

# 一种基于 Tree-LSTM 的句子相似度计算方法

杨萌 李培峰<sup>†</sup> 朱巧明

苏州大学计算机科学与技术学院, 苏州 215006; <sup>†</sup> 通信作者, E-mail: pfli@suda.edu.cn

**摘要** 在浅层句法树和依存关系树的基础上, 提出两种结构化特征: 基于短语的浅层句法树 NPST 和基于短语的依存树 NPDT, 并将它们与 Tree-LSTM 模型相结合, 进行句子相似度计算。实验表明, 使用结构化特征和 Tree-LSTM 会带来性能的提升。

**关键词** 句子相似度计算; Tree-LSTM; 结构化特征

**中图分类号** TP391

## An Approach of Sentence Similarity on Tree-LSTM

YANG Meng, LI Peifeng<sup>†</sup>, ZHU Qiaoming

Department of Computer Science and Technology, Suzhou University, Suzhou 215006; <sup>†</sup> Corresponding author, E-mail: pfli@suda.edu.cn

**Abstract** Based on the shallow tree and dependency tree, the authors introduce the structural representations, NPST (new phrase-based shallow tree) and NPDT (new phrase-based dependency tree) to Tree-LSTM to compute sentence similarity. Experimental results manifest that the proposed approach achieves a higher performance than the baseline.

**Key words** sentence similarity computation; Tree-LSTM; structural representations

相似度计算是自然语言处理的基础工作。句子相似度计算的目标是学习一个得分系统, 给定一对句子, 该系统返回相似度得分。例如, 分数范围为 0~5, 0 代表这对句子含义完全无关, 5 代表含义相同。大多数句子相似度计算方法是将句子对视为一个平面特征向量, 每个特征是某特定方面(词汇、句法和语义等)相似度计算的结果。这种方法存在局限性, 用平面特征向量代表文本对的相似度, 表征性较弱。随着神经网络方法的再度流行, 相关研究表明, Tree-LSTM (long short-term memory) 方法可以充分利用结构化特征得到更多信息<sup>[1-2]</sup>。

### 1 相关工作

目前文本相似度计算方法主要有 4 类: 基于词重叠的方法、基于语料库统计的方法、基于语言学的方法和混合方法。

基于词重叠的方法是通过一个文本对共有的一些词汇, 来计算文本的相似度。Bank 等<sup>[3]</sup>提出 Jaccard 相似系数法, 该方法通过两个文本中词语交集与词语并集的比值, 计算文本的相似度。Metzler 等<sup>[4]</sup>用逆文档频率(IDF)作为两个文本中均出现词语的权重, 改进计算结果。Banerjee 等<sup>[5]</sup>基于短语的长度及使用频率呈 Zipfian 分布的特点, 设计文本相似度计算方法。

基于语料库的方法是将文本对中出现的词语集合作为特征集, 将基于语料库的向量的余弦夹角值作为相似度。Landauer 等<sup>[6]</sup>通过分析一个大型的自然语言语料库, 统计关键词的 TF-IDF 值, 形成文本语义向量, 利用向量的余弦夹角计算文本的语义相似度。Lund 等<sup>[7]</sup>统计词汇之间的共现性, 得到高维向量空间, 计算文本相似度。Allan 等<sup>[8]</sup>提出的 TF-IDF 法是用 TF-IDF 计算单词权重的语义相似度的

国家自然科学基金(61472265, 61772354)资助

收稿日期: 2017-07-18; 修回日期: 2017-11-24; 网络出版日期: 2017-12-05

方法。

基于语言学的方法是利用词汇间的语义关系及语法成分来确定文本的相似度。Mihalcea 等<sup>[9]</sup>基于词语语义相似度度量文本间的相似度,并考虑单词的区分能力,进行文本相似度的计算。Malik 等<sup>[10]</sup>将组成文本对的词之间的相似度总和的最大值被文本长度归一化后的所得值作为文本相似度值。

混合方法是将以上方法混合起来的方法。Yang 等<sup>[11]</sup>将结构化特征与平面特征结合起来,在支持向量回归(support vector regression, SVR)模型中训练,得到句子相似度计算结果。实验结果表明,加入结构化特征可提高句子相似度计算性能,对较短的句子,效果尤为明显。

