



文章编号:1001-9081(2018)07-1831-08

DOI:10.11772/j.issn.1001-9081.2017123009

## 基于注意力与神经图灵机的语义关系抽取模型

张润岩<sup>1</sup>, 孟凡荣<sup>1\*</sup>, 周勇<sup>1</sup>, 刘兵<sup>1,2</sup>

(1. 中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116; 2. 中国科学院电子研究所, 北京 100080)

(\* 通信作者电子邮箱 mengfr@cumt.edu.cn)

**摘要:** 针对语义关系抽取(语义关系分类)中长语句效果不佳和核心词表现力弱的问题, 提出了一种基于词级注意力的双向神经图灵机(Ab-NTM)模型。首先, 使用神经图灵机(NTM)作为循环神经网络(RNN)的改进, 使用长短时记忆(LSTM)网络作为控制器, 其互不干扰的存储特性可加强模型在长语句上的记忆能力; 然后, 构建注意力层组织词级上下文信息, 使模型可以加强句中核心词的表现力; 最后, 输入分类器得到语义关系标签。在 SemEval 2010 Task 8 公共数据集上的实验表明, 该模型获得了 86.2% 的得分, 优于其他方法。

**关键词:** 自然语言处理; 语义关系抽取; 循环神经网络; 双向神经图灵机; 注意力机制

**中图分类号:** TP183    **文献标志码:** A

### Semantic relation extraction model via attention based neural Turing machine

ZHANG Runyan<sup>1</sup>, MENG Fanrong<sup>1\*</sup>, ZHOU Yong<sup>1</sup>, LIU Bing<sup>1,2</sup>

(1. School of Computer Science and Technology, China University of Mining and Technology, Xuzhou Jiangsu 221116, China;

2. Institute of Electrics, Chinese Academy of Sciences, Beijing 100080, China)

**Abstract:** Focusing on the problem of poor memory in long sentences and the lack of core words' influence in semantic relation extraction, an Attention based bidirectional Neural Turing Machine (Ab-NTM) model was proposed. Instead of a Recurrent Neural Network (RNN), a Neural Turing Machine (NTM) was used firstly, and a Long Short-Term Memory (LSTM) network was acted as a controller, which contained larger and non-interfering storage, and it could hold longer memories than the RNN. Secondly, an attention layer was used to organize the context information on the word level so that the model could pay attention to the core words in sentences. Finally, the labels were gotten through the classifier. Experiments on the SemEval-2010 Task 8 dataset show that the proposed model outperforms most state-of-the-art methods with an 86.2% F1-score.

**Key words:** Natural Language Processing (NLP); semantic relation extraction; Recurrent Neural Network (RNN); bidirectional Neural Turing Machine (NTM); attention mechanism

### 0 引言

语义关系抽取(或语义关系分类)任务是指在给定一段文字并标记两个实体词的情况下, 给出两个实体词的语义关系。关系抽取是自然语言处理(Natural Language Processing, NLP)系统的重要组成部分, 准确的语义关系抽取对句意理解有十分重要的作用, 理解句子核心词之间的关系是把握句子整体含义的重要一步, 因此, 近些年语义关系抽取越来越受到国内外研究学者的关注。

机器学习方法在关系抽取上的高效性已被多次证明, 许多传统模型都致力于机器学习和特性设计上, 这些模型通常依赖于一个完备的 NLP 流水线, 同时还需要额外的手工特性或内核<sup>[1-4]</sup>。而近年来, 深度学习方法被广泛应用于关系分类任务, 它能够在保证准确率的前提下大幅减少人为调整的工作。如今有许多基于深度学习的神经网络模型用于该任务, 如卷积神经网络(Convolutional Neural Network, CNN), 循环神经网络(Recurrent Neural Network, RNN), 它们有些使用

最短依赖路径或依赖子树<sup>[5-7]</sup>, 而另一些则减少预训练的操作, 直接输入原始语句来学习隐含特征<sup>[8-9]</sup>。这些方法都已被证明有效, 但它们会平等地考虑句中每个词, 因而不可避免地会被无意义的词所干扰。

针对这一问题, 一些研究提出注意力机制与神经网络相结合的方法<sup>[10-12]</sup>, 该机制使模型能够自动关注句子的核心词, 给予其更高的价值权重。如例句“The news brought about a commotion”, 句中“brought(引起)”一词对“Cause-Effect(因果)”关系有决定性作用, 因此在整个句子的信息时, 模型会自动为“brought(引起)”词分配很高的权重, 以加强其对关系标签的预测能力。此为词层面的注意力机制, 本文将其与神经网络结合以实现一个端到端的模型。

神经图灵机(Neural Turing Machine, NTM), 也可称神经记忆网络(Neural Memory Network), 本文使用循环神经网络作为其控制器, 因而其本质是一个具有额外存储矩阵的循环神经网络(RNN), 而相比长短时记忆(Long Short-Term Memory, LSTM)网络或门控循环神经网络(Gated Recurrent

收稿日期:2017-12-22;修回日期:2018-02-09;录用日期:2018-02-27。    基金项目:国家自然科学基金面上项目(61572505)。

**作者简介:** 张润岩(1994—), 男, 北京人, 硕士研究生, 主要研究方向: 神经网络、自然语言处理; 孟凡荣(1962—), 女, 辽宁沈阳人, 教授, 博士生导师, 博士, 主要研究方向: 智能信息处理、数据库技术、数据挖掘; 周勇(1974—), 男, 江苏徐州人, 教授, 博士生导师, 博士, 主要研究方向: 数据挖掘、无线传感器网络; 刘兵(1981—), 男, 河南永城人, 副教授, 博士, 主要研究方向: 机器学习、模式识别。



Unit, GRU) 等 RNN 上强化记忆能力的改进模型, 神经图灵机拥有更多且互不干扰的存储空间, 它允许 RNN 对输入进行无损的保存, 从而赋予 RNN 更优秀的记忆能力甚至使 RNN 拥有长时记忆(持久记忆), 因此, 本文将其用于关系抽取任务作为循环神经网络的替代, 以获得更好的语境特征提取效果。

综上所述, 本文的研究有以下三个核心点:

1) 使用神经图灵机代替循环神经网络来提取语境特征, 使每个词的高级特征考虑语境信息, 其中本文使用一种简化且更高效的神经图灵机。

2) 在神经图灵机之上加入词层面的注意力机制, 它使模型能够自动识别核心词, 促进核心词对分类的积极作用, 并弱化干扰词的影响, 其中本文使用一种更适合任务的权重评分函数。

