVT 尊享版(201408) 奥德赛 2015 款 2.4L 尊享版	广汽 本田 奥德赛 HG6482BAC4A 2.4 L CVT 尊享 版 201408 奥德赛 2015 款 2.4 L 尊享 版	广汽本田奥德赛 HG6482BAC4A 奥德赛 2015 款尊享

4 总结

本文将深度学习算法和强化学习算法结合起来研究,通过使用强化学习算法来改善 LSTM 模型提取语义时可能的语义丢失现象,在语义相似计算模型 Siamese Network 上取得了很好的效果。但是本文的模型仍然存在一些问题:由于模型是采用强化学习方式训练,LSTM 提取的语义很复杂,模型要想收敛比较好就需要大量的采样,从而大大增加训练的时间,否则很容易陷入局部最优;另外,测试使用的场景和数据有限,只是在比较小的项目数据集上测试,对于很多其他场景的应用没有测试。

参考文献:

- [1] 许飞翔,叶霞,李琳琳,等. 基于 SA-BP 算法的个体概念语义相似度综合计算[J]. 计算机科学,2020,47(1): 199-204.
XU F X, YE X, LI L L, et al. Comprehensive calculation of semantic similarity of ontology concept based on SA-BP[J]. Computer science, 2020, 47(1): 199-204.
- [2] 张克亮,李芊芊. 基于本体的语义相似度计算研究[J]. 郑州大学学报(理学版),2019,51(2): 52-59.
ZHANG K L, LI Q Q. A survey of ontology-based semantic similarity measurement[J]. Journal of Zhengzhou university(natural science edition), 2019, 51(2): 52-59.
- [3] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[C]//The 1st International Conference on Learning Representations. Scottsdale, 2013: 1-12.
- [4] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-80.
- [5] MIKOLOV T, SUTSEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[J]. Advances in neural information processing systems, 2013, 26: 3111-3119.
- [6] CHOPRA S, HADSELL R, LECUN Y. Learning a similarity metric discriminatively, with application to face verification[C]//The IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, 2005: 539-546.
- [7] NECULOU P, VERSTEEGH M, ROTARU M. Learning text similarity with Siamese recurrent networks[C]//Proceedings of the 1st Workshop on Representation Learning for NLP, Berlin, 2016: 148-157.
- [8] MUELLER J, THYAGARAJAN A. Siamese recurrent architectures for learning sentence similarity[C]//The 30th AAAI Confer-

- ence on Artificial Intelligence. Phoenix ,2016: 2786–2792.
- [9] KIM S ,KANG I ,KWAK N. Semantic sentence matching with densely-connected recurrent and co-attentive information [C]//The 33th AAAI Conference on Artificial Intelligence. Honolulu ,2019: 6586–6593.
- [10] ZHANG T Y ,HUANG M L ,LI Z. Learning structured representation for text classification via reinforcement learning [C]//The 32th AAAI Conference on Artificial Intelligence. New Orleans ,2018: 1–8.
- [11] LILLICRAP T P ,HUNT J J ,PRITZEL A ,et al. Continuous control with deep reinforcement learning [J]. Computer science ,2015 ,8(6) : A187.
- [12] SUTTON R ,MCALLESTER S ,SINGH D A ,et al. Policy gradient methods for reinforcement learning with function approximation [J]. Advances in neural information processing systems ,2000 ,12: 1057–1063.
- [13] MNIH V ,BADIA A P ,MIRZA M ,et al. Asynchronous methods for deep reinforcement learning [C]//International Conference on Machine Learning. New York ,2016: 1–8.
- [14] RUMELHART D E ,HINTON G E ,WILLIAMS R J. Learning representations by back-propagating errors [J]. Nature ,1986 ,323(6088) : 533–536.
- [15] KINGMA D P ,BA J. Adam: a method for stochastic optimization [C]//The 3rd International Conference on Learning Representations. San Diego ,2015: 1–15.

A Text Similarity Semantic Computing Model Based on Deep Reinforcement Learning

CHEN Guanlin^{1 2} ,SHI Xiaolong^{1 2} ,ZHOU Liang³ ,WENG Wenyong¹

(1. School of Computer and Computing Science ,Zhejiang University City College ,Hangzhou 310015 ,China;

2. College of Computer Science ,Zhejiang University ,Hangzhou 310027 ,China;

3. Hangzhou Big Data Management Service Center ,Hangzhou 310020 ,China)

Abstract: Semantic similarity computing was a common problem in the field of natural language processing. Most of the existing semantic similarity computing models extracted semantic features based on CNN or LSTM. There was semantic information loss problem with this way of extracting semantic features. Two points were proposed to improve traditional deep learning models. Firstly attention weight model using multiple weighted matrices was improved to weight semantics. And a new method was proposed to calculate weight. Secondly reinforcement learning method in the semantic similarity calculation model was used to improve the semantic loss phenomenon. The reinforcement learning algorithms were studied in the most commonly Siamese Network model ,which could improve the semantic loss of LSTM models when extracting the semantics of sentences. The experiments showed that the improved method had a good effect in processing Chinese sentences.

Key words: deep learning; semantic similarity; reinforcement learning

(责任编辑: 方惠敏)