文章编号：1003-0077(2017)06-0140-07

深度学习中汉语字向量和词向量结合方式探究

李伟康，李 炜，吴云芳

（北京大学 计算语言学教育部重点实验室，北京 100871）

摘 要：该文旨在探究深度学习中汉语字向量和词向量的有效结合方式。我们在以词作为基础语义单元和以字作为基础语义单元这两个方向进行探究，实验了字、词信息多种浅层结合方式和深层结合方式。为了验证该文提出的结合方式的有效性，我们改进了一种 compare-aggregate 模型，并在基于文档的问答系统上进行了实验。实验结果表明，有效的汉语字向量和词向量的结合方式超越了单独的字向量和词向量，提升了基于文档的问答系统的性能，使其结果与目前最好的结果可媲美。

关键词：字向量；词向量；深度学习；问答系统

中图分类号：TP391 **文献标识码**：A

Combination Methods of Chinese Character and Word Embeddings in Deep Learning

LI Weikang, LI Wei, WU Yunfang

(Key Laboratory of Computational Linguistics (Peking University), Ministry of Education, Beijing 100871, China)

Abstract: This paper investigates the combination of Chinese character and word embeddings in deep learning. We propose to do experiments considering shallow and deep combinations based on word and character. In order to demonstrate the effectiveness of combination, we present a compare-aggregate model solving the problem of question answering. Extensive experiments conducted on the open DBQA data demonstrate that the effective combination of characters and words significantly improves the system achieving comparable results with state-of-art systems.

Key words: character embedding; word embedding; deep learning; question answering system

0 引言

近年来，逐渐兴起的深度学习技术越来越多地被用于自然语言处理的各个领域。为了更好地表示自然语言，基于深度学习的多种模型也被提出。这些模型大多将分词工具得到的一个词作为句子的一个语义单元进行分析。对于英语来说，它的最小语义单元是单词，这样的方法比较合适。但对于汉语来说，这样的做法似乎并不合适。首先，由分词工具得到的中文分词结果并非完全正确，不同的分词工具的分词结果也不同；其次，直接用一个词作为语义单元进行表示，也忽略了词内字间的信息。另一方面，汉语中单独的一个字歧义性较大，可能是多个词的组成，若用单独的字作为一个语义单元进行表示，不能准确地表示当前语境的信息。

基于文档的问答系统(DBQA)是自然语言处理的一个热门研究领域，表 1 给出了一个例子。

表 1 一个基于文档的问答例子

文档：腾讯控股有限公司，简称腾讯，是一家民营 IT 企业，总部位于中国广东深圳，于 2004 年 6 月 16 日在香港交易所上市。…… 腾讯在线教育：2014 年 4 月 23 日讯，腾讯低调上线了腾讯课堂页面启用二级域名，腾讯课堂的上线，意味着腾讯在线教育由此达成闭环。……腾讯课堂上线，意味着 QQ 群教育在有了直播工具、支付工具后，其第三个组成部分：课程交易平台也已上线，腾讯在线教育由此达成闭环。……
问题：腾讯在线教育由哪几个部分组成？
答案：腾讯课堂上线，意味着 QQ 群教育在有了直播工具、支付工具后，其第三个组成部分：课程交易平台也已上线，腾讯在线教育由此达成闭环。

收稿日期：2017-09-29 **定稿日期**：2017-10-18

基金项目：教育部人文社会科学研究规划基金(13YJA740060)；国家自然科学基金(61371129,61773026)

从传统的机器学习方法到深度学习方法，研究者进行了大量的研究，提出了许多有效的模型。这些模型大多集中于算法的优化、模型本身的结构构建等，很少对语言表示的粒度进行分析和研究。本文在中文问答系统上进行了基于深度学习的语言表示的粒度分析和研究，即如何将字和词的信息结合起来，以期获得更好的语言表示，并在基于文档的问答系统上获得优良的性能。