3) 将模型应用于语义关系抽取任务, 与该任务的相关研究对比分析, 并在 SemEval-2010 Task 8 这一标准数据集上基本达到最好效果。

## 1 相关工作

虽然半监督或无监督的方法在关系抽取领域有一些应用并得到了可观的效果<sup>[1-2]</sup>, 但本文的研究主要针对有监督的方法, 它一般会得到更好的效果同时更易于验证和评价。

在早期的关系分类研究中, 学者多通过一系列自然语言处理工具进行特征提取, 或是使用一些精心设计的核心<sup>[3]</sup> 加上支持向量机(Support Vector Machine, SVM)等分类器<sup>[4]</sup>, 他们在特征提取或模型设计工作上会花费大量精力, 国内许多研究也多基于此<sup>[13-14]</sup>。另一方面, 针对缺少语料库的问题, 一些远程监督方法将知识库与非结构化文本进行对齐<sup>[5]</sup>。而 SemEval-2010 Task 8<sup>[16]</sup>发布后, 后续研究大多使用它作为标准数据集来测试模型。

传统的方法确实可行且有效, 但必然强烈依赖于模型设计和特征提取的质量, 易引入人为误差。而随着近年深度学习的发展, 许多基于神经网络的模型被用来提取隐藏特征。文献[5]的递归矩阵-矢量(Matrix-Vector Recurrent Neural Network, MV-RNN)模型为 RNN 中的每个节点添加一个矩阵, 以提高整个网络的适应能力; 文献[6]提出了基于因子的组合嵌入模型(Factor-based Compositional embedding Model, FCM), 将句子分解为子结构, 独立提取特征, 并通过汇总层将其组合; 文献[7]提出了一种深度循环神经网络(Deep Recurrent Neural Network, DRNN)模型, 通过解析树的根词将句子分为两部分, 并将其输入到多层 RNN 中, 其中神经元会额外考虑上一层的相邻词信息。上述工作中的模型, 根据语法分析树来设计, 通过最小生成子树整合语句, 它们取得了很好的效果。另一方面, 一些研究者尝试构建端到端模型, 嵌入 NLP 特征来描述词汇, 直接将原始语句作为神经网络的输入。文献[8]提出了 CNN 提取词汇和句子层面的特征。文献[9]提出了扩展中间语境的 CNN, 并使用简单的投票方案将其与双向 RNN 结合。端到端模型更简洁且易于实现, 但不同于基于分析树的模型, 它无法删除语法上不相关的词, 因此可能会受到无意义词的干扰。

近三年, 注意力机制被用于 NLP 领域, 首先由文献[17]用于机器翻译中的文本对齐问题, 而后因其能够自动发现核心词的特性, 注意力机制很快引起人们的关注, 并用于更多任

务; 文献[10]将注意机制与双向 RNN 相结合用于语义关系抽取任务; 文献[11]将句子按两个实体词拆分, 使用分层的基于注意力 RNN 模型; 文献[12]提出了一种基于注意力的 CNN 模型, 用实体来评价每个词的重要程度。这些基于注意力的方法都以原始语句作为输入, 不再进行额外的特征筛选或裁剪, 然而性能却比不使用注意力机制的模型更好。

另一方面, 神经图灵机于 2014 年由 Google DeepMind 实验室<sup>[18]</sup>提出。神经图灵机不能代替循环神经网络, 但它被证明确实比传统的 RNN 及 LSTM 等改进模型有更好的记忆能力, 并能够应付一些复杂序列任务。文献[19]后续又提出一种改进模型, 称其为可微分神经计算机(Differentiable Neural Computer, DNC), 该模型具备高度定制的存储寻址机制, 并出色地完成了地铁线路规划、族谱关系推测、阅读理解等实际问题。

神经图灵机比传统的 RNN 及其改进有更好的记忆能力, 因而它可以更好地完成语句信息提取。本文模型使用神经图灵机提取语句信息, 而后使用注意力机制, 强化核心词的作用。最终本文的模型可以有效地从原始语句中提取关键特征, 并出色地完成语义关系分类任务。

## 2 模型描述及实现

本文研究的模型, 称其为基于注意力的神经图灵机(Attention-based Neural Turing Machine, Ab-NTM)模型。如图 1 所示, 该模型通过词嵌入层获得每个词的数值化表示, 使句子文本转化为一个二维的数值矩阵表示, 然后将该矩阵输入一组双向神经图灵机中获得每个词的高级特征, 之后将所有词通过注意力层进行合并得到句子特征, 最后输入到分类器进行分类。本文模型两个关键点如下:

1) 双向神经图灵机。神经图灵机在形式上类似于循环神经网络, 以同样的方式接受序列输入, 因而本文模型以双向 RNN 的方式构建一组双向神经图灵机。

2) 词级注意力层。模型对所有词使用注意力机制进行加权合并, 从而有针对性的提取句子特征, 同时词级的注意力层可以体现每个词的重要程度。

### 2.1 词嵌入层

词嵌入层用于将词语文本转化为数值表示, 其中数值应在一定程度上包含词语信息。本文选择三种属性对词语进行数值化表示, 分别为 word2vector 的词向量表示(简称词向量)、词与实体的位置关系以及词性信息。此外, 还进行了一个只使用 word2vec 词向量属性作为词语表示的对比实验, 其效果稍差。

词向量定义如下, 给定一个句子  $S = (w_1, w_2, \dots, w_n)$ , 通过查找词向量矩阵  $\mathbf{W}_{\text{word}} \in \mathbb{R}^{|V| \times d_w}$  将每个词转换成词向量, 其中:  $d_w$  是词向量的维度;  $|V|$  是词向量矩阵的语料集中词汇数量。通过矩阵查找, 每个单词被映射到一个行向量  $w_i^d \in \mathbb{R}^{d_w}$ , 即该词的词向量表示。对于无法在词向量矩阵中找到的词, 本文使用随机初始化方式。

对于多义性问题, 本文使用词性(Part of Speech, POS)标签来增强词的语义表示。通过查找 POS 嵌入矩阵, 将每个 POS 标签映射到预先训练的向量  $w_i^{\text{pos}} \in \mathbb{R}^{d_{\text{pos}}}$ , 其中  $d_{\text{pos}}$  是超参数。

本文还使用位置嵌入来描述两个实体和每个单词之间的



相对距离,相对两个实体的距离是分开计算的,仿照一些之前的研究<sup>[8]</sup>。例如,在“That < e1 > machine </e1> makes a lot of < e2 > noise </e2>”句中,单词“makes”到实体“machine”和实体“noise”的相对距离是1和-4,因此,将获得位置信息的两个向量表示  $w_i^{p1}, w_i^{p2} \in \mathbb{R}^{d_p}$ ,其中  $d_p$  是超参数。

