博士学位论文

基于深度神经网络的文本表示及其应用

DEEP NEURAL NETWORKS FOR TEXT REPRESENTATION AND APPLICATION

户保田

哈尔滨工业大学

2016 年 6 月

国内图书分类号：TP391.1　　学校代码：10213
国际图书分类号：004.8　　密级：公开

工学博士学位论文

基于深度神经网络的文本表示及其应用

博 士 研 究 生：户保田

导　　师：陈清财教授

申 请 学 位：工学博士

学　　科：计算机应用技术

所 在 单 位：深圳研究生院

答 辩 日 期：2016 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: TP391.1
U.D.C: 004.8

Dissertation for the Doctoral Degree in Engineering

DEEP NEURAL NETWORKS FOR TEXT REPRESENTATION AND APPLICATION

Candidate: Baotian Hu
Supervisor: Prof. Qingcai Chen
Academic Degree Applied for: Doctor of Engineering
Specialty: Computer Application Technology
Affiliation: Shenzhen Graduate School
Date of Defence: June, 2016
Degree-Conferring-Institution: Harbin Institute of Technology

摘 要

近年来，深度神经网络在诸如图像分类、语音识别等任务上被深入探索并取得了突出的效果，表现出了优异的表示学习能力。文本表示一直是自然语言处理领域的核心问题，传统的文本表示的维数灾难、数据稀疏等问题，已经成为大量自然语言处理任务性能提高的瓶颈。近年来，通过深度神经网络对文本学习表示逐渐成为一个新的研究热点。然而，由于人类语言的灵活多变以及语义信息的复杂抽象，深度神经网络模型在文本表示学习上的应用更为困难。本文旨在研究深度神经网络对不同粒度的文本学习表示，并将其应用于相关任务上。

首先，对词向量的学习进行了研究。提出了一种基于动名分离的词向量学习模型。该模型将词性引入到词向量的学习过程，同时保持了词序信息。受人类大脑的动名分离结构的启发，在学习词向量的过程中，该模型根据词性标注工具得到的词性，动态的选择模型顶层的网络参数，从而实现模型的动名分离。与相关向量学习方法进行实验对比，结果显示该模型能够以相对较低的时间复杂度，学习得到高质量的词向量；通过其得到的常见词的相似词更为合理；在命名实体识别和组块分析任务上的性能，显著地优于其它对比的词向量。

其次,对语句的表示学习进行了研究。提出了基于深度卷积神经网络的语句表示模型。该模型不依赖句法分析树，通过多层交叠的卷积和最大池化操作对语句进行建模。语句匹配对自然语言处理领域的大量任务非常重要。一个好的匹配模型，不仅需要对语句的内部结构进行合理建模，还需要捕捉到语句间不同层次的匹配模式。基于此，本文提出了两种基于深度卷积神经网络的语句匹配架构。架构一，首先通过两个卷积神经网络分别对两个语句进行表示，然后通过多层感知机进行匹配。架构二，则是对两个语句的匹配直接建模，然后通过多层感知机对匹配表示进行打分。两种匹配架构都无需任何先验知识，因此可被广泛应用于不同性质、不同语言的匹配任务上。在三种不同语言、不同性质的语句级匹配任务上的实验结果表明，本文提出的架构一和架构二远高于其他对比模型。相比架构一，架构二更能够有效地捕捉到两个语句间多层次的匹配模式，架构二在三种任务上取得了优异的性能。

第三，对统计机器翻译中短语对的选择进行了研究。提出了上下文依赖的卷积神经网络短语匹配模型。该模型对目标短语对进行选择，不仅考虑到了

源端短语与目标端短语的语义相似度，同时利用了源端短语的句子上下文信息。为了有效的对模型进行训练，提出使用上下文依赖的双语词向量初始化模型，同时设计了一种“课程式”的学习算法对模型进行从易到难、循序渐进的训练。实验表明，将该模型对双语短语的匹配打分融入到一个较强的统计机器翻译系统中，可以显著提高翻译性能，BLEU 值提高了 1.0%。

第四，对摘要自动生成进行了研究。构建了一个较高质量的大规模中文短文本摘要数据集，该数据集包括 240 多万的摘要，同时构造了一个高质量的测试集。提出使用基于循环神经网络的编码 -解码架构从大规模数据集中自动学习生成摘要，构建了两个基于循环神经网络的摘要生成模型。模型一通过使用循环神经网络对原文进行建模，并将其最后一个状态作为原文段落的表示，利用另一个循环神经网络从该表示中解码生成摘要。模型二在模型一的基础上，通过动态的从编码阶段的循环神经网络的所有状态中综合得到上下文表示，然后将当前的上下文表示传递给解码循环神经网络生成摘要。两种模型都是产生式模型，无需任何人工特征。实验表明，两种模型能够对原文进行较为合理的表示，生成具有较高信息量的摘要文本。特别地，模型二生成的摘要文本质量显著优于模型一。

综上所述，本文以深度神经网络为手段，以文本表示为研究对象，对自然语言中不同粒度的文本即词、句、段的表示学习及其应用进行了深入研究。本文将所提出的方法应用到了序列标注、语句匹配、机器翻译以及自动文摘生成问题上，并取得了良好的效果。

关键词：深度学习；语言表示；词向量；语义匹配；自动文摘

Abstract

In recent years, deep neural networks (DNN) have been explored on various of tasks such as image classification and speech recognitions, on which DNN achieves state of the art performances and shows the powerful ability for representation learning. Language representation is the core issue of natural language processing. The over simplified bag of word hypothesis has been the bootle neck of various tasks because of the curse of dimension, data sparsity. In recent years, DNN have dominant the research on language representation learning. However, the flexibility and rich semantic information of language make the representation learning via DNN more difficult. This thesis aims to research on the language representation learning and its applications via DNN.

Firstly, the word representation learning is studied. The Continuous Dissociation between Nouns and Verbs Model (CDNV) is proposed. CDNV integrates pos tag information into the word embeddings learning process, while preserving the word order. Inspired by the principle the dissociation between nouns and verbs, the model can dynamically choose the connection on the output layers according to the pos tag information. Comparisons to most of public word embeddings show that CDNV is able to learn high-quality word embeddings with relatively low time complexity. The nearest neighbors of some representative words derived from the CDNV word embeddings are more reasonable. The performance improvements on F1 measure from CDNV word embeddings are significantly greater than other word embeddings on NER and Chunking.

Secondly, the sentence modeling is studied. The deep convolutional neural network sentence model is proposed. This model need not rely on parsing tree and can represent the hierarchical structures of sentences with layer-by-layer convolution and pooling. Semantic matching is of central importance to many natural language tasks. A successful matching algorithm needs to adequately model the internal structures of language objects and the interaction between them. As a step toward this goal, we propose two deep convolutional neural network based sentence matching architecture. Architecture I gets two sentence representations via two different convolutional neural networks, and then the multiple layer perceptron is used to match them. While Architecture II models the matching of two sentences directly, and score the matching representation via multiple

layer perceptron. The two architectures require no prior knowledge, and can hence be applied to matching tasks of different nature or languages. The empirical study on a variety of matching tasks demonstrates the efficacy of the proposed architectures and its superiority to competitor models. Architecture II is superior to Architecture I on capturing the hierarchical matching patterns of two sentences. Architecture II achieves state of the art performances on three tasks.

Thirdly, the bilingual phrases selection is studied. The Context-Dependent Convolutional Neural Network Bilingual Phrases Matching Model is proposed. It encodes not only the semantic similarity of the translation pair, but also the context containing the phrase in the source language. In order to train the model efficiently, we initialize the word embeddings by the pre-trained context dependent bilingual word embeddings. A curriculum learning algorithm is proposed to train the model, which can gradually train the model from easy to difficult. Experimental results show that our approach significantly outperforms the baseline system by 1.0 BLEU points.

Fourthly, The automatic text summarization is studied. This thesis constructs a large scale Chinese short text summarization which consists of over 2.4 million data. A high-quality test set is also constructed. The recurrent neural network (RNN) encoder-decoder summarization generation architecture is proposed and two models are constructed. Model I uses one RNN to model the short text (RNN-encoder) and the last hidden state is used to represent the short text. The another RNN is used to generate summary from the short text representation (RNN-decoder). Based on Model I, Model II dynamically construct the context from all hidden states of RNN-encoder. The two models require no prior knowledge. Experiment results show that the two model can generate informative summary. Especially, the generated summaries of Model II are better than Model I significantly.

To sum up, this thesis used DNN to study the representation of different text grain i.e word, sentence and paragraph. The proposed models and methods are applied on the sequence labeling, sentence matching, machine translation and automatic text summarization tasks. And some works reach the state of the art performances.

Keywords: Deep Learning, Language Representation, Word Vector, Semantic Matching, Automatic Text Summarization

目 录

Contents

第 1 章 绪论

1.1 课题的研究背景与意义

构建并成功训练深层的神经网络模型[1]是机器学习领域一直的研究热点，多年来并没有取得突破的进展[2]。直到 2006 年，加拿大多伦多大学 Geoffrey Hinton 和 Ruslan Salakhutdinov 在《科学》上发表了一篇利用深度神经网络进行数据降维的文章[3]，掀起了深度神经网络研究的浪潮。该工作主要传递了两方面的信息：1. 通过多层非线性运算，深度神经网络表现出了突出的表示学习能力，能够准确的捕捉到数据更为本质的信息，更有利于数据的分类；2. 通过无监督学习对深度的神经网络采用逐层自动编码（Auto Encoder）的方式进行预训练，能够有效地降低模型的训练难度。这种深度神经网络又被称之为深度学习。

Bengio 将深度模型定义为具有多层非线性运算的模型，其典型代表就是深度神经网络。深度模型可以学习到数据不同抽象层次的表示。与之相对应的模型称之为浅层模型（例如，支持向量机模型）[2]。相关研究表明人类大脑同样是按照层次结构进行组织的。给定一个输入，大脑可以抽象出不同层次的表示，不同层次的表示对应到大脑皮层不同区域[4]。大脑在处理信息的过程中，需要在不同阶段对信息进行变换，以得到不同抽象层次的表示。近年来，深度模型在多种任务上取得了突破，微软和谷歌先后采用深度神经网络架构将语音识别错误率降低了 20%~30%[5]，其是语音识别领域近年来取得的最大的突破性进展。2012 年，Alex Krizhevsky 等利用深度卷积神经网络在图像识别领域取得了惊人的效果，与之前最好的结果相比，将 ImageNet 评测前五选错误率从 25% 降低到 17%[6]。深度学习引起了工业界与学术界的高度关注。美国国防部于 2010 年对深度学习进行了项目资助，包括斯坦福大学、纽约大学等著名高校参与了项目。谷歌在 2011 年启动了“谷歌大脑”项目，该项目利用约 16,000 台电脑集群模拟人类大脑的部分活动，通过大量的图片数据的训练识别动物“猫”，该项目的技术随后被广泛用于谷歌的相关产品上，如谷歌图片搜索、安卓平台语音识别等。随后，Facebook、百度等公司也纷纷成立研究院进行深度学习领域的研究。

随着人们在互联网上的活动日益频繁，计算机需要处理的数据规模也急剧增长。以往的经验表明当数据规模超过一定的量级后，浅层的机器学习模型比

深层的机器学习模型更加适用。一方面，以往的一些复杂的深度模型并没有能力学习到大数据中复杂的本质信息。另一方面，受限于计算设备的性能，一些有效的深度模型并没有被有效训练。图形处理器 (Graphics Processing Unit,GPU) 的应用大幅度提高了计算能力。深度模型在大量任务上的突破性进展有力地证明了深度模型的潜力。当前，研究人员更倾向于从深度模型的角度寻求解决大数据挖掘和处理问题。近年来，取得突破性进展的工作大多是在海量数据上使用深度模型，例如，ImageNet 包括了 1500 万张图片 [6]，战胜围棋世界冠军李世乭的 AlphaGo 在 3000 万个落子数据上学习 [7]，同时通过自我对弈，又极大的增加了模型可见的数据规模。越来越多的实践证明深度模型与大数据结合可以学习得到更为本质的数据信息。

