文章编号：1003-0077（2017）00-0000-00

基于深度学习的文本蕴含关系识别

刘永志[1] 黄生斌[1,2] 乔春庚[1] 王洪俊[1]
（1.北京拓尔思信息技术股份有限公司，北京 100101；2.北京信息科技大学 计算机学院，北京 100101）

摘要：现有基于深度学习的中文文本蕴含关系识别方法存在模型复杂且效果提升不明显的问题，该文提出了一种基于卷积神经网络的中文文本蕴含识别方法。在 CCL2018 文本蕴含测评任务中，摒弃了复杂的深度学习模型，而仅仅运用了改进的卷积神经神经网络对文本蕴含关系进行识别，在 test 数据集上正确率达到了 82.38%,并最终摘得测评桂冠。该文从测评任务出发，对实现思想和模型进行详细介绍，并与其它模型的测评结果进行了对比。实验结果表明，该方法在文本蕴含识别上简单而有效。

关键词：文本蕴含；卷积神经网络；深度学习

中图分类号：TP391 **文献标识码：**A

0 引言

文本蕴含识别（recognizing textual entailment, RTE）是自然语言处理领域的一项基础研究，是一种单向关系识别问题。文本蕴含给定两个文本分别记为 T(Text)和 H(Hypothesis)，如果能从 T 推出 H 为正确的话，则称 T 蕴含 H。准确地来说，蕴含关系分三种：

1) T 推出 H 为真，称 T 和 H 的关系为蕴含关系（Positive Textual Entailment）。
2) T 推出 H 为假，则称 T 和 H 的关系为矛盾关系(Contradiction)。
3) H 和 T 没有任何的关系，则称 T 和 H 的关系为中性关系(Neutral)。

文本蕴含识别可以很好地辅助自然语言处理的其他领域，具有丰富的应用场景。比如，在机器翻译领域，文本蕴含关系识别可以用来对比翻译的文本和标准答案之间的匹配程度，从而评价翻译质量；在问答系统中，可以运用文本蕴含技术对语料库句子进行简单推理直接生成答案或对答案进行筛选排序进而提高回答的正确率；文本蕴含还可以应用于句法分析评价、个人智能助理等领域。

在大数据与深度学习快速发展的背景下，近年来，深度学习在文本蕴含领域的应用也得到了突破性的进展。Lyu 等人[1]首次将受限波尔兹曼机（Restricted Boltzmann Machines, RBM）模型应用到文本蕴含领域；随后，RNN、CNN、LSTM 等相继应用到文本蕴含识别，目前，基于深度学习的文本蕴含识别方法虽取得了比较好的效果。

本文提出了一种基于 CNN 的文本蕴含识别方法，方法对 CNN 进行了加强和改进，首先对文本进行分词、去除停用词后用 word2vec 进行词向量训练，这些预处理完成后对模型进行训练，训练完后将文本送入卷积神经网络，经过嵌入层、卷积层、池化层对特征进行提取后再送入一个融合层对提取的特征进行融合，最后连接一个全连接层，通过 Softmax 函数预测分类，最终得到识别结果。该方法在 CNLI2018 测试集上正确率达 82.38%，高出第二名 4.1 个百分点。

1 基于深度学习的文本蕴含识别

深度学习技术条件的日趋成熟促使了基于深度学习的文本蕴含识别技术的发展。

递归神经网络(Recursive neural network)出现后被应用到自然语言处理的多个领域，如：句法分析、情感分析等。Bowman 等人[2]首次将递归神经网络应用到文本蕴含领域，其方法为先对两个文本建模，将得到的结果送入下一层进行比

收稿收稿日期：2017-03-16；**定稿日期**：2017-04-26 六号
基金项目：基金名（基金号）；基金名(基金号)
六号，核实准确完整的基金名称

较，最后通过 Softmax 函数得到分类结果。

卷积神经网络（Convolutional Neutral Networks）的出现以及在图像处理方面的重大突破促使了其在自然语言处理领域的尝试和研究。在文本蕴含方面，Yin 等人[3]提出了利用 CNN 来处理 RTE 问题，该方法在识别蕴含关系的过程中，在对一个文本建模的过程中参照了另一文本的信息，从而取得了不错的效果。

随后，各种基于卷积神经网络的文本蕴含识别方法相继出现，有基于句法树的卷积神经网络等，这些网络相对来说要训练的参数较多，模型相对复杂，付出的代价较大但提升效果不太明显，基于这个问题，Ankur P. Parikh 等人[4]提出了一种可分解的模型对文本蕴含关系进行识别，特点是模型简单、参数少并且能达到其他复杂模型的相同甚至更好的效果。该方法主要运用了 Attention 机制[5-6]、应用在机器翻译领域的 alignment 机制[7]以及自然语言推理 NLI[8-10]，方法首先将句子做分词，转换为词向量然后利用 Attention 机制训练 Attention 矩阵得到每个词的对齐字短语，随后计算比较每个词与其对齐字短语的相似程度，最后合并比较结果并将其送入 Softmax 函数得到最终分类结果。

LSTM 在自然语言处理领域应用相对广泛，同样，不少学者提出了基于 LSTM 的文本蕴含识别方法。Bowman 等人[11]首次将 LSTM 模型应用到 RTE 领域，Attention 机制依然在 LSTM 模型中起到了关键性作用，随后又有带 Attention 机制的双向 LSTM 模型[12]、mLSTM 模型[13]、LSTMN 模型[14]、增强的 LSTM 模型[15]等。

基于深度学习的文本蕴含方法是目前的主要研究方向，方法相对传统的基于相似度对比的识别方法、基于对齐的识别等方法来说具有较高的识别准确率、并且可移植性强的特点，但是，一般神经网络需要训练的参数大，同时需要大量的预料才能得到比较好的效果。

2 基于 CNN 的中文文本蕴含识别

本文方法首先对文本进行分词、去除停用词等预处理，之后将文本嵌入到低维的稠密向量中，再使用 CNN 对文本进行提取特征，接着将提取到的特征进行特征融合[16-18]，最后使用全连接层进行分类实现最终的识别结果。具体模型如图 1 所示。

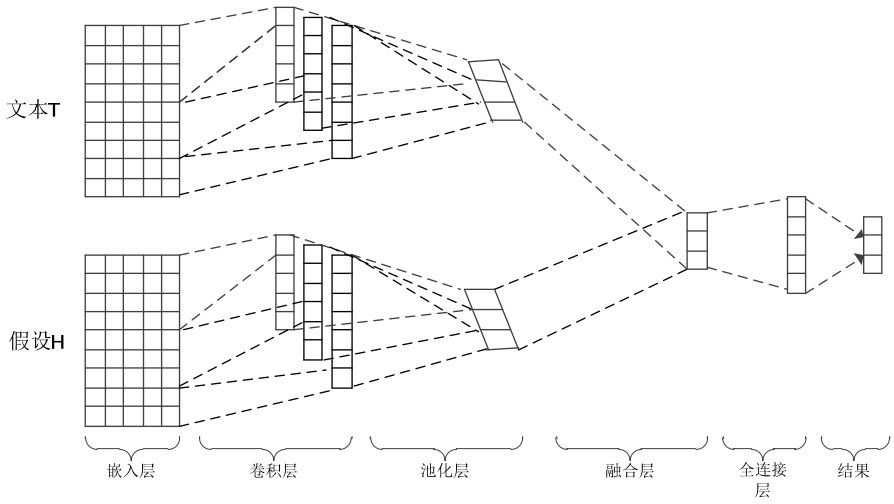


图 1 基于 CNN 的中文文本蕴含识别模型图

2.1 预处理

2.1.1 语料拆分

由于语料中数据 ID、蕴含关系、文本 T 和假设 H 是在同一行中用制表符隔开保存的，文本系统需要对文本 T 和假设 H 分开进行特征提取，为了处理的方便性，本文首先将语料中的数据 ID、蕴含关系、文本 T 和假设 H 拆分成单独一行的表示形式。

2.1.2 中文分词

在英文文本中，单词之间是以空格作为自然分界符的，而中文文本中只有字、句和段落之间能够通过明显的标识或分界符来进行简单的区分，词语之间没有一个标准的分界符，本文使用北京拓尔思信息技术股份有限公司内部研发的分词系统对语料进行分词处理。

2.1.3 去除停用词

人类语言中包含很多功能词，与其他词相比，功能词没有什么特殊的含义。如英文中的“a”、“the”、“that”等限定词，中文中的“是”、“的”等词。这些词在语料中的词频一般较高且很少能够单独表达语句的相关信息，因此在信息检索中常常被作为停用词[19]处理。为了减少停用词对语句造成的噪声干扰，文本在进行提取特征之前进行去除停用词处理。

2.1.4 文本对齐

为方便后续的卷积操作，本文需要将分词后的文本进行对齐操作。本文预设的文本最大长度为 100 个词，如果语料中的所有文本的长度都没有超过预设的最大长度，那么系统将以语料中所有文本的最大长度为最终的最大长度，超过最大长度的部分将会被丢弃。在将词语映射为唯一 id 时，词的下标从数字 1 开始编号，数字 0 作为对齐时补充不足部分的编号。

2.1.5 词嵌入

本文将经过分词后的文本 T 和假设 H 混合保存在同一个文档中，然后使用 word2vec 工具对该文档进行词向量的训练，并将训练完成后的结果文件作为整个系统的输入文件之一。