最后,本文将每个词的三种特征拼接在一起,构成输入,即  $w_i^f = [w_i^d, w_i^{pos}, w_i^{p1}, w_i^{p2}]$ 。

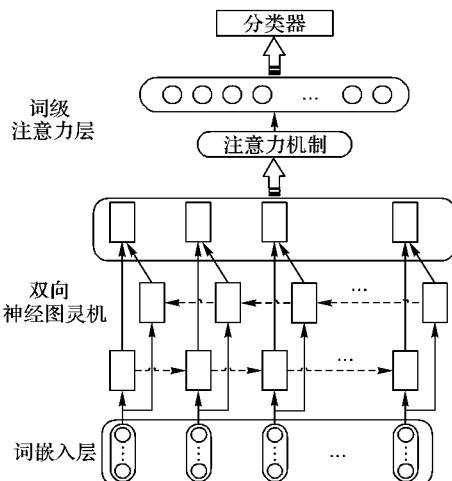


图1 本文模型概览

Fig. 1 Architecture of the proposed model

## 2.2 双向神经图灵机

### 2.2.1 长短时记忆(LSTM)网络

本模型采用的神经图灵机其本质是一个配备额外存储矩阵的循环神经网络(使用LSTM单元),模型使用LSTM单元作为神经图灵机控制器(Controller),用于保持内部状态并生成存储矩阵的地址信息。

循环神经网络(RNN)多用于有时序的数据,它对当前时刻输入进行特征提取时会考虑之前时刻的特征信息,因而RNN可以完整地考虑整个序列信息。对于自然语言文本,语句可以视为一个文字序列,因此RNN被广泛用于许多自然语言处理(NLP)任务中。为了增强RNN在长序列上的性能,大多数模型使用LSTM作为传统RNN(BasicRNN)的替代。LSTM由文献[20]提出。其主要思想是利用线性存储单元存储每次迭代的信息,并利用三个门单元进行读取、写入和擦除。此外还有许多LSTM变种被设计用于特定的任务作相应的改进,在本文中,采用由文献[21]提出的LSTM变体,这是一种被广泛使用的变体形式。

LSTM包含一个存储单元  $c_t$ ,该存储单元贯穿每个时间步,允许信息在没有交互的情况下流过时间步。为了改变存储器单元中的信息,LSTM设置三个逻辑门:用于写入的输入门  $i_t$ ,用于擦除的遗忘门  $f_t$  和用于读取的输出门  $o_t$ 。每个门的计算由输入  $x_t$  和前一个时间步的隐层值  $h_{t-1}$  共同决定,并最终得出一个操作信息的比值。详细计算如下所示:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1}^l + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1}^l + b_f) \quad (2)$$

$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1}^l + b_c) \quad (3)$$

$$c_t = i_t g_t + f_t c_{t-1} \quad (4)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1}^l + b_o) \quad (5)$$

$$h_t^l = o_t \tanh(c_t) \quad (6)$$

其中: $W$  和  $b$  均为神经网络待训练的参数,下标表示所属计算单元。

### 2.2.2 神经图灵机

神经图灵机(NTM)使用LSTM作为控制器,LSTM内部的存储单元作为内部状态,而在LSTM外部本文额外使用一个二维的存储矩阵,用于保存外部状态,其整体架构如图2所示。

与文献[19]提出的可微分神经计算机(DNC,一种最新的神经图灵机改进)相比,本文精简了NTM的寻址机制,直接利用注意力机制,使用LSTM的隐层以及存储-输入相似度来生成,计算写入和读取的存储地址;而DNC在寻址上引入连续存储、先进先出等机制,这适用于其研究的序列复制、线路规划等任务,但对于关系抽取的研究任务并不十分适用,而且引入这些机制会使模型变得十分复杂。实验表明,本文精简的模型达到了相当的效果,并且训练耗时更短。

在图2中,标有“时序”的虚线表示模型在时间序列中的状态维持,即按时间顺序从头至尾传递;标有“LSTM”的实线是常规LSTM的工作机制;标有“NTM”的实线是神经图灵机的改进之处,每个时间步存储矩阵获得输入将其存放在合适的地址位置上,然后读取一行内容分别送入输出单元和下一个时间步的控制器,标有“地址”的点状线是存储地址的生成,本文使用LSTM隐层以及输入-存储矩阵相似度(图中未标出)来计算地址。图中下方为输入地址,上方为输出地址,两者分别计算。

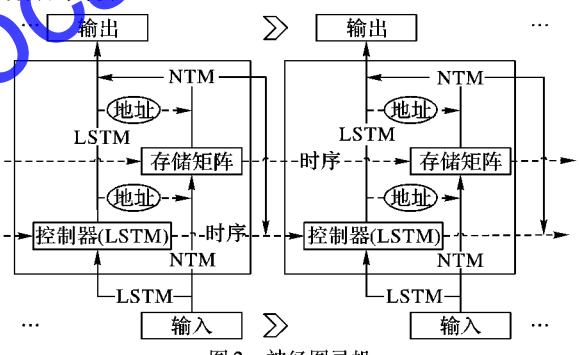


Fig. 2 Architecture of neural Turing machine

存储矩阵的地址计算公式如下:

$$\cos_t = \text{softmax}(\text{cosine}(M_{t-1}, x_t)) \quad (7)$$

$$ha_t^i = \text{softmax}(W_{ha}^i h_t^l + b_{ha}^i) \quad (8)$$

$$ha_t^o = \text{softmax}(W_{ha}^o h_t^l + b_{ha}^o) \quad (9)$$

$$addr_t^i = \alpha * \cos_t + (1 - \alpha) * ha_t^i \quad (10)$$

$$addr_t^o = \alpha * \cos_t + (1 - \alpha) * ha_t^o \quad (11)$$

其中: $addr_t^i$  和  $addr_t^o$  分别为输入和输出地址; $h_t^l$  是LSTM的隐藏层输出; $x_t$  是输入向量; $M \in \mathbb{R}^{d_{m\_len} \times d_{m\_size}}$  是存储矩阵;softmax是softmax函数,即非线性激活并作归一化;cosine是余弦相似度计算函数; $\alpha$  是一个超参数,用于调节隐层和相似度的比例。模型计算出的存储地址是一个一维向量,其长度等于存储矩阵的长,数值的大小表示存储矩阵对应位置的采用率。

