

文章编号: 1003-0077(2021)07-0030-11

## 融合词义信息的文本蕴涵识别方法

杜倩龙<sup>1,2</sup>, 宗成庆<sup>1,2</sup>, 苏克毅<sup>3</sup>

(1. 中国科学院自动化研究所 模式识别国家重点实验室, 北京 100190;

2. 中国科学院大学 人工智能学院, 北京 100049;

3. 台湾“中央研究院”资讯科学研究所, 台湾 台北 11529)

**摘要:** 文本蕴涵识别是对两个文本之间语义关系的有向推理, 而词汇的词义对理解文本的语义以及推理文本之间的语义蕴涵关系有着重要作用。因此, 为了有效利用词汇的词义信息推断文本之间的语义蕴涵关系, 该文提出一种融合词义信息的文本蕴涵识别方法。该方法首次提出将原始的词汇转化为对应的目标词义, 然后利用词汇的词义信息改善文本的语义表示和文本间语义关系的推理。实验表明, 该文所提出的方法可以有效改善文本间语义关系的推理, 从而提升文本蕴涵识别的准确率。

**关键词:** 词义推断; 文本蕴涵识别; 语义推理

**中图分类号:** TP391

**文献标识码:** A

## Incorporating Word Sense Information for Recognizing Textual Entailment

DU Qianlong<sup>1,2</sup>, ZONG Chengqing<sup>1,2</sup>, SU Keh-Yih<sup>3</sup>

(1. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China;

2. School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China;

3. Institute of Information Science, “Academia Sinica”(Taiwan), Taipei, Taiwan 11529, China)

**Abstract:** The task of recognizing textual entailment detects whether a given text passage can be inferred from another passage. During inference process, the sense of each word plays an important role in understanding the meaning of the passages and predicting the relationship of the passage-pair. In this paper, we propose a novel approach to incorporate word sense information into the inference mechanism. We first use a word sense disambiguation system to generate the sense of each content word, and then use the information of word sense to improve the representations of the passages and enhance the capability of predicting the entailment relationship of the passage-pair. Experimental results show that our proposed approach can improve the performance effectively.

**Keywords:** word sense disambiguation; recognizing textual entailment; semantic inference

## 0 引言

文本蕴涵识别(recognizing textual entailment, RTE)旨在利用各种分析手段有效理解文本的语义, 然后推理出文本之间的语义蕴涵关系。如表 1 所示, 该任务通常给定一个前提文本 P(premise)和一个假设文本 H(hypothesis), 然后判定是否可以从前提文本 P 的语义中推理出假设文本 H 的含义, 从而推断出文本 P 和文本 H 之间的关系是蕴涵

(entailment)、中立(neutral)还是矛盾(contradiction)。文本蕴涵识别有效解决了语言表达的多样性和歧义性问题, 被广泛应用于自动问答<sup>[1]</sup>和文本摘要<sup>[2]</sup>等自然语言处理任务中。为了有效评估文本蕴涵识别系统的性能, 在过去的十几年间学界组织了大量的文本蕴涵识别相关评测, 例如, RTE<sup>[3]</sup>和 SemEval-2014<sup>[4]</sup>等。早期的文本蕴涵识别相关研究都是在小规模的数据集之上, 基于人工定义的规则和特征, 利用传统的统计学方法、规则方法和逻辑推理方法等进行的文本间蕴涵关系的判定<sup>[5-6]</sup>。

收稿日期: 2020-10-12 定稿日期: 2020-12-16

基金项目: 国家自然科学基金(U1836221)

表 1 文本蕴涵识别样例

| ID  | 前提文本(P)   | 假设文本(H)                                   | 关系 |
|-----|---|---|----|
| (1) | Tom will join the board of the film company as a nonexecutive director. | Tom is a film director.                   | 中立 |
| (2) | The new rights are nice enough.   | Everyone really likes the newest benefits | 中立 |
| (3) | The computer is very cheap.   | The computer is quite inexpensive.        | 蕴涵 |
| (4) | I don't know what the overnight low was.                                | I don't know how cold it got last night.  | 蕴涵 |
| (5) | The boy in a blue shirt shook his head sadly.                           | The boy was very happy.                   | 矛盾 |

随着深度神经网络方法的快速发展和大规模文本蕴涵识别数据集 SNLI<sup>[7]</sup> 和 MultiNLI<sup>[8]</sup> 的出现, 基于神经网络的文本蕴涵识别方法<sup>[7-17]</sup> 开始大量涌现并取得了很好的效果。神经网络文本蕴涵识别方法主要采用神经网络结构对两个输入文本进行编码, 从而生成文本的向量表示, 然后对这两个输入文本的向量表示进行比较, 最终判定文本之间的语义蕴涵关系。在上述方法中, 为了更好地推理文本间的蕴涵关系, 有学者提出了基于文本间信息交互的神经网络文本蕴涵识别方法, 该类方法采用注意力机制对文本间不同粒度的语义信息进行对齐和比较, 并将不同粒度间的比较信息进行融合, 从而预测文本之间的语义蕴涵关系<sup>[11-14, 17]</sup>。基于文本间信息交互的识别方法在文本蕴涵识别数据集上取得了最优的效果, 并受到学者越来越多的关注。

然而, 当前基于信息交互的神经网络文本蕴涵识别方法<sup>[11-14, 17]</sup> 主要关注采用大规模的无监督数据对模型和词向量进行预训练, 或者构建更加复杂的网络结构对文本间的语义信息进行对齐和比较, 而忽略了词义信息对文本蕴涵识别的作用。

下面以表 1 中的文本对(1)为例:

P: Tom will join the board of the film company as a nonexecutive **director**.

H: Tom is a film **director**.