本文的主要贡献有：①探究了深度学习方法中汉语字和词信息的结合方式；②在中文问答领域对深度学习方法中字和词信息的结合方式的有效性进行了验证。

本文后续部分的结构如下：第一部分对相关领域的工作进行了介绍；第二部分介绍了本文采用的面向文档问答系统的神经网络模型，第三部分详细探究字向量和词向量的多种结合方式；第四部分通过实验证明了该方法的有效性并进行了实验分析；最后一部分总结了本文工作并提出了可改进的几个方面。

1　相关工作

随着互联网的发展，自然语言的处理显得愈发重要。目前逐渐兴起的基于文档的问答系统就是自然语言处理的一个研究领域。基于文档的问答系统可以快速地回答用户基于一篇文章请求的问题，类似于中文文章的阅读理解，具有十分广泛的应用前景。深度学习是目前蓬勃发展的一个领域，已在语音和图像领域取得巨大进展，在自然语言处理领域近些年也在迅猛发展。

基于文档的问答系统是根据问题从文档里抽出相关的句子或词组作为该问题的答案，本文提到的场景为句子层面。这一问题本质上是探究两个句子的相关性，即通过问题和文档句子本身的相似性来进行答案的选择。对于该类问题，深度学习中采用卷积神经网络(CNN)[1]或循环神经网络(RNN)[2]将问题和文档的句子进行编码表示，然后判断文档中的每个句子是否和问题相关。其中常用的计算相似度的方法为夹角余弦[3]、对应元素相乘[4]及张量运算[5]。对文档中句子的简单编码表示，忽略了问题和句子的重点对应词信息。为此，前人提出注意力机制(attentive network)对句子进行编码，即在编码句子时考虑问题本身的信息[6]。为了将文档中句子上下文的信息融于当前句子，前人通过记忆网络来解决问题匹配时的信息缺失[7]。compare-aggregate 网络[8]借助句子中词层面的信息对句子本身进行信息的整合和选择。

利用深度学习解决汉语的自然语言问题时，大多数模型往往将词作为一个语义单元，并通过 word2vec 方法[9]或 GloVe 方法[10]训练得到词向量。由于中文本身是一种象形文字，一些学者试图通过研究词本身的形态特征来获得词向量表示。文献[11]提出了一种基于元素的神经语言模型，将每一个词视作一种特殊的元素词向量。文献[12]提出通过循环神经网络(RNN)提取词的形态语义学特征，进而获得词向量的表示。除了中文词本身的形态特征之外，词义消歧也得到了许多学者的关注。文献[13]提出对一个词进行多种词向量的训练，以得到不同语义下一个词的词向量表示。为了利用词内部和外部的信息，文献[14]提出了一种联合学习词向量的方法。前人的大多数方法重点考虑如何融合词的内部和外部信息来构建词本身的信息表示，很少关注字向量和词向量如何结合。

2　基于文档的问答系统的深度网络模型

本文着重探索神经网络框架下汉语字向量和词向量的有效结合方式，为了验证相关结合方式的有效性，我们选取了目前研究较为火热的基于文档问答系统的这一任务。本文采用了一种性能优良的前人模型，并结合该任务的特点进行了模型改进。

2.1　基础模型

在英文 WikiQA 数据[15]上，目前效果较好的是 compare-aggregate 模型[16]，这是一种解决两个句子相似度匹配问题的方法。它借助注意力机制计算句子间的关联信息，并通过信息融合(compare)过程充分挖掘这些关键词的信息，进而利用信息整合(aggregate)过程将得到的关键词信息进行整合。compare-aggregate 模型[16]由以下四部分组成。

(1) 预处理

通过预处理获得包含上下文信息的问句 Q 和答案句子 A 的表示，句子中的每个词将得到一个全新的词向量表示。预处理过程采用循环神经网络(RNN)进行，这里使用的是一种 LSTM/GRU 的变种，如式(1)～(2)所示。

$$\overline{Q}=\sigma(w^{i}\odot Q+b^{i}\otimes e_{Q})\odot\tanh(w^{u}\odot Q+b^{u}\otimes e_{Q}) \tag{1}$$

$$\overline{A} = \sigma(w^i \odot A + b^i \otimes e_A) \odot \tanh(w^u \odot A + b^u \otimes e_A) \quad (2)$$