本文在 Yang 等<sup>[11]</sup>的结构化特征方法基础上,对其进行优化。将句子对处理成两种基本结构化表示:NPST (new phrase-based shallow tree)和 NPDT (new phrase-based dependency tree),并使用 Tree-LSTM 进行相似度计算。

## 2 基于 Tree-LSTM 的句子相似度计算

本文使用 Yang 等<sup>[11]</sup>提出的浅层句法树和依存关系树作为基本结构化表示,并分别对这两种结构进行优化(如增加句法结构信息、删除非重要信息等),得到基于短语的浅层句法树 PST 和基于短语的依存树 PDT,以期更好地表示句子句法和语义等信息。

### 2.1 使用结构化特征与 Tree-LSTM 的动机

句法结构等结构化信息在自然语言处理中非常重要。Yang 等<sup>[11]</sup>的实验结果表明,结构化特征能提高相似度计算性能,尤其对短句子的性能提高更大。但是, Yang 等<sup>[11]</sup>是通过支持向量回归 SVR 模型进行相似度计算,由于 SVR 中的树核函数仅计算两个树形结构中相同子树的个数,没有充分利用句子的语义等信息,所以实验结果中结构化特征对长句子性能的提升效果不明显。

LSTM 是一种具有更复杂计算单元的循环神经网络(recurrent neural networks, RNN),能够解决 RNN 的后面时间节点对于前面时间节点感知力下降的缺点。然而, LSTM 只能探索线性链结构的信息,基于此, Tai<sup>[1]</sup>提出 Tree-LSTM 方法(LSTM 树形结构的一种推广),可以使用句法树等树形结构,与 SVR 的树核函数相比,能更好地从结构化特征中得到语义等信息。

在此基础上,本文探索可应用于 Tree-LSTM 的结构化特征,以期提高句子相似度计算的性能。

### 2.2 句子相似度计算模型

首先,将一个文本对表示成  $h_L$  和  $h_R$ ,二者均为对应文本的 PDT 或 PST 对应 Tree-LSTM 的根节点  $o$  的隐藏状态  $h_o$ 。将  $h_L$  和  $h_R$  的距离和角度输入单层神经网络,获得  $h_s$ ,通过 softmax 进行归一化处理,得到概率  $p$ ,进而获得一对文本的相似度得分  $\tilde{y}$ 。

$$h_x = h_L \odot h_R, \quad (1)$$

$$h_+ = |h_L - h_R|, \quad (2)$$

$$h_s = \sigma(W^{(x)}h_x + W^{(+)}h_+ + b^{(h)}), \quad (3)$$

$$\hat{P}_\theta = \text{softmax}(W^{(p)}h_s + b^{(p)}), \quad (4)$$

$$\tilde{y} = r^T \hat{P}_\theta, \quad (5)$$

其中,  $r^T = [1, 2, \dots, 5]$ 。

本模型的目标是使基于分布  $\tilde{P}_\theta$  的相似度计算得分与人工评分更加接近,即  $\tilde{y} = r^T \tilde{P}_\theta \approx y$ 。本文定义稀疏目标分布  $p$  满足  $y = r^T p$ ,

$$p_i = \begin{cases} y - \lfloor y \rfloor, & i = \lfloor y \rfloor + 1, \\ \lfloor y \rfloor - y + 1, & i = \lfloor y \rfloor, \\ 0, & \text{其他}, \end{cases} \quad (6)$$

其中,  $1 \leq i \leq 5$ 。

损失函数  $J(\theta)$  是  $P$  与  $\hat{P}_\theta$  的 KL 散度:

$$J(\theta) = \frac{1}{m} \sum_{k=1}^m \text{KL} \left( P^{(k)} \parallel \hat{P}_\theta^{(k)} \right) + \frac{\lambda}{2} \|\theta\|_2^2, \quad (7)$$