语音、图像和文字是人类交流的主要途径。使用计算机对文本进行处理和理解（自然语言处理，NLP）是人工智能领域的重要分支。经过多年的发展，统计机器学习模型已经成为自然语言处理研究领域的主流。但以往自然语言处理领域的机器学习方法，大多属于浅层模型。使用浅层模型需要靠人工经验从数据中抽取特征，机器学习模型主要负责分类或预测，对数据的表示学习能力较弱。靠人工提取的特征的质量往往决定了系统的性能。因此，研究人员不得不在数据的标注、观察和特征提取上耗费大量的精力。提取有效的特征需要研究人员具有丰富的经验并对数据足够的理解。对不同的任务，又需要重新提取特征。较为常用的浅层模型包括基于支持向量机的分类模型 [8]、基于条件随机场的序列标注模型 [9] 等。虽然浅层机器学习方法在大量应用上取得了一定效果。但在一些较为复杂的问题（例如，自动问答，语义理解）上效果并不理想。通常这些问题需要模型对文本语义进行很好的表示。

人工神经网络特别是深度神经网络在自然语言处理领域并没有得到深入的研究。其中一个重要原因是传统文本表示导致维数灾难、数据稀疏等问题。在使用神经网络时，其输入结点通常非常大，在没有足够训练数据的情况下，深度神经网络难以被有效训练。近些年，深度模型在语音和图像领域上表现出的优异的表示学习能力、计算设备性能的大幅提高、以及人类活动产生的大量文本数据，使得研究人员越来越对深度模型在自然语言处理领域的应用感兴趣。其主要集中在对词、句和篇的表示学习及其相关应用上。虽然深度神经网络模型在语音和图像上取得了突破性进展，但现有深度模型直接运用到自然语言处理任务上并没有取得人们预想的效果。首先，相比于语音和图像，语言是唯一的非自然信号，是人类文明进程中，由大脑产生和处理的符号系统，是人类文明智慧的高度体现。语言的变化性和灵活度远远超过图像和语音。其二，图像

和语音具有明确的数学表示，例如图像通常为数值矩阵，每个点的值表示一定的灰度色彩值。而以往自然语言处理领域过于简单的词袋假设导致文本表示存在维数灾难、高度稀疏以及语义信息损失等严重问题。

近年来，针对文本的特点，大量研究人员转向通过神经网络模型对文本学习表示。Bengio 等使用神经网络模型得到一种名为词嵌入（Word Embedding）或词向量（Word Vector) 的向量表示，其是一种低维、稠密、连续的向量表示，同时包含了词的语义以及语法信息。词向量不仅有效的避免了传统词表示的维数灾难和数据稀疏问题，而且词与词之间可以计算其语义相关性。词向量的学习方法研究成为近几年来的热点，大量的工作涌现 [10–14]。当前，基于神经网络的自然语言处理方法大都基于词向量的输入。当前方法的基本思路是通过无监督方式在大规模的数据上学习词向量，而忽略了人类学习语言的特点以及语言的一些固有属性。在词向量的基础上，研究人员设计深度神经网络模型学习语句的向量表示，具体工作包括递归神经网络（Recursive Neural Network）或递归自动编码 (Recursive Auto-encoder) 的方式、循环神经网络（Recurrent Neural Network，RNN）、卷积神经网络等。语句表示被应用于大量的自然语言处理任务上，并在一些任务上取得了较为突出的效果，例如，机器翻译 [15, 16]、情感分析等 [17, 18]。而从语句的表示到篇章的表示学习仍然非常困难，相关工作也相对较少。

总的来说，虽然深度模型已成为自然语言处理领域的研究热点，但其相对于语音和图像，文本表示远没有被深入研究。虽然在一些较为基础的任务（比如，命名实体识别、词性标注 [10] 等）上的效果超过了传统人工特征方法的性能，但是在一些较为复杂而困难的任务上的性能并不理想。例如，NLP 中的语义匹配、文摘自动生成、自动对话等。这些问题都需要模型能够对文本语义进行深度理解而不是简单处理。需要深度模型能够抽象出数据的本质信息的表示，并根据具体的问题合理的利用这些表示，例如，复述检测需要模型能够捕捉到两个文本间关键而细致的差异；自动文摘生成不仅需要深度模型对文档进行合理表示，还需要从文档的表示中生成简洁的摘要文本。

本人博士期间的研究旨在摆脱传统的浅层模型以及繁杂的特征工程，从深度神经网络的角度对文本表示及其应用进行深入研究。针对具体任务，设计与提出有效的深度模型从大规模的文本数据中自动学习数据的本质信息，从而提高文本表示的质量，在相关任务上取得突出的性能。

从语言学的角度看，词是语言中能够独立存在且表达一个具体语义的最小单位。对词如何表示直接决定了具体任务的模型构建。当前大多数的深度模型

都是以词向量作为输入。虽然当前的词向量学习方法可以学习到富含词的语义以及语法信息的低维向量表示，但当前的词向量学习方法大都是基于词在大规模数据集中的统计信息，并没有充分考虑到词的一些固有属性。例如，词性对大多数语言来说是一种重要的属性，其决定了词的基本语义倾向性，但以往的学习方法没有将词性引入到词向量的学习过程中。人工神经网络理论的提出是受人类大脑神经结构的启发。然而当前的词向量学习模型几乎没有考虑到大脑在学习语言时的特点。因此，本文首先对词向量的学习方法进行深入研究，并利用词向量对自然语言处理领域的相关任务进行提高。

语句是由多个词按照特定语言的语法习惯或规则组合而成，且能够表达较为完整的语义信息的单位。自然语言处理领域存在大量的语句级的任务，例如，问句分类、情感分析以及机器翻译等。对语句进行合理的建模能够改善这些任务的性能。同时，很多任务可以看作是语句匹配问题，例如，复述检测、智能对话等。对语句级的语义匹配合理建模能够将这些任务统一到同一个架构下。然而，以往的匹配模型大多是通过人工特征进行，不能够对语句间的多层匹配关系进行很好的建模。因此，通过构造深度神经网络模型对语句进行建模，并在此基础上，构建深度模型对语句级的语义匹配进行深入研究具有重要的理论与应用价值。基于此，本文从深度模型的角度出发在词向量的基础上对语句表示进行深入研究，并针对相关语义匹配任务，如对话系统、机器翻译等提出基于深度模型的语句匹配架构。

段落通常由一个或多个语句组成，其表达一个较为完整的观点信息。从词、句的表示学习研究到段落的表示学习研究是一种自然的延伸。使计算机能够对段落或篇章进行深入的理解，一直是研究人员孜孜追求的目标，其可以被应用于诸多任务和应用上。特别地，自动文摘生成是一种段落表示应用的典型任务。一个好的文摘生成系统需要对篇章进行深度的理解，并生成流畅、简要、忠实于原文主要信息的摘要文本。对篇章或段落进行合理表示是文摘系统的重要一步。然而，以往的自动文摘生成研究对段落的表示学习并没有突破词袋假设模型。一方面，由于以往的文摘研究缺乏大规模的数据，从而限制了机器学习模型可以发挥的空间。另一面，以往对篇章表示方法通常采用词袋模型，其将文本看作无序的词集合，丢失了大量的信息，同时表示向量存在高维度、高稀疏的缺点。基于此，本文从深度模型的角度出发，针对文摘生成问题，利用互联网自然标注信息构建一个大规模的中文短文本摘要数据集，然后，设计深度模型对段落进行表示，并从段落的表示中解码生成相应的摘要。

1.2 基于神经网络模型的文本表示及应用的研究现状

自然语言处理首先面临的一个问题就是表示问题。文本表示是指将语言的符号文本转换成计算机可以计算的数学形式，通常这种形式是向量。文本表示向量一般包含语言的一些重要特征，例如，语法、语义等。自然语言处理经过长久的发展，研究人员提出了多种文本表示方法。以往的文本表示通常是通过研究人员抽取相关任务文本的若干基础特征，例如，词性、词频信息等。向量的每个维度都对应着文本的某一重要特征。然而，这种方式一方面需要对语言进行深入的研究，其过程需要耗费大量的人力物力。另一方面，通过这种方法提出的特征往往并不能适应多种任务。深度模型的优势在于对数据的表示学习能力，无须人工的过多干预。本节针对近年来基于深度神经网络对词、句和段落进行表示学习的研究进行具体介绍。

1.2.1 词向量表示学习

以往的研究中，对词的表示往往比较简单而粗暴。一般将词表示为一个与词典大小一致的高维向量。这个高维向量的每一个位置对应词典中的特定词。对特定词的表示是将该词对应位置置为 1，其他位置置为 0。这种表示方法存在两个问题，其一，其表示具有非常高的维度和极大的稀疏性，其二，这种表示方法不能表示语言复杂的语义信息，例如，词与词之间的相似度。其高维度和稀疏性导致了自然语言处理中常见的“维数灾难”（Curse of Dimensionality）[19] 问题，例如，将一个维度为 500,000 的向量输入一个深度神经网络模型，即使对于简单的应用其计算代价都会非常高。这种表示同时使语言中比较基本的语义相关性都无法计算，例如，利用这种方法分别对“哈工大”和“哈尔滨工业大学”两个词进行表示，“哈工大”$= (1, 0, 0, ..., 0, 0, 0)$ 和“哈尔滨工业大学”$= (0, 1, 0, ..., 0, 0, 0)$，从语义上看，这两个词表示的是同一个学校，然而，通过常用的余弦相似度计算，这两个词的相似度为 0.

为了克服传统的表示方法的缺点，近些年来，一种新的词表示形式被提出。这种形式被称为词嵌入（Word Embedding）或词向量（Word Vector）。词向量的概念最早是由加拿大蒙特利尔大学的 Yoshua Bengio 等人提出 [20]。所谓词向量，是一种低维的（通常在 50-1000 之间）、连续的实值向量 [21]。词向量不仅可以有效的解决传统词表示的“维数灾难”问题，而且词与词之间的语义关联性可以通过向量距离计算。图 1-1 给出了通过 t-SNE [22] 得到的第二章提出的

图 1-1 通过 t-SNE 得到的词向量空间示意图

Fig. 1-1 The t-SNE visualizations of word embeddings

模型的部分词向量的空间示意图。从图中可以看出，语义或语法相关的词在空间中的位置较为靠近，例如，图中被标记出的一个簇与其它词相距较远，放大后可以看出，其主要是英文中的月份词。根据模型的训练策略，当前词向量学习模型可以大体分为两类：基于熵准则（Entropy Criterion-based）的模型和基于 Pairwise 排序（Pairwise Ranking-based）[10] 策略的模型。以下分别对这两类模型进行介绍。

基于熵准则的词向量学习模型, 由 Bengio 等人提出的四层前馈神经网络模型 [20] 是一种典型的基于熵准则的模型。该模型包含输入层、线性映射层、非线性隐藏层以及 softmax 输出层。该模型本质上是一个基于神经网络的语言模型。其主要思想是通过将当前词的前 n 个词作为模型输入预测当前词。其基本结构如图 1-2所示。

模型的输入是当前词 w_t 的前 n 个词。$w_{t-n+1},...,w_{t-1}$ 是它们在词典中的索引位置。通过线性映射层将 $w_{t-n+1},...,w_{t-1}$ 映射到对应的实值词向量，其操作为 $[C(w_{t-n+1}),...,C(w_{t-n+1})]$。$C$ 是一个 $|V| \times m$ 的实值矩阵。其中 $|V|$ 是词典的大小，m（一般在 50-1000 之间）是对应的词向量的维度。然后通过神经网络的非线性运算以及 softmax 输出层得到当前词的条件概率 $P(w_t|w_{t-n+1},...,w_{t-1})$。这个模型包括两部分参数 (C,θ)，其中 C 即为需要学习得到的词向量，θ 是模型的其他参数。从以上描述我们可以看出，该类模型的最终优化目标可以通过