其中，$\odot$表示矩阵中对应元素相乘的操作，$\otimes$表示外积的操作，w^i、w^u、b^i、b^u是神经网络中需要学习的参数。

(2) 注意力机制

这里采用标准的注意力机制，答案中的每个词的隐层向量表示h_i是该位置的词向量a_i对问题中各个词向量q_j的加权表示。i、j分别表示答案和问题中各个词的位置。计算如式(3)～(4)所示。

$$G = \text{softmax}((w^g \odot \overline{Q} + b^g \otimes e_Q)^{\mathrm{T}} \overline{A} \quad (3)$$

$$h = \overline{Q} \odot G \quad (4)$$

其中，w^g、b^g表示网络中需要训练的参数。

(3) 信息融合

通过一些操作将答案中的一个词的词向量表示a_i和经过注意力机制得到的向量h_i进行融合，得到最后的向量表示t_i。融合的操作有：前馈神经网络的连接、两个向量的张量计算、两个向量的夹角余弦相似度或欧氏距离、两个向量相减或相乘等。本文采用向量相乘方式进行信息融合。

(4) 信息整合

利用卷积神经网络(CNN)提取信息融合过程得到的向量t_i的特征向量，并将其用于最后的分类运算。

2.2 改进模型

对于基于文档的问答，原有模型未考虑文档中句子间的上下文信息。本文进行了改进，通过循环神经网络(RNN)对篇章中的每个句子再次进行编码表示，以期将更多的上下文信息融合于当前句子中。

每个已经被编码成为固定长度的单个句子向量会被作为输入加入到双向循环神经网络中。上下文信息通过隐层之间的联系获得共享，从而使得文档中的每个句子不再只是作为单独的句子单元呈现，而是通过上下文连接起来。我们将RNN的隐层节点作为对句子的表示，并在此基础上进行预测。