2.2 网络结构

2.2.1 嵌入层

嵌入层通过加载预处理中得到的词向量文件，将文本中的每个句子映射到低维的向量表示形式，如图 1 中所示，每一行对应句子中的一个词，列数表示嵌入的向量维度，即一个包含 n 个词的句子经嵌入到 d 维的向量中后，在计算机中可以表示成 $n\times d$ 的矩阵。

2.2.2 卷积层

二十世纪六十年代，Hubel 和 Wiesel 在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现其独特的网络结构可以有效地降低反馈神经网络的复杂性，提出了感受野的概念[20]。1998 年，纽约大学的 Yann LeCun 提出了卷积神经网络（Convolutional Neural Networks，简称 CNN），并成功地将其应用到了手写数字识别系统中[21]。近年来，卷积神经网络在模式分类、运动分析、自然语言处理等方面均有突破。

与传统技术相比，CNN 具有良好的容错能力、并行处理能力和自学习能力。CNN 的精华之处主要包括局部感受野、权值共享和降采样三大技术特点，以及卷积和池化两种核心操作。在卷积层中，一个神经元只与部分邻层神经元相连。在 CNN 的一个卷积层中，每一个卷积滤波器（卷积核）都重复的作用于整个感受野中，对输入对象进行卷积操作，从而提取出对象的局部特征。卷积滤波器在整个感受野提取特征的过程中其权值是不变的，通过这种权值共享的方式，使要学习的卷积神经网络模型参数的数量大大降低。

如图 1 中所示，卷积层使用多个 $m\times d$ 的滤波器（m 为滤波器的窗口大小，d 为嵌入的向量维度）来对嵌入层进行卷积操作，通过使用不同窗口大小以及多个同一窗口的滤波器可以使系统自动地提取到文本中的不同特征，再将每个卷积核提取的特征连接起来作为整个卷积层的输出。本文为了降低同一个语句对（文本 T 和假设 H）之间相互造成的信息干扰，分别用不同的多个卷积核进行提取特征。

2.2.3 池化层

池化操作是一种非线性降采样方法，常用的有平均池化和最大池化。本文使用最大池化在保证卷积得到的结果的最重要特征不丢失的前提下来对卷积特征进行降维，以此减少计算量增强网络的鲁棒性。

2.2.4 融合层

在卷积层和池化层中，对文本 T 和假设 H 是分开进行操作的，为了获取语句对之间的语义关系，需要对池化层的输出进行融合。

2.2.5 全连接层

全连接层是神经网络的重要组成部分，在本文中全连接层的输入为融合了文本 T 与假设 H 的特征组合，中间使用一个隐藏层，输出层使用 softmax 激活函数得到最终的结果。

3 实验

3.1 实验数据与评价指标

本文使用 CCL2018 中文文本蕴含测评任务的数据进行实验，用于训练的数据有 90000 条，验证数据 10000 条，测试数据 10000 条，数据具体情况统计如表 1 所示

表 1 实验数据统计

	蕴含	矛盾	中性	合计
训练集	29738	28937	31325	90000
验证集	3485	3417	3098	10000
测试集	-	-	-	10000

评价的准确率（Accuracy）为 $\text{Accuracy}=\frac{\sum \mathrm{D}_{\mathrm{correct}}}{\sum D_{all}}\times 100\%$

$\sum \mathrm{D}_{\mathrm{correct}}$ 表示数据集中预测正确的数据总和，

$\sum D_{all}$ 表示数据总量。

召回率 $recall_i = \frac{\sum T_i}{\sum(Ti+F_i)}$ ，$\sum T_i$ 表示正确识别成类别 i 的总数，分母表示正确识别与将 i 类识别成其他类的总数。

3.2 参数设置

本文使用的词向量是通过 word2vec 训练得到的，其它参数设置如下：

表 2 参数设置

参数	数值
词向量维度	100
学习速率	0.001
Dropout 值	0.5
L2 正则值	0.00001
卷积核大小	1,2,3,4,5
卷积核数量	128,128,128,128,128

3.3 实验结果及分析

为对比分析不同网络结构在文本蕴含关系识别的准确程度，本文构造了另外两种网络结构以及测评方提供的 baseline 模型进行了对比，其中召回率为三个类别召回率的平均值。

没有进行改进的 CNN：相对本文的方法，没有进行融合，也就是没有共享参数。

BiLSTM:仅使用双向 LSTM 对蕴含关系进行识别，最后通过分类的到识别结果。

Baseline——可分解的 attention 模型：将文本对其分解成子短语再比较、分类得到结果。

为将 CNN 与本文改进的 CNN 进行区别，这里将本文的 CNN 起名为 HCNN，对以上模型在验证集上进行试验结果如下：

表 3 本文方法与其它不同模型方法对比

模型	正确率	召回率	F 值
CNN	67.41%	66.85%	67.13%
BiLSTM	65.20%	64.57%	64.88%
Baseline-Decomposable	69.35%	68.26%	68.80%
HCNN	**76.83%**	**76.96%**	**76.89%**

经过后期的不断优化，本文使用的方法在测试集上最终取得第一的好成绩，前 8 名的测评结果如下：

表 4 前 8 团队的最终测评结果

排名	团队名	准确度
1	**water**	**0.8238**
2	zznlp2018	0.7828
3	-	0.7692
4	GDUFSER	0.7618
5	ray_li	0.7425
6	INTSIG_AI	0.7303
7	Yonseiiii	0.7242
8	Baseline	0.7222

以上结果表明,本文使用的方法提升效果明显，也证明本文的方法是非常有效的。

4 结论与展望

面对现有基于深度学习的文本蕴含方法存在模型复杂、训练参数大、效果提升不明显问题，本文提出了一种仅基于卷积神经网络的文本蕴含识别方法，该方法在将数据进行训练或预测前，先将数据做分词等预处理并映射成对应的词向量送入模型，经过卷积层、池化层后，连接一个融合层对特征进行融合，即获取两个文本之间语义关系，最后接入一个全连接层进行分类，得到最终结果。方法需要训练的参数相对较少，模型简单但提升效果明显，整个模型非常关键的一步在于融合层,实验结果表明,对两个文本进行融合，能有效的提取文本之间的语义关系，对识别两个文本之间的蕴含关系起到了非常大的作用。在 CCL2018 文本蕴含测评任务中，测试结果集上的正确率达 82.38%,在所有比赛队伍中稳居第一，表明该方法是有效的。

由于本次测评数据 T 和 H 两个文本大部分都是比较简短的句子且没有涉及到一些特定领域的文本蕴含识别，因此，今后可从大文本或者特定领域文本出发，利用深度学习技术来探索识别蕴含关系，使文本蕴含关系识别更好地应用到实际场景。

参考文献

[1] Lyu C, Lu Y, Ji D, et al. Deep learning for textual entailment recognition//Proceedings of the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI). Vietri sul Mare, Italy, 2015: 154-161

[2] Bowman S R, Potts C, Manning C D. Recursive neural networks can learn logical semantics. arXiv: 1406. 1827, 2014

[3] Yin W, Schütze H, Xiang B, et al. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. arXiv: 1512. 05193, 2015

[4] Ankur P.Parikh, Oscar Täckström, et al. A Decomposable Attention Model for

Natural Language Inference. arXiv: 1606.01933, 2016
[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings ofICLR.
[6] Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with LSTM. In Proceedings ofNAACL.
[7] Philipp Koehn. 2009. Statistical machine translation. Cambridge University Press.
[8] Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In Proceedings ofEMNLP.
[9] Johan van Benthem. 2008. A brief history ofnatural logic. College Publications.
[10] Bill MacCartney and Christopher D. Manning. 2009. An extended model of natural logic. In Proceedings ofthe IWCS.
[11] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal, 2015: 632-642
[12] Liu Y, Sun C, Lin L, et al. Learning natural language inference using bidirectional LSTM model and inner-attention. arXiv preprint arXiv: 1605. 09090, 2015
[13] Wang S, Jiang J. Learning natural language inference with LSTM. arXiv preprint arXiv: 1512. 08849, 2015
[14] Cheng J, Dong L, Lapata M. Long short-term memory-networks for machine reading. arXiv preprint arXiv: 1601. 06733, 2016
[15] Qian Chen, Xiaodan Zhu, et al. Enhanced LSTM for Natural Language Inference. arXiv: 1609. 06038, 2017
[16] 张辰，冯冲，刘全超，等. 基于多特征融合的中文比较句识别算法[J]. 中文信息学报，2013，27(6)：110-116.
[17] 倪耀群，许洪波，程学旗.基于多特征融合和图匹配的维汉句子对齐[J].中文信息学报. 2016,30(4):124-133.
[18] 奠雨洁，金 琴，吴慧敏. 基于多文本特征融合的中文微博的立场检测[J]. 计算机工程与应用,2017,53(21):77-84.
[19] W.Bruce Croft,Donald Metaler,Trevor Strohman 著刘挺,秦兵,张宇,等译.搜索引擎信息检索实践[M].北京：机械工业出版社,2009：54-55.
[20] Hubel David Hunter，Wiesel Torsten Nils.Receptive fields,binocular interaction and functional architecture in the cat's visual cortex[J].Journal of Physiology,1962,160(1)：106-154.
[21] LeCun Yann,Bottou Léon，Bengio Yoshua，et al.Gradient-based learning applied to document Recognition[J].Proceedings of the IEEE,1998,86(11)：2278-2324.