对于该文本对, 在对文本 P 和文本 H 之间的蕴涵关系进行判定时, 现有基于文本间信息交互的识别方法<sup>[11-14, 17]</sup> 通常采用注意力机制对文本之间的词汇表示进行对齐, 然后将文本中的词汇与其在另一个文本中的对齐信息进行比较, 最终通过融合所有的比较信息预测文本之间的关系标签。在上述方法中, 词汇“director”在文本 P 和文本 H 中具有相同的初始化向量。如果现有文本蕴涵识别模型无法有效地对词汇“director”在这两个文本中的词义表示进行建模, 模型将错误地将假设文本 H 和前提文本

P 之间的关系判定为“蕴涵”(假设文本 H 中所有词汇的表示向量均可以在前提文本 P 中找到其匹配项)。

针对上述问题分析之后发现, 文本中词汇的词义信息可以有效改善词汇在文本中的语义表示, 从而改进文本间语义关系的推理。例如, 对于文本对(1)中的词汇“director”, 其在文本 P 和文本 H 中对应着不同的词义。对于该词义, 其对应的语义分别为:

(a) member of a board of directors

译文: 董事会成员

(b) the person who directs the making of a film

译文: 导演

(a)和(b)分别为词汇“director”在文本 P 和文本 H 中的词义在 WordNet<sup>[18]</sup> 中对应的释义信息, 该释义信息很好地描述了目标词汇在对应上下文中的语义。因此, 为了有效构建词汇“director”在文本 P 和文本 H 中的向量表示, 本文首次提出了利用词义(a)和词义(b)对应的释义信息生成其表示向量, 并将该表示向量作为词汇“director”在对应文本中的初始化向量。在此基础上, 本文进一步提出了利用词义之间的关联信息(例如, 上下位关系和蕴涵关系等)改善文本之间的语义关系推理。综上所述, 本文提出了一个融合词义信息的文本蕴涵识别模型(incorporating word sense information for recognizing textual entailment, IWSI-RTE), 该模型利用词汇的词义信息改善文本中词汇的语义表示和文本间语义关系的推理, 从而有效提升了文本蕴涵识别的准确率。

## 1 相关工作

随着深度神经网络方法的快速发展和大规模文

本蕴涵识别数据集 SNLI<sup>[7]</sup> 和 MultiNLI<sup>[8]</sup> 的出现, 基于深度神经网络的文本蕴涵识别方法开始大量涌现, 并显著提升了文本蕴涵识别系统的性能。依据是否对文本间的语义信息进行交互, 可以将现有的神经网络文本蕴涵识别方法划分为两类: ①基于句向量的文本蕴涵识别方法<sup>[7,10]</sup>, 该类方法首先利用神经网络模型分别生成前提文本和假设文本的句向量表示, 然后对这两个文本的句向量表示进行比较; ②基于文本间信息交互的识别方法<sup>[11-14]</sup>, 该类方法利用注意力机制对文本间不同粒度的语义信息进行对齐和比较, 并将比较后的信息进行融合, 从而推断出文本之间的蕴涵关系。

在基于句向量的文本蕴涵识别方法中, Bowman 等人<sup>[7]</sup> 提出了利用长短期记忆网络 (long short-term memory, LSTM) 分别生成前提文本 (P) 和假设文本 (H) 的向量表示, 然后利用一个三层的前向网络对文本 P 和文本 H 的向量表示进行比较, 从而预测最终的关系标签。此外, 为了更好地抽取文本中词汇之间的依存信息, Mou 等人<sup>[10]</sup> 提出了基于树结构的神经网络识别方法, 该方法利用树结构的卷积神经网络对输入文本进行编码, 然后对编码后的向量表示进行比较。

在基于文本间信息交互的文本蕴涵识别方法中, Chen 等人<sup>[12]</sup> 于 2017 年提出了一个增强型序列推理模型 (enhanced sequential inference model, ESIM)。在该模型中, Chen 等人利用双向 LSTM 网络结构对两个输入文本进行处理, 从而生成文本中所有词汇的向量表示, 然后利用注意力机制计算两个文本中所有词汇之间的对齐权重。对于任意词汇, 利用对齐权重计算其在另一个文本中的相关语义向量, 并将该词汇的向量与其相关语义向量进行比较, 从而生成该词汇对应的比较信息。最终, 通过融合所有词汇的比较信息预测文本对之间的关系标签。在此基础上, Chen 等人<sup>[13]</sup> 于 2018 年进一步提出了一个基于结构知识的推理模型 (knowledge-based inference model, KIM), 该模型利用人工构造的知识资源 WordNet 抽取词汇之间的关联信息, 然后将其添加到 ESIM 模型的词汇对齐、局部推理和推理信息融合的过程中, 从而进一步提升了文本蕴涵识别的准确率。在利用 WordNet 抽取词汇之间的关联信息时, 该模型不对词汇的词义进行判定, 而是直接对词汇所有词义之间的关系进行比对。此外, 在 Transformer<sup>[19]</sup> 结构出现以后, Devlin 等人<sup>[20]</sup>

提出了利用深度双向 Transformer 结构编码文本表示的模型, 该模型在一个深层 Transformer 结构上利用大规模无监督数据对模型参数进行预训练, 然后在文本蕴涵识别任务上对模型参数进行微调。

现有识别方法主要关注如何构造更加复杂的网络结构对文本中的语义信息进行对齐和比较, 或者利用大规模的无监督数据对模型和词向量进行预训练, 而忽视了传统语言学信息在判定文本间语义蕴涵关系时的作用。不同于上述方法, 本文将探究传统语言学信息是否可以有效改善文本蕴涵识别的效果, 以及如何在神经网络模型中有效融合词汇的语言学信息。基于此, 本文提出了一种融合词义信息的文本蕴涵识别方法, 该方法利用输入文本中词汇的词义信息改善文本的语义表示和文本间语义关系的推理, 从而有效提升了文本蕴涵识别的准确率, 且具有很好的可解释性。不同于 KIM 模型中利用 WordNet 抽取词汇所有词义间的关联信息, 然后利用该关联信息改善词汇对齐和关系推理的方法, 本文提出了直接将输入文本转换为对应的词义序列, 然后对词义序列中的词义进行对齐和比较。此外, 为了更好地表示词义序列中的词义, 本文提出了利用 WordNet 中词义的释义信息生成词义的向量表示, 并利用词义间的关联信息改善词义间的关系推理。

## 2 融合词义信息的文本蕴涵识别方法

### 2.1 预处理

为了有效利用文本中词汇的词义信息推断文本间的语义蕴涵关系, 对于输入的前提文本 (P) 和假设文本 (H), 本文首先利用词义消歧工具对文本中所有的词汇进行处理, 从而推断出每个词汇对应的词义, 并利用 WordNet<sup>[18]</sup> 中的词义标签表示该词义。特别地, 对于在 WordNet 中不存在词义的词汇 (例如, 介词、冠词和情态动词等), 本文直接利用词汇本身表示其词义。如图 1 所示, 本文依据上述方法将原始的词义序列转换为一个新的词义序列, 该序列中的词义标签对应着目标词汇的词义。对于词义序列中标定的词义, 本文进一步提出了利用 WordNet 中词义的释义信息对其进行表示。表 2 展示了文本蕴涵识别样例中部分词汇的词义及其在 WordNet 中对应的释义信息。