此外，本文将信息融合过程中的卷积神经网络CNN改为了循环神经网络RNN，以期获取更丰富的序列特征。

本文所采用的神经网络结构如图1所示。

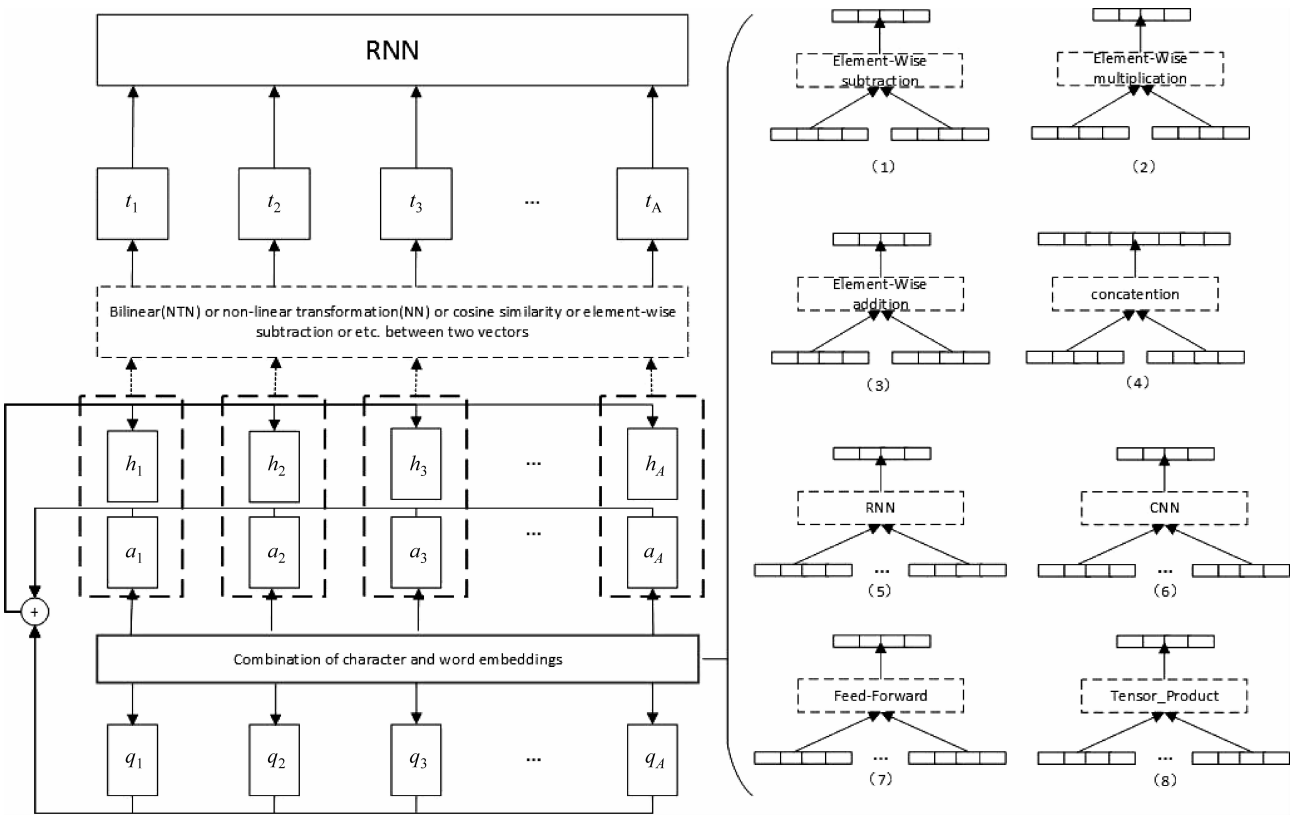

图1 本文采用的神经网络结构

这里，a^M表示答案中的一个语义单元，q^N表示问题中的一个语义单元，M和N分别表示答案和问题中语义单元的个数。图中标有“Combination of Character and word embeddings”的实线框右边

部分即为我们对字向量和词向量如何进行结合的探究，连续的矩形框表示向量。下一节将对此进行详细分析和介绍。

3　词向量和字向量的结合方式

基于中文语言本身的特点，本文将从两个方向探究字信息和词信息的结合方式：一是以词作为最小语义单元，考虑如何通过词内的字的信息得到该词语义单元的表示，或将通过字得到的词的表示和原始词的表示进行结合；另一个方向是以字作为最小语义单元，考虑如何将句子中某字所在的词的信息和该字的信息进行结合。

3.1　以词作为基本语义单元的字、词信息的结合方式

首先，我们考虑如何通过词内字的信息计算得到该词的语义向量表示。设汉语中的一个词 w_i 的词向量表示为 $\hat{v}_i^{\mathrm{w}}$，其由若干个字 c_j 组成，每个字的字向量表示为 $\hat{v}_j^{\mathrm{c}}$。通过字向量信息得到词的词向量表示 $\bar{v}_i^{\mathrm{w}}$ 的计算方法如式(5)～(11)所示。

(1) 字向量间取平均：

$$\hat{v}_i^{\mathrm{w}} = \frac{1}{n}\sum_{j=0}^{n} \hat{v}_j^{\mathrm{c}} \tag{5}$$

(2) 字向量间取最大值：

$$\bar{v}_i^{\mathrm{w}} = \max(\hat{v}_{j=0}^{\mathrm{c}}, \hat{v}_{j=1}^{\mathrm{c}}, \cdots, \hat{v}_{j=n}^{\mathrm{c}}) \tag{6}$$

(3) 字向量间取最小值：

$$\bar{v}_i^{\mathrm{w}} = min(\hat{v}_{j=0}^{\mathrm{c}}, \hat{v}_{j=1}^{\mathrm{c}}, \cdots, \hat{v}_{j=n}^{\mathrm{c}}) \tag{7}$$

(4) 字向量间并连：

$$\bar{v}_i^{\mathrm{w}} = \mathrm{concat}(\hat{v}_{j=0}^{\mathrm{c}}, \hat{v}_{j=1}^{\mathrm{c}}, \cdots, \hat{v}_{j=n}^{\mathrm{c}}) \tag{8}$$

(5) 利用前馈神经网络对字向量进行加权表示：

$$\bar{v}_i^{\mathrm{w}} = w \cdot \hat{v}_j^{\mathrm{c}} + b \tag{9}$$

(6) 利用卷积神经网络(CNN)提取字向量的特征：

$$\bar{v}_i^{\mathrm{w}} = \mathrm{CNN}(\hat{v}_{j=0}^{\mathrm{c}}, \hat{v}_{j=1}^{\mathrm{c}}, \cdots, \hat{v}_{j=n}^{\mathrm{c}}) \tag{10}$$