公式 1-1表示。

$$(\theta, C) \mapsto max \sum_{k=0}^{M} logP(w_k|w_{k-1}, ..., w_{k-n}) \tag{1-1}$$

式中 M———训练语料的大小；

C———词向量;

θ———除词向量外的其他参数

该模型最为突出的问题是时间复杂度高，主要原因包括两方面，其一，模型过于复杂、运算量大。由于考虑了当前词的上文信息的语序信息，$[C(w_{t-n+1}), ..., C(w_{t-n+1})]$ 的维度为 $n \times m$。其二，使用传统的 Softmax 作为输出层会导致最后一层的运算量大，因为最后一层的输出结点为词典大小 $|V|$（通常规模大于 100,000）。为了克服该类模型的缺点，后续的研究做了较多工作。Morin 等人根据 WordNet [23] 中先验语义知识 [13] 对词典中的词构造了一个二叉树，然后使用二叉树中从根结点到每个叶结点的路径表示对应的词，从而对输出层进行分解。虽然该模型将传统的模型效率提升了 200 倍，但是学习到的词向量的质量有所降低。Mnih 等人使用从大规模无标签数据集中得到统计信息构造了词的层次二叉树 [12] 对输出层进行分解，该方法不仅降低了时间复杂度，而且提升了词向量的质量。2013 年 Mikolov 等提出了两种基于三层前馈神经网络的学习模型 CBOW 和 Skip-gram [11]。这两种模型相比于 2003 年 Bengio 提出的模型，做了大量简化，去除了非线性隐藏层、忽略了局部窗口的词序。同时，通过使用层次 Softmax 分类器作为输出层，降低模型的时间复杂度。这两种模型相比于以前的工作，不仅模型学习效率高，而且学习得到的词向量质量较高。第 2 章将会对 CBOW 模型进行详细介绍，作为本文提出的基于动名分离的词向量学习模型的对比模型。

基于 Pairwise 排序策略的模型，基于 Pairwise 排序策略的词向量学习模型是由 Collobert 和 Weston 提出的 [10]。该类模型的主要特点是通过判断当前文本片段是不是自然的语言片段作为优化目标学习词向量。其优化目标的数学形式可以描述为公式1-2。

$$(\theta, E) \mapsto min \sum_{k=0}^{M} \sum_{w'_k \in V} \{\alpha - s(w_{k-n}, ..., w_k, ..., w_{k+n}) + s(w_{k-n}, ..., w'_k, ..., w_{k+n})\} \tag{1-2}$$

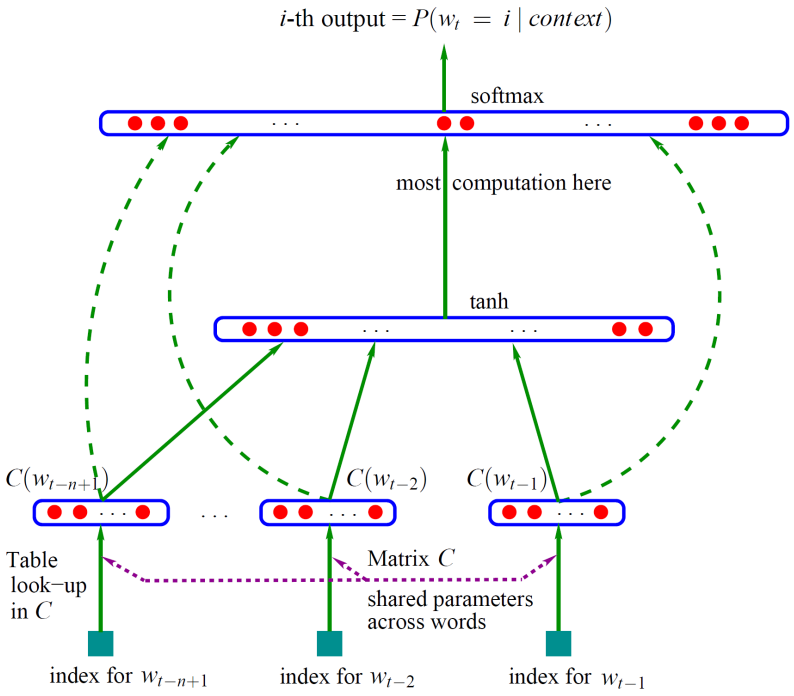


图 1-2 由 Bengio 等提出的典型的基于熵准则的词向量学习模型示意图 [20]

Fig. 1-2 The Classical entropy criterion-based Model proposed by Bengio et.al [20]

式中 $s(w_{k-n},...,w_k,...,w_{k+n})$———对自然片段 $(w_{k-n},...,w_k,...,w_{k+n})$ 的打分；

w'_k———从词典中随机选取的一个词;

α———为了使自然片段的得分比随机替换掉其中的词 w_k 后得到的非自然片段的得分大的最小差值；

根据该类模型的训练策略，模型一次迭代既要计算自然片段的得分，又要计算一次其对应的非自然片段的得分。虽然该类模型避免了基于熵准则模型的输出结点过多的问题。但是其监督信号相对较弱，对于一个自然片段，通常需要随机产生上百个非自然片段。其训练时间也非常漫长，例如，Collobert 和 Weston 在路透社和维基百科数据集上，对他们的模型训练了超过七周才得到相对理想的词向量。在 Collobert 和 Weston 模型的基础上，Huang 等人利用无标签数据集中的全局统计信息（也即每个词的 tf·idf 信息），对每个词学习得到了多尺度的词向量 [14]。该模型学习到的词向量在 WordSim353 [24] 词相似度数据上的性能超过了其他对比词向量的性能，但是他们的词向量在自然语言处理的组块分析和命名实体识别任务上的表现较差，其结果可以参见本文的第 2 章的实验部分。

1.2.2 语句表示学习

所谓语句表示学习，是指通过机器学习模型将语句的语义以及语法信息编码到一个数学表示形式（通常是向量）的过程。近年来，使用神经网络在词向量的基础上对语句进行表示的研究成为热点。目前较为流行的语句表示模型主要包括三种类型，递归神经网络、深度卷积神经网络、以及循环神经网络。

递归神经网络 考虑到语句具有句法层次和递归结构，Socher 等提出递归神经网络的方式对短语或语句进行表示学习 [17, 25, 26]。递归神经网络的输入层包含两部分，分别是左子节点的向量表示和右子节点的向量表示。两个子节点的向量表示通过递归神经网络后生成父节点的向量表示，同时生成一个打分判断父节点表示的可信度。父节点的向量又可以与其它子节点通过组合得到更大的父节点。依次递归，直至生成整个语句的根节点表示。叶节点是语句中的词，词的表示是通过在大规模的数据上训练得到的词向量，词向量通过递归神经网络进行更新。使用递归神经网络可以得到语句中不同短语以及整个句子的表示，可以显式的得到整个语句的递归路径。Socher 等使用递归神经网络显著的提高了句法分析的性能 [25]。其后，Socher 等对递归神经网络架构进行改进，提出了递归神经张量网络，并在情感分类任务上取得了突出的性能 [17]。将自动编码与递归神经网络结合便演化出了递归自动编码模型。标准的递归自动编码模型是在给定句法树的基础上进行的。Socher 等提出贪心自动编码算法，该算法无须预先给定句法树 [27]。其基本思想是给定序列 (x_1, x_2, x_3, x_4)，对所有的相邻词 $(x_1, x_2), (x_2, x_3), (x_3, x_4)$ 进行组合得到其表示 y_1, y_2, y_3，并记录下其重构误差，选择重构误差最低的组合进行编码，这里假设选择 $(x_2, x_3) -> y_2$，使用 y_2 替换序列中的 (x_2, x_3) 得到 (x_1, y_2, x_4)，然后在新的序列上继续上述过程，最终得到根节点的语句表示。在无监督递归自动编码模型的基础上，Socher 等针对情感分析任务结合语句中词、短语以及语句的情感分析类别，提出了以情感类标为部分监督信号的半监督递归自动编码模型，预测语句级的情感类别分布 [27]。标准的递归自动编码模型在重构输入的过程中，仅仅是重构其直接孩子节点，Socher 等提出在重构阶段重构当前节点的所有子树节点，从而对语句中的短语以及整个语句进行表示，其将得到的语句和短语表示应用到复述检测任务上，取得了优异的性能 [28]。通过对递归自动编码模型进行改进，近年来其他研究者也提出了不少变形并应用于其它任务上，例如，多文档文摘的语句排序 [29]、统计机器翻译 [30] 等。

卷积神经网络 卷积神经网络是另一种较为流行的语句建模架构。卷积神

经网络早期主要应用于图像识别[31]。卷积神经网络中的卷积操作可以有效的获得数据中不同层次的表示，而最大池化操作能够有效的提取数据中的重要特征。近年来，将卷积神经网络应用到自然语言处理任务上涌现出了大量工作。Ronna Collebert 等通过将语句中所有词的词向量顺序排列，并使用滑动窗口对窗口中的词进行一层卷积操作，然后使用全局最大池化的方式得到固定维度的向量表示语句，然后使用 Softmax 分类器对语句进行分类，从而在语义角色标注任务上取得了很好的效果[10]，Kim 等通过类似的卷积神经网络在多种语句分类任务上进行了实验，得到了较为理想的效果[32]。Zeng 等人利用该模型对语句中特定的两个名词间的关系进行分类，配合人工提取的词级别特征显著地超过了以往的方法[33]。然而，这些模型的层次较浅，只有一层卷积操作和一层池化操作，对于复杂的任务不能有效的获得丰富的语义信息，同时全局最大池化操作导致了词序信息的丢失。通过该模型完成分类任务大多还需配合人工提取的其他特征[10, 33]。Nal 等人通过多层卷积操作以及巧妙设计动态窗口大小的最大池化层对语句进行表示，其在语句情感和问题分类等任务上取得了较好的效果[34]。该方法不需要依赖句法树，可以较好的应用于其他语言。由于该模型对最大池化操作做了巧妙地设计，也带来其不宜扩展到除了分类任务以外的其他类型的任务的问题，例如，语义匹配问题。除了以上具有代表性的工作。近年来的基于卷积神经网络语句建模的工作还被大量用于复述检测[35]、文本分类[36–38]、机器翻译[39, 40]等任务。本章提出的基于深度卷积神经网络的语句表示模型，不需要依赖任何句法树，通过多层交叠的卷积和局部池化操作对语句进行建模，相比于 Nal 提出的模型，该模型更为通用。另外，也更适合于我们第 3 章所研究的语句级匹配问题。