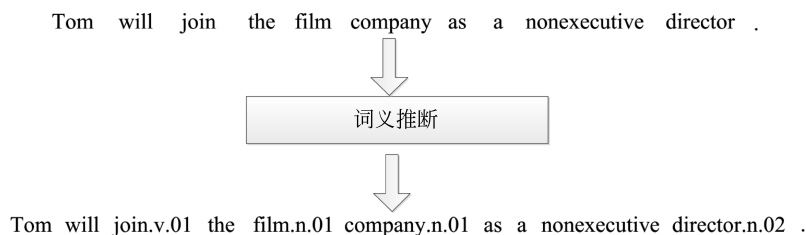


图 1 原始词汇序列转换为词义序列

表 2 输入文本中词汇的词义及其在 WordNet 中对应的释义

| 样例   | 词义            | 词义在 WordNet 中的释义                                 | 关系 |
|--|---------------|--|----|
| P: Tom will join the board of the film company as a nonexecutive <u>director</u> . | director.n.02 | member of a board of directors                   | 中立 |
| H: Tom is a film <u>director</u> .   | director.n.04 | the person who directs the making of a film      |    |
| P: The computer is very <u>cheap</u> .   | cheap.a.01    | relatively low in price or charging low prices   | 蕴涵 |
| H: The computer is quite <u>inexpensive</u> .                                      | cheap.a.01    | relatively low in price or charging low prices   |    |
| P: The boy in a blue shirt shook his head <u>sadly</u> .                           | sadly.r.02    | with sadness; in a sad manner                    | 矛盾 |
| H: The boy was very <u>happy</u> .   | happy.a.01    | enjoying or showing or marked by joy or pleasure |    |

## 2.2 模型方法

对于输入的文本 P 和文本 H,通过对其进行预处理分别生成对应的词义序列。为了有效推理文本间的语义蕴涵关系,本文将分别对输入文本的词汇序列和词义序列进行比较,并将比较信息进行融合,从而预测最终的关系标签。

图 2 展示了本文提出的 IWSI-RTE 模型的整体框架,该模型主要包含三层网络:①编码层,该层分别对文本的词汇序列和词义序列进行编码,从而分别生成词汇和词义的向量表示;②交互层,该层分别对两个文本的词汇序列和词义序列进行对齐和比较,从而分别生成对应的比较向量;③推理层,该层利用词汇及词义与另一个文本间的比较向量生成对应的局部蕴涵信息,然后将所有的局部蕴涵信息进行融合,从而预测文本间的蕴涵关系。

### 2.2.1 编码层

对于文本 P 和文本 H,该层分别对其词汇序列和词义序列进行编码,从而生成每个词汇和词义的向量表示。

#### (1) 编码词汇序列

定义两个输入文本的词汇序列分别为  $P = (p_1, p_2, \dots, p_m)$  和  $H = (h_1, h_2, \dots, h_n)$ 。  $p_i$  表示词汇序列 P 中的第  $i$  个词汇,而  $h_j$  则表示序列 H 中的第  $j$  个词汇。 $m$  和  $n$  为这两个序列对应的词汇

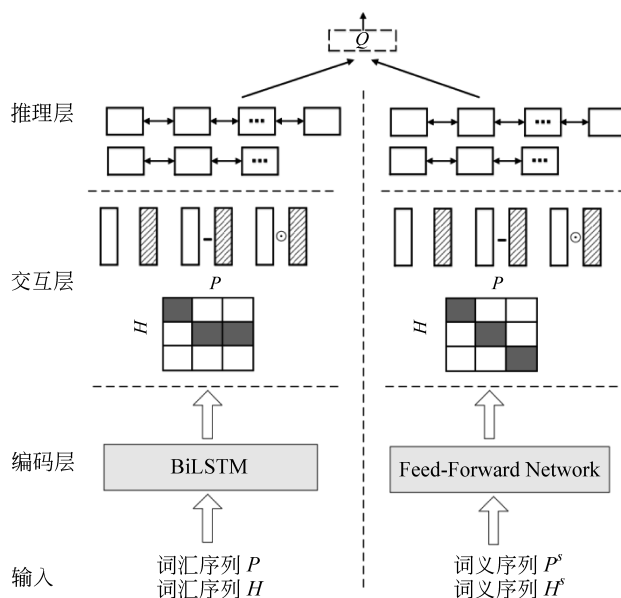


图 2 IWSI-RTE 模型的整体结构框架

数目。对于词汇序列 P 中的任意词汇  $p_i$ , 利用  $\hat{p}_i$  表示其对应的初始化向量,进而得到词汇序列 P 的初始化向量序列为  $\hat{P} = (\hat{p}_1, \hat{p}_2, \dots, \hat{p}_m)$ 。同理,得到序列 H 的初始化向量序列  $\hat{H} = (\hat{h}_1, \hat{h}_2, \dots, \hat{h}_n)$ 。

在对词汇进行初始化时,初始化向量  $\hat{p}_i$  和  $\hat{h}_j$  主要包含以下信息:

**词向量:** 本文采用 GloVe<sup>[21]</sup> 词向量对所有词汇



进行初始化。

**词汇特征：**受到 Chen 等人<sup>[22]</sup>工作的启发，本文在对词汇进行初始化时融合了一些语言学特征，包含了词性、命名实体和归一化的词频，并将其分别表示为 one-hot 向量。

**词汇匹配特征：**本文采用了三种文本间词汇的匹配特征，这些特征表示另一个文本中是否存在目标词汇的原始形式、小写形式和词干形式，并用一维的 0 或者 1 值表示这些特征。

将上述特征向量拼接得到词汇的初始化向量  $\hat{p}_i$  和  $\hat{h}_j$  后，将其分别输入到一个双向 LSTM 网络中编码词汇对应的向量表示，具体如式(1)、式(2)所示。

$$\{\tilde{p}_1, \dots, \tilde{p}_m = \text{BiLSTM}(\{\hat{p}_1, \dots, \hat{p}_m\})\} \quad (1)$$

$$\{\tilde{h}_1, \dots, \tilde{h}_n = \text{BiLSTM}(\{\hat{h}_1, \dots, \hat{h}_n\})\} \quad (2)$$

其中， $\tilde{p}_i \in R^{2d}$  和  $\tilde{h}_j \in R^{2d}$  分别为  $p_i$  和  $h_j$  在双向 LSTM 网络中对应的隐层向量。 $d$  表示隐层向量的维度。 $\tilde{p}_i$  和  $\tilde{h}_j$  分别表示词汇  $p_i$  和词汇  $h_j$  融合上下文信息后的向量表示。