得到地址后,存储矩阵的写入和读取计算公式如下:

$$v_t = \text{relu}(W_{xm}x_t + b_{xm}) \quad (12)$$

$$M_t = addr_t^i \cdot v_t + (1 - addr_t^i) * M_{t-1} \quad (13)$$



$$\mathbf{h}_t^m = \mathbf{M}_{t-1} \cdot \text{addr}_t^o \quad (14)$$

其中:  $\mathbf{M}_t$  是  $t$  时刻完成写入操作后的存储矩阵, 本文使用非线性映射后的输入向量  $v_t$  进行写入;  $\mathbf{h}_t^m \in \mathbb{R}^{d_{m\_size}}$  从存储矩阵中读取出的信息, 为一个一维向量, 本文将其与 LSTM 的隐层输入合并在一起作为神经图灵机的完整输出, 计算公式如下:

$$\mathbf{h}_t = \text{relu}(\mathbf{W}_h [\mathbf{h}_t^l, \mathbf{h}_t^m] + \mathbf{b}_h) \quad (15)$$

最终模型得到一系列神经图灵机的隐层输出, 并且它们与最初的输入一一对应, 即每个输出对应句中的一个词, 这可认为是该词的深层特征。本文将它们拼接在一起  $[\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_t, \dots]$ , 然后输入到后续网络。

### 2.2.3 双向网络

通常, 一个单词的语义信息会同时受到前文和后文的影响, 而单向的循环神经网络只考虑前文信息, 因此, 本文用构造双向 RNN 的方法构建双向神经图灵机, 它由两个神经图灵机组成, 分别使用正向和反向序列作为输入, 然后将两个网络的隐层输出拼接在一起作为双向网络的输出, 最后的输出将包含整个句子的语义信息。

### 2.3 词级注意力层

考虑到每个单词可能对分类任务有不同的贡献, 即有些词起到关键作用而有些词作用不大, 因而本文引入了单词层面的注意力层。利用注意力机制, 本文将每个单词对应的高级特征和整个句子的特征作为驱动, 为每个单词计算一个评分权重, 然后将所有词的高级特征加权合并, 因此, 注意机制会更多地关注对预测有重要意义的单词, 并赋予它们更大的权重。

词级注意力层的网络结构如图 3 所示。合并每个单词对应的隐层输出  $\mathbf{h}_t$  和代表整句信息的最终时刻隐层  $\mathbf{h}_{last}$ , 对其进行非线性映射并作归一化, 从而模型可求解出一个权值  $a_t$ , 然后将词的高级特征加权求和, 得到句子的最终特征。

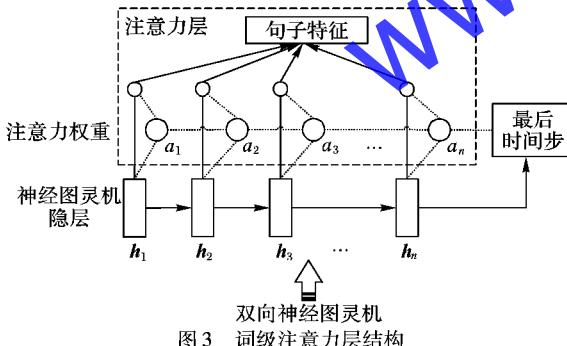


图 3 词级注意力层结构

Fig. 3 Architecture of word-level attention layer

受文献[20]的启发, 本文同样考虑了三种不同的评分函数, 如式(19)所示, 用于注意力中  $\mathbf{h}_t$  和  $\mathbf{h}_{last}$  的组合, 并测试其性能。最终注意力层的计算如下所示:

$$\mathbf{h}_a = \mathbf{W}_{atn} \sum_{t=1}^n a_t \mathbf{h}_t \quad (16)$$

$$a_t = \exp(e_t) / \sum_{j=1}^n \exp(e_j) \quad (17)$$

$$e_t = k * \text{score}(\mathbf{h}_t, \mathbf{h}_{last}) \quad (18)$$

$$\text{score}(\mathbf{h}_t, \mathbf{h}_{last}) = \begin{cases} \mathbf{h}_t^T \mathbf{W}_a \mathbf{h}_{last}, & \text{dot 函数} \\ \text{cosine}(\mathbf{h}_t, \mathbf{h}_{last}), & \text{cosine 函数} \\ \mathbf{W}_a [\mathbf{h}_t : \mathbf{h}_{last}] + \mathbf{b}_a, & \text{concat 函数} \end{cases} \quad (19)$$

其中:  $\mathbf{h}_a$  是注意力层的输出, 即句子的最终特征;  $a_t$  对应于  $t$

位置单词的权重, 是通过 score 函数计算并进行归一化的实数。

### 2.4 正则化与训练

dropout 方法在训练时按比例随机屏蔽一些神经元, 在测试时打开所有神经元进行预测, 从而减缓过拟合速度, 使神经网络更充分地自我学习, 已经有诸多实验证明 dropout 方法是切实有效的。文献[23]进行了改进, 提供了一种在循环神经网络上使用 dropout 的方法。在本模型中, 采用类似的方法对神经图灵机单元进行封装, 并在神经图灵机层和输出层分别使用 0.6 和 0.4 的 drop 比例。

另外, 本文使用 L2 范式对参数矩阵进行限制, L2 范式可以有效地提高矩阵的稀疏性。以  $\lambda$  和  $\lambda_2$  作为学习率,  $J(\theta)$  作为损失函数, 本文使用交叉熵函数计算损失。

## 3 实验结果及分析

### 3.1 数据集和参数设置

#### 3.1.1 数据集

本文使用常用的标准数据集 SemEval-2010 Task 8<sup>[14]</sup> 来评估模型, 该数据为每个样本注释一种关系, 包含 9 种类型以及 1 个不属于任何一种关系的“Other”类型, 9 种关系类型分别为 Cause-Effect(原因-效果)、Instrument-Agency(工具-使用者)、Product-Producer(产品-生产者)、Content-Container(内容-容器)、Entity-Origin(实体-源头)、Entity-Destination(实体-归宿)、Component-Whole(部分-整体)、Member-Collection(成员-集体) 和 Message-Topic(消息-话题)。考虑到关系存在方向, 即主动方和被动方的区别, 该数据集额外注释了 9 种关系的方向(“Other”类型没有方向), 因此数据集标签一共有 19 种。该数据集有 8000 个训练样本和 2717 个测试样本, 本文直接使用这一样本分配方案。SemEval-2010 Task 8 官方提供了一个评分程序, 本文使用其中的宏(Macro)平均 F1 分数方法来评估本文模型的表现, 其他学者在本任务的研究也多采用该评分方式。