CCL2018 评测任务文本蕴含识别模型说明

刘鹏程 穆玲玲 杨朝阳

（郑州大学 信息工程学院，河南 郑州 450000）

摘要：近年来，随着人工智能的不断发展，自然语言处理的应用与日俱增。文本蕴含是自然语言处理领域的常见任务，属于机器翻译、阅读理解的基础性工作。CCL2018 评测任务中首次出现了中文文本蕴含识别任务，本文主要介绍用于该任务的注意力对齐模型。该模型主要由注意力层、对比层和聚合层三部分组成，经训练调整后，在评测任务测试集上达到了 78.28%的准确率。

关键词：文本蕴含；注意力机制；词向量；机器学习

0 引言

近年来，随着人工智能的不断发展和我国信息化建设的快速推进，人工智能的一个重要分支——自然语言处理的应用也越来越广泛。文本蕴含则是自然语言处理领域的常见任务，也是机器翻译、阅读理解的基础性工作。

文本蕴含研究即是识别、找出和生成两个文本中单向的推理关系——蕴含关系。因此文本蕴含同时有文本蕴含识别、文本蕴含抽取、文本蕴含生成三个研究领域。文本蕴含识别是给出一个文本对，让机器去判断这两个文本之间是否存在蕴含关系。目前有关文本蕴含的研究主要集中在文本蕴含识别方面。

第十七届中国计算语言学学术会议（The Seventeenth China National Conference on Computational Linguistics, CCL 2018）评测任务[1]中首次加入了中文文本蕴含识别任务。本文的实验就是在该评测任务的基础上进行的。本次评测将中文文本蕴含识别看作一个分类问题：每个输入样本为 2 个句子，分别是“前提句 Premise”和“假设句 Hypothesis”，要求参评系统判断两者之间的蕴含类别，共有蕴含、矛盾和无关三类。

评测提供了两个基线系统（Baseline）：Decomposable-Att[1]和 ESIM[2]，在开发集上的准确率分别达到了 69.35%和 73.57%。在正式评测的测试数据上 Baseline 的准确率达到了 72.22%。

在这次评测任务中，本模型采用了基于注意力机制的神经网络模型对所给出的数据集进行分类，该模型可以分解为注意力层、对比层和聚合层三个主要部分。注意力对齐模型训练时经过部分调整，最后在评测任务的测试集上进行分类，达到了 78.28%的准确率。

1 方法

注意力对齐模型基于传统的自然语言推理方法[3]，即基于对齐的文本蕴含关系识别，该方法将两个句子中词的表示构成一个对齐矩阵，通过该矩阵把两个文本中相似的部分找出来进行对齐操作，以对齐的程度作为判断是否构成蕴含关系的依据[4]。

1.1 数据集

本次实验使用的是 CCL2018 评测任务的中文文本蕴含识别数据集。数据集分两个批次发布，第一批发布的训练集为19999条，开发集9273条；第二批发布的训练集为 90000 条，开发集增补到 10000 条。训练集加上开发集一共 12 万条数据。

数据集的示例见图 1.1，其中每行的数据被 Tab 键划分为三列，第一列是前提句 P，第二列是假设句 H，第二列为蕴含类别 L，L 共有三类：

蕴含(E, entailment)：P 能合理推理得到 H，反之不一定；

矛盾(C，contradiction)：P 和 H 不能同时成立；

无关(N, neutral)： P 和 H 不存在必然的联系。

[1] http://www.cips-cl.org/static/CCL2018/call-evaluation.html#task3

孩子从秋千上倒挂下来。	孩子坐直，用脚推地。	contradiction↓
毛茸茸的棕色狗正在草地上奔跑。	狗在床上睡觉。	contradiction↓
一群人正在参加一个仪式。	它们是为了相同的目的而聚集的。	neutral↓
两名女子走在拥挤的街道上。	两名女子走在街上。	entailment↓
长颈鹿的嘴巴闭上了。	长颈鹿的嘴巴张开	contradiction↓
登山者正在攀登一座山崖。	一名登山者正在攀登山峰	entailment↓
球队解决对方的球载体。	小组对付对方的球载体。	entailment↓
三个小女孩坐在草地上。	有三个人。	contradiction↓

图 1.1 数据集示例

例如图 1.1 中第一行实例：前提句 P 为“孩子从秋千上倒挂下来。”;假设句 H 为“孩子坐直，用脚推地。”；标签 L 为“contradiction”。

上述实例中 L 为“contradiction”就表示 P 和 H 存在矛盾关系，即“孩子”的“倒挂”动作和“坐直”动作是不能同时成立的。

中文文本语料与英文语料不同，词与词之间没有空格分割，因此中文文本在进行分类之前，一般都要经过分词等预处理。本实验使用开源的 jieba-0.39 中文分词工具[2]分词。

本次实验文本数据使用分布式向量表示，文本向量采用基于 Word2Vec 模型预训练的中文融合词向量(skip-gram model with negative sampling, SGNS)，SGNS 结合了维基百科、百度百科、人民日报等多种中文数据进行训练，能较好地表示汉语中词与词之间的关系[5]。词向量的维数为 300 维。

1.2 注意力对齐模型

模型主要由三个部分组成：注意力层、对比层和聚合层。

前提 P 和假设 H 被转换成词向量序列 a=(a_1, … a_i)、b=（b_1,… b_j），其中 i、j 分别是 a、b 的长度，a_i为 a 中第 i 个词的向量表示，b_j为 b 中第 j 个词的向量表示。a_m, b_n∈$\boldsymbol{R^{300}}$，其中 m∈[1,i]，n∈[1,j]。标签$\boldsymbol{l}$ =（l_1, l_2, l_3），分别对应中立 N、蕴含 E 和矛盾 C 三类。

1.2.1 注意力层

注意力层在接受词向量序列 a、b 后，分别经过一个前馈神经网络 F 提取特征，再进行对齐操作和注意力计算，得到一个初步的特征权重矩阵$\boldsymbol{e_{ij}}$。F 为两层全连接神经网络。

对齐操作即是使用 a、b 序列分别作为矩阵的行和列，用序列中的词或子句构成一个 i×j 的矩阵。

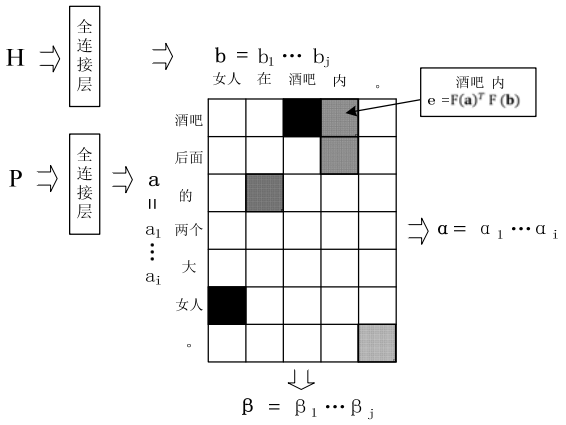


图 1.2 注意力层示意图

通过权重矩阵$\boldsymbol{e_{ij}}$，分别计算出 b 中的词向量和 a 中的词向量逐个对齐后的注意力权重β_i与 a 中的词向量和 b 中的词向量逐个对齐后的注意力权重α_j。得到词与词或子句与子句之间的相似部分。如图 1.2 所示，图中 P 和 H 分别代表前提句和假设句的词向量序列。计算权重矩阵的过程形式化表示如式（1.1）~（1.3）所示。

$$\boldsymbol{e_{ij}} = \mathrm{F}(\mathbf{a})^T \mathrm{F}(\mathbf{b}) \qquad (1.1)$$

$$\beta_i = \sum_1^j \frac{\exp(\boldsymbol{e_{ij}})}{\sum_{k=1}^j \exp(\boldsymbol{e_{ik}})} b_j \qquad (1.2)$$

$$\alpha_j = \sum_1^i \frac{\exp(\boldsymbol{e_{ij}})}{\sum_{k=1}^j \exp(\boldsymbol{e_{jk}})} a_i \qquad (1.3)$$

式中：F 代表前馈神经网络的操作；β_i代表 b 的子句和 a 对齐后的注意力权重；α_j代表 a 的子句和 b 对齐后的注意力权重。

1.2.2 对比层

对比层将 a、b 两个序列中的向量a_i、b_j和在注意力层得到的子句权重β_i、α_j分别进行拼接。通过一个前馈神经网络层 G 对a_i和β_i，b_j和α_j两对向量进行对比，得到对应每个序列对中子句的权重向量$\boldsymbol{v_{1,i}}$、$\boldsymbol{v_{2,j}}$。对比操作的形式化表示如式(1.4)~（1.5）所示。

$$\boldsymbol{v_{1,i}} = \mathrm{G}([\beta_i, a_i]) \qquad (1.4)$$

$$\boldsymbol{v_{2,j}} = \mathrm{G}([\alpha_j, b_j]) \qquad (1.5)$$

[2] https://github.com/fxsjy/jieba

式中：G 代表前馈神经网络的操作；[]代表向量的拼接操作。

1.2.3 聚合层

聚合层即是首先将词的权重向量$\boldsymbol{v_{1,i}}$、$\boldsymbol{v_{2,j}}$分别聚合成代表整个句子的权重向量$\boldsymbol{v_1}$、$\boldsymbol{v_2}$，然后将$\boldsymbol{v_1}$、$\boldsymbol{v_2}$进行拼接操作，使用一个前馈神经网络H进行分类，得到向量$\mathbf{v} \in \boldsymbol{R^3}$，最后对$\mathbf{v}$通过argmax函数输出最终的标签$\tilde{l}$。聚合层的形式化表示如式（1.6）~（1.9）所示。

$$\boldsymbol{v_1} = \sum_1^i \boldsymbol{v_{1,i}} \tag{1.6}$$

$$\boldsymbol{v_2} = \sum_1^j \boldsymbol{v_{2,j}} \tag{1.7}$$

$$\mathbf{v} = \mathrm{H}([\boldsymbol{v_1}, \boldsymbol{v_2}]) \tag{1.8}$$

$$\tilde{l} = \mathrm{argmax(L)} \tag{1.9}$$

式中：H 代表分类器；[]代表向量的拼接操作；$\tilde{l}$ 代表最终得到的标签。

2 实验与结果分析

本节主要介绍注意力对齐模型在评测数据集上的表现和对比实验。

2.1 实验设置

本次实验环境为 python3.6、Tensorflow-GPU-1.4.0。主要超参数见表 1.1。

表 1.1 主要超参数

学习率	0.0004
dropout	0.35
batch size	32

在模型前两部分使用的前馈神经网络 F、G 均为两层全连接神经网络，激活函数为 Relu[6]；最后一部分使用的前馈神经网络 H 为单层全连接神经网络，激活函数为双曲正切函数。优化方法采用 AdamOptimizer 算法，loss 函数采用二次交叉熵函数。