## (2) 编码词义序列

如 2.1 节所述，通过预处理过程本文分别得到文本 P 和文本 H 的词义序列。定义  $P^s = (p_1^s, p_2^s, \dots, p_m^s)$  和  $H^s = (h_1^s, h_2^s, \dots, h_n^s)$  分别表示文本 P 和文本 H 的词义序列。 $p_i^s$  和  $h_j^s$  分别对应文本 P 和文本 H 中第  $i$  个和第  $j$  个词汇的词义。此外，定义  $\hat{P}^s = (\hat{p}_1^s, \hat{p}_2^s, \dots, \hat{p}_m^s)$  和  $\hat{H}^s = (\hat{h}_1^s, \hat{h}_2^s, \dots, \hat{h}_n^s)$  分别表示文本 P 和文本 H 词义序列的初始化向量序列， $\hat{p}_i^s$  和  $\hat{h}_j^s$  为词义  $p_i^s$  和  $h_j^s$  对应的初始化向量。

在初始化时，词义所对应的初始化向量  $\hat{p}_i^s$  和  $\hat{h}_j^s$  包含以下两个部分：

**词义向量：**对于词义序列中的每个词义，其在 WordNet<sup>[18]</sup>中对应着一个释义，本文通过对该释义中所有词汇的词向量采用最大值池化操作生成词义的向量表示。此外，对于无法利用 WordNet 标注词义的词汇，本文采用 GloVe<sup>[21]</sup>词向量对其进行初始化。

**词义特征：**本文采用了四种描述文本间词义关系的特征，这些特征表示另一个词义序列是否存在目标词义的同义词、下位义(hypernym)、上位义(hyponym)和存在蕴涵关系的词义。这四类特征表示为一维的 0 或者 1 值。此外，本文还添加了词义在对应序列中的归一化频率，并将该特征表示为一

个 one-hot 向量。

将词义向量和词义特征向量拼接得到词义的初始化向量  $\hat{p}_i^s$  和  $\hat{h}_j^s$  后，模型将对文本 P 和文本 H 的词义序列进行编码，从而生成每个词义对应的向量表示。由于词义序列中的词义表示已经包含了对应词汇在该文本中的语义信息，因此在对词义序列进行编码时模型将不再利用双向 LSTM 结构融合其上下文信息。对于初始化后的词义序列，本文将其输入到一个前向网络中进行处理，从而为每个词义生成一个新的表示，具体如式(3)、式(4)所示。

$$\tilde{p}_i^s = \text{Relu}(\mathbf{W}_e \hat{p}_i^s + \mathbf{b}_e) \quad (3)$$

$$\tilde{h}_j^s = \text{Relu}(\mathbf{W}_e \hat{h}_j^s + \mathbf{b}_e) \quad (4)$$

其中， $\tilde{p}_i^s \in R^d$ 、 $\tilde{h}_j^s \in R^d$  分别表示词义  $p_i^s$  和  $h_j^s$  的向量。 $\mathbf{W}_e \in R^{d \times d}$  和  $\mathbf{b}_e \in R^d$  为要学习的参数，Relu 表示激活函数 relu。

## 2.2.2 交互层

利用编码层分别生成词汇序列和词义序列的向量表示后，交互层将分别对文本 P 和文本 H 的词汇序列和词义序列进行对齐和比较，从而生成每个词汇及词义与另一个文本之间的比较向量。特别地，在比较时本文对词汇序列和词义序列采用相同的对齐和比较策略。下面将对文本间语义信息对齐和比较的具体过程进行介绍。

对于词汇序列中的词汇  $p_i$  和  $h_j$ ，首先利用向量的内积操作计算其对齐权重，具体计算如式(5)所示。

$$e_{i,j} = \tilde{p}_i^T \tilde{h}_j \quad (5)$$

其中， $e_{i,j}$  表示词汇  $p_i$  和词汇  $h_j$  之间的对齐权重。 $\tilde{p}_i$  和  $\tilde{h}_j$  分别为词汇  $p_i$  和  $h_j$  编码后的向量表示，具体如式(1)和式(2)所示。

计算得到词汇序列  $P$  和词汇序列  $H$  间所有词汇的对齐权重后，对于词汇  $p_i$ ，模型将利用该对齐权重对词汇序列  $H$  中所有的词汇信息进行融合，从而计算词汇  $p_i$  的相关语义向量  $u_i$ 。同理，利用对齐权重将词汇序列  $P$  中的所有词汇信息融合，从而计算词汇  $h_j$  的相关语义向量  $v_j$ 。

$$u_i = \sum_{j=1}^n \frac{\exp(e_{i,j})}{\sum_{k=1}^n \exp(e_{i,k})} \tilde{h}_j \quad (6)$$

$$v_j = \sum_{i=1}^m \frac{\exp(e_{i,j})}{\sum_{k=1}^m \exp(e_{k,j})} \tilde{p}_i \quad (7)$$

其中， $\tilde{p}_i$  和  $\tilde{h}_j$  表示词汇编码后的向量表示。

相关语义向量  $u_i$  融合了词汇  $p_i$  在词汇序列  $H$  中的相关语义信息,而  $v_j$  则融合了词汇  $h_j$  在序列  $P$  中的相关语义信息。

计算得到词汇的相关语义向量后,对于词汇  $p_i$ ,将其词汇向量  $\tilde{p}_i$  与其相关语义向量  $u_i$  进行比较,从而生成词汇  $p_i$  和词汇序列  $H$  间的比较向量;同理,将  $h_j$  的词汇向量和相关语义向量也进行比较,具体计算如式(8)、式(9)所示。

$$p_i^C = G([\tilde{p}_i; u_i; \tilde{p}_i - u_i; \tilde{p}_i \odot u_i]) \quad (8)$$

$$h_j^C = G([\tilde{h}_j; v_j; \tilde{h}_j - v_j; \tilde{h}_j \odot v_j]) \quad (9)$$

其中,  $G$  表示激活函数为  $\text{relu}$  的单层前向网络。“;”表示向量的拼接操作,“-”表示向量的减法操作,而“ $\odot$ ”则表示向量的点乘操作。 $p_i^C$  表示词汇  $p_i$  和词汇序列  $H$  间的比较向量; $h_j^C$  则表示词汇  $h_j$  和词汇序列  $P$  之间的比较向量。

与词汇序列间的对齐和比较策略相同,对于词义序列  $P^s$  中的词义  $p_i^s$ ,将其与词义序列  $H^s$  中的所有词义进行对齐和比较,从而得到  $p_i^s$  与词义序列  $H^s$  之间的比较向量  $p_i^{s-C}$ ;同理,计算得到词义  $h_j^s$  和词义序列  $P^s$  之间的比较向量  $h_j^{s-C}$ 。