#### 3.1.2 参数设置

本文使用在维基百科上训练的 word2vec skip-gram 模型<sup>[21]</sup> 将单词转化为数值化的向量表示, 即词向量, 该模型作为公共数据集被许多研究使用。此外, 本文还尝试了 senna 模型<sup>[22]</sup> 和 glove 模型<sup>[24]</sup> 进行词向量嵌入, 同时测试了不同词向量维度下模型的效果。本文对词性(POS)特征和位置特征作了预处理将其转化为  $d_{pos}$  和  $d_p$  维度的数值化向量, 使用 Stanford POS Tagger 工具包自动进行词性分析。本文使用高斯分布来随机初始化权重矩阵, 使用 Adam 优化方法<sup>[25]</sup> 迭代更新参数, 训练神经网络。表 1 列出了其他一些超参数的详细设置。

### 3.2 对比实验分析

#### 3.2.1 不同模型对比

表 2 展示了本文模型与语义关系抽取领域相关研究的 F1 分数比较, 本文模型的性能基本达到了最优水平。为便于分析, 本文将本领域相关研究分为 4 类。

本文实验结果评价指标使用 F1 值计算方法, 如式(20)所示。本文采用的数据集有 9 个子类别和 1 个“Other”类别, 总体 F1 指标使用子类别平均 F1 值计算方式:



$$F1 = \frac{召回率 \times 准确率}{召回率 + 准确率} \times 100\% \quad (20)$$

表1 部分超参数的设置

Tab. 1 Partial hyperparameters setting

参数	参数解释	值	参数	参数解释	值
$d_w$	词向量维度	50/100/ 200/640	$d_a$	注意力层 输出维度	80
$d_{pos}$	词性维度	25	$\lambda$	学习率	0.01
$d_p$	相对位置维度	25	$\lambda_2$	L2 学习率	0.0001
$d_n$	神经图灵机 隐层维度	200			

表2 SemEval 2010 Task 8 数据集上的不同模型对比

Tab. 2 Different models comparison on dataset SemEval 2010 Task 8

模型类别	分类器	F1 分数/%
非神经网络模型	SVM <sup>[4]</sup>	82.2
基于最短 依赖树的 模型	MVRNN <sup>[5]</sup>	82.4
	FCM <sup>[6]</sup>	83.0
	DRNN <sup>[7]</sup>	84.1
端到端模型	CNN <sup>[8]</sup>	82.7
	ER-CNN + R-RNN <sup>[9]</sup>	84.2
基于 注意力 的模型	Att-BLSTM <sup>[10]</sup>	84.0
	Hier-BLSTM <sup>[11]</sup>	84.3
	Attention-CNN <sup>[12]</sup>	85.9
本文模型	Ab-NTM(word2vec) (word dim = 100)	86.2
	Ab-NTM (word2vec) (word dim = 640)	85.2
	Ab-NTM (seme) (word dim = 50)	85.4
	Ab-NTM (glove) (word dim = 100)	85.7
	Ab-NTM (glove) (word dim = 200)	85.9

1) 非神经网络模型。本文选择了一个使用支持向量机(SVM)分类器的代表性方法<sup>[4]</sup>,该方法考虑了3组原始特征:词汇(Lexical)、语法依赖(Dependency)、PropBank、FrameNet、Hypernym、NomLex-Plus、NGram和TextRunner。该研究将这8种特征进行数值化表示用以表示词或句子,然后将它们放入SVM进行分类,该研究发现这些特征都有助于分类。该方法以82.19%的分数夺得SemEval任务的冠军,但后续的关系分类领域研究超越了这一分数。这是因为最近的研究主要是通过神经网络对原始信息进行深层特征挖掘,而不是手动地精心挑选原始特征,即后续的研究将重点放在了神经网络结构的设计上,而不是特征挑选,挖掘特征表示的任务交给神经网络来完成,因为人为的或通过预处理引入的特征可能存在偏差或干扰,而足够强大的神经网络可以自动寻找特征关系进行分类。也有越来越多的研究表明,通过隐层特征提取,少量的NLP特征就足以完成分类。

2) 最短依赖树(Shortest Dependency Path, SDP)模型。SDP是检测语法结构和语法逻辑的一个有效方法,它通过构建语法树上两实体的最短生成树,消除了两个实体联系之外的不相关词,即剩余词语是与两实体的表达直接相关的(在语法上)。根据这个树框架,父节点对子节点有直接影响,树中的词语都是有直接联系的,因此,基于SDP的模型可以忽略无意义的单词,并且输入序列可以依据语法结构建立;但这只是理想状态,实际情况下,SDP可能并不总是准确,许多语义上相关甚至极为重要的词,在语法上可能并不相连,这些词

如果删去就会丢失信息;并且另一方面,语法树的解析时间会随着句子长度增加而成倍地增长,所以语法树预处理在长句子上会消耗大量时间。尽管如此,从实验结果来看,基于SDP的模型具有优势,SDP是一个十分有效的改进。而本文模型使用注意力机制,这在一定程度上可以代替SDP删除无意义词的工作。

3) 端到端模型。随着深度学习的发展,一些研究者寄希望于通过网络自动提取语句特征、探寻语句内在联系,不再专注于挑选特征或进行人为的结构重建,而是改善网络结构,加强网络表达能力,使其表现力更强、更加健壮。换句话说,端到端的模型旨在尽量减少人为干预,同时也减轻工作量,把预训练的任务交给神经网络完成,真正实现输入端(原始语句)到输出端(分类结果)的网络构建。根据实验结果,这些模型比过去的研究具有一定优势,模型虽计算量增大但更易于实现。而与SDP的神经网络模型相比,不相上下,这是因为即使深层网络结构也存在表现能力的极限,不引入人工先验知识虽然减少了误差,但也为模型减少了帮助,所以性能上还有待改进。

4) 基于注意力的模型。通过注意机制可为输入的每个单词提供权重,句子完整特征通过此权重加权地整合,这能减少噪声词的干扰,并将更大的权重赋予核心词以加强它们对分类预测的影响。从表2中实验结果可看出,由于对词语影响力调节,基于注意力的模型比端到端模型更胜一筹。与SDP模型相比,对于不相关的词,注意力模型不是将其从句子中直接去除,而是为它们赋予较小的权重,这可以减轻错误判断的情况。然而,如果探讨注意力机制的计算方法,就可发现注意力机制的权重评分十分依赖于上一层网络的隐层输出,因此首先使用更加有效的神经网络进行高级特征提取,可以改善模型整体的性能,这也就是本文模型进行的一个改进。