(7) 利用循环神经网络(RNN)构建字向量的特征：

$$\bar{v}_i^{\mathrm{w}} = \mathrm{RNN}(\hat{v}_{j=0}^{\mathrm{c}}, \hat{v}_{j=1}^{\mathrm{c}}, \cdots, \hat{v}_{j=n}^{\mathrm{c}}) \tag{11}$$

其中，i 表示一个句子中词的位置，j 表示一个词中字的位置，n 表示一个词中字的个数。w、b 表示神经网络中需要学习的参数。

其次，我们考虑如何将上述通过字信息计算得到的词向量 $\bar{v}_i^{\mathrm{w}}$ 和原始词向量 $\hat{v}_i^{\mathrm{w}}$ 进行结合，以期得到一个语义单元的最佳信息表示 v_i^{w}。计算如式(12)～(14)所示。

(1) 词向量和词向量相加：

$$v_i^{\mathrm{w}} = \hat{v}_i^{\mathrm{w}} + \bar{v}_i^{\mathrm{w}} \tag{12}$$

(2) 词向量和词向量相乘：

$$v_i^{\mathrm{w}} = \hat{v}_i^{\mathrm{w}} \cdot \bar{v}_i^{\mathrm{w}} \tag{13}$$

(3) 词向量和词向量并连：

$$v_i^{\mathrm{w}} = \mathrm{concat}(\hat{v}_i^{\mathrm{w}}, \bar{v}_i^{\mathrm{w}}) \tag{14}$$

3.2　以字作为语义单元的字、词信息的结合方式

由于汉语中单字包含完整且真实的语义，除了以词作为语义单元外，本文探究了以字作为语义单元时，句子中的每个字向量与其所在词的词向量的结合方式。用 $\hat{v}_i^{\mathrm{c}}$ 表示一个句子中第 i 个位置的字的初始字向量，$\hat{v}_i^{\mathrm{w}}$ 表示该句中该字所在词的词向量，w、b 为神经网络中需要学习的参数，相关计算如式(15)～(19)所示。

(1) 字向量和词向量相加：

$$v_i^{\mathrm{c}} = \hat{v}_i^{\mathrm{c}} + \bar{v}_i^{\mathrm{w}} \tag{15}$$

(2) 字向量和词向量相乘：

$$v_i^{\mathrm{c}} = \hat{v}_i^{\mathrm{c}} \cdot \bar{v}_i^{\mathrm{w}} \tag{16}$$

(3) 字向量和词向量并连：

$$v_i^{\mathrm{c}} = \mathrm{concat}(\hat{v}_i^{\mathrm{c}}, \bar{v}_i^{\mathrm{w}}) \tag{17}$$

(4) 利用前馈神经网络对字向量和词向量进行加权表示：

$$v_i^{\mathrm{c}} = w \cdot (\hat{v}_i^{\mathrm{c}}, \bar{v}_i^{\mathrm{w}}) + b \tag{18}$$

(5) 利用张量对字向量和词向量进行抽象表示：

$$v_i^{\mathrm{c}} = \mathrm{tensor\_product}(\hat{v}_i^{\mathrm{c}}, \bar{v}_i^{\mathrm{w}}) \tag{19}$$

4　实验

4.1　实验数据

本文的实验采用了 NLPCC 2016 的 DBQA 数据[17]。该数据的格式为：对于一篇文档，给出一个问题，训练数据中文档的每个句子和问题都进行了人工的标记，若标记为 1，表示该句子和问题相关，可以作为该问题的回答；若标记为 0，表示该句子与问题无关。测试数据需要系统给出问题和文档的每个句子是否相关的判断。由于该数据集无开发集，本文随机从训练数据集拿出一部分作为开发集进行参数调整。数据分布如表 2 所示。