本次评测使用准确率 P 作为判定标准，即文本蕴含识别的分类的标签正确率，计算时对每个标签进行微平均统计，如式（1.10）所示。

$$\mathrm{P} = \frac{\tilde{l}_{correct}}{l} \tag{1.10}$$

式中：$\tilde{l}_{correct}$代表不分类别所有分类正确的标签；l为数据集的原始标签。

2.2 实验结果

该模型训练的结果在评测数据集上准确率达到了 78.12%。

2.2.1 评测测试集准确率

评测任务中的各模型的准确率对比如表 1.2 所示：

表 1.2 模型对比

模型	训练集（%）	测试集（%）
注意力对齐模型	73.83	78.28
ESIM	76.91	72.22
LSTM+CNN	——	82.38

其中 LSTM+CNN[7]的方法是目前在该测试集上最好的成绩，达到了 82.38%。ESIM 是评测的基准模型 Baseline。

注意力对齐模型通过全连接层提取句子每个词的特征，计算蕴含文本对之间的注意力权重，然后进一步提取特征得到标签，结合了句子间的联系信息。其神经网络的结构比 ESIM 模型更简单，计算上存在速度优势。

2.2.2 超参数调整

在第一批 2 万条数据的基础上，实验测试了注意力对齐模型在训练集上不同超参数的表现，以决定在评测测试集上的超参数。以 dropout 率为例，调整方式见表 1.3。

表 1.3 超参数调整

dropout	训练集（%）	测试集（%）
0.2	86.68	57.96
0.3	70.28	65.83
0.4	67.41	55.33

可以看出在 dropout 率较小的情况下，模型倾向于过拟合；dropout 率较大时，模型产生欠拟合。所以根据在第一批数据的基础上的实验结果，本文在评测时模型采用的 dropout 率为 0.35。

2.3 结果分析

本文统计了模型在开发集上分类错误的错误率和错误实例，结果如表 1.4 和表 1.5 所示。表中 N2E 表示中立错判为蕴含，N2C 表示中立错判为矛盾，E2N 表示蕴含错判为中立，以此类推。错误率计算如式（1.11）所示。

$$P_{error} = \frac{\tilde{l}_{error}}{l} \tag{1.11}$$

式中：$\tilde{l}_{error}$为各个类别中分类错误的标签；l为数据集的原始标签。

表 1.4 错误率统计

错判类别	错判数量	错误率
N	1146	30.16%
E	999	26.29%
C	1654	43.54%

可以从表中看出矛盾关系的识别是最困难的，矛盾关系的错误率达到了 43.54%。而中立和蕴含的错误率分别只有 30.16%和 26.29%。

表 1.5 错误实例

输入文本	真实标签	分类标签
P:滑板运动员在电线杆前做空中跳跃。 H：一个人在玩电脑。	N	E
P:红线是倾斜的。 H：黑色线条曲折。	C	E
P:一个女孩骑在秋千上。 H：一位女性正坐在外面。	E	C

由表 1.5 中第一对例子是中立被误分为蕴含，能看出因为有“在”、“做”等虚词或动词比较相似，导致判断错误，可以考虑结合虚词关系或去除虚词进行进一步实验。第二、三对例子是矛盾被误分为蕴含和蕴含被误分为矛盾，原因是“线”、“红色”、“黑色”、“一个”、“女孩”等名词的词向量之间的相似度较高，两句子句之间的注意力权重较高，使模型难以分清矛盾和蕴含的界限。矛盾关系的判断也是文本蕴含识别中的一个难点[8]，这符合上述有关错误率的分析。未来工作可着重于矛盾关系判断的改进，可使用宏平均统计计算各类别准确率。

3 结论

本文针对 CCL2018 文本蕴含评测任务采用了注意力对齐模型。该模型采用结合注意力的对齐操作，可以提取蕴含文本对的子句特征并计算文本对之间的注意力权重，来识别蕴含关系的类别。在测试集上该模型达到了 78.28%的准确率。

下一步工作重点可以放在改善矛盾关系的分类效果上，比如加入词性特征或句法依存关系等知识、改进模型，以更好地解决文本蕴含识别任务。

参考文献

[1] Parikh A P, Täckström O, Das D, et al. A Decomposable Attention Model for Natural Language Inference [J]. 2016:2249-2255.

[2] Chen Q, Zhu X, Ling Z, et al. Enhanced LSTM for Natural Language Inference [J]. 2016:1657-1668.

[3] Maccartney B. Natural language inference [M]. Stanford University, 2009.

[4] 郭茂盛, 张宇, 刘挺. 文本蕴含关系识别与知识获取研究进展及展望[J]. 计算机学报, 2017, 40(4):889-910.

[5] Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., & Du, X. (2018). Analogical Reasoning on Chinese Morphological and Semantic Relations. arXiv preprint arXiv:1805.06504

[6] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C]// International Conference on Artificial Intelligence and Statistics. 2011:315-323.

[7] Cheng, J., Dong, L., & Lapata, M. (2016). Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733.

[8] MacCartney B, Manning C D. Natural language inference [M]. Stanford: Stanford University, 2009.

文章编号：1003-0077（2017）00-0000-00

基于混合注意力机制的中文文本蕴含模型

吴晓晖[1] 尹存祥[1] 骆金昌[1] 钟辉强[1]
（1.无）

摘要：文本蕴含是自然语言处理领域中的一项重要的基础研究和研究热点。该文针对文本蕴含任务，提出一种基于注意力机制的神经网路模型的文本蕴含关系推理方法。该文基于 ESIM 模型，结合注意力机制，从模型结构、数据增强、数据预处理等多个维度进行改进。实验结果表明，所提出的改进机制均有提升，在 CNLI 数据集上的关系推理准确率达到了 76.92%。

关键词：文本蕴含；自然语言理解；人工智能

中图分类号：TP391 **文献标识码：**A

0 引言

正文用“本文”。在 8000 字左右为宜，参考文献按文中出现顺序引用。[1]

正文中，图表须注明中文图题和表题，且在正文中应明确提及。其中图的编号和图题置于图下方的居中位置，表的编号和表题置于表上方的居中位置。

示图使用黑白绘图，请确保图表中文字清晰。

公式用 5 号字体，其中变量的符号应采用斜体，向量、矢量、矩阵用黑斜体表示。函数（单词）用正体小写，第一个字母小写；单个字母斜体。

1 相关工作

1.1 标题

正文

1.1.1 标题

正文

2 基于注意力机制的中文文本蕴含模型

2.1 基于字级别的模型

预处理阶段，使用现有的分词工具对文本语料进行分词。但是，分词工具在分词时会引入分词误差，为了避免分词误差，本文引入了 Soft Word 方法。将文本以字序列形式表示，而词作为每个字的辅助特征，从而每个文本都可以转化为字词对序列。

单纯的字向量没有单词和词组的信息，因此我们使用拼接词向量的方式来增强字向量的表示：

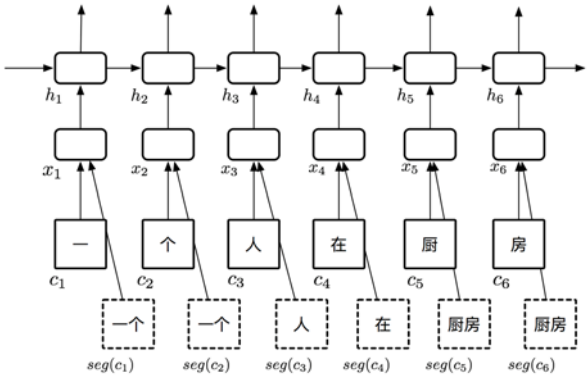


图 1 基于字+词的模型

$$x_j = [e^c(c_j); e^w(seg(c_j))]$$

其中，e^w代表词向量矩阵，e^c代表字向量矩阵。c_j代表第 j个字，而$seg(c_j)$就代表包含了c_j的分词

收稿收稿日期：2017-03-16；**定稿日期**：2017-04-26
基金项目：基金名（基金号）；基金名(基金号)