$m$  表示训练句子对的个数,  $k$  表示第  $k$  个句子。

一对文本的相似度计算过程如图 1 所示。首先将文本 1 和文本 2 中所有的单词表示成语义实数向量;然后将文本对表示为结构化特征 NPST 或 NPDT,使用对应的 Child-Sum Tree-LSTM 或 N-ray Tree-LSTM 模型进行计算,获得根节点的隐藏状态  $h_L$  和  $h_R$ ;最后通过  $h_L$  和  $h_R$ ,使用文本相似度计算模块进行计算,获得相似度计算结果。

### 2.3 NPDT 和 Child-Sum Tree-LSTM

依存关系树 DT (dependency tree)可以表示句子中词语间的依存关系。Yang 等<sup>[11]</sup>使用基于短语的依存关系树(phrase-based dependency tree)来获得更多句法信息。图 1 为文本句子相似度的计算过程。

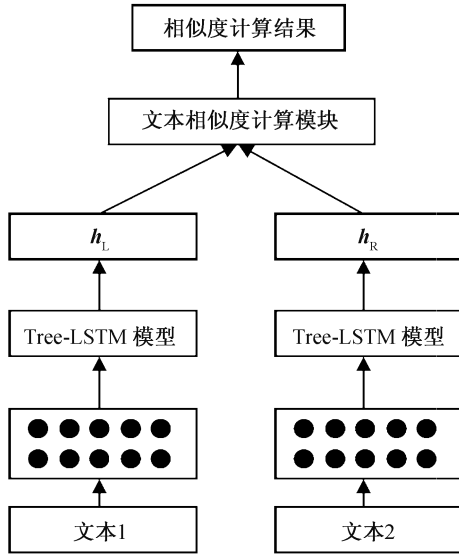


图 1 文本相似度的计算过程  
Fig. 1 Similarity calculation process of text

对于 Tree-LSTM 模型, 单词节点需要输入对应的语义实数向量, 语法节点因无法将语法信息表示为语义实数向量而无需输入语义向量。由于 PDT 中每个节点都有可能是单词节点或语法节点, 因此无法确定哪个节点可以接受单词语义实数向量, 哪个节点不可以。所以, Tree-LSTM 模型的计算难以实现。

本文将属于同一短语的单词节点连接到同一个依存关系头结点上, 得到新的基于短语的依存关系树 NPDT (NewPDT)。由于 NPDT 不考虑短语内部的依存关系, 只考虑短语间的依存关系, 可以保证

每个节点均为单词节点, 更利于获得语义信息, 所以 NPDT 可应用于 Tree-LSTM 模型。图 2 是将文本 “a man plays the keyboard with his nose” 依存关系树转换为 NPDT 的例子。

本文用 Child-Sum Tree-LSTM 表示 PDT, 方法如下:

$$\tilde{h}_j = \sum_{k \in C(j)} h_k,$$

$$i_j = \sigma(W^{(i)}x_j + U^{(i)}\tilde{h}_j + b^{(i)}),$$

$$f_{jk} = \sigma(W^{(f)}x_j + U^{(f)}h_k + b^{(f)}),$$

$$o_j = \sigma(W^{(o)}x_j + U^{(o)}\tilde{h}_j + b^{(o)}),$$

$$u_j = \tanh(W^{(u)}x_j + U^{(u)}\tilde{h}_j + b^{(u)}),$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k,$$

$$h_j = o_j \odot \tanh(c_j),$$

其中, 孩子节点和 Tree-LSTM 单元表示为一个向量的集合: 输入门  $i_j$ 、遗忘门  $f_{jk}$ 、输出门  $o_j$ 、记忆单元  $c_j$  和隐藏状态  $h_j$ 。对于节点  $j$ ,  $C(j)$  表示节点  $j$  的孩子集合,  $k \in C(j)$ 。  $x_j$  表示节点  $j$  的输入向量 (文本中为树形结构中每个节点对应单词的语义向量表示),  $\sigma$  表示 sigmoid 函数,  $\odot$  表明向量元素依次相乘。  $W$  和  $U$  均为权重矩阵,  $b$  为阈值向量。  $W$ ,  $U$ ,  $b$ ,  $h$  中的所有元素均初始化为 -1.0~1.0 之间的随机数。

