基于卷积神经网络和词重叠模型的问答排序

孟令涛[1] 段新宇[1] 徐晓刚[2] 赵洲[1]

（1.浙江大学计算机科学与技术学院人工智能所，浙江 杭州 310027
2.浙江大学信息与电子工程学院，浙江 杭州 310027）

摘要：本文旨在研究问答领域中的答案排序任务，提出了一个结合卷积神经网络和词重叠的模型。本模型可对某一问题下的所有答案进行排序，所得排序结果趋近人工的标注准确度。本文使用词向量模型表达原始的问答文本数据，利用卷积神经网络提取问题-答案对的特征，将每一个问题与每一个答案映射在特定的语义空间中，并计算每一个问题-答案对之间的关联性。根据关联性高低，对某一问题下的答案进行质量排序。与此同时，我们根据不同词向量之间的余弦距离，设计了近义词表，并在此基础上建立了词重叠模型，计算问题与答案的文本关联性。通过综合利用卷积神经网络与词重叠模型，我们在已有的数据集上进行测试，发现我们设计的模型表现较好。
关键词：卷积神经网络；词重叠模型；问答排序

A Question Answer Ranking Method Based On Convolutional Neural Network And Wordoverlap

MENG Ling-tao[1], DUAN Xin-yu[1], XU Xiao-gang[2], ZHAO Zhou[1]

（1. Institute of Artificial Intelligence, College of Computer Science, Zhejiang University，Hangzhou, Zhejiang, 310027)
（2. College of Information and Science and Electronic Engineering, Zhejiang University，Hangzhou, Zhejiang, 310027)

Abstract: This paper aims at solving the problem of answer ranking in the question answering research. We propose a method which combines the model of convolutional neural network (CNN) and wordoverlap. The proposed method is able to rank the quality of the answers w.r.t a certain question, and the experimental results reach the level of human's operation. We use Word2Vec to represent all the textual data. Then we employ a CNN model to extract features of each question and answer and map all these questions and answers into a shared semantic space. The CNN model is able to rank the answers on the basis of the relevance between question-answer pair. In addition, we construct a homoionym list according to the cosine similarity between different words in Word2Vec and propose a wordoverlap model to calculate the textual relevance between question-answer pair. By leveraging CNN and wordoverlap model, we conducted experiments on some existing dataset. The results demonstrates the effectiveness of our method.
Keywords: CNN; wordoverlap; answer ranking;

1. 引言

面向智能问答的答案排序任务一直是自然语言处理和人工智能领域的一个前沿研究课题。其目标是为用户通过自然语言表述的问题直接提供精确答案。在本文中，针对 CCIR 和搜狗搜索联合主办的“面向智能问答的篇章排序”智能问答评测比赛上的任务，我们提出了一个

作者简介：孟令涛（1992 年 4 月），男，硕士，deep learning and natural language processing
段新宇（1991 年 5 月），男，博士，deep learning and natural language processing
徐晓刚（1996 年 1 月），男，学士，deep learning and natural language processing
赵洲（1988 年 1 月），男，副教授，deep learning and natural language processing

基于卷积神经网络（CNN）[1]以及词重叠模型（wordoverlap）的章节排序方法。

近些年来，深度神经网络在人工智能领域已经表现出十分良好的效果。在这里，鉴于 CNN 强大的特征提取能力以及词向量模型（word2vec）的空间语义映射能力，我们成功构建了一个能够将某个问题-答案对映射成一个用来衡量其关联性的分数的 CNN 模型。该卷积神经网络模型产生的分数应该符合以下的特性：该分数应该符合人工标注的时候的准确趋势，即问题和答案之间的匹配程度越高，该分数应该越高。

除此之外，传统的基于统计的自然语言处理模型已经在多年的研究中表现出具有很高的使用价值。在一般的问题和答案下，存在许多的相同词或者是近义词。这些特征是匹配的问题和答案之间十分重要的依据。词重叠算法[2]的本质在于使用问题和答案之间的近义词的数目以及他们的相似度来衡量他们之间的匹配程度， 从而能够得到一个对应的分数。在这里我们使用不同词向量之间的余弦相似度首先来构建一个近义词表，以便可以更好地处理问题和答案之间的相似度并且得到较为准确的结果。