结果。将两者的向量拼接，得到每个字的向量表示x_j。

在$x_1, x_2, \ldots, x_j$上使用一个双向 LSTM，从而得到前向序列$\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_j$和反向序列$\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_j$。因此，得到的隐藏层向量表示为：

$$h_j = [\vec{h}_j; \overleftarrow{h}_j]$$

2.2 数据增强

在数据样本中，前提文本和假设文本中存在一些字面上匹配但对蕴含推理无关的文本，这些文本会引导模型将其他关系误判为蕴含的关系。因此，我们将每个数据样本中的最长公共子序列（LCS）去除，把剩余的文本作为新的数据样本，从而将数据量增加了一倍。

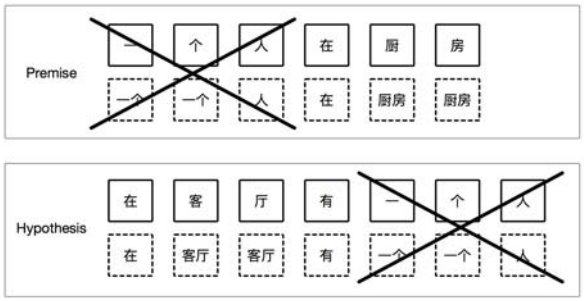
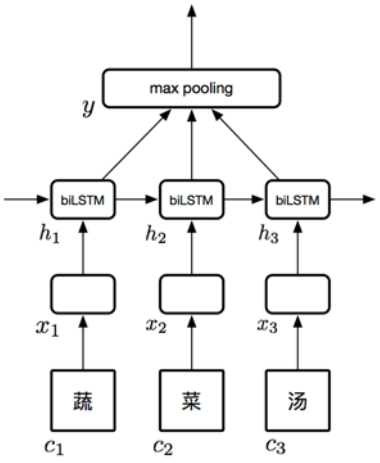


图 2

2.3 未登录词向量

通常，我们会使用预训练的字向量和词向量作为模型的嵌入层初始化。而预训练的词向量通常难以覆盖所有的词语，在目标任务上会出现没有预训练词向量的情况。本文利用预训练的字向量和词向量，构造一个词向量生成模型（OOV 模型）。



对于 OOV 模型的输入，使用预训练的字向量。

$$x_j = e^c(c_j)$$

本文将词向量的生成看成是字序列的向量表示问题，使用双向 LSTM 来整个字序列的向量表示。将 LSTM 学习出来的向量表示进行最大池化：

$$y = \max_m \begin{bmatrix} h_{11} & h_{12} & \ldots & h_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ h_{d1} & h_{d2} & \ldots & h_{dm} \end{bmatrix}$$

词语中的第m个向量表示为$h_m \in \mathbb{R}^d$，而该向量的元素表示为h_{dm}。按照时序维度取最大值，生成词向量表示$y \in \mathbb{R}^d$。

本文采用 MSE 损失函数和 L2 正则项作为训练模型的目标函数：

$$J(\theta) = \frac{1}{N}\sum_{n=0}^{N-1} \|y_i(\theta) - e^w([c_1; \ldots; c_m]_i)\|^2 + \frac{\lambda}{2}\|\theta\|^2$$

其中，$y_i(\theta)$通过模型生成的词向量，$e^w([c_1; \ldots; c_m]_i)$表示已有的词向量，$N$为训练样本的个数，$\lambda$为正则化系数，$\theta$为模型中所需要学习的参数。

2.4 Co-Attention 模型

在 ESIM 模型中，有一个局部推理的模块，，在句子经过 LSTM 层的编码后，将前提文本和假设文本的时序进行对齐，其中使用的方式是软对齐，计算 hidden state 的相似度，计算公式是：

$$s_{ij} = \overline{a}_i^T \overline{b}_j$$

在其中，a_i和b_j分别是通过 LSTM 层计算得到的。在前提文本中每个时序和假设文本中的每个时序的相关性，可以通过计算s_{ij}来表示。我们将s_{ij}分别按照两个维度进行归一化，得到两个权重矩阵后，再计算交互后的向量表示。

$$\tilde{a}_i = \sum_{j=1}^{l_b} \frac{\exp(s_{ij})}{\sum_{k=1}^{l_b}\exp(s_{ik})}\overline{b}_j, \qquad i = 1, \ldots, l_a$$

$$\tilde{b}_j = \sum_{i=1}^{l_a} \frac{\exp(s_{ij})}{\sum_{k=1}^{l_a}\exp(s_{kj})}\overline{a}_i, \qquad j = 1, \ldots, l_b$$

为增强两个文本时序间的交互，通过计算$< a, \tilde{a} >$和$< b, \tilde{b} >$之间的差异，比如计算差，点对点乘积。

$$a_i = [\overline{a}_i; \tilde{a}_i; \overline{a}_i - \tilde{a}_i; \overline{a}_i \odot \tilde{a}_i], \qquad i = 1, \ldots, l_a$$
$$b_j = [\overline{b}_j; \tilde{b}_j; \overline{b}_j - \tilde{b}_j; \overline{b}_j \odot \tilde{b}_j], \qquad j = 1, \ldots, l_b$$

得到两个文本交互后的表示，我们需要进行交互后的信息组合。因此再次使用 LSTM 来计算输入。

$$h_a = LSTM(a)$$
$$h_b = LSTM(b)$$

在组合交互信息后，我们需要将信息压缩成固定大小的向量表示，因此使用池化的方式来获取固定大小的向量。这里使用最大池化和平均池化的方法，最终拼接所有的向量得到最终向量表示。

$$r_{a,ave} = \sum_{i=1}^{l_a} \frac{h_{a,i}}{l_a}, \qquad r_{a,max} = \max_{i=1}^{l_a} h_{a,i}$$

$$r_{b,ave} = \sum_{j=1}^{l_b} \frac{h_{b,j}}{l_b}, \qquad r_{b,max} = \max_{j=1}^{l_b} h_{b,j}$$

$$r = [r_{a,ave}; r_{a,max}; r_{b,ave}; r_{b,max}]$$

随后，将r输入到前向神经网络分类器中，分类器有一层隐藏层用 tanh 激活，输出层用 softmax 进行分类，使用交叉熵作为损失函数，整个模型是端到端训练的。

2.5 Self-Attentive 模型

co-attention 机制更加强调文本中局部信息的交互，而缺乏从整体信息来进行匹配。因此，本文将使用 Self-Attentive 机制来进行文本整体的匹配。

$$\bar{A} = (\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_{l_a}) \in \mathbb{R}^{d \times l_a}$$
$$\bar{B} = (\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_{l_b}) \in \mathbb{R}^{d \times l_b}$$

我们的目的是将不定长的句子压缩成固定大小的向量表示。我们计算 attention 矩阵：

$$U_a = softmax(W_{s2}\tanh(W_{s1}\bar{A}))$$
$$U_b = softmax(W_{s2}\tanh(W_{s1}\bar{B}))$$

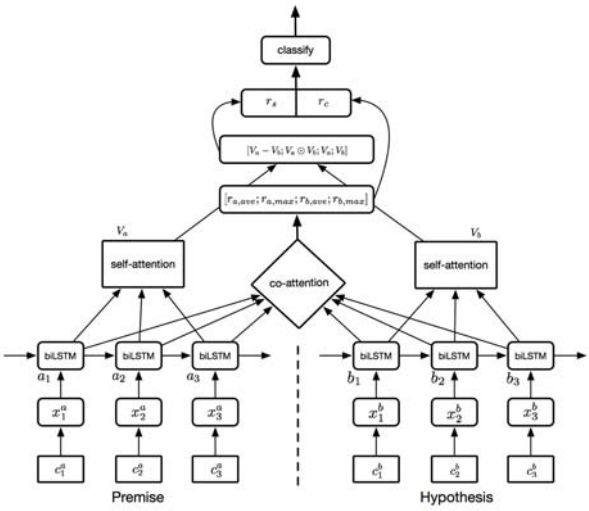
我们使用两层的前向神经网络来计算 attention 矩阵，其中$W_{s1} \in \mathbb{R}^{s \times d}$, $W_{s2} \in \mathbb{R}^{r \times s}$是需要学习的参数。其中$softmax$运算是计算句子长度维度的 softmax，$U_a \in \mathbb{R}^{r \times l_a}, U_b \in \mathbb{R}^{r \times l_b}$。

$$V_a = U_a \bar{A}^T$$
$$V_b = U_b \bar{B}^T$$
$$M = [V_a - V_b; V_a \odot V_b; V_a; V_b]$$

其中，$V_a \in \mathbb{R}^{r \times d}, V_b \in \mathbb{R}^{r \times d}$，它们是经过 attention 矩阵加权后得到的固定大小的矩阵。M是计算两个文本矩阵表示的匹配度矩阵。随后将M摊开成为向量，输入到前向神经网络中，得到 Self-attentive 模型的匹配表示。

2.6 最终模型

该模型的总的结构如图所示，混合了 co-attention 和 self-attention 机制，co-attention 机制负责文本局部信息的交互匹配，self-attention 负责文本全局信息的匹配。文本经过 LSTM 来获取上下文相关的向量表示，随后输入到混合注意力中，分别得到两种注意力匹配的表示，最终拼接输入到最终的分类器中。



3 实验

3.1 实验设置

数据
分词
词向量

3.2 实验过程

正文

3.3 实验设置

正文

4 结论与未来工作

4.1 标题

正文

参考文献

参考文献小五号，只列举最主要的，必须是公开发表的书刊才能列入，最少不得少于 5 条。文献按文章中出现先后顺序排列

（各类文献严格按照主页上的《参考文献规范》）

[1] 专著:[序号] 作者.题名[M].出版地：出版者，出版年：起止页码.