表 2　实验数据说明

	训练数据	测试数据
问题数(篇章数)	8 772	5 997
“问题-答案”数	181 882	12 253

4.2 实验设置

由于句子长度不同，无论是以词作为语义单元还是以字作为语义单元，我们均将句子长度（即句中语义单元的个数）固定为相同，根据句子长度的分布，该值设为200。另外，组成一个词的字的个数不同，本文均将字的个数设定为4。本文借助jieba分词工具[18]进行分词，利用word2vec训练方法在中文语料上训练得到字向量表和词向量表。训练得到的字向量表中字的个数为8 009，词向量表中词的个数为316 056，向量的维度均为300维。

本文的实验代码通过Python及TensorFlow实现。神经网络模型中，学习方法采用Adam，学习率为0.001。RNN的内核采用LSTM结构。实验中批处理的“问题-答案”数为10，经过对训练数据的若干轮训练得到最佳模型，并在测试数据上进行测试。

答案选择的任务最终通过打分排序的方法进行处理。因此，评测时我们采用在信息检索领域广泛使用的指标，包括MAP、MRR、Precision、Recall、F1。

4.3 实验结果与分析

（1）模型的结果

我们对比了本文融合上下文信息的改进模型和基础模型在以单独的词向量作为语义表示时的实验结果，并基于改进模型对单独的词向量表示和单独的字向量表示进行了性能对比。结果如表3所示。

表3 模型结果对比

	MAP	MRR	Precision	Recall	F1
基础模型（基于词）	0.746 8	0.747 8	0.712 8	0.493 7	0.583 4
改进模型（基于词）	0.753 8	0.762 3	0.665 5	0.650 3	0.657 9
改进模型（基于字）	**0.807 3**	**0.808 2**	**0.724 4**	**0.705 2**	**0.714 7**

从以上实验结果可知，以单独词向量作为语义表示时，本文的改进模型比基础模型在关键指标MAP、MRR、F1上表现优良，证明句子上下文信息可以帮助捕捉正确答案。令人惊喜的是，以字向量作为语义表示时的效果在各项指标上均显著优于词向量，在关键指标MAP上提高了5.45%。这一方面是因为汉语的字确实携带了重要的语义信息，另一方面是因为我们处理的语料大多来自网络文本，语言使用不太规范，自动分词器会有很多分词错误，影响了词向量的准确度。同时，由于字的个数远远小于词表的个数（8 009 vs. 316 056），基于字的模型速度也远远高于基于词的模型。

（2）以词作为语义单元的结果

首先，我们通过词内字的信息来计算得到词的信息，实验结果如表4所示。

表4 用字的信息计算得到词的向量表示

	MAP	MRR	Precision	Recall	F1
Word	0.753 8	0.762 3	0.665 5	0.650 3	0.657 9
Char-Concat	0.766 4	0.767 3	0.671 6	0.654 3	0.662 8
Char-Max	0.724 3	0.725 2	0.618 0	0.596 0	0.606 8
Char-Min	0.708 6	0.709 7	0.593 2	0.565 6	0.586 6
Char-Mean	**0.788 6**	**0.789 3**	**0.703 9**	**0.683 8**	**0.693 7**
Char-FF	0.748 6	0.749 2	0.653 7	0.632 3	0.642 8
Char-CNN	0.774 4	0.775 3	0.695 4	0.655 7	0.675 0
Char-RNN	**0.788 7**	**0.789 7**	**0.701 1**	**0.683 0**	**0.691 7**

从以上实验结果可知，以词作为语义单元时，通过有效地结合词内字的字向量来计算词的信息要优于词本身的词向量。其中，浅层结合方式中并连操作（Char-Concat）和取平均操作（Char-Mean）均比原词的词向量要好，深层结合方式中CNN操作和RNN操作也比原词的词向量要好。表现最好的是RNN操作，较词本身向量提升了3.49%。但对比表3和表4可知，用字信息计算得到词的向量表示仍然不敌单独基于字的模型。