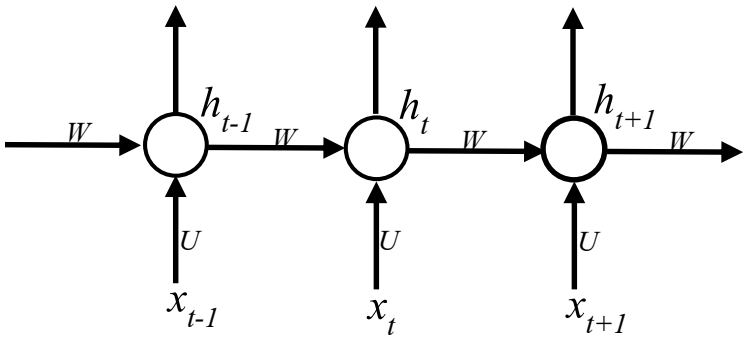


图 1-3 循环神经网络的基本架构图

Fig. 1-3 The overall Architecture of Recurrent Neural Network

循环神经网络 在卷积神经网络模型中，每层的运算单元的输出只传向下一层不同的运算单元，语句中的各个词的运算是独立的，这种网络结构称之为

前馈神经网络。循环神经网络与前馈神经网络不同，每一个神经元的输出不仅与该神经元的输入有关，还与其相邻的前一个神经元的输出有关。如一个语句$(x_0, x_1, ..., x_T)$，通过典型的循环神经网络的架构对其进行建模的过程如图 1-3 所示。给定输入 x_t 得到 t 时刻的输出状态 h_t 的公式见 1-3，其中 h_{-1} 一般初始化为全 0 向量，$\phi(\cdot)$ 为非线性函数如 sigmoid。这里 W, U, b 为循环计算单元的参数，对于每个时刻 t，它们都是相同的。因此，理论上时刻 T 的状态 h_T 包含了整个句子的信息，可以将 h_T 看作是语句表示向量。Sutskever 等人通过基于 LSTM 运算单元的多层循环神经网络，将翻译问题中的源语言语句表示成固定长度的向量，然后使用另一个循环神经网络从该向量中解码生成目标语言，取得了突出的效果 [16]。类似地，Cho 等人通过一个 RNN 编码源端短语，另一个 RNN 解码得到目标端短语，通过训练得到短语的表示向量。其分析表明得到的表示向量保持了短语的语义和句法结构 [41]。当前，大多数语句建模架构都是通过有监督任务对模型进行优化，因此其学习得到的语句向量是具体任务依赖的。Kiros 等人在提出了一种通过无监督学习方法得到通用语句向量的方法。其通过一个循环神经网络对篇章中的语句进行编码得到其向量表示，然后再分别通过两个不同的循环神经网络，从编码端得到的语句向量中生成当前语句的前一个语句和后一个语句。通过大规模的语料训练，将给定语句输入给编码循环神经网络，输出该语句的通用向量表示，他们称该模型为 Skip-thoughts，而得到的句子向量为 Skip-thoughts 向量 [42]。

$$h_t = \phi(W \cdot h_{t-1} + Ux_t + b) \tag{1-3}$$

1.2.3 段落表示学习

对段落进行表示是自然语言处理领域重要的问题，其可以被应用于多种任务，例如，摘要生成、主题挖掘、文本分类等。相比词和句子，对段落的表示学习更为困难。一方面，由于段落相对较长，通常包含多个语句，当前的深度模型研究对于长序列的数据建模仍然存在较大的困难，另一方面，段落包含了丰富而复杂的语义信息，人类丰富的语言技巧以及书写技巧导致段落中存在大跨度的信息依赖，例如，段落的开头有可能是段落结尾要表达的核心思想的铺垫，另外语句间也存在复杂的关系，例如，转折、强调等。当前对段落的表示学习的研究仅处于探索阶段。

Quoc 和 Mikolov 等对段落随机初始化向量，然后将随机初始化的段落向量与段落中连续的几个词连接，通过 Softmax 分类器预测下一个词，更新段落向

量和词向量，最终得到段落的向量表示[43]。该方法完全是无监督的学习算法，但模型相对较为浅层，其过程是通过段落向量不断的预测下一个词将该词的信息编码到段落向量中，因此无法对整个段落充分理解后，得到段落的全局信息，另一方面，对新的段落得到其向量仍然需要执行训练的过程[42]。Quoc 和 Mikolov 等在论文中报告得到的段落向量在相关文本分类任务上得到了突出效果，然而其性能仍然存在较大的争议性[1]。Li 等则通过循环神经网络对段落中的语句进行编码，然后通过循环神经网络解码得到输入的段落[44]，通过这种方法训练编码端的循环神经网络，对于新的段落可以利用编码端循环神经网络得到其向量表示。其验证了三种模型，a) 标准的序列到序列的模型，通过一层 LSTM 对段落编码，将编码 LSTM 的最后一个状态输入给另一个 LSTM 进行解码；b) 层次序列到序列模型，使用 LSTM 对语句进行编码，并将最后一个状态作为语句向量，在语句向量的基础上，再使用另一个 LSTM 对段落进行编码，解码过程中，也同样使用层次的 LSTM 架构；c) 带有 attention 机制的层次序列到序列模型，该模型是在 b) 的基础上增加了 attention 机制，对编码端的第二层 LSTM 的状态进行动态选择。其通过在 50 万的段落数据集上的训练结果显示其提出的模型可以保持句法、语义以及论述的连贯性。本文第 5 章对短文本摘要生成的研究，也使用了基于循环神经网络的模型，与之不同的是我们并没有使用复杂的层次循环神经网络，而是使用单层的循环神经网络进行编码和解码。Tang 等则通过使用卷积神经网络或 LSTM 对语句进行建模，然后在语句向量表示上，使用循环神经网络得到文档的表示，在有监督的文档级情感分类任务上验证了模型的性能[45]。Denil 等则首先通过卷积神网络得到语句的表示向量，继续通过卷积神经网络在语句表示的基础上得到文档向量表示，其在有监督的 IMDB 情感分类等任务上验证了模型的性能[46]。

1.3 论文的主要研究内容及创新点

1.3.1 研究内容概述

通过上文的分析可以看出，以往自然语言处理领域的研究将大量精力集中在特征工程上。虽然深度模型已经在图像、语音领域取得较大进展，但针对自然语言处理领域的具体问题的研究仍非常不完善，特别是对文本的表示以及在相关问题上的应用。文本组成的基本单位包括词、句、段（篇），它们分别代表

[1]https://groups.google.com/forum/#!msg/word2vec-toolkit/Q49FIrNOQRo/ToAU2sYrPjYJ

了文本的不同粒度的形式，它们的表示之间具有层次递进的关系，也即语句由词组成，段由语句组成。一方面，通过深度神经网络分别对词、句、段进行表示学习的方法和方式存在较大的差异，它们可以被应用于不同类型的任务上。另一方面，它们之间又具有一定的关系。具体而言，词的表示学习是基础，不仅语句和段落的表示学习都会用到词的表示，同时词的表示也可以被集成到具体的任务（例如，命名实体识别、问答系统等）中提高其性能。对语句的表示学习通常是在词表示的基础上设计深度模型解决语句级的相关任务（例如，问题分类、句子情感分类等）。而对段落的表示学习既可以以语句表示为基础，也可以完全从词的表示出发，设计深度模型学习段落的表示，并将其用于篇章级的任务上，如摘要生成、文档理解等。本文以词、句、段的顺序，分别对它们的表示学习进行深入研究，并将其应用到自然语言处理领域具体问题上。

为了便了解本文的结构，图 1-4 描述了本文研究的基本框架，以及各章间的关系。在文本表示研究的内容上，本文的研究按照词、句、段的表示学习递进展开。具体地，第 2 章针对文本领域中的词的表示进行研究。第 3 章、第 4 章则在词向量的基础上，对语句表示学习以及语句匹配进行深入研究，其中，第 3 章将语句表示学习应用于同一种语言的语句匹配任务上，而第 4 章针对翻译问题，将语句表示学习应用于双语匹配任务上。第 5 章则对文本中段的表示学习及其应用进行研究。在模型方法上，各章提出的模型复杂度以及深度也递进增加，例如，第 2 章提出的词向量学习方法的模型具有三层结构，其在一个局部窗口上进行运算，第 3 章和第 4 章提出的模型和方法的深度通常在 5 层以上，其运算在整个语句或短语上进行，而第 5 章提出的模型需要针对长度在 100 个字左右的序列上循环进行，其需要对包含数句话的序列进行建模。每章解决的应用任务的难度存在较大差异。第 2 章由于是对词进行表示学习，其可以被集成到传统的任务上解决命名实体识别、组块分析等基础性任务。第 3 章和第 4 章的方法则可以独立作为一种新的方法应用于相对高级而复杂的语义任务上，例如，对话匹配、复述检测、双语短语对的选择等问题。第 2 、3 、 4 章解决的问仍属于分类问题，而第 5 章研究的问题更为困难，其不仅需要对较长的文本进行建模，还需要解决语言生成问题，其被公认为是自然语言处理领域最为困难的任务之一。以下对每章的研究内容进行具体描述：

首先，本文针对自然语言的基本独立语义单位词的表示进行研究。一方面，语言学的研究发现，语法特别是词性对语言的理解具有重要作用；另一方面，神经心理学发现人类大脑对动词和名词的处理区域不同。基于此，我们将词性和动名分离引入到词向量的学习过程中，提出了基于动名分离的词向量

学习模型。该模型是一个具有三层结构的前馈神经网络。在模型的输入端保持了语言的词序特征。同时，该模型将词性工具自动标注的词性信息整合到词向量的学习过程中，实现模型对动词和名词的分离。受益于以上两种改进，1）该模型的时间复杂度相对较低，与目前效率较高的词向量学习模型 CBOW 和 Skip-gram 相当，比其它模型更为高效。2）该模型学习到的词向量质量高，实例分析表明通过该模型学习得到的词向量得到的特定词的相近词，比其他它向量得到的结果更为合理。3）将该模型学习得到的词向量加入到基于条件随机场的命名实体识别和组块分析序列标注任务上，其性能提高的幅度显著地高于对比的其它词向量。

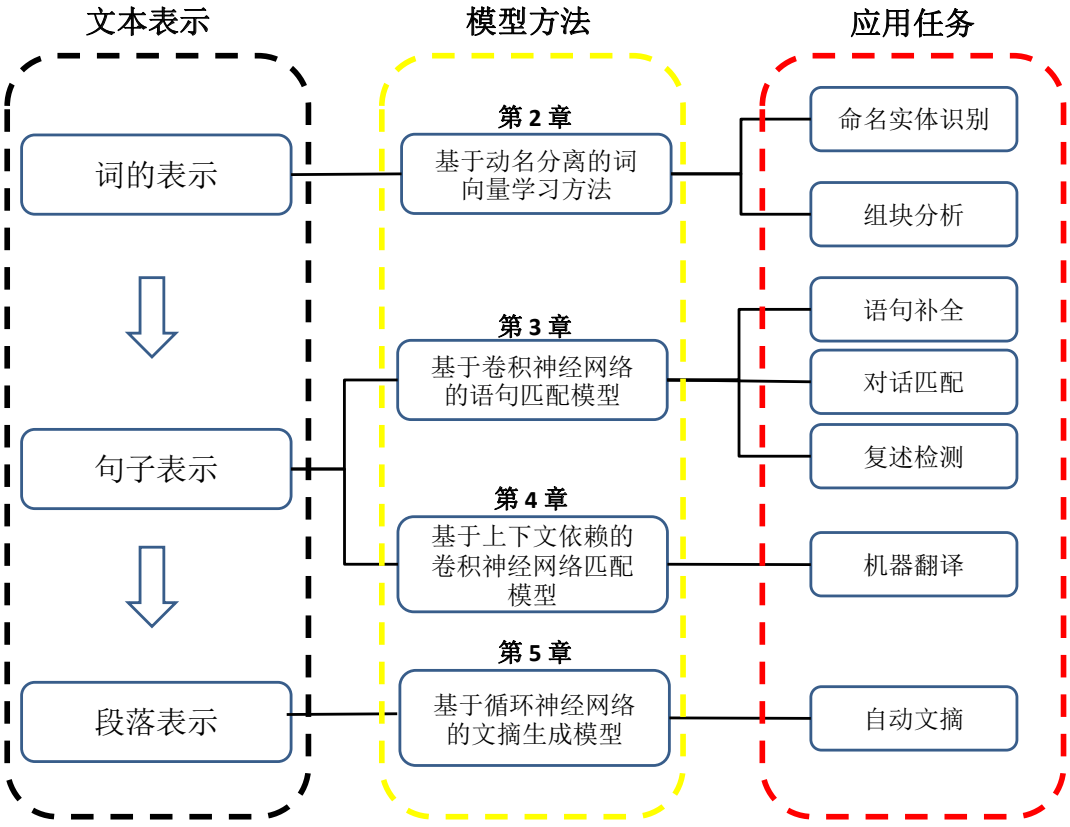


图 1-4 论文主要研究内容

Fig.1-4 The main research problems of the thesis

其次，本文研究了基于深度神经网络的语句表示方法。提出了一种基于深度卷积神经网络的语句表示模型，该模型不需要依赖句法分析树，可以通过深度卷积神经网络的卷积以及局部最大池化操作，根据具体任务的监督信号学习得到语句的表示向量。鉴于自然语言中存在大量的语句级匹配问题，我们从深度模型的角度出发，对语句匹配问题进行了深入研究，提出了两种基于深度卷