[2] 期刊:[序号] 作者(多作者用逗号分开，超过 3 个者用“,等”代替).文章题目[J].刊物名称,年,卷(期):起止页码.

[3] 会议论文集:[序号] 作者.题名[C]//编者.论文集名.出版地：出版者，出版年：起止页码.

[4] 英文会议：[序号] 作者.题名[C]//Proceedings of 会议名称.出版地：出版者，出版年：起止页码

[5] 学位论文：[序号] 作者.题名[D].保存单位 XX 学位论文，年份.

[6] 报告:[序号]作者.题名[R].保存地点：保存单位，年份.

[7] 报纸文章:[序号] 作者.题名[N].报纸名，出版日期(版次).

[8] 标准:[序号]标准编号，标准名称[S].出版地：出版者，出版年.

[9] 专利: [序号]专利所有者.专利题名[P].专利国别:专利号,公开日期.

[10] 电子文献:主要责任者.电子文献题名[电子文献标识/载体类型]. [发表或更新日期].电子文献的出处或可获得地址.

[11] 电子文献标识:[DB]-数据库 [CP]-计算机程序 [EB]-电子公告

[12] 电子文献载体类型:[OL]-联机网络 [MT]--磁带 [DK]-磁盘 [CD]-光盘

文章编号：1003-0077(2017)00-0000-00

基于多种注意力机制的中文文本蕴涵识别

戚昆逊, 杜剑峰, 曹子旋, 刘汉锋, 胡裕鹏

广东外语外贸大学 信息科学与技术学院，广东 广州 510000

摘要：文本蕴涵识别是一项具有挑战性的自然语言处理任务，是近年来国内外研究热点之一。该文针对中文文本蕴涵识别任务，提出了一个基于多种注意力机制结合的神经网络模型，具有以下特点：1. 该模型使用预训练的深度上下文词表征模型 ELMo 生成输入的词向量；2. 该模型结合了多种注意力机制，用于捕捉两个句子之间的交互信息。实验结果表明，该模型在第十七届中国计算语言学大会(CCL2018)中文文本蕴涵任务的测试数据集上取得 76.2%的准确率。

关键词：中文文本蕴涵；神经网络；注意力机制

中图分类号：TP391 **文献标识码：**A

Recognition of Chinese Textual Entailment Based on Multiple Attention Mechanisms

Kunxun Qi, Jianfeng Du, Zixuan Cao, Hanfeng Liu and Yupeng Hu

School of Information Science and Technology,

Guangdong University of Foreign Studies, Guangzhou, Guangdong 510006, China

Abstract: Recognition of textual entailment is a challenging task in natural language processing. This paper proposes a neural network model based on multiple attention mechanisms. It has the following characteristics: First, a pre-trained deep contextualized word representation model called ELMo is used to generate the input word embedding; Second, multiple attention mechanisms are employed to express the interaction between two sentences. Experimental results show that the model achieved 76.2% accuracy on the test set about recognition of textual entailment in Chinese from the 17th China Computational Linguistics Conference.

1 引言

文本蕴含识别(Recognition of Textual Entailment, RTE)，又称自然语言推断(Natural Language Inference, NLI)是自然语言处理中一项具有挑战性的任务。文本蕴含的定义为：判断一个前提文本(Premise)和一个假设文本(Hypothesis)之间的关系。两个文本之间的关系包括蕴涵关系(Entailment)、矛盾关系(Contradiction)以及中立关系(Neutral)。若假设 H 的语义可以被前提 P 的语义所推断出来，则认为前提 P 蕴涵假设 H，记为蕴涵关系；若前提 P 的语义与假设 H 的语义相违背，则认为前提P与假设H相互矛盾，记为矛盾关系；若前提 P 的意义与假设 H 的意义无关，则记为中立关系。例如：

(1)蕴含关系：

前提文本：穿红衬衫的男人和拿着白色袋子的女人正在交谈。

假设文本：两个人在交谈。

(2)矛盾关系：

前提文本：有涂鸦的棕色建筑物。

收稿日期：定稿日期：

基金项目：国家自然科学基金(61876204)；广州市科技计划(201804010496)

假设文本：一座红房子。

(3)中立关系：

前提文本：一名女子正在焊接电子设备。

假设文本：一名女子戴着防护装备。

近年来，随着深度学习的发展，国内外研究趋向于使用构建神经网络模型来处理文本蕴涵问题。目前大部分的文本蕴涵识别工作在斯坦福自然语言推理数据集(Stanford Natural Language Inference, SNLI)和多类型自然语言推理数据集(Multi-Genre Natural Language Inference, MutliNLI)中进行训练和评估。在深度学习模型中，已有两种主流框架用于判断两个文本之间的蕴含关系。它们分别是孪生网络(Siamese)框架和匹配聚集(Matching-aggregation)框架。孪生网络框架使用两个权重共享的句子编码器获取两个文本各自的句子向量表示 v_1 和 v_2，并使用这两个向量表示构造类似于(v_1; v_2; $v_1 \circ v_2$; v_1- v_2)的特征向量，用于预测分类。该框架具有结构简单、易于训练等特点。代表性方法包括 BCNN[1]，InferSent[2]以及 SWEMs[3]等。但该框架忽略计算两个文本之间的交互信息，往往不能达到最高水平的性能。为了捕捉句子交互信息，匹配聚集框架在孪生网络框架的基础上加入了匹配机制与聚集机制。该框架通常使用注意力机制对句子编码的不同粒度如词级别粒度进行匹配，然后使用聚集机制产生匹配后的句子表示。代表性方法包括 ABCNN[1]，BiMPM[4]，ESIM[5]和 MwAN[6]等。本文在匹配聚集框架的基础下，提出了一种基于多种注意力机制的神经网络模型。该模型有以下特点：

(1)使用深度上下文词表征模型 ELMo[7]产生的词向量作为模型的输入。近年来，词向量被广泛应用于多种自然语言处理任务上，并取得了较好的效果。在多种词向量之中，较为常见的词向量有 Word2vec[8]和 GloVe[9]。Word2vec 和 GloVe 通过训练大规模文本语料，产生每个单词的向量表示，这些词向量具有一定的语义信息。然而，在实际应用中，这种通过预训练方式产生的词向量，不能覆盖到句子中的每一个单词，因此存在集外词(Out of vocabulary, OOV)问题：没有被词向量词表所覆盖的单词缺少有效的向量表示。OOV 问题的常见处理方法是使用随机生成的向量表示集外词。这种方法一定程度上缓解了集外词的问题，但是随机生成的向量表示缺少语义信息。为了解决 OOV 问题，我们训练了一个深度上下文词表征模型 ELMo，并将其应用于输入句子的词向量生成。ELMo 是一种新型的神经语言模型，其通过输入单词的字符和单词所在句子中的上下文，计算每个词的词向量，因此每个输入的单词都有相应的词表示，解决了 OOV 问题。更重要的是，ELMo 所产生的词向量包含了句子中的上下文信息，已被证明其在多种自然语言处理任务中具有较高的性能[7]。

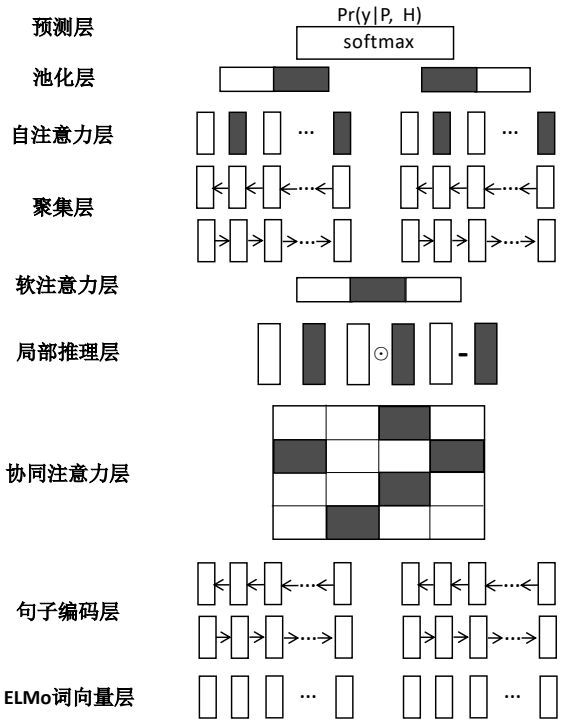


图 1. 模型结构图

(2)本文提出的模型使用了多种注意力机制，包括协同注意力机制(Co-attention)、软注意力机制 (Soft-attention) 以及自注意力机制(Self-attention)。其中，协同注意力机制应用在计算两个文本中每个单词粒度的交互信息。软注意力机制用于合成两个文本中单词与单词之间的局部推理信息。自注意力机制用于捕捉聚集计算后句子序列内部的交互特征。

我们在第十七届中国计算语言学大会(CCL2018)中文文本蕴涵任务的数据集上训练及验证了本文提出的模型。该数据的训练集包含 9 万个中文句子对，验证集包含 1 万个句子对，测试集包含 1 万个句子对。我们的模型在测试集中取得了 76.2%的准确率。

2 主要方法

图 1 为本文提出模型的结构图。其中，第一层为 ELMo 模型生成的词向量。第二层为句子编码层，本文使用双向门控循环单元 (Gated Recurrent Unit, GRU)[10]产生句子序列编码。第三层为协同注意力层，本文使用线性注意力机制计算