### 3.2.2 对比不同词向量

一般来说,更大的词向量维度可以为词义的表现提供更多空间,然而这种表现力不会随着维度增加而无限增长。因为通常合适维度的向量空间已足够词义表示,更大的维度反而可能因词向量难以训练,导致训练不完全而引入噪声,同时也会使网络更复杂、稳定性变差,使训练变得困难。从表2的实验结果也可看出,使用640维比使用100维词向量的模型效果要差不少(本文没有在公共数据中找到两者中间的维度)。然而,适当地增加词向量维度,可以增强词语的表现力,以此提高模型整体的性能。如表2所示,具有100维度的word2vec skip-gram模型<sup>[21]</sup>效果最好,许多已有研究也同样使用该词向量模型,因此本文使用该词向量进行下面的实验。

## 3.3 注意力机制的实验分析

### 3.3.1 注意力评分函数的比较

本节比较了注意力机制中不同评分函数的效果,计算公式如式(19)所示。评分函数接受两个输入,分别为每个单词的隐层特征和整个句子的隐层特征,输出为一个实数,用于表示该词的得分,即该词的权重。不同的评分函数会影响到整个注意力层的好坏,从而影响到模型的效果,图4显示了分别使用3个评分函数的模型的训练损失曲线,评分函数如式(19)所示。

如图4所示,3个模型在下降速度上几乎没有差别,只有



concat 的计算方式略微慢于另外两个,这或许是因为该计算方式需要更大的参数矩阵  $W_a$ ,即待训练参数更多,因而需要更多时间进行训练;但总体来看,三者的训练速度可以认为是相同的。

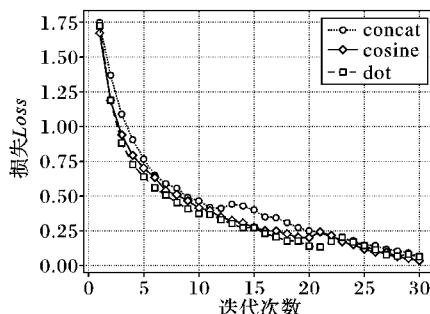


图4 注意力评分函数的训练 Loss

Fig. 4 Loss variation of attention scoring functions

在模型性能上, dot、cosine、concat 三个方法的 F1 得分分别为 86.2%、85.9% 和 85.6%, concat 方法的得分较低,因为 concat 方法实际上就是一个单层的神经网络,通过单层神经网络往往很难有效地提取特征,进而单词权重的评分也不会太高,本文同样尝试了多层神经网络(Multi-Layer Perceptron, MLP)来代替单层神经网络,但效果提升不明显。

### 3.3.2 词级注意力的可视化展示

词级注意力一个优势就在于可以查看模型对哪些词语进行了加强,对哪些词语进行了削弱,因为注意力机制会为每个词提供一个权值,查看这些权值就可以了解模型“偏袒”了哪些词,相应地,本文可以推断出模型认为这些词更有助于分类任务。

图5展示了两个样本句的注意力权重分布情况,其中图5(a)是注意力机制有效的情况,而图5(b)则是一个反例。

如图5(a)所示,对于样本的句子“The most common <e1> audits </e1> were about <e2> waste </e2> and recycling. (最常见的 <e1> 审计 </e1> 是关于 <e2> 浪费 </e2> 和再循环。)”,两个实体间的关系是“Message-Topic(消息-话题)”。实际上,通过句中的“about(关于)”一词就基本足以断定两个实体的关系。而从图上也可看出,模型为“about(关于)”一词赋予了很高的权重,即模型同样认为该词十分重要。这说明注意力机制在语义关系抽取任务上是十分有效的,同时它鼓励模型以抓关键词的方法进行句意理解,这在实际情况下通常也是十分有效的。

然而,如图5(b)所示,注意力机制也不能应付全部句子,比如对于“My <e1> shoe </e1> <e2> laces </e2> stay tied all the time. (我的 <e1> 鞋 </e1> <e2> 带 </e2> 总是系得很紧。)”,句中两个实体的关系是“Component-Whole(部分-整体)”,单从句子很难获得“鞋”和“鞋带”的关系,实际上判断两者的关系更多的是通过常识,或者说通过两个词自身的意思,所以在注意力层,模型也很难区分出哪个词更为重要,但从图上看,至少模型得出“My(我的)”单词贡献不大。

综上所述,词级注意力层可促使模型着重理解关键词,从而完成分类,这在大多数情况下是十分有效的。而在少数情况下,句中没有词对分类起到显著作用,则注意力机制效果不

明显,这也是本文模型的一个主要失分点,相信在以后研究中,更准确的词向量表示或更强大的模型结构可以改善这一问题。

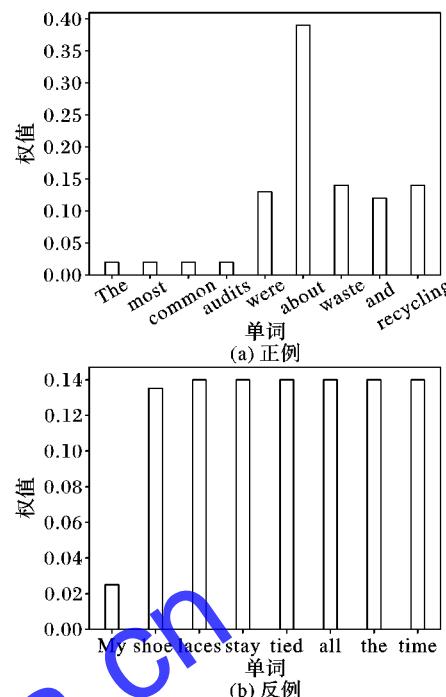


图5 注意力层权重分布

Fig. 5 Distribution of attention weights

### 3.4 神经图灵机的实验分析

#### 3.4.1 神经图灵机与 LSTM 对比

考虑在关系抽取任务上使用的神经图灵机,其存储矩阵十分庞大,不便于展示,并且很难给出一个明确定义指明怎样使用存储器是正确的,即对于在每次计算中使用多少信息是很困难设定正确结果的(神经网络本来就是黑盒计算),所以本文使用一个相对简单且有明确答案的例子来证明神经图灵机的有效性,同时将展示它如何使用内部的存储矩阵。