其次，我们用较好的词内字向量的结合方式（浅层结合中的并连操作及深层结合中的RNN操作）计算的词向量和原词的词向量进行结合，实验结果如表5所示。

表5 以词作为语义单元的字结合词的实验结果

	MAP	MRR	Precision	Recall	F1
Word	0.753 8	0.762 3	0.665 5	0.650 3	0.657 9
Word_Concat_Char-Concat	**0.815 6**	**0.818 6**	**0.727 3**	**0.708 5**	**0.720 0**
Word_Plus_Char-Concat	0.787 2	0.788 0	0.699 6	0.679 3	0.689 3
Word_Multi_Char-Concat	0.729 8	0.731 0	0.643 0	0.609 6	0.625 9
Word_Concat_Char-RNN	**0.823 9**	**0.824 9**	**0.746 8**	**0.731 7**	**0.739 2**

从以上实验结果可知，通过词内字的字向量和词向量的并连操作得到该词的最终表示比原词的词向量要好。其中，词内字的结合方式为并连时，比单独以词向量作为表示时的结果提升了 6.18%；词内字的结合方式为 RNN 时，效果提升了 7.01%。对比表 3 和表 5 可知，用字的信息计算得到词的向量表示，再与原词向量结合，其性能超越了单独基于字的模型。

(3) 以字作为语义单元的结果

进一步，我们将以字作为语义单元，对字向量和词向量的结合方式进行探究，实验结果如表 6 所示。

表 6　以字作为语义单元的实验结果

	MAP	MRR	Precision	Recall	F1
Char	0.807 3	0.808 2	0.724 4	0.705 2	0.714 7
Char_Multi_Word	0.520 9	0.522 2	0.398 0	0.332 8	0.362 5
Char_Plus_Word	**0.825 1**	**0.826 1**	**0.760 9**	**0.727 4**	**0.743 7**
Char_Concat_Word	**0.834 6**	**0.835 6**	**0.759 0**	**0.747 2**	**0.753 0**
Char_FF_Word	**0.821 8**	**0.822 8**	**0.751 7**	**0.725 4**	**0.738 3**
Char_TensorProduct_Word	0.560 9	0.562 1	0.428 1	0.396 9	0.411 9

从以上实验结果分析可知，通过字的字向量和其所在词的词向量的结合，可有效提升单独字向量的表示。这是因为，汉语单字的意义歧义性和模糊性较大，而加入词层面信息可以有效地消解部分歧义。浅层结合方式中并连操作(Char_Concat_Word)和相加操作(Char_Plus_Word)是有效的，相乘操作(Char_Multi_Word)是无效的；深层结合方式中的前馈神经网络连接(Char_FF_Word)的操作是有效的，张量操作是无效的。其中，浅层结合中字向量和词向量的并连操作的效果最好，效果较单独字向量提升了 2.73%。

目前，在 NLPCC 2016 的 DBQA 评测中，表现最好的两个结果为文献[18]和文献[19]，表 7 为本文最好的结果和当前最好的结果的对比。

表 7　与前人结果的对比

	MAP	MRR	Precision	Recall	F1
Char_Concat_Word	**0.834 6**	**0.835 6**	**0.759 0**	**0.747 2**	**0.753 0**

续表

	MAP	MRR	Precision	Recall	F1
Rank 1	**0.859 2**	**0.858 6**	**0.790 6**	—	—
Rank 2	0.826 9	0.826 3	0.738 5	—	—

从以上的实验结果知，我们最好的结果在该任务上的表现非常满意。由于 NLPCC 2016 的 DBQA 为评测任务，参赛队伍可借助多种资源提高实验结果。文献[18]借助外部知识中文维基百科进行词向量的训练，并引入词共现的外部特征，且其论文中指出这些操作对其最终结果有很重要的影响。文献[19]则通过从各个维度提取特征的特征工程方法进行系统构建。我们的方法仅仅依赖训练数据本身，并未刻意进行特征的筛选和集成，仍取得了相媲美的结果。

5　结语

本文探究了深度学习中汉语字向量和词向量的结合方式，发现对字向量和词向量的有效结合可提高基础语义单元的信息表示，并对基于文档的问答系统有很好的性能提升。

通过字向量和词向量的结合得到的新的向量表示比单独的字向量或词向量的效果好。在以词为语义单元时，通过对词内字进行 RNN 编码表示的新词向量和原词向量并连的结合方式最为有效，效果最好；在以字为语义单元时，通过字向量和其所在词的词向量的并连结合方式最为有效，效果最好。此外，本文以字为语义单元通过字向量和词向量的有效结合使得我们在基于文档的问答(DBQA)问题中取得了满意的结果。