两个句子编码中词与词之间的注意力值，并使用该注意力值重新调整句子中每个单词的隐藏层表示。第四层为局部推理层，本文使用点乘和减计算局部推理表示。第五层为软注意力层，本文使用软注意力机制产生 4 种局部推理表示的注意力值，并使用该注意力值对局部推理表示进行加权平均。第六层为聚集层，本文使用双向门控循环单元对重新调整后的局部推理表示进行聚集。第七层为自注意力层，本文使用多头注意力机制(Multi-Head Attention)[11]对经过聚集计算后的序列表示进行调整。第八层为池化层，本文使用平均池化和最大池化拼接的方法产生两个文本的向量表示。第九层为预测层，本文使用两层前馈网络对两个文本表示的拼接进行分类预测。

2. 1 ELMo 词向量层

本文使用 3.3GB 中文维基百科语料训练 ELMo 模型，并使用 ELMo 模型生成两个句子中每个词的单词表示。由 ELMo 模型产生的单词表示维度为 512 维。在训练过程中我们不对该词向量进行调整。

2. 2 句子编码层

本文使用双向门控循环单元 BiGRU 对两个句子序列中的单词表示进行建模。其中，序列中的每一个隐藏层大小为 300：

$$h_i = BiGRU(h_{i-1}, w_i) \tag{1}$$

$$p_j = BiGRU(p_{j-1}, w_j) \tag{2}$$

其中，h_i为假设文本 H 中每个单词的隐藏状态表示，p_j为前提文本 P 中每个单词的隐藏状态表示。w_i为文本 H 中单词的 ELMo 词向量表示，w_j为文本 P 中单词的 ELMo 词向量表示。

2. 3 协同注意力层

本文使用协同注意力机制产生两个文本序列的交互注意力值，并根据该注意力值调整句子编码层产生的序列表示。

目前主流模型，如 ESIM 等，使用点积操作计算两个向量表示的注意力值。然而，点积等相似度方法难以更准确地捕捉单词隐藏状态之间的交互关系。因此，本文使用多种局部特征组合并经过线性映射计算注意力值的方法，具体如下：

$$e_{ij} = v^T\left[h_i; p_j; h_i \circ p_j; h_i - p_j\right] \tag{3}$$

$$\tilde{h}_i = \sum_{j=1}^{N_p} softmax\left(e_{ij}\right) p_j \tag{4}$$

$$\tilde{p}_j = \sum_{i=1}^{N_h} softmax\left(e_{ij}\right) h_i \tag{5}$$

其中，v^T为可训练参数，∘为向量对位点乘操作，$-$为向量相减操作，$\tilde{h}_i$为调整后的 H 文本序列的隐藏状态表示，$\tilde{p}_j$为调整后的 P 文本序列的隐藏状态表示。

2. 4 局部推理层

与 ESIM 方法相同，本文使用对位点乘和相减操作构建局部推理表示，具体表示如下：

$$m_h = \left\{h,\ \tilde{h},\ h \circ \tilde{h},\ h - \tilde{h}\right\} \tag{6}$$

$$m_p = \left\{p,\ \tilde{p},\ p \circ \tilde{p},\ p - \tilde{p}\right\} \tag{7}$$

2. 5 软注意力层

本文使用软注意力机制对集合m_h和m_p集合进行调整，具体如下：

$$\alpha_h^k = v_1^T tanh\left(W^1 m_h^k + W^2 v_2\right) \tag{8}$$

$$\alpha_p^k = v_1^T tanh\left(W^1 m_p^k + W^2 v_2\right) \tag{9}$$

$$\bar{h} = \sum_{k=1}^{4} softmax\left(\alpha_h^k\right) m_h^k \tag{10}$$

$$\bar{p} = \sum_{k=1}^{4} softmax\left(\alpha_p^k\right) m_p^k \tag{11}$$

其中 k 为集合 m 中第几个元素，共有 4 个值。v_1^T，W^1，W^2，v_2为可训练的参数。$\bar{h}$为加权平均后的假设文本 H 的隐藏状态表示，$\bar{p}$为加权平均后的假设文本 P 的隐藏状态表示。

2. 6 聚集层

本文使用双向门控循环单元 BiGRU 对经过软注意力层加权平均后的 H 和 P 文本序列的隐藏状态表示进行聚集。其中，序列中的每一个隐藏层大小为 300：

$$a_i^h = BiGRU\left(a_{i-1}^h, \bar{h}_i\right) \tag{12}$$

$$a_j^p = BiGRU\left(a_{j-1}^p, \bar{p}_j\right) \tag{13}$$

2. 7 自注意力层

本文使用[10]中的多头自注意力机制对经过聚集操作后的序列隐藏状态表示进行调整：

$$Att(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{14}$$

$$MultiHead(Q,K,V) = Concat(head_1, \ldots, head_h)W^O$$

$$head_i = Att\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{15}$$

$$\tilde{H} = MultiHead(A^h, A^h, A^h) \tag{16}$$

$$\tilde{P} = MultiHead(A^p, A^p, A^p) \tag{17}$$

其中，Q、K、V 是三个文本序列表示的矩阵，在自注意力机制中，我们令 Q=K=V。$\sqrt{d_k}$为 K 矩阵中列向量的大小。A^h是聚集计算后 H 文本序列的隐藏状态矩阵。A^p是聚集计算后 P 文本序列的隐藏状态矩阵。W_i^Q，W_i^K，W_i^V，W^O为可训练的参数。本文将头数 h 设为 8，W_i^Q，W_i^K，W_i^V隐藏层大小为 32。

2.8 池化层

本文使用平均池化和最大池的方法产生两个文本序列的表示，并将四个向量拼接构造用于预测的特征向量：

$$v_f = \left[v_{avg}^h; v_{max}^h; v_{avg}^p; v_{max}^p\right] \quad (18)$$

2.9 预测层

本文使用两层前馈网络，第一层前馈网络激活函数为 tanh。第二层前馈网络使用 softmax 函数输出每个蕴涵类标的预测概率。训练过程的目标为最小化交叉熵(cross entropy)损失。

3 实验与结果

3.1 实验设置

表 1. 测试集中的实验结果

所属框架	模型	验证集准确率	测试集准确率
孪生网络框架	BCNN	67.5%	67.3%
	InferSent	70.9%	69.7%
	SWEMs	70.9%	70.5%
匹配聚集框架	ABCNN-2	71.4%	71.3%
	BiMPM	72.6%	71.9%
	ESIM	73.5%	72.2%
	Our model	**76.4%**	**75.1%**
	Our model (ensemble)	**77.1%**	**76.2%**

本文在 CCL2018 中文文本蕴涵评测数据中训练及验证我们的模型。我们使用结巴[1]分词对数据集进行中文分词。我们使用 3.3GB 维基百科语料训练的深度上下文词表征模型 ELMo 生成输入的词向量，词向量维度为 512。我们在训练过程中不更新词向量。所有双向 GRU 网络的隐藏状态大小为 300。我们在每一层间设置 droupout，比例为 0.3。我们使用 Adam 方法作为最优化方法，学习率设置为 0.0003。我们取验证集上性能最好的模型对测试集进行预测。

3.2 实验结果

从表 1 可以看出，聚集匹配框架下各方法相比孪生网络框架下各方法具有较高的性能。其中 BCNN 为使用卷积神经网络作为句子编码器的孪生网络，测试集准确率为 67.3%。InferSent 为使用双向 LSTM 作为句子编码器的孪生网络，准确率为 69.7%。SWEMs 是使用池化操作作为句子编码器的孪生网络，准确率达到 70.5%。ABCNN-2 是在 BCNN 基础上加入注意力机制的模型，性能达到 71.3%。BiMPM 是一个多感受视野的双向匹配模型，准确率达到 71.9%。ESIM 模型结果为评测方发布的基线模型结果，准确率达到 72.22%。本文所提出的模型在单模型上达到了 75.1%的准确率，在使用集成方法后(ensemble)，准确率达到了 76.2%。

表 2. 模型简化测试结果

模型	准确率
Our model	76.4%
(1)使用点积计算协同注意力值	74.9%
(2)去除软注意力层	76.0%
(3)去除自注意力层	75.1%
(4)使用 word2vec 代替 ELMo	75.3%

表 2 展示了在验证数据集上的模型简化测试(Ablation Study)结果。由于测试数据需要上传到评测平台进行验证才能获取预测性能，因此在比赛结束后我们使用验证集进行模型简化实验。(1)中我们使用 ESIM 模型中的点积操作计算协同注意力值。(2)中我们去除原始模型中的软注意力层。(3)中我们去除原始模型中的自注意力层。(4)我们使用 word2vec 词向量代替原始模型中预训练 ELMo 产生的词向量。

4 总结

本文提出了一种基于多种注意力机制结合的文本蕴含识别模型。该模型使用深度上下文词表征模型 ELMo 产生输入的词向量，并结合了多种注意力机制，包括协同注意力机制，软注意力机制，自注意力机制等，用于表达两个句子的交互信息。我们对模型进行了集成，在第十七届中国计算语言学大会(CCL2018)中文文本蕴涵任务的数据集上，取得 76.2%的测试集准确率。

[1] https://github.com/fxsjy/jieba

参考文献

[1] Yin, W., Schütze, H., Xiang, B., et al.: ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs [J]. Transactions of the Association for Computational Linguistics, 4, 2016, 259-272.

[2] Conneau A, Kiela D, Schwenk H, et al. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data [C] //Processing of Conference on Empirical Methods in Natural Language Processing, 2018: 670-680.