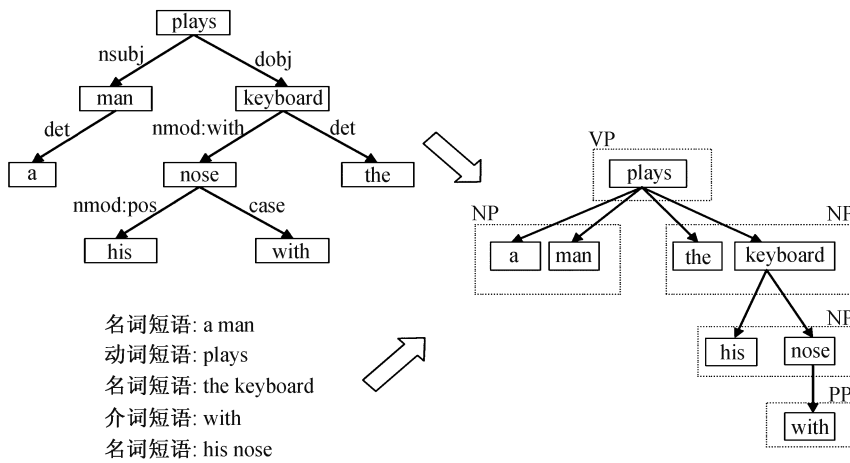


图 2 基于短语的依存关系树构造方法  
Fig. 2 Construction method of dependency tree based on phrase

直观地看,可以认为上述公式中权重矩阵的值表示 Child-Sum Tree-LSTM 节点为内所有向量、输入向量  $\mathbf{x}_j$  和孩子单元的隐层状态  $\mathbf{h}_k$  之间的关系。例如,当 Child-Sum Tree-LSTM 单元得到重要语义单词(例如动词)输入时,模型可以学习到对应于输入门  $i_j$  的参数值接近 1。如果输入为不重要语义单词(例如定冠词),对应参数值则接近 0。

Child-Sum Tree-LSTM 单元的状态由其所有的孩子隐藏状态向量  $\mathbf{h}_k$  决定。Child-Sum Tree-LSTM 适合于有高分支数或者孩子节点无序的树形结构, NPDT 中可能有许多依存关系涉及同一个单词,即某节点拥有很多孩子节点,且这些孩子节点是无序的,适合使用 Child-Sum Tree-LSTM。在使用 NPDT 时,  $\mathbf{h}_L$  和  $\mathbf{h}_R$  均表示为对应文本的 Child-Sum Tree-LSTM 中根节点  $o$  的隐藏状态  $\mathbf{h}_o$ 。

## 2.4 浅层句法树

ST 的基本形式是深度为 3 的树,如 Yang 等<sup>[11]</sup>所述。转换方法如下:先将所属同一个句法成分的词节点及其词性节点组织到同一个节点(chunker),再通过根节点组织起来,最后,在结构化表示中裁剪掉定冠词和连词及其父亲节点(词性节点),以减少非重要信息对计算结果的影响,最终获得 PST。与 Yang 等<sup>[11]</sup>构造的 PST 不同,本文构造的 NPST (NewPST)减少了加入语义信息的步骤。图 3 为文本“the girl sing into a microphone”的 NPST 结构化表示。

将 PST 转换成 NPST 的原因是, PST 的叶子节点为单词节点,其余节点为句法和词性节点。对于 Tree-LSTM 模型, PST 树中仅叶子节点可以接受语

义向量,其他节点加入的语义信息(如命名实体识别结果等)无法转化为语义实数向量,所以该步骤无意义。

对于 NPST,本文使用 2-ray Tree-LSTM。2-ray Tree-LSTM 与 Child-Sum Tree-LSTM 基本上相同,区别在于 2-ray Tree-LSTM 适用于每个节点孩子数最多为 2 的树形结构,并且孩子节点有序。对于每个节点  $j$ ,其第  $k$  个孩子节点的隐层状态和记忆单元分别为  $\mathbf{h}_{jk}$  和  $\mathbf{c}_{jk}$ 。2-ray Tree-LSTM 的表示方法如式(8)~(13)所示。