从上述的描述和分析中可以看出，词重叠模型是基于统计的模型，具有较好的稳定性，但是在实际中由于分词的方法以及其他的特殊字符的出现，问题和其匹配的答案之间却可能存在很少的相同或者近义词，但是这并不意味着这个问题答案对之间的关联性就很低，所以在这种情况下，我们不适合使用词重叠模型来衡量。而 CNN 模型是基于深度学习的一个能够高效地提取问题答案对语义特征的模型。它能够自动地很好提取问题和答案的在语义空间中的定量特征，并且进行依据此进行匹配程度的分析。但是这个模型的效果收到我们训练的方式，以及训练数据的多少的影响。在很多情况下的鲁棒性不如 wordoverlap 模型。为了结合两者的优势并且互补他们之间的短处，在这里我们提出了一个联合 CNN 和 wordoverlap 模型的方法。在问题和答案之间存在比较多的相同词汇或者近义词的时候，我们使用鲁棒性较好的 wordoverlap 模型；当问题和答案之间的相同词或者近义词的数目较少的时候，可以使用 CNN 模型来获得较好的效果。在算法设计的最后，我们也使用已经公开的训练数据集上的数据进行测试，发现我们这个联合模型的效果是比较好的。

下面将详细阐述 CNN 模型的设计，CNN 与 wordoverlap 联合模型的设计，以及我们在实现的时候，编写的代码描述。

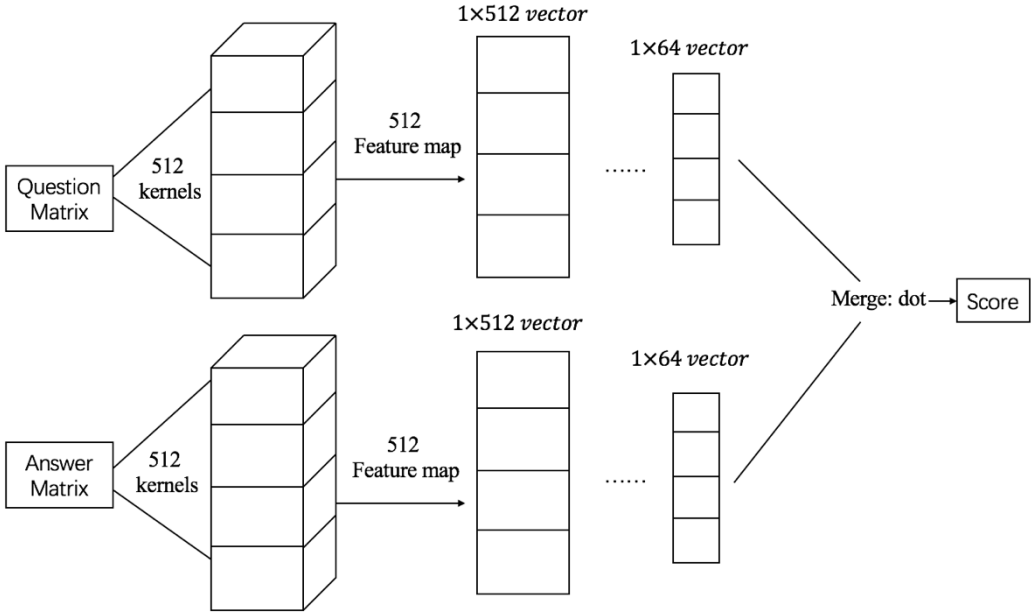
2. CNN 模型设计

1） Word2vec 的训练

首先我们需要将给出的训练数据集以及测试数据集中的中文单词映射成为向量的形式，也就是一个 embedding 的过程。在这里我们使用的方法是首先使用 nlpir 的包进行分词，并且用分好的词来训练一个词向量（word2vec）模型[3]。词向量模型是一个高效的，能够将单词映射到特定的语义空间中的词向量的机器学习模型。我们在这个模型的训练过程中使用了维基百科的训练数据[4]和本次比赛给出的训练数据以及测试数据。在训练完成之后，我们可以将分好词的单词映射成为相同维度的向量，在这里我们选择将每一个单词都映射称为 128 维的向量。我们可以看到的是，问题和答案将被各自映射成为一个二维矩阵形式。针对本次的数据的特点，一般来说问题能够分出的单词较少，答案能够分出的单词数较多，所以这个答案矩阵的高度明显高于问题矩阵。我们认为这是合理的，因为问题对于其下的每一个答案来说都是一样的。那么其高度并不会很大程度上影响最后的匹配分数的计算。

2）CNN 模型的设计

在使用训练好的 word2vec 模型之后，我们已经可以得到一个问题矩阵和答案矩阵。由此，该数据已经映射成为固定维度的矩阵，可以放入卷积神经网络模型进行特征提取。问题矩阵和答案矩阵分别通过两路卷积神经网络，并且分别输出一个向量，即我们的需要提取的关于问题和答案的语义特征。得到了特征之后，就可以将其映射成为衡量问题-答案对相似度的分数，在这里是将两者的向量进行 merge 的操作，也就是通过点积两个向量的方式。下图中说明了我们的 CNN 网络的基本设计框架：