积神经网络的语句匹配架构。架构一分别通过深度卷积神经网络对两个语句进行建模，然后对两个语句的表示进行匹配；架构二则直接对两个语句的匹配表示，通过深度卷积神经网络进行建模。架构二可以捕捉到语句间更为细致的匹配关系。我们将两种匹配架构应用到三种匹配任务上，这些任务涵盖了不同语言、不同类型的匹配问题。实验表明，两种架构不仅具有较强的通用性，同时较大幅度的超过了其他对比的方法。特别地，相比于架构一，架构二在三种任务上取得了较优的性能。

第三，本文对统计机器翻译中双语短语对的选择进行了研究。统计机器翻译解码阶段非常重要的一步是对翻译短语对进行选择，而传统的方法忽略了源端短语的上下文信息。本文在第 3 章提出的深度卷积神经网络语句匹配架构一的基础上，提出了上下文依赖的卷积神经网络短语匹配模型，将源端短语的上下文以一种较为自然的方式引入到了深度匹配架构中。提出了通过上下文依赖的双语词向量对模型进行初始化，以捕捉到两种语言间词级别的匹配关系，考虑到短语间存在不同层次的语义匹配，对训练样例显式的分为三个不同难度等级：“容易”、“中等”和“困难”。提出了“课程式”学习的训练算法，该算法按照从易到难、循序渐进的方式训练模型。最终实验结果表明，该模型可以有效的利用源端短语的上下文选取对应的目标端短语，同时在一个较强的基准系统上，显著地提高了统计机器翻译系统的性能（BLEU 值提升了 1.0 个 BLEU）。

第四，本文对自动文摘生成进行了深入的研究。自动文摘生成是段落表示学习最具代表性的任务之一。自动文摘生成是自然语言处理领域极具挑战性的问题。不仅需要对原文进行合理的表示，同时需要从表示中生成符合原文核心主题的摘要文本。由于缺少大规模的生成式文摘数据集，文摘自动生成的研究一直进展缓慢。本文首先利用互联网社交平台上用户产生的大规模的带有自然标注的数据，构建了一个大规模的中文短文本摘要数据集。该数据集包含 240 多万的中文短文本摘要数据。为了检测数据的质量，人工标注了随机采样的大约 10,000 个样本，标注结果表明该数据集具有较高的质量。数据集中短文本长度大多集中在 100 个汉字左右，并包含多个语句，描述了一个较为完整的事件或主题，其可以看做是一个段落级的摘要数据集。为了便于其他研究的需要，本文人工构造了一个标准的测试集。提出使用基于循环神经网络的编码 -解码架构自动生成短文本摘要，并引入了两种模型。模型一通过一个循环神经网络对短文本进行建模，并将其最后一个状态看做短文本的表示，使用另一个循环神经网络解码生成摘要。模型二则在模型一的基础上，在解码阶段动态的从编码阶段的所有时刻的状态中生成解码当前词的“上下文”表示。在测试集上的

实验结果表明模型一和二对短文本的表示保持了短文本中的重要语义以及语法信息，模型一和二无须任何人工特征，是一种端到端的机器学习系统。通过大规模的数据训练，模型一和二可以有效的生成摘要文本。特别地，模型二生成的文摘具有较高质量，并在 Rouge 评价体系下取得了最高的性能。

1.3.2 论文创新点

本文的主要贡献和创新点归纳如下：

（1）针对词的表示学习，提出了一种基于动名分离的词向量学习方法。该模型以较低的时间复杂度，学习得到高质量的词向量。将其应用到命名实体识别和组块分析任务上，显著地优于所对比的词向量方法。

（2）针对语句表示学习，提出了一种基于深度卷积神经网络的语句表示模型。该模型不依赖于句法分析树可以根据具体任务学习到有针对性的语句向量。针对语句级匹配问题，提出了两种基于深度卷积神经网络的语句匹配架构。在不同语言、不同类型的匹配任务上，两种匹配架构显著地超过了其他对比方法。

（3）针对统计机器翻译系统中的目标短语选择问题，提出了一种上下文依赖的卷积神经网络短语匹配模型。针对模型的词向量初始化，提出了基于上下文依赖的双语词向量学习方法。针对模型的训练，提出了一种“课程式”训练算法，该算法可以从易到难、循序渐进的对模型进行训练。在一个较强的统计机器翻译系统上，显著地提高了系统性能，整体提高了 1.0 个 BLEU。

（4）针对文摘自动生成，构建了一个大规模的中文短文本摘要数据集。该数据已授权来自清华大学、香港大学、台湾中央研究院、卡内基梅隆大学等众多研究机构使用。提出使用基于循环神经网络的编码 -解码架构从大规模数据集中学习生成摘要，并构建了两种文摘生成模型。实验表明构建的两种架构能够对短文本进行合理表示，并从中解码生成具有较高质量的摘要。

1.4 论文组织结构

本文按照对文本的词，句，段的表示研究以及应用的顺序安排，其余章节的具体安排如下：

第 2 章： 本章主要对词的表示学习进行研究，主要介绍了提出的一种基于动名分离的词向量学习模型，该模型有效地利用了通过词性标注工具得到的词性信息，并保持了语言的词序信息。对模型得到的词向量的质量通过实例进行了分析，通过将词向量加入到基于条件随机场的命名实体识别和组块分析序

列标注任务上，验证了模型的性能。

第 3 章： 本章主要介绍了对语句的表示学习以及语句级匹配问题的研究，详细介绍了提出的基于深度卷积神经网络语句表示模型，以及基于深度卷积神经网络的语句匹配架构。并对匹配架构一和二的性能以及关系进行了深入分析。最后，介绍了将匹配架构应用于三种不同的语义匹配任务上。

第 4 章： 本章主要对统计机器翻译系统中，源短语与目标端短语的匹配选择的研究进行了介绍。首先详细描述了提出的上下文依赖的卷积神经网络短语匹配模型，随后对模型的训练以及初始化进行了介绍，针对该模型的训练提出了一种“课程式”的训练算法，从易到难的训练模型。最后，介绍了将上下文依赖的卷积神经网络短语匹配模型对源端短语与目标端短语的打分，融入到统计机器翻译系统中的实验以及分析。

第 5 章： 本章主要对文摘自动生成方法进行了深入研究，首先详细介绍了为了解决文摘自动生成领域数据集不足的问题，利用自然标注信息从互联网上构建的一个大规模的短文本中文摘要数据集。并对数据集的过滤，标注以及数据集的统计信息进行了详细的描述。详细描述了构建的两种基于循环神经网络的文摘自动生成方法。并在本章构造的大规模中文短文本摘要数据集上，验证了提出的模型的性能。

第 2 章 基于动名分离的词向量学习方法

2.1 引言

当前，大多数词向量学习模型是在大规模的无标签文本数据上进行学习，人类对语言的一些先验知识很少被有效利用。近些年来，一些研究开始使用一些人工构建的语义知识[13] 或者全局统计特征 [14] 指导词向量的学习过程。尽管语法知识对于语言理解非常重要，然而，在词向量学习的过程中，目前还没有工作将语法知识引入到词向量的学习过程中。特别地，词性是词的基础属性[47]，对于中文或英文来说，根据词的上下文，可以非常明确的确定一个词是动词还是名词以及其他。动词和名词对于一种语言来言占有极其重要的位置。动词和名词不仅在词性和用法上截然不同，他们在语义层面也具有极大区别，动词通常表示一种动作实施，名词通常表示一种实体[48]。在语言中，经常会出现一个词在不同的上下文中被用作动词或者名词的情形。例如，英文单词“love”在不同的语境中既可以被用作动词“love/VB”，意义为“爱，喜欢”，也可以被用作名词“love/NN”, 意义为“爱情，恋爱”。语言学家的研究发现，词性对于语言理解占据非常重要的地位。低年级的学生会以不同的方法和方式学习“love/VB”和“love/NN”，因为，它们的意义和用法完全不同[49]。然而，现有的词向量学习方法基本都忽视了语言的这一重要基础属性。

另一方面，神经语言学通过病例观察以及脑部生理解剖实验发现，人类在学习语言的过程中，动词和名词激发的是大脑的不同的区域。具体来说，当人类脑区的左颞叶侧部和下部受到不同程度的损伤时，对于名词的识别和认识能力也相应会受到不同程度的下降，当人类脑区左颞叶的前上部和左额叶的后下部受到不同程度的损伤时，对于动词的识别和认识能力也相应会受到不同程度的下降。最早发现这类现象的学者是来自意大利的哲学家 Giovanni Battista Vico(1688-1744) [50]。这种现象被称为动名分离现象（DNV, Dissociation between Nouns and Verbs)。经过几百年的发展，语言学家和神经解剖学家一直在试图解释为什么在人类身上会存在动名分离现象，但是一直没有找到一个统一的解释。比较有代表性的解释分为三类：其一，语法解释，动词和名词的分离现象反映了它们在语法和句法复杂度上的巨大差异；其二，语义概念解释，动词和名词表达的意义分别是动作性的和实体性的语义，而这两种不同的语义信息存储在大脑的不同部位；其三，词汇学解释，动词和名词被存储在大脑的不同部

位，与它们的语义内容是无关的[48]。尽管如此，人类在学习与使用语言时，动名分离现象是确定存在的[50–52]。受人类大脑的动名分离结构的启发，我们将动名分离引入到词向量学习过程中，从而提出基于动名分离的词向量学习模型（Continuous Dissociation Between Nouns and Verbs Model, CDNV）。该模型通过将分词系统自动生成的词性信息融合到一个三层的神经网络模型中指导词向量的学习。同时，该模型保持了语言的局部词序信息。该模型可以以较低的时间复杂度学习到高质量的词向量。实验表明：1）该模型在 10 亿词规模的语料上训练消耗大约 1.5 个小时，其时间复杂度与目前主流的模型 CBOW 和 Skip-gram 模型相当；2）通过 CDNV 模型学习得到的词向量，得出的一些具有代表性的常见词的相关词汇比其他词向量更为合理；3）CDNV 的词向量在以条件随机场为基础的组块分析和命名实体识别序列标注任务上的性能，显著地超过了对比模型的性能。

本章的组织结构如下：章节 2.2、详细介绍了提出的基于动名分离的词向量学习方法；章节 2.3、详细介绍了基于动名分离的词向量学习模型的训练方法；章节 2.4、对 CDNV 模型的时间复杂度进行了分析；章节 2.5、介绍了实验设置、评价指标、以及通过 CDNV 模型学习得到的词向量，得到的一些具有代表性的常用词上的质量，并与其他词向量作了对比分析。然后介绍了在基于条件随机场的组块分析（Chunking）和命名实体识别（Named Entity Recognition）任务上的实验对比，最后给出了本章的结论。

2.2 基于动名分离的词向量学习模型

本章提出的基于动名分离的词向量学习模型是以 Mikolov 等提出的连续词袋模型（Continuous Bag-of-Words Model，CBOW）为基础的。为了便于理解 CDNV 模型的特性以及与 CBOW 模型的不同。下面我们首先简要介绍 CBOW 模型的网络结构，然后详细介绍 CDNV 在 CBOW 的基础上做的两个重要改进：1）通过将 CBOW 模型中“向量加”操作替换为“向量连”操作，在输入层保持了词序特征；2）CDNV 使用动名分离特性指导二叉树的构建，根据动名分离的特性，一个词最多有三个不同的编码分别是名词编码，动词编码或其他词性编码。根据该词在语境中的词性，选择一个编码作为输出，指导模型参数以及其上下文中词的词向量的更新。需要特别强调的是，针对一个词，CDNV 只学得到一个向量，不同的词性仅被用作指导顶层对应的神经连接。

2.2.1 连续词袋模型

连续词袋模型（CBOW）是一个三层的前馈神经网络模型，该模型是2013年由Mikolov等人提出[11]的。图2-1给出了该模型的网络结构图，图中“*PADDING*”表示句子的结束标示，d表示词向量的维度，“BP”表示后向传播算法，$C(*)$表示单词$*$的输出结点。模型分为三个部分，输入层（Input Layer）、映射层（Projection Layer）和输出层（Output Layer）。