$$\mathbf{i}_j = \sigma \left( \mathbf{W}^{(i)} \mathbf{x}_j + \sum_{\ell=1}^2 \mathbf{U}_{\ell}^{(i)} \mathbf{h}_{j\ell} + \mathbf{b}^{(i)} \right), \quad (8)$$

$$\mathbf{f}_{jk} = \sigma \left( \mathbf{W}^{(f)} \mathbf{x}_j + \sum_{\ell=1}^2 \mathbf{U}_{k\ell}^{(f)} \mathbf{h}_{j\ell} + \mathbf{b}^{(f)} \right), \quad (9)$$

$$\mathbf{o}_j = \sigma \left( \mathbf{W}^{(o)} \mathbf{x}_j + \sum_{\ell=1}^2 \mathbf{U}_{\ell}^{(o)} \mathbf{h}_{j\ell} + \mathbf{b}^{(o)} \right), \quad (10)$$

$$\mathbf{u}_j = \tanh \left( \mathbf{W}^{(u)} \mathbf{x}_j + \sum_{\ell=1}^2 \mathbf{U}_{\ell}^{(u)} \mathbf{h}_{j\ell} + \mathbf{b}^{(u)} \right), \quad (11)$$

$$\mathbf{c}_j = \mathbf{i}_j \odot \mathbf{u}_j + \sum_{\ell=1}^2 \mathbf{f}_{j\ell} \odot \mathbf{c}_{j\ell}, \quad (12)$$

$$\mathbf{h}_j = \mathbf{o}_j \odot \tanh(\mathbf{c}_j), \quad (13)$$

其中  $\ell$  表示该节点的第  $\ell$  个孩子节点,  $k=1, 2$ 。

由于 2-ray Tree-LSTM 的左、右孩子节点有序,本文将 2-ray Tree-LSTM 用于二叉树化的 NPST。当且仅当该节点  $j$  为叶子节点时,接受输入向量  $\mathbf{x}_j$ 。在使用 NPST 时,  $\mathbf{h}_L$  和  $\mathbf{h}_R$  均表示为对应文本的 2-ray Tree-LSTM 根节点  $o$  的隐藏状态  $\mathbf{h}_o$ 。

在式(8)~(13)中,对每个孩子节点  $k$  分别使用参数矩阵,使 2-ray Tree-LSTM 可以比 Child-Sum Tree-LSTM 更细粒度地学习孩子节点的状态。例如,将一个 NPST 应用于 2-ray Tree-LSTM,这个 NPST 中某个节点的左孩子对应一个名词单词的输入,右孩子对应一个动词单词的输入。在动词单词比名词词组包含语义更强的情况下,  $\mathbf{U}_{k\ell}^{(f)}$  可以通过训练使对应于  $f_{j1}$  的参数接近 0,表示“忘记”该单词,使对应于  $f_{j2}$  的参数接近 1,表示“记住”该单词。

式(9)中,对节点  $j$  第  $k$  个孩子节点的遗忘门  $f_{jk}$ ,定义非对角线参数矩阵  $\mathbf{U}_{k\ell}^{(f)} (k \neq \ell)$ 。这个参数矩阵使得二叉化的 NPST 的左孩子节点的隐层状

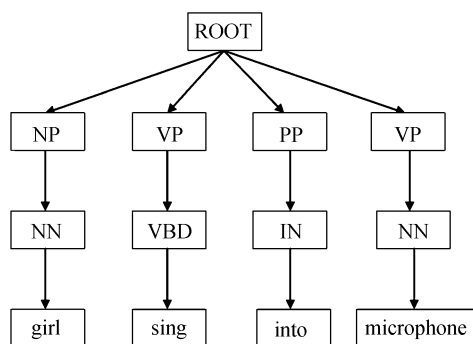


图 3 在浅层句法树上加入短语节点并修剪非重要节点得到 PST

Fig. 3 PST by adding phrase nodes and cutting non-important nodes in the shallow tree