[3] Shen, D., Wang, G., Wang, W., et al. Baseline Needs More Love: On Simple Word-Embedding-Based Models and Associated Pooling Mechanisms [C] //Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018: 440–450 .

[4] Wang, Z., Hamza, W., Florian, R. Bilateral Multi-Perspective Matching for Natural Language Sentences [C] //Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017: 4144-4150.

[5] Chen, Q., Zhu, X., Ling, Z., et al. Enhanced LSTM for Natural Language Inference [C] //Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 1657-1668 .

[6] Tan, C., Wei, F., Wang, W., et al. Multiway Attention Networks for Modeling Sentence Pairs [C] //Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018: 4411-4417.

[7] Peters, M., 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: Human Language Technologies, 2018: 2227–2237 .

[8] Mikolov, T., Sutskever, I., Chen, K., et al. Distributed Representations of Words and Phrases and their Compositionality [C] //Proceedings of the 27th Annual Conference on Neural Information Processing Systems, 2013: 3111-3119.

[9] 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 .

[10] Cho, K., Merrienboer, B., Gulcehre, C., et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation [C] // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014: 1724–1734.

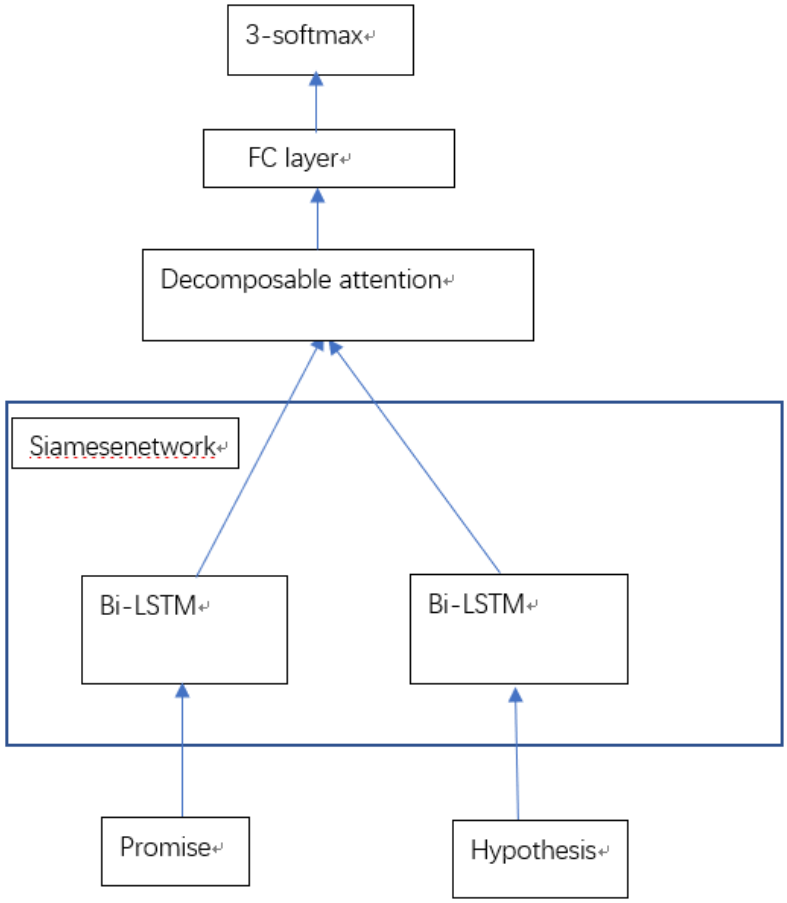
[11] Vaswani, A., Shazeer, N., Parmar, N., et al. Attention is all you need [C] // /Proceedings of the 31st Annual Conference on Neural Information Processing Systems, 2017: 6000-6010.

团队：__503

队员：严明，张真练，任函

系统描述：

本系统主要采用 Siamese network 和 decomposable attention 来判断两个句子存在的关



系（蕴含，矛盾，中性），实际上是一个三分类问题。本系统尝试采用字符向量来表示文本语义信息，利用相同权重的 Bi-LSTM 对 promise 和 hypothesis 进行建模，提取文本特征，并在模型中加入 decomposable attention，得到两个句子之间的语义交互信息，并选用 softmax 作为分类器，得到最后的分类结果。

1， promise 和 hypothesis 使用 Siamese network 用 Bi-LSTM 进行编码。

$$\bar{a}_i := BiLSTM(a,i), \forall \in [1,2,3,\dots\dots l_a]$$

$$\bar{b}_j := BiLSTM(b,j), \forall \in [1,2,3,\dots\dots l_j]$$

其中$\bar{a}_i$， $\bar{b}_j$ 使用共同参数的 Bi-LSTM。Bi-LSTM 是由两个 LSTM 上下反向叠加在一起组成的，避免句子过长造成的信息冗余、信息丢失等长期依赖问题。当输入一个子句的时，模型会分别从前向和后向两个方向的进行运算，这样既能保存子句中过去的语义信息，又能考虑到子句未来的语义信息，从而更充分的利用子句的上下文信息。

Siamese network 通过两个 Bi-LSTM 共享权值来实现，可以来衡量 promise 和 hypothesis

两个输入的语义相似程度。

2，decomposable attention，注意力机制用来提取重要的信息，并选择性的忽略无用的信息，对重要的信息做放大处理，最后将这些信息的向量组合在一起作为输出，从而进一步提升模型的性能。因为 self-attention 只能获得单个句子自己的权重信息，无法获得两个句子之间的交互信息，所以我们采用 decomposable attention

$$e_{ij} := F(\bar{a}_i, b_j) := F(\bar{a}_i)^T F(\bar{b}_j)$$

$$\beta_i := \sum_{j=1}^{l_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_b} \exp(e_{ik})} \bar{b}_j$$

$$\alpha_j := \sum_{i=1}^{l_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_a} \exp(e_{kj})} \bar{a}_i$$

e_{ij}是 promise 和 hypothesis 中每个词之间的交互权重矩阵。

α_j就是句子 hypothesis 中的每个词$\bar{b}_j$的 self-attention 权重和句子 promise 中每个词$\bar{a}_i$加权得到的结果。就是α对应 b 中的词，β对应 a 中的词。这样就能体现两个句子中每个词在相互之间的重要程度，attention 选择性的忽略和放大比较重要的信息，从而达到两个句子之间的交互。

再进行加权后的句子与原句子进行比较。

$$V_{1,i} := G([\bar{a}_i, \beta_i])$$
$$V_{2,j} := G([\bar{b}_j, \alpha_i])$$

[，]是进行拼接，就是句子中第 i 个词和另一个句子里所有词与它比较的加权词向量，G 是一个前馈神经网络。

然后把V_1，V_2中的每个元素进行求和，

$$V_1 := \sum_{i=1}^{l_a} V_{1,i}$$

$$V_2 := \sum_{j=1}^{l_b} V_{1,j}$$

最后将求和后V_1，V_2进行拼接

$$\mathrm{y} := \mathrm{H}([V_1, V_2])$$

3，将 y 连接一层全连接层，然后丢入分类器中进行三分类，判断其所属哪一类别。

$$\hat{y} := argmax_i\, y_i$$

$\hat{y}$即为最后预测的标签。

文章编号：1003-0077（2017）00-0000-00

中文句子蕴含模型

朱洪银 [1]

(1. 中国科学院 自动化研究所，北京市 100190)

摘要：中文文本蕴含推理旨在根据给定的自然语言描述的前提，判断是否能推出结论。在英语语言上有许多相关研究，但是中文需要分词，而且分词的准确性受分词工具的影响，因此我们在模型中采取了词语和字符两种粒度的输入，以抵消某些分词错误带来的影响。我们的模型包含 4 个 LSTM-CNN 模型，分别对应词语和字符粒度的输入。由于句子的字面相似性能很大程度能够决定最终结果，因此我们使用了联合训练的重复词向量和重复字向量。我们的模型在测试集上取得了 69.?%的正确率。

关键词：LSTM-CNN；重复词向量； 重复字向量

中图分类号：TP391 **文献标识码：**A

0 引言

本文提出了一种基于词语和字符向量的中文句子匹配模型。如*图 1* 所示，模型包含 4 个由 Bi-LSTM-CNN 组成的相同的结构，从左到右依次是前提的词序列，前提的字序列，结论的词序列和结论的字序列。输入的序列，映射为词向量经过 Bi-LSTM 编码之后，由 CNN[1]进行特征提取，最终表示为一个向量。4 个网络的表征拼接为一个整体。同时序列的前提和结论网络会分别求相似度得分。最终的表示会输入 softmax 分类器。

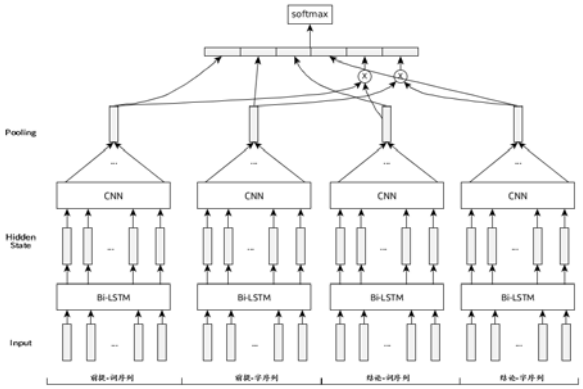


图 1 模型结构

参考文献

[1] Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).

收稿**收稿日期**：2017-03-16；**定稿日期**：2017-04-26 六号

基金项目：基金名（基金号）；基金名(基金号)

六号，核实准确完整的基金名称