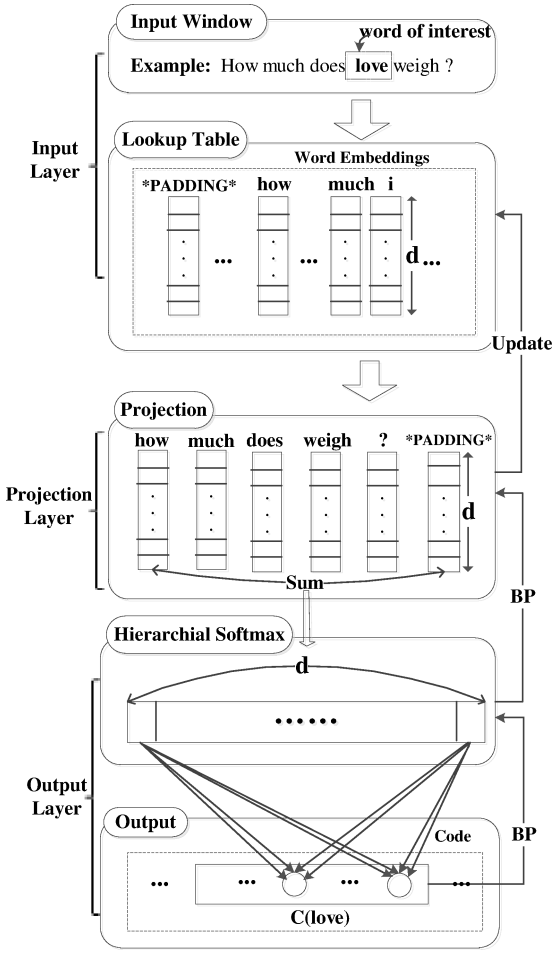


图 2-1 CBOW 模型的网络结构

Fig.2-1 The architecture of the CBOW

以图2-1中句子为例，假设单词“love”是当前要预测的词。首先需要获取“love”的上下文，假设我们选的模型训练的上下文窗口为[-3,3]，它的上文包括它前面三个词“How much does”，它的下文包括它后面二个词“weigh ?”。由于“love”在句子中的位置使得其下文只有两个单词，然而由于前馈神经网络要求模型的输入维度为统一大小的向量，因此，对于其上文或下文长度不足窗

口大小的情况，选择使用占位符“*PADDING*”来填充，直到待预测词的上文和下文的个数与窗口大小一致为止。输入层的参数为所有词的词向量的矩阵，该矩阵在模型训练之前，被随机初始化为 $|V| \times d$ 的浮点数矩阵。其中，$|V|$ 表示词典的大小，d 为词向量的维度。

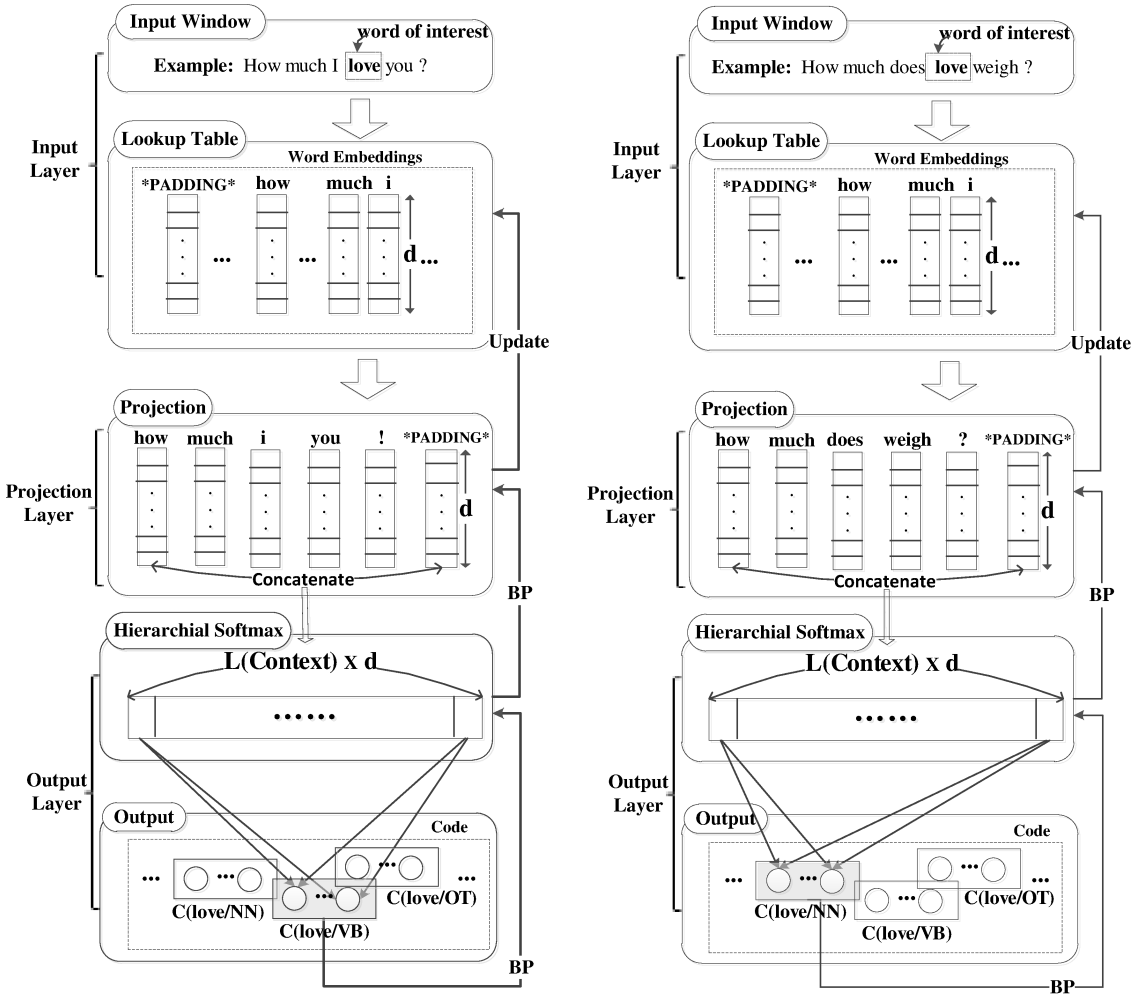
映射层首先需要根据待预测词的上下文查找词向量矩阵，取出对应的上下文中的词的词向量。根据词袋模型的假设，CBOW 忽略了待预测词的上下文中的词序，因此，映射层将待预测词的上下文中词的词向量通过“向量加”操作得到一个向量，并将该向量输入给输出层。

输出层的作用是通过待预测词的上下文预测该词，传统的 Softmax 分类器的目标输出结点数为词典的大小，因此输出层的参数矩阵为 $d \times |V|$，从而导致了输出层的矩阵运算量非常大。为了加速模型的运算，Mikolov 等采取了层次 Softmax 分类器。Mikolov 等人通过观察发现，词的频率对于基于神经网络的语言模型的词的类别划分具有重要作用[53]。较为合适的做法是选择使用哈夫曼编码表示词。具体做法是首先根据词典中所有词的频数统计，对词典中的词构建一个哈夫曼树，词典中的词位于对应的哈夫曼树的叶结点，词被表示为一个从该词到根节点长度的 0/1 哈夫曼编码。根据哈夫曼编码的特点，频率高的词其编码长度较短，频率低的词的编码长度较长。在输出层，对每个待预测的词，只有该词编码的相应结点被激活。这样，每次迭代在理想情况下如果得到的哈夫曼树为平衡二叉树，那么一个输入窗口对应的激活的输出层的结点个数就可以从 $|V|$ 降低到 $log_2(|V|)$。通过预测该词，CBOW 将误差传递到输入层，对输入层的词向量矩阵中对应的上下文中词的向量进行更新，通过在大规模的训练语料上进行训练，得到词典中所有词的词向量。由于 CBOW 在输出层使用了基于频率的哈夫曼编码，去除了隐藏层的运算，忽略了输入窗口中的词序，因此 CBOW 模型非常高效。

2.2.2 保持上下文的词序

CBOW 模型的高效性部分得益于它的输入层没有考虑局部上下文的词序信息。然而，局部上下文的词序特征对于区分窗口中心词的意义具有重要作用，大量的自然语言处理任务证明词序特征的重要性。传统任务中，考虑词序遇到的主要困难是词序可以明显增加模型的时间复杂度，另外由于缺少大规模的训练数据，词序特征可能引起数据稀疏。一方面，由于 CBOW 模型不仅去除了非线性隐藏层的运算，同时在输出层使用了层次 Softmax，使得该模型的时

间复杂度较低。另一方面，词向量的训练数据集来自于自然的文本数据，不需要额外的人工标注，而大规模的文本数据非常容易获得。因此，在保持模型的时间复杂度不至于太高的前提下，在大规模的无标注文本数据上将词序信息引入模型成为可能。



(a) Example of "love/VB" (b) Example of "love/NN"

图 2-2 基于动名分离的词向量学习模型的示例图

Fig. 2-2 CDNV with an example of one word in different context

本文将局部词序特征引入到词向量学习模型。图2-2描述了保持局部词序特征的基于动名分离的词向量学习模型的网络结构。如图中所示，CDNV 将 CBOW 模型映射层中的"向量加"操作替换为了"向量连"操作。严格地，给定单词 w_i 的上下文 $(w_{i-k},...,w_{i+k})$，定义 $e^{(j)}$ 表示单词 w_j 的词向量，通过映射层得到的输出为 h_i 可以通过公式2-1给出。这里局部窗口的大小为 $2k$，h_i 的维度为 $2kd$。

$$h_i = (e^{(i-k)}, ..., e^{(i+k)}) \tag{2-1}$$

2.2.3 引入动名分离特性

受人类学习语言过程中的动名分离现象的启发，CDNV 的输出层与 CBOW 也存在重要差异。根据一个词在不同语境下的词性，对一个词存在多个编码。为了得到每个词在大规模文本数据集中的所有词性，以及每个词的各种词性在大规模文本中出现的频数，传统的分词工具是一个适当的选择。虽然，当前分词工具在非规范文本上具有较大的错误率，但通常词向量的训练数据来源于维基百科和路透社新闻语料，因此文本的规范性较高，传统的词性标注工具在这些数据集上的准确率较高。由于动词和名词在语言中的重要地位，以及受神经语言学发现的启发，我们这里主要考虑动词和名词的分离情况。传统词性标注的工具对词进行标注时，给出的类别非常丰富，包括动词、名词、介词、形容词、副词等，而且对每种词性还有较为细致的分类，例如，动词性词可以分为“VB”、“VBP”、“VBZ”、“VBD”、“VBN”和“VBG”，这些是动词不同语境下表现出的语法变形，名词词性可分为，“NN”、“NNP”、“NNS”和“NNPS”，其中“NN”和“NNP”分别表示普通名词和专有名词，而“NNS”和“NNPS”分别表示其对应的复数形式。为了实现动名分离，CDNV 将词性标注工具的词性类别划分为三个大类，分别是动词类（VB）、名词类（NN）和其他类（OT）。三种类别与传统词性标注工具的类别的对应关系如下表2-1 所示。

表 2-1 词性标注工具对词标注的类别与 CDNV 所用的词性类别的对应关系

Table2-1 The relation between the pos category of Pos Tagger with the Pos type used in the CDNV

Our Class	Pos Tagger
VB	‘VB’、‘VBP’、‘VBZ’、‘VBD’、‘VBN’、 ‘VBG’
NN	‘NN’、‘NNP’、‘NNS’、 ‘NNPS’
OT	‘CD’、‘JJ’、 ‘MD’、 ‘IN’、 ‘CC’、‘RB’、‘PRP’、‘RBR’
	‘RP’、‘PRP$’、‘JJS’、‘WP’、‘JJR’、‘RBS’、etc.