态对右孩子节点的遗忘门有兴奋或抑制的作用，使信息从孩子节点到父亲节点的传播更灵活。

3 实验

3.1 实验设置

本文的基准系统是 Yang 等<sup>[11]</sup>将平面特征与树形特征相结合，使用 SVR 进行相似度学习的系统。该系统共使用 40 个平面特征，来自 SemEval 会议 2012 年评测任务“文本语义相似度(STS)”中两个最佳系统 Takelab<sup>[8]</sup>、UKP 使用的平面特征<sup>[9]</sup>和 Yang 等<sup>[11]</sup>提供的一个特征。本文的对比系统还加入结构化特征 PST 和 PDT，使用 SVR 进行句子相似度计算。

本文实验使用 STS-2012 提供的 SMTeuroparl, MSRvid 和 MSRpar 语料，其中的文本对来自新闻标题、图片说明和公众论坛的留言等。这 3 种语料的说明见表 1。本文使用 SICK 数据集<sup>[10]</sup>的 500 对句子作为开发集，5000 对句子作为训练集，补充在 3 个文本集中。

在参数设置方面，隐藏层大小为 50，所有单词表示为由 Word2Vec 训练得到的 300 维向量。使用 AdaGrad 方法<sup>[11]</sup>进行训练，以学习率为 0.05、批尺度为 25 更新参数。模型参数以 L2 正则强度为  $10^{-4}$  进行正则化。实验结果评估由测试文本中句子对的人工得分与本文系统计算得分的皮尔逊相关性系数由给出，系数小于 0.4 表示显著弱相关，系数在 0.4~0.75 之间表示中等相关，系数大于 0.75 表示强相关。

3.2 实验结果及分析

表 2 为基于浅层句法树和基于词短语依存关系树的对比实验结果，可以看出，基于 Tree-LSTM 模型的文本相似度计算在长文本对集 SMTeuroparl 和 MSRpar 上性能更佳。例如长文本对的词组“recently”和“period of time”，两个词组含义相同，但其浅层句法树与依存关系树完全不同。用 SVR

表 2 基于浅层句法树和基于词短语依存关系树的对比实验结果

Table 2 Result of PST and PDT

方法	MSRvid	MSRpar	SMTeuroparl
PST+ SVR	<b>0.908</b>	0.732	0.571
NPST+ SVR	0.896	<b>0.783</b>	0.584
NPST+ Tree-LSTM	0.889	0.777	<b>0.613</b>
PDT+ SVR	<b>0.886</b>	0.776	0.528
NPDT+ SVR	0.847	0.754	0.507
NPDT+ Tree-LSTM	0.881	<b>0.799</b>	<b>0.568</b>

计算得到的相似度很低，因为相同子树的个数为零；Tree-LSTM 使用高维度的语义向量，更容易获得文本深层次的语义，从而得到更高的相似度。在使用 NPST 时，与 Tree-LSTM 模型相比，SVR 在长文本集 MSRpar 上实验效果更佳。例如，文本对“The problem will be solved”和“He said the problem should be solved”的语义相似，且大部分句法结构相似，但是基于 Tree-LSTM 模型的相似度得分较低。由于 SVR 模型通过计算相同子树个数得到高相似度计算得分，使用 Tree-LSTM 模型时，如果文本对的主语不同，通过学习后的参数会给主语更高的权值，主语语义对结果的影响更大，所以计算结果偏低。基于 SVR 的相似度计算在短文本对集 (MSRvid) 上效果更好，这是因为 SVR 除提供结构化特征外，还提供 40 个平面特征。如果 SVR 与 Tree-LSTM 相同，仅使用结构化特征，实验结果则远差于 Tree-LSTM。

从表 2 可以看出，对于使用 Tree-LSTM 效果更佳的两个文本对集 MSRpar 和 SMTeuroparl，NPST 更适合于不符合句法规则的长文本集 SMTeuroparl，NPDT 更适合于合句法规则的长文本集 MSRpar。对于不符合文法的文本，例如“Cuz i went to home”中的 cuz 不符合文法，是 because 的口语化表示，通过依存关系分析，得到“Cuz i went to home”与“Because i went to home”的依存关系，可知不符合文法的文本依存关系的分析是错误的。因此，对于不符合文法文本的语法分析树和依存关系树，可能得到错误的结构，导致实验结果偏低。同时，这个文本的浅层句法树结构简单，且基本的浅层句法树无需使用句法信息，在此基础上做出修改也仅可能将单词错分到周围的短语里，得到的 NPST 与正确