本文使用“随机多次复制”的任务来对比神经图灵机和LSTM。该任务给模型一串随机长度的数字以及一个随机的复制次数,期望的模型输出结果是复制了相应次数的该数字串。该任务需要模型有十分牢靠的记忆能力,在多次复制后仍能保持数字串的准确性;同时还需要模型有一定的泛化能力,因为数字串长度和复制次数是随机的,即模型需要理解并使用自身记忆,而不是单纯地输入输出。

如图6(a)所示,为便于展示,本文将输入的01序列转化为像素方格(按列分成不同时间步输入),灰色代表1,白色代表0,实验目标是连续复制并输出10次该输入内容。相比LSTM,神经图灵机在较长的时间序列上也保持了良好的记忆能力,如图6(a)中NTM模型的结果,每个时间步的图形与输入基本一致,而只在最后几步有细微变化;但使用LSTM,在长时间迭代后,记忆信息发生了很大的改变。

分析其原因,是因为LSTM只有一块存储单元,对于不同时间步的信息,需要反复写入或擦除存储单元。如果记忆的保存和使用间隔较短,这种方式是直接且有效的;但如果任务需要模型进行较长时间的记忆,或模型需要同时维持多段记忆,则LSTM存储单元中的信息会交叉干扰,难以无障碍地传



播到较远时刻。而 NTM 在存储上使用二维的存储矩阵, 使用注意力机制进行地址选择, 这可以较好地解决多记忆维持问题, 不同地址的存储信息相互隔离, 记忆可以不经修改地穿过不相关的时间步, 并且可以有顺序地存放和使用, 如图 6(b) 所示。

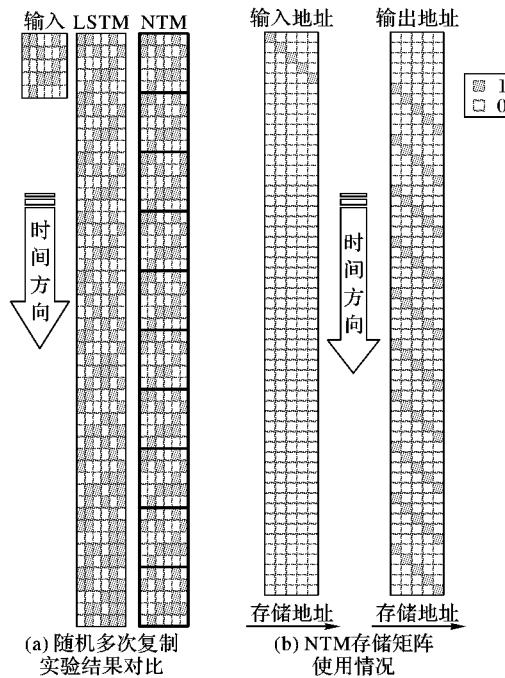


图 6 “随机多次复制”实验效果

Fig. 6 Results of the “repeat copy” task

对于语义关系抽取任务, 本文使用 NTM 来为每个词加入上下文信息, 这与“随机多次复制”任务是相似的, 同样是多记忆同时维持的任务。因为句子中不是每个单词之间都有联系的, 更多情况下应把单词分成几组, 不同组应使用不同的记忆单元, 即对应 NTM 中不同地址上的存储单元。通过图 7, 可以发现在语义关系抽取任务上 NTM 的 F1 结果确实比 LSTM 好。

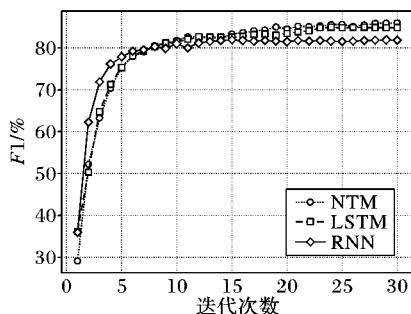


图 7 不同模型的抽取任务效果对比

Fig. 7 Results comparison of different models

### 3.4.2 模型在各分类上的 F1 分数

表 3 展示了本文的模型在语料 9 个类别中分类的 F1 分数, 可看出在一些分类中(如 Cause-Effect(原因-效果) 和 Entity-Origin(实体-源头)) 表现较好, 而在一些分类中(如 Component-Whole(部分-整体) 和 Instrument-Agency(工具-使用者)) 表现较差。

本文分析了这些分类的句子特点, 发现在表现较好的分类中, 句子常出现一些明确的关键词, 如“原因-效果”类别的

句子中常出现“cause”“result”等词以及“实体-源头”类别的句子中常出现“from”等词, 这些关键词在对应分类中往往词性相同、语义相近。相对的, 表现不好的分类, 其句子表述可能存在多种形式, 且差异很大, 模型较难从中提取相同特征; 或者与其他分类表述形式类似, 易造成干扰, 如“部分-整体”和“成员-集体”有些表述是一致的。

得益于神经图灵机的记忆能力优化, 本文模型可以更准确地获取上下文信息, 从表 3 中准确率的分布情况可知, 各类别准确率较平均, 即本文的模型没有表现非常差的类别, 这意味着模型在一定程度上理解了语义关系抽取这个任务, 并对每个子类别都有一套分类方式; 但是总体来看, F1 分数还有近 14% 的上升空间, 此模型还存在改进之处以得到更好的效果。

表 3 本文模型在各类别上的 F1 分数 %

Tab. 3 F1 score on subclasses by the proposed model %

类别	F1 分数
Cause-Effect(原因-效果)	89.02
Component-Whole(部分-整体)	83.51
Content-Container(内容-容器)	86.98
Entity-Destination(实体-归属)	85.33
Entity-Origin(实体-源头)	88.91
Instrument-Agency(工具-使用者)	83.94
Member-Collection(成员-集体)	86.20
Message-Topic(消息-话题)	86.16
Product-Producer(产品-生产者)	85.25

## 4 结语

本文在语义关系抽取任务上, 针对核心词表现力弱和长语句效果不佳的问题, 提出一种基于词级注意力的双向神经图灵机模型。其中, 词级注意力层可增强句中关键词的影响力, 促使模型以抓关键词的方法理解句意, 这可以解决端到端模型(即输入语句不作裁剪或结构调整)易被不相关词干扰的问题; 神经图灵机可视为一种循环神经网络的改进, 拥有一个额外的存储矩阵, 允许网络通过存储地址进行信息记忆, 使得模型可以进行多方面的记忆(不同方面放于不同地址), 且各方面之间互不干扰, 相比常用的 LSTM, 这可以增强词语高级特征提取的效果。本文将两者进行分层组合, 使其共同服务于关系抽取任务。

在语义关系抽取常用的公共数据集 SemEval-2010 Task 8 上测试了模型, 并分成非神经网络模型、最短依赖树模型、端到端模型、基于注意力的模型 4 组与该领域相关研究作比较, 进行了详细对比分析。此外, 还单独分析了词级注意力层和神经图灵机的有效性以及它们在关系抽取任务上的适应性。本文的模型最终实验结果以 86.2% 的 F1 值优于目前大多数相关研究, 但仍存在一些需要改进的地方, 未来工作中, 更准确的词向量表示或一组更合适的网络超参数可以带来更好的效果。

## 参考文献 (References)

- [1] LIU S, REN F. Relation extraction from Wikipedia articles by entities clustering [C]// Proceedings of the 2012 International Conference on Cloud Computing and Intelligent Systems. Berlin: Springer, 2012: 1491–1495.