### 2.2.3 推理层

该层分别利用词汇和词义的比较向量生成对应的局部蕴涵信息,然后将所有的局部蕴涵信息进行融合,从而预测文本间的关系标签。

#### (1) 序列间局部蕴涵信息推理及融合

为了有效利用词汇序列之间的比较向量生成对应的蕴涵信息,将词汇序列  $P$  和  $H$  对应的比较向量序列分别输入到双向 LSTM 中进行处理,从而得到每个词汇所对应的局部蕴涵信息,具体计算如式(10)、式(11)所示。

$$\{p_1^I, \dots, p_m^I = \text{BiLSTM}(\{p_1^C, \dots, p_m^C\})\} \quad (10)$$

$$\{h_1^I, \dots, h_n^I = \text{BiLSTM}(\{h_1^C, \dots, h_n^C\})\} \quad (11)$$

其中,  $p_i^I$  和  $h_j^I$  分别表示词汇  $p_i$  和词汇  $h_j$  所对应的词汇级别蕴涵信息。计算得到词汇级别的蕴涵信息后,本文将利用均值和最大值池化操作分别对序列  $P$  和序列  $H$  中的词汇级别蕴涵信息进行融合,具体计算如式(12)、式(13)所示。

$$\left. \begin{aligned} O_{P, \text{avg}} &= \sum_{i=1}^m p_i^I / m, & O_{P, \text{max}} &= \max_{i=1}^m p_i^I \\ O_P &= [O_{P, \text{avg}}; O_{P, \text{max}}] \end{aligned} \right\} \quad (12)$$

$$\left. \begin{aligned} O_{H, \text{avg}} &= \sum_{j=1}^n h_j^I / n, & O_{H, \text{max}} &= \max_{j=1}^n h_j^I \\ O_H &= [O_{H, \text{avg}}; O_{H, \text{max}}] \end{aligned} \right\} \quad (13)$$

其中,  $O_P$  和  $O_H$  表示将词汇级别蕴涵信息融合后得到的序列  $P$  和序列  $H$  之间的推理向量,且分别代表着不同的比较方向。 $O_P$  表示将词汇序列  $P$  中的词汇与序列  $H$  进行比较,  $O_H$  则表示将词汇序列  $H$  中的词汇与序列  $P$  进行比较。

得到上述推理向量后,将这两个向量进行拼接并输入到一个前向网络中,从而计算词汇序列  $P$  和词汇序列  $H$  间的全局推理向量,具体计算如式(14)所示。

$$O = F([O_P; O_H]) \quad (14)$$

其中,  $F$  是一个激活函数为  $\text{relu}$  的单层前向网络。 $O$  表示词汇序列  $P$  和词汇序列  $H$  之间的全局推理向量。

同理,对于词义序列  $P^s$  和  $H^s$  中的词义  $p_i^s$  和  $h_j^s$ ,本文利用与式(10)中相同的方法对其比较向量  $p_i^{s-C}$  和  $h_j^{s-C}$  进行处理,从而生成词义  $p_i^s$  和  $h_j^s$  所对应的推理向量。然后将所有词义对应的推理向量进行融合(如式(12)~式(14)所示),从而生成词义序列  $P^s$  和  $H^s$  之间的全局推理向量  $O^s$ 。

#### (2) 结果预测

利用上述操作分别得到词汇序列和词义序列间的全局推理向量后,将这两种不同类型的全局推理向量进行融合,并输入到一个多层前向网络  $Q$  中,从而预测文本之间的语义蕴涵关系,具体计算如式(15)所示。

$$\hat{y} = Q([O; O^s; \max(O, O^s); \text{avg}(O, O^s)]) \quad (15)$$

其中,  $Q$  表示一个两层的前向网络,其激活函数分别为  $\text{relu}$  和  $\text{softmax}$ 。 $O$  和  $O^s$  分别表示词汇序列和词义序列间的全局推理向量,  $\max$  和  $\text{avg}$  表示向量间的最大值和均值操作。 $\hat{y}$  表示最终的预测向量,向量维度为 3。

## 3 实验

### 3.1 实验数据

本文采用了两个数据集对模型性能进行测试,分别为: SNLI<sup>[7]</sup> 和 MultiNLI<sup>[8]</sup>。下面分别对其进行介绍。

**SNLI** 该数据集是 Bowman 等人<sup>[7]</sup>于 2015 年公开的第一个大规模文本蕴涵识别数据集,包含了 570 152 个英文文本对。在该数据集中,文本间的关系标签被标注为蕴涵、矛盾或者中立。特别地,由于标注者在部分文本对的关系上无法达成一致,

这些文本对间的关系标签被标注为“-”。在实验时,本文将标注为“-”的文本对删除。因此在真实的训练过程中,训练集包含了 549 367 个文本对,而开发集和测试集则分别包含了 9 842 和 9 824 个文本对。

**MultiNLI** 该数据集是由 Williams 等人<sup>[8]</sup>于 2018 年公开的大规模英文文本蕴涵数据集,一共包含了 43.3 万个文本对,且覆盖了多个领域的英文数据,包括书面语类、口语类和非正规书写类等。此外,为了更好地评估模型的性能,Williams 等人依据是否与训练数据来自相同的领域,将开发集和测试集划分为领域内数据(matched)和领域外数据(mismatched)两种类型。

### 3.2 实验设置

在对模型进行训练时,本文采用 Adam 算法对参数进行优化,并利用 300 维的 GloVe<sup>[21]</sup>词向量对输入文本进行初始化。基于相应的开发集,本文将批大小设置为 32,并采用大小为 0.000 4 的学习率,同时将 SNLI、matched 和 mismatched 数据集上网络结构的隐层维度分别设置为 400、400 和 450。此外,为了避免过拟合,本文还采用了 dropout 策略,并将概率值设置为 0.2。在预处理步骤对输入文本中词汇的词义进行标注时,本文采用了 Du 等人<sup>[23]</sup>的方法标注文本序列中词汇的词义。该方法利用词汇的局部上下文信息和词汇之间的结构依存信息推断目标词汇的词义,从而生成整个句子的词义序列。该方法采用词义消歧数据集 Semcor 对模型进行训练,然后对词汇的词义进行判定。

### 3.3 结果分析

表 3 展示了模型在 SNLI 数据集上的测试结果。在表 3 展示的模型中,Parikh 等人<sup>[11]</sup>提出了一个可分解注意力模型[模型(2)],该模型将文本蕴涵识别划分为对齐、比较和融合三个子任务进行处理。在此基础上,模型(6)提出了 ESIM 模型,该模型利用双向 LSTM 网络对文本中的词汇进行编码从而生成词汇的向量表示,然后利用句间注意力(inter-attention)机制对文本之间的词汇表示进行对齐和比较,最终预测文本间的蕴涵关系。该模型成为文本蕴涵识别领域的代表性模型。此外,模型(9)在 ESIM 模型的基础上进一步提出了一种利用 WordNet 中人工构造的结构知识改善文本间语义信息对齐和推理的方法,该方法不对文本中词汇的词义进行标注,而是直接利用词汇所有词义间的关