虽然本文从两个方面多个角度对字向量和词向量的结合方式进行了探究，但并未考虑中文的句法结构信息。后续的研究中可尝试在结合字向量和词向量时考虑句法结构信息，深入挖掘中文语言本身的信息。另外，本文提出的方法在处理句子间共现词较多时表现并不好，后续的工作可考虑将句间浅层语义特征融合于深度学习的深层表示中。

参考文献

[1] Feng M, Xiang B, Glass M R, et al. Applying deep learning to answer selection: A study and an open task [C]//Proceeding of Automatic Speech Recognition and

Understanding (ASRU), 2015 IEEE Workshop on. IEEE, 2015: 813-820.
[2] Tan M, Santos C, Xiang B, et al. LSTM-based deep learning models for non-factoid answer selection[J]. arXiv preprint arXiv:1511.04108, 2015.
[3] Tan M, Dos Santos C, Xiang B, et al. Improved representation learning for question answer matching [C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016.
[4] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks[J]. arXiv preprint arXiv: 1503.00075, 2015.
[5] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference [J]. arXiv preprint arXiv:1508.05326, 2015.
[6] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional attention flow for machine comprehension[J]. arXiv preprint arXiv:1611.01603, 2016.
[7] Sukhbaatar S, Weston J, Fergus R. End-to-end memory networks[C]//Advances in Neural Information processing systems, 2015: 2440-2448.
[8] Wang S, Jiang J. Learning natural language inference with LSTM[J]. arXiv preprint arXiv: 1512.08849, 2015.
[9] Goldberg Y, Levy O. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method[J]. arXiv preprint arXiv:1402.3722, 2014.
[10] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//EMNLP. 2014, 14: 1532-1543.
[11] Alexandrescu A, Kirchhoff K. Factored neural language models[C]//Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 2006: 1-4.
[12] Luong T, Socher R, Manning C D. Better word representations with recursive neural networks for morphology[C]//Proceedings of CoNLL 2013, 2013: 104-113.
[13] Huang E H, Socher R, Manning C D, et al. Improving word representations via global context and multiple word prototypes[C]//Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, 2012: 873-882.
[14] Chen X, Xu L, Liu Z, et al. Joint learning of character and word embeddings[C]//Proceedings of the 24th International Joint Conference on Artificial Intelligence. 2015.
[15] Yang Y, Yih W, Meek C. WikiQA: A challenge dataset for open-domain question answering[C]//Proceedings of the EMNLP 2015. 2015: 2013-2018.
[16] Wang S, Jiang J. A compare-aggregate model for matching text sequences[J]. arXiv preprint arXiv: 1611.01747, 2016.
[17] Duan N. Overview of the NLPCC-ICCPOL 2016 Shared task: open domain Chinese question answering [C]//Proceedings of International Conference on Computer Processing of Oriental Languages. Springer InternationalPublishing, 2016: 942-948.
[18] Sun J. 'Jieba' Chinese word segmentation tool[CP/OL]. 2012. https://github.com/whtsky/jieba/
[19] Fu J, Qiu X, Huang X. Convolutional deep neural networks for document-based question answering [C]//Proceedings of International Conference on Computer Processing of Oriental Languages. Springer International Publishing, 2016: 790-797.
[20] Wu F, Yang M, Zhao T, et al. A hybrid approach to DBQA[C]//Proceedings of International Conference on Computer Processing of Oriental Languages. Springer International Publishing, 2016: 926-933.

李伟康(1993—),硕士,主要研究领域为智能问答系统,计算语言学。
E-mail: wavejkd@pku.edu.cn

李炜(1993—),硕士,主要研究领域为智能问答系统,计算语言学。
E-mail: liweitj47@pku.edu.cn

吴云芳(1973—),通信作者,博士,副教授,主要研究领域为智能问答系统,计算语言学。
E-mail: wuyf@pku.edu.cn