相比于 CBOW 模型，CDNV 模型的一个重要特点是在预测词的时候，会根据该词的上下文所决定的词性动态的选择词的编码。而在 CBOW 模型中，一个词只有一个唯一哈夫曼编码。这个编码的生成根据的是在大规模文本中该词的出现频数，与该词的上下文无关。CDNV 模型中对词典中所有词的二叉树的构建过程，需要考虑到该词在大规模文本中可以用作的词性类型，同时考虑了该词在用作这种词性时在文本中出现的频数。因此，对词进行编码的过程

综合考虑了动名分离以及频数统计。如果一个词在不同的语境中可以用作不同的词性，那么它就有多个编码。为了下文描述的方便，我们定义以下符号，$C(w_i, p_i) = (c_1^{(i)}, ...c_n^{(i)})$ 表示单词 w_i 在用作词性 p_i 时的编码，$p_i \in [VB, NN, OT]$。$c_j^{(i)}$ 取值为“0”或者“1”。n 是 $C(w_i, p_i)$ 编码的长度，$\theta(w_i, p_i)$ 表示根据编码 $C(w_i, p_i)$ 在二叉树中的路径选择的层次 Softmax 的参数。图2-2给出了单词“love”在不同的语境下用作不同词性时，不同的编码被激活的例子。单词“love”通常在不同的语境下被用作动词或者名词。因此在 CDNV 模型中，该词就有两个编码 $C(love, VB)$ 和 $C(love, NN)$。在图2-2 （a）中，给定一个英文句子“How much I love you ?”，由于词性标注工具在该语境中将“love”标注为“VBP”，将该词的上下文“How much I you ? *PADDING*”输入给 CDNV 模型，在输出层通过选择层次 Softmax 的参数 $\theta(love, VB)$ 来预测编码 $C(love, VB)$，然后通过优化算法更新“love”的上下文“How much I you ? *PADDING*”的词向量。类似地，在图2-2 （b）的句子“How much does love weigh ?”中单词“love”通过词性标注工具标注为“NN”，故其编码 $C(love, NN)$ 被激活。将该词的上下文“How much does weigh ? *PADDING*”输入给 CDNV，CDNV 在输出层使用与图2-2 （a）中不同的输出节点与神经连接（也即参数 $\theta(love, NN)$）识别 $(love, NN)$。通过这个例子，我们可以发现，CDNV 模型在输出层对动词和名词进行了分离。

虽然受大脑结构中的动名分离特性的启发，本文提出的模型对名词和动词进行了分离。然而，动词和名词具体需要如何进行分离，以及分离的程度级别并没有唯一的指导原则。基于此，本文提出与验证了三种具有代表性的方法对动词和名词进行分离，具体反映在它们的编码上。如图2-3(b)、(c)、(d) 显示了本文提出的三种不同编码方式分别对应 DNV-I、DNV-II 和 DNV-III，为了便于说明动词和名词分离的三种不同方案与 CBOW 模型的区别，2-3(a) 展示了 CBOW 模型的编码方式。CBOW 编码根据单词在数据集中的出现频数构造哈夫曼二叉树，通过从根结点到叶结点单词的路径表示该单词的编码，从图2-3(a) 中可以看出 CBOW 编码对于一个词无论其词性有多少种该词只有一个编码。与 CBOW 不同，DNV-I 则考虑了同一个词的不同词性，也即单词和词性 (w_i, p_i) 一起作为唯一的符号，最后通过其在文本中的频数构造哈夫曼二叉树，对一个单词的不同词性做到了分离，对三种词性之间并没有做特殊的分离，DNV-II 与 DNV-I 不同，DNV-II 则将“NN”和“VB”与“OT”首先做了分离，而“NN”和“VB”之间并没有做完全的分离，DNV-III 则在 DNV-II 的基础上对“NN”和“VB”做进一步的分离，从而在较大程度上对三种词性做了分离。

从以上的描述可以看出本文提出的三种动名分离方案，分别对应了对动词和名词的不同分离程度，其分离程度从小到大进行排列为 DNV-II、DNV-I、DNV-III。DNV-II 强制使动词和名词在右子树中进行编码，使得所有的动词和名词都拥至少有一个相同的编码位，同时对不同词的动词和名词词性进行混合编码。虽然 DNV-II 做到了对同一个词的不同词性进行了分离，但动词和名词依然存在较大的共同路径；DNV-I 虽然没有将动词和名词强行放到一个子树上进行编码，但是 DNV-I 对动词和名词进行了混合编码，因此存在大量的不同词的动词和名词词性的编码拥有较长的共同路径的情况。DNV-III 虽然也将动词和名词在同一个右子树中进行编码，但在右子树中，DNV-III 将动词和名词进行了完全分离，使得即使不同词之间的动词词性和名词词性都没有共同的路径，因此其分离程度最大。通过使用三种不同的编码方式可以得出本文提出的 CDNV 模型的三种具体形式，我们称之为 CDNV-I、CDNV-II 和 CDNV-III。

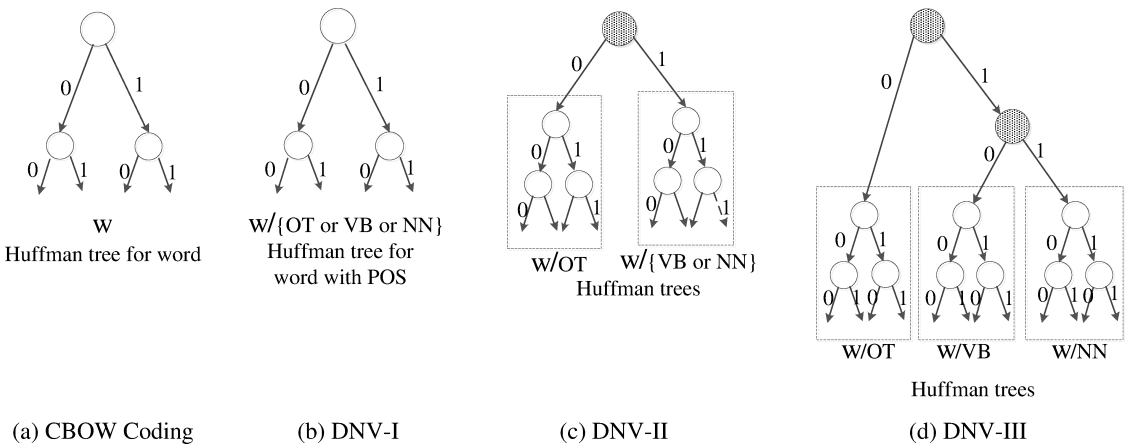


图 2-3 生词的编码的不同方案

Fig.2-3 Different ways of generating codes of words

2.3 模型训练

在模型的学习过程中，除了中心词 w_i，其局部上下文中的其他词 $w_{i-k},...,w_{i+k}$ 都映射到其对应的词向量 $e^{(i-k)},...,e^{(i+k)}$。然后，通过将这些词向量进行连接操作得到层次 Softmax 的输入 h_i，从而识别带有词性的中心词，其编码为 $C(w_i,p_i)$。因此，给定局部上下文 $w_{i-k},...,w_{i+k}$，识别 (w_i,p_i) 的条件概率表示为 $P(C(w_i,p_i)|w_{i-k},...,w_{i+k})$。CDNV 的优化目标函数为对数似然函数见公式2-2：

$$J = \frac{1}{M}\sum_{i=k}^{M-k} logP(C(w_i, p_i)|w_{i-k}, ..., w_{i+k}) \tag{2-2}$$

式中 M———表示训练集中所有的局部上下文的个数；

$P(C(w_i, p_i)|w_{i-k}, ..., w_{i+k})$ 的计算见公式2-3：

$$\begin{aligned} P(C(w_i, p_i)|w_{i-k}, ..., w_{i+k}) &= P((c_1^{(i)}, ...c_n^{(i)})|w_{i-k}, ..., w_{i+k}) \\ &= \prod_{j=0}^{n} P(c_j^{(i)}|c_0^{(i)}, ..., c_{j-1}^{(i)}, w_{i-k}, ..., w_{i+k}) \end{aligned} \tag{2-3}$$

条件概率 $P(c_j^{(i)}|c_0^{(i)}, ..., c_{j-1}^{(i)}, w_{i-k}, ..., w_{i+k})$ 的计算见公式2-4：

$$P(c_j^{(i)}|c_0^{(i)}, ..., c_{j-1}^{(i)}, w_{i-k}, ..., w_{i+k}) = c_j^{(i)}\frac{1}{1+e^{\theta_j^{(i)}h_i}} + (1-c_j^{(i)})\frac{e^{\theta_j^{(i)}h_i}}{1+e^{\theta_j^{(i)}h_i}} \tag{2-4}$$

这里参数 $\theta_j^{(i)}$ 的选择是根据 $c_j^{(i)}$ 在二叉树中的路径决定的。因此CDNV模型的参数包括层次Softmax的参数 $\theta = (..., \theta^{(i)}, ...)$ 和所有的词向量 $E = (..., e^{(i)}, ...)$，这里 $i \in [0, |V|]$。对于单词 w_k 的局部上下文 $(w_{i-k}, ..., w_{i+k})$，$\theta(w_i, p_i) = (\theta_1^{(i)}, ...\theta_n^{(i)})$ 和 $(e^{(i-k)}, ..., e^{(i+k)})$ 被选择性更新，它们都是整个模型参数 (E, θ) 的一小部分。因此，对于一次迭代，CDNV的连接非常稀疏，这就意味着我们可以非常容易的将CDNV并行化。

在训练的过程，本文选择后向传播[54]的随机梯度下降法更新模型参数。其数学描述见公式2-5：

$$\begin{cases} h_i \leftarrow h_i + \alpha\frac{\partial logP(c_j^{(i)}|c_0^{(i)}, ..., c_{j-1}^{(i)}, w_{i-k}, ..., w_{i+k})}{\partial h_i} \\ \\ \theta_j^{(i)} \leftarrow \theta_j^{(i)} + \alpha\frac{\partial logP(c_j^{(i)}|c_0^{(i)}, ..., c_{j-1}^{(i)}, w_{i-k}, ..., w_{i+k})}{\partial \theta_j^{(i)}} \end{cases} \tag{2-5}$$

2.4 时间复杂度

一方面，基于CBOW和Skip-gram[11]是当前效率较高的两个模型，另一方面，CDNV与CBOW和Skip-gram的相关性较大。本文选择这两个模型作为对比。与[11]类似，我们选择模型在一次迭代中需要访问的参数数量来描述模型的时间复杂度。CBOW的模型一次迭代的时间复杂度见公式2-6，Skip-gram模型的时间复杂度见公式2-7，CDNV模型的时间复杂度见公式2-8。

$$O_1 = 2kD + Dlog_2(|V|) \tag{2-6}$$

$$O_2 = 2kD + 2kDlog_2(|V|) \tag{2-7}$$

$$O_3 = 2kD + 2kDlog_2((1+\lambda)|V|) \tag{2-8}$$

式中 $2k$———局部上下文的窗口大小（通常取值 4 或者 10）；

D———词向量的维度（通常取值为 50-200）；

$|V|$———词典的大小;

λ———使用动名分离后在层次 Softmax 层增加的结点的个数比例;

在公式 2-6, 2-7和 2-8中，第一项 $2kD$ 表示输入层需要访问的参数，第二项为输出层层次 Softmax 需要访问的参数。

由以上公式，我们可以得到 CDNV 与 CBOW 的时间复杂度的比值为 R_1 如公式2-9 所示，CDNV 与 Skip-gram 的时间复杂度比值为 R_2 如公式2-10所示。

$$R_1 = \frac{2k + 2klog_2(|V|) + 2klog_2(1+\lambda)}{2k + log_2(|V|)} \tag{2-9}$$

$$R_2 = \frac{1 + log_2(|V|) + log_2(1+\lambda)}{1 + log_2(|V|)} \tag{2-10}$$

由于使用动名分离，在输出层一个单词可能对应多组输出编码，因此增加了层次 Softmax 层的总的结点数量。理论上 λ 的最小上界为 $sup(\lambda) = 3$，最大下界为 $inf(\lambda) = 0$。然而，在十亿词规模的数据集上的统计结果显示，$\lambda \approx 0.3$。从以上公式 2-9和 2-10可以看出 λ 并不能显著的增加输出层的结点数量。由于在映射层，CDNV 将“向量加”操作替换为了“向量连”操作，也在一定程度上增加了参数数量。最终，我们可以得到 CDNV 与 CBOW 模型的时间复杂度比值 $R_1 \approx 6.5$，CDNV 与 Skip-gram 模型的时间复杂度的比值 $R_2 \approx 1.0$。因此，CDNV 的时间复杂度确实比 CBOW 高，但是与 Skip-gram 的时间复杂度相当。