联信息改善文本间词汇的对齐和关系推理。模型(10)则在 ESIM 模型的基础上提出了利用大规模无标注数据改善词汇的向量表示,从而提升了文本蕴涵识别系统的推理性能。

表 3 模型在 SNLI 测试集上的准确率

| 模型   | 准确率/<br>% |
|--|-----------|
| (1) mLSTM word-by-word attention model <sup>[24]</sup> | 86.1      |
| (2) decomposable attention model <sup>[11]</sup>       | 86.8      |
| (3) RHDTM <sup>[25]</sup>                              | 87.4      |
| (4) BiMPM <sup>[17]</sup>                              | 87.5      |
| (5) re-read LSTM <sup>[26]</sup>                       | 87.5      |
| (6) ESIM <sup>[12]</sup>                               | 88.0      |
| (7) DIIN <sup>[27]</sup>                               | 88.0      |
| (8) DR-BiLSTM <sup>[28]</sup>                          | 88.6      |
| (9) KIM <sup>[13]</sup>                                | 88.6      |
| (10) ESIM+ELMo <sup>[16]</sup>                         | 88.7      |
| (11) Human Performance <sup>[27]</sup>                 | 87.7      |
| (12) IWSI-RTE  | 88.6      |

不同于上述方法,本文在 ESIM 模型的基础上提出了融合词义信息的文本蕴涵识别模型 IWSI-RTE,该模型利用词汇的词义信息改善文本的语义表示和文本间语义关系的推理。从表 3 中可以看出,IWSI-RTE 模型在 SNLI 数据集上的准确率达到 88.6%,优于表中大多数模型。特别地,相比于基线模型 ESIM,IWSI-RTE 模型的性能有明显提升。此外,相比于模型(9)中利用结构知识改善词汇对齐和词汇间关系推理的方法及模型(10)中采用大量无标注数据改善文本语义表示的方法,IWSI-RTE 模型在性能上也是可比的。相比于人工标注的准确率[模型(11)],IWSI-RTE 模型在 SNLI 数据集上也取得了更好的效果。

表 4 展示了不同模型在 MultiNLI 数据集上的测试结果。模型(1)和模型(2)为基于句向量的文本蕴涵识别模型。模型(3)为基于文本间信息交互的 ESIM 模型,该结果为 Williams 等人<sup>[8]</sup>发布 MultiNLI 数据集时所公布的 ESIM 模型在该数据集上的测试结果。模型(4)和模型(5)同样为基于句向量的文本蕴涵识别模型。在模型(4)中,Chen 等人<sup>[29]</sup>采用了句内门注意力(intra-sentence gated-attention)机制生成文本的向量表示。模型(5)则利用堆 LSTM 及一种跳跃式传递机制生成文本的向量表示。模型(6)为基于词汇依存三元组间独立比较的模型。模型(7)则在 ESIM 模型的基础上利用大量无标注数

据改善输入词汇的向量表示。此外,在模型(8)中,Chen 等人<sup>[13]</sup>利用外部知识改善文本间语义信息的比较和推理。在模型(9)中,Tay 等人<sup>[14]</sup>采用了一种因式分解网络对文本间的关系进行压缩,然后利用压缩后的信息预测文本间的关系标签。模型(10)则提出了利用卷积神经网络对文本间的语义信息进行交互和比较。

从表 4 中可以看出,本文提出的 IWSI-RTE 模型在 MultiNLI 的领域内(matched)测试集上优于所有的相关模型。此外,在 MultiNLI 的领域外(mismatched)测试集上,IWSI-RTE 模型的性能明显优于模型(1)~模型(8),且与模型(9)和模型(10)的性能也是可比的。特别地,相比于模型(7)中采用大量无标注数据改善词汇表示的方法,以及模型(8)中同样利用 WordNet 中的结构信息改善词汇对齐

和关系推理的方法,IWSI-RTE 模型在这两个数据集上的性能具有明显的提升。

### 3.4 模型分析

表 5 展示了模型中不同策略对最终结果的影响。模型(1)为本文最终的 IWSI-RTE 模型。模型(2)表示在对 IWSI-RTE 模型中的词义表示进行初始化时,该模型仅采用了词义向量,而没有添加词义特征信息。相比于 IWSI-RTE 模型,模型(2)的结果在这三个数据集上分别下降了 0.3%,1.0%和 0.4%。结果表明,本文所采用的词义特征信息可以改善文本间语义关系的推理。模型(3)表示在模型(2)的基础上继续去除词义向量间的比较信息,仅利用词汇信息对文本间的语义信息进行对齐和比较。从表中可以看出:相比于模型(2),当在模型(3)中仅利用词汇信息判定文本间的蕴涵关系时,模型的性能分别下降了 0.1%,1.0%和 1.0%。这表明本文所采用的词义向量信息也可以改善文本间语义关系的推理,且与词汇特征信息对推理性能的影响大致相当。模型(4)在模型(3)的基础上继续去除了词汇特征和词汇匹配特征信息。模型(4)为本文的基线系统,即 ESIM 模型,所呈现的结果为本文在实现时对 ESIM 模型部分参数进行调整后得到的结果。为了测试词义信息编码时不同网络结构对识别效果的影响,在模型(5)中本文展示了利用 BiLSTM 网络对词义序列进行编码时模型的识别性能。从表中可以看出,相比于 IWSI-RTE 模型,模型(5)的性能在所有数据集上均有所下降。这表明,采用前向网络结构能够更好地生成词义的向量表示,且具有更好的可解释性。

表 4 模型在 MultiNLI 测试集上的准确率

| 模型  | 准确率/%   |            |
|---|---------|------------|
|   | matched | mismatched |
| (1) BiLSTM <sup>[8]</sup>                   | 67.0    | 67.6       |
| (2) Inner Att <sup>[30]</sup>               | 72.1    | 72.1       |
| (3) ESIM <sup>[8]</sup>                     | 72.3    | 72.1       |
| (4) Gated-AttBiLSTM <sup>[29]</sup>         | 73.2    | 73.6       |
| (5) Shortcut-stacked Encoder <sup>[9]</sup> | 74.6    | 73.6       |
| (6) RHDTM <sup>[25]</sup>                   | 75.1    | 74.7       |
| (7) BiLSTM+ELMo+Attn <sup>[20]</sup>        | 76.4    | 76.1       |
| (8) KIM <sup>[13]</sup>                     | 77.2    | 76.4       |
| (9) CAFÉ <sup>[14]</sup>                    | 78.7    | 77.9       |
| (10) DIIN <sup>[27]</sup>                   | 78.8    | 77.8       |
| (11) Human Performance <sup>[27]</sup>      | 88.5    | 89.2       |
| (12) IWSI-RTE                               | 78.9    | 77.4       |