在这个卷积神经网络中，输入的数据是之前已经映射成功的问题矩阵和答案矩阵。在他们分别通过第一个卷积层之后，分别输出了 512 个特征图。之后再次通过卷积的操作，将新得到的特征图进行拼接，而且拼接称为一个 512 维的向量。向量经过多层全连接层的神经网络之后，就得到了我们需要的 64 维的特征向量。问题和答案的两路卷积神经网络，一共会输出两个特征向量，分别代表问题和答案在高层语义空间中的特征。两者的点积就成为最后的，衡量问题和答案之间相似度的分数。该分数应该符合人工标注的时候的趋势，即问题和答案之间的匹配程度越高，该分数应该越高。

3. CNN 和 wordoverlap 模型联合

wordoverlap 模型是基于统计的自然语言处理模型。尤其是问题和答案之间存在较多的相同的词的时候，具有较好的表现。wordoverlap 算法的本质在于使用问题和答案之间的近义词的相似度来衡量他们之间的匹配程度， 从而能够得到一个对应的分数。CNN 模型和 wordoverlap 模型各自具有优势而且在数据集上都取得了很好的效果。在这里我们考虑将他们两者联立。以期能够获得更好的效果。

我们将两个模型联合的方法是这样的：首先我们需要构建近义词表，若对于测试数据的问题及答案，有在近义词表中的单词，则作为相同词对待。遍历所有测试数据中的问题及答案，若其中的相同词及近义词的对数超过一个阈值，则使用 wordoverlap 的方式产生对于该问题的问题答案对分数；若其中的相同词及近义词的对数少于一个阈值，则使用 CNN 模型的方法来产生对于该问题的问题答案对分数。除此之外，我们还引入了其他的判别机制，以期能够有更加合理的安排以及更好的效果。在实际的最后的得分中，我们采取如下的做法：因为在分词的时候，有些单词的数目特别少，或者问题和答案之间实在太相近，我们模型得出的分数对于不同的问题相差很少的时候，按照一般的人工的思想，我们最后应该对其进行随机排序。以上的思想和模型经过我们对于实际的，已经公开的训练数据集上的测试之后，发现效果是十分好的。

4. 代码描述

对于训练及测试数据的解析编码部分，该部分是通过如下两个步骤完成：

1)通过 get_words_for_test_json.py 这份程序完成对于输入的训练及测试数。 数据的读入、解析、分词功能。

2)通过 train_word2vec_model.py 这份程序

完成对于分好的单词进行进行训练，训练出对应的 wor2vec 模型。

3) 对于分好词的输入数据，通过 get_CNN_wordoverlap_score.py 进行处理，处理方法如下：

1> 通过求解余弦距离[5]建近义词表，若对于测试数据的 query 及 answer，有在近义词表中的单词，则作为相同词对待。遍历所有测试数据中的问题及答案，若其中的相同词及近义词的对数超过一个阈值，则使用 wordoverlap 的方式产生对于该问题的问题答案对分数；若其中的相同词及近义词的对数少于一个阈值，则使用 CNN 模型的方法来产生对于该问题的问题答案对分数。

2> wordoverlap.py 完成对于相应问题采用 wordoverlap 的方式得到相应问题答案对应分数的功能。

3> 完成 CNN 的部分:CNN 的代码执行分为训练与测试两个部分：
i)训练：执行 Final_CNN_Model_train.py 进行 CNN 模型的训练;
ii)测试：执行 Final_CNN_Model_test.py 对测试集得到每一个问题对应的不同答案的分数。

4)对于上面的两种方法分别得到对应的问题的答案排序，并存入结果文件中，完成比赛的要求任务。

5. 结论

本文提出了一个基于卷积神经网络(CNN)和词重叠（wordoverlap）模型联合的章节排序方法。CNN 模型在近几年中已经表现出强大的特征提取能力，在这里结合词向量模型的映射，能够很好地提取出问题-答案对之间的高层语义关系。词重叠模型作为基于统计的自然语言处理模型，能够根据问题和答案中的相同词和近义词来衡量它们之间的匹配程度。通过余弦相似度来建立近义词表，在最后的联立模型中，我们统计问题和答案中的相同词和近义词个数来判断应该使用哪个模型。通过综合利用卷积神经网络与词重叠模型，我们在已有的数据集上进行测试，发现我们设计的模型表现较好。

6. 参考文献

[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C];Advances in neural information processing systems. 2012: 1097-1105.

[2] Mehdad Y, Magnini B. A word overlap baseline for the recognizing textual entailment task[J]. Unpublished manuscript, 2009.

[3] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[4] Wikimedia Downloads.https://dumps.wikimedia.org/

[5] Steinbach M, Karypis G, Kumar V. A comparison of document clustering techniques[C];KDD workshop on text mining. 2000, 400(1): 525-526.