表 1 语料说明

Table 1 Corpus description

语料	训练文本对数	测试文本对数	文本特点
SMTeuroparl	750	750	文本长度长，不符合语法
MSRvid	750	750	文本长度最短，结构简单
MSRpar	734	459	文本长度长，结构复杂，符合语法

的 NPST 差异很小, 因此对计算的准确性影响较小。对于符合文法规则的文本对集(MSRpar), NPDT 和 NPST 均提供了有效的短语结构信息, 与 NPST 相比, NPDT 还提供了以短语为粒度的依存关系信息, 更有利于相似度计算。使用 NPST 的 2-ray Tree-LSTM, 对每个孩子节点分别使用参数矩阵, 可以更精确地学习孩子节点的状态。

使用相同的树形结构(NPST 或 NPDT), 基于相同的相似度计算模型(SVR 或 Tree-LSTM)的实验均是在短文本对集(MSRvid)上的结果最佳, 符合句法规则的长文本集(MSRpar)次之, 最次是不符合句法规则的长文本集(SMTeuroparl)。说明对于短文本和符合句法规则的文本, 相似度计算模型获得其语义和句法信息的能力更强。

因此, 基于 SVR 的相似度计算适用于较短的文本对, Tree-LSTM 则适用于长文本对相似度计算。书面化、符合句法规则的文本对适合使用 NPDT, 口语化、不符合句法规则的文本对适合使用 NPST。

## 4 总结

本文在浅层句法树和依存关系树的基础上, 将文本对处理成 NPST 和 NPDT 树形结构, 并通过 Tree-LSTM 进行相似度计算。实验结果显示, 合适的结构化表示与 Tree-LSTM 可以提高长文本相似度计算的性能。在下一步研究中, 将对结构化特征做进一步研究, 拟将文本的语义信息表示为结构化特征, 以期进一步提高文本相似度计算结果的精确性。

## 参考文献

- [1] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [EB/OL]. (2015-02-28)[2017-02-20]. <https://arXiv.org/abs/1503.00075>
- [2] Chen Qian, Zhu Xiaodan, Ling Zhenhua, et al. Enhancing and combining sequential and tree LSTM for natural language inference [EB/OL]. (2016-09-20)[2017-06-01]. <https://arXiv.org/abs/1609.06038>
- [3] Bank J, Cole B. Calculating the jaccard similarity coefficient with map reduce for entity pairs in Wikipedia [R]. 2008[2017-06-01]. <http://www.infosci.cornell.edu/weblab/papers/Bank2008.pdf>
- [4] Metzler D. Generalized inverse document frequency // ACM Conference on Information and Knowledge Management. New York, 2008: 399-408
- [5] Banerjee S, Pedersen T. Extended gloss overlaps as a measure of semantic relatedness // Proceedings of the 25th International Joint Conference on Artificial Intelligence. Acapulco, 2003: 805-810
- [6] Landauer T K, Foltz P W, Laham D. Introduction to latent semantic analysis. *Discourse Processes*, 1998, 25(3): 259-284
- [7] Lund K, Burgess C. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods*, 1996, 28(2): 203-208
- [8] Allan J, Wade C, Bolivar A. Retrieval and novelty detection at the sentence level // Proceedings of Association for Computing Machinery Special Interest Group on Information Retrieval Conference on Research and Development in Informaion Retrieval. Dresden, 2003: 314-321
- [9] Mihalcea R, Corley C, Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity // Proceedings of the 21st Association for the Advancement of Artificial Intelligence. Boston, 2006: 775-780
- [10] Malik R, Subramaniam V L, Kaushik S. Automatically selecting answer templates to respond to customer emails // Proceedings of the 25th International Joint Conference on Artificial Intelligence. Hyderabad, 2007: 1659-1664
- [11] Yang Meng, Li Peifeng, Zhu Qiaoming. Sentence similarity on structural representations // Proceedings of the 5th Conference on Natural Language and Chinese Computing. Kunming, 2016: 481-488