表 5 消融分析

| 模 型  | 测试集准确率/% |         |            |
|--|----------|---------|------------|
|  | SNLI     | matched | mismatched |
| (1) IWSI-RTE   | 88.6     | 78.9    | 77.4       |
| (2) baseline + additional features + sense embedding | 88.3     | 77.9    | 77.0       |
| (3) baseline+additional features                     | 88.2     | 76.9    | 76.0       |
| (4) baseline   | 88.0     | 76.3    | 75.8       |
| (5) encode sense emb with BiLSTM                     | 88.3     | 78.8    | 76.5       |
| (6) concat inference vector                          | 88.3     | 78.1    | 76.7       |
| (7) concat inference vector + avg                    | 88.0     | 77.9    | 77.3       |
| (8) concat inference vector + max                    | 88.5     | 78.2    | 76.7       |
| (9) sense embedding (sum)                            | 88.3     | 78.1    | 76.7       |
| (10) sense embedding (avg)                           | 88.4     | 77.9    | 76.4       |



模型(6)~模型(8)展示了不同的词义信息融合策略对测试结果的影响。如式(15)所示,在对词汇序列以及词义序列间的全局推理向量进行融合时,IWSI-RTE模型首先对词汇序列间的全局推理向量和词义序列间的全局推理向量进行均值操作和最大值操作,从而生成两个全局推理向量间的均值向量和最大值向量。然后,模型将该均值向量和最大值向量与对应的全局推理向量进行拼接并输入到一个前向网络中,从而预测文本间的语义蕴涵关系。不同于上述方法,模型(6)直接将词汇序列间的全局推理向量与词义序列间的全局推理向量进行拼接,然后预测文本间的关系标签。模型(7)则在模型(6)的基础上添加了全局推理向量间的均值操作。同理,模型(8)在模型(6)的基础上添加了全局推理向量间的最大值操作。从表中可以看出,IWSI-RTE模型中融合三类推理向量(原始推理向量、推理向量间均值和推理向量间最大值)的方法取得了最好的效果。

模型(9)和模型(10)则展示了词义向量的生成策略对测试结果的影响。在IWSI-RTE模型中,本文利用最大值池化操作对词义释义信息中所有词汇的向量表示进行处理,从而生成词义的向量表示。不同于该方法,模型(9)采用求和池化操作对词义释义信息中所有词汇的向量表示进行融合,而模型(10)则采用均值池化操作对词义释义信息中所有词汇的向量表示进行融合。从表中可以看出,IWSI-RTE模型中采用最大值池化操作的方法取得了最好的效果。

词汇的词义信息可以有效改善文本中多义词及同义词的向量表示,从而改善文本间语义信息的对齐和比较。为了分析词义信息对不同类型文本对间蕴涵关系判定的影响,本文在SNLI和MultiNLI领域内数据集<sup>①</sup>不同类型文本对上对所提模型的性能进行了测试。如表6所示,本文展示了SNLI开发集中不同类型文本对的数目及模型的测试结果。表中的模型“baseline + additional features”表示未融合词义信息的识别模型。从表中可以看出,当融合

表6 SNLI 开发集上不同标签类型的测试结果

|                                | 整体    | 标签类别  |       |       |
|--------------------------------|-------|-------|-------|-------|
|                                |       | 蕴涵    | 矛盾    | 中立    |
| Num                            | 9 842 | 3 329 | 3 278 | 3 235 |
| baseline+additional features/% | 88.8  | 90.6  | 90.4  | 85.4  |
| IWSI-RTE/%                     | 89.2  | 91.4  | 90.5  | 85.7  |

词义信息时,IWSI-RTE模型在不同类型文本对上的性能均有一定的提升。特别地,在语义关系为“蕴涵”的文本对上,模型IWSI-RTE的性能提升要优于另外两种类型的文本对。

此外,表7展示了模型在MultiNLI领域内开发集上的测试结果。从表中可以看出,模型IWSI-RTE的性能在开发集中不同类型文本对上的提升大致相当。整体来说,在该数据集中词义信息对不同类型文本对的性能提升没有明显的差异。综上所述,多义词或同义词问题与具体的文本对类型没有一定的关联性,而是与数据集中的样本分布相关。

表7 MultiNLI 领域内开发集上不同标签类型的测试结果

|                                | 整体    | 标签类别  |       |       |
|--------------------------------|-------|-------|-------|-------|
|                                |       | 蕴涵    | 矛盾    | 中立    |
| Num                            | 9 815 | 3 479 | 3 213 | 3 123 |
| baseline+additional features/% | 77.9  | 79.0  | 80.9  | 73.5  |
| IWSI-RTE/%                     | 78.6  | 79.7  | 81.7  | 74.1  |

### 3.5 词义作用分析

如表8所示,为了验证词义消歧准确率对文本蕴涵识别的影响,本文展示了采用不同词义消歧方法时文本蕴涵识别模型的性能。IWSI-RTE(MFS)表示采用最大频率词义(the most frequent sense, MFS)消歧方法对文本中词汇的词义进行判定。MFS词义消歧方法在词义消歧标准数据集 senseval-2 和 senseval-3 上分别达到了 61.9% 和 62.4% 的准确率,而 IWSI-RTE 方法中采用的 Du 等人<sup>[23]</sup>的词义消歧方法则分别达到了 66% 和 69% 的词义消歧准确率。实验结果表明,当词义消歧的准确率提升时,文本蕴涵识别系统的推理性能也有明显提升。由于现有的词义消歧方法还无法取得很好的效果,这也成为制约本文所提出的文本蕴涵识别方法的瓶颈。然而,论文中的实验结果表明,在词义消歧结果不是很理想的情况下,文本蕴涵的识别性能仍然有所改进。该结果再次表明,词汇的词义对判定文本之间的语义关系有着重要作用。在未来的工作中,我们将继续探究如何改善词义的判定效果,从而进一步提升文本蕴涵识别系统的推理性能。