- [2] CHEN Y, LU Y, LAN M, et al. A semi-supervised method for classification of semantic relation between nominals [C]// Proceedings of the 2010 International Conference on Asian Language Processing. Washington, DC: IEEE Computer Society, 2010: 146–149.
- [3] KAMBHATLA N. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations [C]// Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions. Stroudsburg, PA: Association for Computational Linguistics, 2004: 22.
- [4] RINK B, HARABAGIU S. UTD: classifying semantic relations by combining lexical and semantic resources [C]// Proceedings of the 2010 International Workshop on Semantic Evaluation. Stroudsburg, PA: Association for Computational Linguistics, 2010: 256–259.
- [5] SOCHER R, HUVAL B, MANNING C D, et al. Semantic compositionality through recursive matrix-vector spaces [C]// Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Stroudsburg, PA: Association for Computational Linguistics, 2012: 1201–1211.
- [6] YU M, GORMLEY M, DREDZE M. Factor-based compositional embedding models [C]// Proceedings of the 2014 NIPS Workshop on Learning Semantics. Cambridge, MA: MIT Press, 2014: 95–101.
- [7] XU Y, JIA R, MOU L, et al. Improved relation classification by deep recurrent neural networks with data augmentation [C]// Proceedings of the 2016 International Conference on Computational Linguistics. [S. l.]: The COLING 2016 Organizing Committee, 2016: 1461–1470.
- [8] ZENG D, LIU K, LAI S, et al. Relation classification via convolutional deep neural network [C]// Proceedings of the 2014 International Conference on Computational Linguistics. New York: ACM, 2014: 2335–2344.
- [9] VU N T, ADEL H, GUPTA P, et al. Combining recurrent and convolutional neural networks for relation classification [C]// Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: Association for Computational Linguistics, 2016: 534–539.
- [10] ZHOU P, SHI W, TIAN J, et al. Attention-based bidirectional long short-term memory networks for relation classification [C]// Proceedings of the 2016 Meeting of the Association for Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2016: 207–212.
- [11] LIU M X C. Semantic relation classification via hierarchical recurrent neural network with attention [C] // Proceedings of the 26th International Conference on Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2016: 1254–1263.
- [12] SHEN Y, HUANG X. Attention-based convolutional neural network for semantic relation extraction [C]// Proceedings of the 26th International Conference on Computational Linguistics. Stroudsburg, PA: Association for Computational Linguistics, 2016: 2526–2536.
- [13] 刘丹丹, 彭成, 钱龙华, 等. 词汇语义信息对中文实体关系抽取影响的比较[J]. 计算机应用, 2012, 32(8): 2238–2244. (LIU D D, PENG C, QIAN L H, et al. Comparative analysis of impact of lexical semantic information on Chinese entity relation extraction [J]. Journal of Computer Applications, 2012, 32(8): 2238–2244.)
- [14] 甘丽新, 万常选, 刘德喜, 等. 基于句法语义特征的中文实体关系抽取[J]. 计算机研究与发展, 2016, 53(2): 284–302. (GAN L X, WAN C X, LIU D X, et al. Chinese named entity relation extraction based on syntactic and semantic features [J]. Journal of Computer Research and Development, 2016, 53(2): 284–302.)
- [15] MINTZ M, BILLS S, SNOW R, et al. Distant supervision for relation extraction without labeled data [C]// Proceedings of the 2006 Joint Conference of Meeting of the ACL and International Joint Conference on Natural Language. Stroudsburg, PA: Association for Computational Linguistics, 2009: 1003–1011.
- [16] HENDRICKX I, SU N K, KOZAREVA Z, et al. SemEval-2010 task 8: multi-way classification of semantic relations between pairs of nominals [C]// Proceedings of the 2009 Workshop on Semantic Evaluations: Recent Achievements and Future Directions. Stroudsburg, PA: Association for Computational Linguistics, 2009: 94–99.
- [17] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate [EB/OL]. [2017-10-20]. <https://arxiv.org/abs/1409.0473>.
- [18] GRAVES A, WAYNE G, DANIELKA I. Neural Turing machines [EB/OL]. [2017-10-18]. <https://arxiv.org/abs/1410.5401>.
- [19] GRAVES A, WAYNE G, REYNOLDS M, et al. Hybrid computing using a neural network with dynamic external memory [J]. Nature, 2016, 538(7626): 471.
- [20] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735.
- [21] ZAREMBA W, SUTSKEVER I, VINYALS O. Recurrent neural network regularization [EB/OL]. [2017-10-25]. <https://arxiv.org/abs/1409.2329>.
- [22] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [EB/OL]. [2017-11-01]. <https://arxiv.org/abs/1301.3781>.
- [23] COLLOBERT R, WESTON J, KARLEN M, et al. Natural language processing (almost) from scratch [J]. Journal of Machine Learning Research, 2011, 12(1): 2493–2537.
- [24] PENNINGTON J, SOCHER R, MANNING C. Glove: global vectors for word representation [C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: Association for Computational Linguistics, 2014: 1532–1543.
- [25] KINGMA D, BA J. Adam: a method for stochastic optimization [EB/OL]. [2017-11-02]. <https://arxiv.org/abs/1412.6980>.

This work is partially supported by the Surface Program of National Natural Science Foundation of China (61572505).

**ZHANG Runyan**, born in 1994, M. S. candidate. His research interests include neural network, natural language processing.

**MENG Fanrong**, born in 1962, Ph. D., professor. Her research interests include intelligent information processing, database technology, data mining.

**ZHOU Yong**, born in 1974, Ph. D., professor. His research interests include data mining, wireless sensor network.

**LIU Bing**, born in 1981, Ph. D., associate professor. His research interests include machine learning, pattern recognition.