① 由于 MultiNLI 数据集未公开测试集中文本对之间的语义蕴涵关系,因此本文在开发集上对模型在不同类型文本对上的结果进行分析。

表 8 词义作用分析

| 模型            | 开发集准确率/% |         |            |
|---------------|----------|---------|------------|
|               | SNLI     | matched | mismatched |
| IWSI-RTE(MFS) | 88.9     | 77.8    | 78.0       |
| IWSI-RTE      | 89.2     | 78.6    | 79.1       |

## 4 结束语

为了有效利用词汇的词义信息改善文本的语义表示和文本间语义关系的推理,本文提出了一种融合词义信息的文本蕴涵识别方法。通过对文本中词汇的词义进行标注,该方法将原始的词汇序列转化为新的词义序列进行对齐和比较。特别地,为了更好地表示词汇在对应文本中的词义,本文提出了从 WordNet 中抽取目标词义的释义信息并利用该释义信息生成词义的向量表示。此外,本文还提出了在词义表示中融合文本间的词义关系特征,从而提升了文本间语义关系的推理性能。实验表明,本文所提出的方法可以有效改善文本蕴涵识别系统的推理性能,且具有很好的可解释性。

## 参考文献

- [1] 王宝鑫,郑德权,王晓雪,等.基于文本蕴含的选择类问题解答技术研究[J]. 北京大学学报(自然科学版), 2016, 52(1), 134-140.
- [2] Li H R, Zhu J N, Zhang J J, et al. Ensure the correctness of the summary: Incorporate entailment knowledge into abstractive sentence summarization [C]// Proceedings of the 27th International Conference on Computational Linguistic, 2018:1430-1441.
- [3] Dagan I, Glickman O, Magnini B. The Pascal recognising textual entailment challenge [C]// Proceedings of the International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment, 2005: 177-190.
- [4] Marelli M, Bentivogli L, Baroni M, et al. Semeval-2014 task1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment [C]// Proceedings of the 8th International Workshop on Semantic Evaluation, 2014:1-8.
- [5] 刘茂福,李妍,姬东鸿.基于事件语义特征的中文文本蕴含识别[J]. 中文信息学报, 2013, 27(5):129-137.
- [6] 张晗,盛雅琦,吕晨,等.基于短文本隐含语义特征的文

本蕴涵识别[J]. 中文信息学报, 2016, 30(3):163-171.

- [7] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference [C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015: 632-642.
- [8] Williams A, Nangia N, Bowman S R. A broad-coverage challenge corpus for sentence understanding through inference [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, 2018: 1112-1122.
- [9] Nie Y, Bansal M. Shortcut-stacked sentence encoders for multi-domain inference [C]// Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP. Copenhagen, Denmark: Association for Computational Linguistics, 2017:41-45.
- [10] Mou L L, Li G, Zhang L, et al. Convolutional neural networks over tree structures for programming language processing [C]// Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016:1287-1293.
- [11] Parikh A, Tackstrom O, Das D, et al. A decomposable attention model for natural language inference [C]// Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016: 2249-2255.
- [12] Chen Q, Zhu X, Ling Z H, et al. Enhanced LSTM for natural language inference [C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 1657-1668.
- [13] Chen Q, Zhu X, Ling Z H, et al. Neural natural language inference models enhanced with external knowledge [C]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne, Australia: Association for Computational Linguistics, 2018: 2406-2417.
- [14] Tay Y, Tuan L A, Hui S C. A compare propagate architecture with alignment factorization for natural language inference [C]// Proceedings of the CoRR, 2018.
- [15] Rocktäschel T, Grefenstette E, Hermann K M, et al. Reasoning about entailment with neural attention [C]// Proceedings of the International Conference on Learning Representations (ICLR), 2016.
- [16] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, 2018:2227-2237.
- [17] Wang Z G, Hamza W, Florian R. Bilateral multi-perspective matching for natural language sentences

- [C]//Proceedings of the International Joint Conference on Artificial Intelligence, 2017: 4144-4150.
- [18] Miller G A. WordNet: A lexical database for English [J]. Communications of the ACM, 1995, 38(11):39-41.
- [19] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C]//Proceedings of the 31st International Joint Conference or Neural Information Processing Systems, 2017:5998-6008.
- [20] Devlin J, Chang M, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 2019: 4171-4186.
- [21] Pennington J, Socher R, Manning C D. GloVe: Global vectors for word representation [C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014: 1532-1543.
- [22] Chen D, Fisch A, Weston J, et al. Reading Wikipedia to answer open-domain questions [C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 1870-1879.
- [23] Du Q L, Zong C Q, Su K Y. Integrating structural context with local context for disambiguation word sense [C]//Proceedings of the 5th Conference On Natural Language Processing and Chinese Computing & The 24th International Conference On Computer Processing of Oriental Languages, 2016: 3-15.
- [24] Wang S, Jiang J. Learning natural language inference with LSTM [C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016.
- [25] Du Q L, Zong C Q, Su K Y. Adopting the word-pair-dependency-triplets with individual comparison for natural language inference [C]//Proceedings of the 27th International Conference on Computational Linguistics (COLING), Santa Fe, New-Mexico, USA, 2018: 414-425.
- [26] Sha L, Chang B B, Sui Z F, et al. Reading and thinking: Reread LSTM unit for textual entailment recognition [C]//Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, 2016: 2870-2879.
- [27] Gong Y C, Luo H, Zhang J. Natural language inference over interaction space [C]//Proceedings of the International Conference on Learning Representations, 2018.
- [28] Ghaeini R, Hasan S A, Datla V, et al. DR-BiLSTM: Dependent reading bidirectional LSTM for natural language inference [C]//Proceedings of the NAACL-HLT, 2018: 1460-1469.
- [29] Chen Q, Zhu X D, Ling Z H, et al. Recurrent neural network-based sentence encoder with gated attention for natural language inference [C]//Proceedings of the 2nd Workshop on Evaluating Vector-Space Representation for NLP, 2017: 36-40.
- [30] Balazs J A, Marrese-Taylor E, Loyola P, et al. Reining raw sentence representations for textual entailment recognition via attention [C]//Proceedings of the 2nd Workshop on Evaluating Vector-Space Representations for NLP, 2017: 51-55.



杜倩龙(1989—), 博士研究生, 主要研究领域为词义消歧和自然语言推理。

E-mail: qianlong.du@nlpr.ia.ac.cn



苏克毅(1955—), 博士, 研究员, 主要研究领域为机器翻译、机器阅读和自然语言理解。

E-mail: kysu@iis.sinica.edu.tw



宗成庆(1963—), 通信作者, 博士, 研究员, 主要研究领域为自然语言处理理论与方法、机器翻译、文本数据挖掘和人机对话系统。

E-mail: cqzong@nlpr.ia.ac.cn