2.5 实验

2.5.1 实验设置

词向量的训练需要大规模的无标注自然文本数据，本文采用维基百科英文语料（截止 2013 年 8 月的快照）[1] 和路透社 RCV1[55] 数据集训练 CDNV 模型。为了去除数据集中的噪音，我们首先需要对数据做规范化处理，具体步骤如下：1、删除数据集中较短（长度小于 5 个单词）的句子；2、去除非正常的句子（句子中小写英文字母 a-z 的比率小于 90%）；3、将所有的大写字母转化为小写字母；4、将所有的阿拉伯数字转换为符号“D”，例如，“1998”转换后为“DDDD”；5、将数据集中出现较少（频数小于 30）的词统一转化为符号

[1] http://corpus.byu.edu/wiki/

“UNKNOWN”字符。最终将两个数据集整合到一起得到一个拥有 120,000,000 个符号的纯文本数据集，其词典大小为 300,000。为了得到词的词性，本文使用斯坦福词性标注工具 Stanford POS-tagger[2] 对整个数据集进行标注。为了便于与公开的词向量进行对比，本文选择词向量的维度大小为 50，窗口的大小为 [-5,5]。

为了更好的验证 CDNV 模型的特性，本文选择 CBOW，Skip-gram 和本文构造的 CCWM (Continuous Concatenating Word Model) 作为基准模型。CCWM 模型是通过将 CBOW 模型中映射层的“向量加”操作替换为“向量连”操作得到，该模型是为了验证单纯引入词序特征对词向量性能的影响。另外，本文也将 CDNV 训练得到的词向量与大多数公开的词向量做对比。这些公开的词向量包括 HLBL（hierarchical log-bilinear）词向量[12]、Turian 的词向量[21]、C&W 的词向量[10] 以及 Huang 的词向量[14]。所有这些词向量都可以从公开连接中下载获得。其中，HLBL 和 Turian 的词向量是在包含 37,000,000 个单词的路透社数据集 RCV1 上训练得到的，C&W 的词向量是在拥有 853,000,000 个单词的路透社 RCV1 和维基百科数据集上训练获得，Huang 的词向量是在包含 990,000,000 个单词的维基百科数据上训练获得的。从上述的描述可以发现，这些模型的训练数据集大小并不一致。导致这个问题的主要原因，是它们当中的一些模型训练非常耗时，将这些模型在新的数据集上重新训练非常困难。以 C&W 的模型为例，在 852,000,000 大小的数据上训练词典大小为 130,000 个单词 50 维的词向量，耗费大约 2 个月的时间。因此，一些工作为了与这些词向量的做对比，通常会选择直接从相应连接下载对应的词向量进行对比[21, 56]。

2.5.2 评价指标

词向量的质量评测常用方法是通过将其作为额外特征加入到传统的自然语言处理任务中，观察其对系统的提升幅度来判断。与大多数评价词向量的方法一致，本文将词向量作为额外特征加到两个基于条件随机场的序列标注系统，组块分析和命名实体识别中。本文选择 CRFsuite[57] 作为条件随机场的实现。评价系统性能的标准采用 F1 值 2-11。

$$F = \frac{2PR}{(R+P)} \tag{2-11}$$

[2]http://nlp.stanford.edu/downloads/tagger.shtml

式中 P———组块或命名实体的准确率；

R———组块或者命名实体的召回率;

为了验证词向量对命名实体识别和组块分析任务的性能的影响是否是显著性差异的，本文做了显著性分析验证。由于不同模型正确和错误标注的样本并不能直接对比[58]，因此，对不同模型的 F1 值做显著性分析比较困难。Yeh[59] 使用随机产生测试集的方法进行显著性分析，具体的方法是在测试集上进行多次重复采样方法（bootstrapping）。威尔科森符号秩检验（Wilcoxon signed rank test）[60] 是一种针对匹配对的非参数显著性检验方法。其普遍被应用于序列标注问题的显著性检验问题中[61, 62]。本文基于多次重复采样方法（bootstrapping）的结果做显著性检验。从命名实体识别和组块分析测试集中随机采样 100 个句子 200 次，作为多次重复采样数据集。对每个多次重复采样的数据集，分别对所有模型计算 F1 值。然后，使用威尔科森符号秩检验方法判定两种方法的结果是否是具有显著性差异（$p - value < 0.05$）。

2.5.3 实例分析

为了验证通过 CDNV 模型学到的词向量的质量，我们通过人工筛选一些具有代表性的词作为样本，通过词向量计算得到这些样本词的最临近的 4 个单词。通过观察分析这些最邻近词的质量得出 CDNV 模型的性能。选取样本词的标准是样本需要比较常见，并且可以被用作不同的词性，例如，英语单词“laugh”在文本中既可以被用作名词“笑声”，也可以被用作动词“发笑”，“like”既可以用被用作动词“喜欢，希望”，也可以被用作介词“如，比如”，也可以被用作副词“好像”。为了对比的完整性，一些只有单个词性的词也被选定，例如，“organization”通常只被用作名词“组织，团体”，“create”通常只被用作动词“创造，建立”。

这些词的四个最近邻词语，是通过计算词典中所有词与这些样本之间的余弦相似度得到的。其结果如表2-2 所示。从表2-2 中可以看出，虽然单词“laugh”和“like”是非常常见的两个单词，但是，基于 pairwise ranking 的模型（C&W,Turian and Huang）得到的其 4 个最近邻词语的质量并不理想。例如，对于 C&W 词向量，“haircut”与“laugh”，“bless”与“like”非常相近，然而从语义上它们的意义完全不同。导致 C&W 模型得到这样结果的原因可能来自于 pairwise ranking-based 模型的训练策略。例如，“bless”和“like”很多时候上下文比较相近，例如，“I like you”和“I bless you”。通过 pairwise ranking-based

表 2-2 对于选定的一些英文单词，从不同的词向量模型中得到它们的余弦距离相近的词

Table2-2 Nearest neighbors of words derived from different word embeddings

Model	laugh	like	organization	create
C&W	haircut, look, shout, audition, laughs	with, bless, especially, peddled, as	organization, agency, entity, institution, center	perform, incorporate, develop , generate
Huang	hug, shout, joke, cheer, amuse	by, including, fearlessly, such, inside	initiative, agency, institution, organization, entity	allow, develop, generate, maintain, describe
Turian	grin, walk, slap, skirt, fool	when, from, between, with, recognised	package,scheme, position, plan, obligation	pursue, promote, deliver, facilitate, establish
HLBL	arose, nedded, metamorphosis, jokes, ringstrasse	prefer, afford, through, manage, chose	organisation, consortium, authority, administration, society	creating, creates, establish, introduce, establishing
CBOW	kiss, flinch, squirm, screaming, laughs	despise, detest, liken, admire, unlike	organisation, plan, initiative, community, organizing	introduce, develop, establish, add, incorporate
Skip-gram	everybody, shit, crap, laughs, suff	unlike, jumpstyle, admire, dan, prefer	organisation, initiative, plan, community, entity	introduce, incorporate, deliver, establish, develop
CCWM	clap, shout, smile, kiss, weep	resembling, including, unlike, in, besides	organization, ngo, initiative,institution, entity	introduce, add, develop, establish, produce
CDNV-I	kiss, smile, giggle, laughs, fool	resembling, despise, prefer, likes, unlike	organization, ngo, community, institution, association	establish, develop, creates, produce, introduce
CDNV-II	smile, kiss, laughs, giggle,fool	resembling, including, prefer, likes, unlike	organisation, institution, initiative, ngo, association	introduce, develop, establish, devise, adapt
CDNV-III	smile, giggle, laughs, laughing, weep	resembling, despise, prefer, unlike, including	organisation, institution, ngo, association, federation	introduce, recreate, devise, develop, establish

准则训练模型时，随机选取的负例信号不足以将它们两个分开。从表2-2中还可以看出，无论是忽略了词序，还是不考虑动名分离的基于熵准则的模型（HLBL, CBOW, Skip-gram 和 CCWM）得到的结果也不理想。相比于 CDNV 模型，虽然 CBOW 和 Skip-gram 模型得到的结果从语义上是相关的，但是语法和语义相似性并不理想。例如，对于单词“laugh”的四个最相关词，“everybody”和“kiss”都比“laughs”排序靠前。导致这种结果的原因，可能来自于 CBOW 和 Skip-gram 忽略了词序特征。相比于其他模型，CDNV 的结果更为理想，例如，在“laugh”的最邻近词中，“smile”排在了其它候选词的前面，从“simle”和“laugh”的语义和用法上分析得知，这两个词非常相似。语法上，“smile”既可以用作动词，也可以用作名词，与“laugh”的用法相似，语义上“smile”用作动词时译为“微笑”，用作名词时译为“笑容”，其与“laugh”也较为相近。“laugh”和“like”的结果显示，CDNV 对可以用作多个词性的单词能够学到更好的词向量。与“laugh”和“like”不同，“organization”和“create”通常只用作一种词性。表2-2显示通过 CDNV 模型的词向量，对“organization”和“create”得到的结果在语义和语法方面，与它们都较为相近。对于 HLBL 模型，虽然“creates”和“creating”与“create”的语义上非常相近，它们都是“create”的不同变形。然而，它们与“create”的用法截然不同。CDNV 模型将“creating”和“creates”、“organizing”排在较为靠后的位置，是对语义和语法相似度做了一种折中处理。

2.5.4 命名实体识别任务

命名实体识别（Named Entity Recognition）是自然语言处理中典型的序列标注问题，其目的是识别出文本中的命名实体的边界和类型。其不仅对信息抽取、问答系统、机器翻译等多个自然语言处理任务能够起到基础性作用，同时这些年来常被用于评测序列标注系统的性能以及词向量的质量[21]。本文采用标准的命名实体识别的公开评测任务 CoNLL2003[3]评价词向量的质量。该任务的数据集来源于路透社新闻 [63]。数据集被划分为三个部分训练集、开发集和测试集。训练集包括 204,000 个词语、14000 个句子、946 个文档；开发集包括 51000 个词语、3300 个句子、216 个文档；测试集包括 46000 个词语、3500 个句子、231 个文档。命名实体识别数据集中，对于一个句子，其中大多数的词都不是命名实体，只有少量的词是命名实体，例如，数据样例如 S，

[3]http://www.cnts.ua.ac.be/conll2003/ner/

S :***West/B-MISC Indian/I-MISC*** *all-rounder/O* ***Phil/B-PER Simmons/I-PER*** *took/O four/O for/O 38/O on/O Friday/O as IN I-PP O* ***Leicestershire/B-ORG*** *beat VBD I-VP O* ***Somerset/B-ORG*** *by IN/O an DT/O innings/O and/O 39/O runs/O in/O two/O days/O to/O take/O over/O at/O the/O head/O of/O the/O county/O championship/O ./O*

从中可以看出只有少量的单词为命名实体的类别。本文通过训练集训练基于条件随机场的命名实体识别模型，通过开发集选择模型参数，最后在测试集上评估模型结果。

表 2-3 基于条件随机场的命名实体识别系统用到的特征

Table2-3 Features used in the CRF-based NER systems

特征	特征描述
Embedding features[if applicable]: $e_{t+i}[d]$, for i in {-2,-1,0,1,2} features: $f[t+i]$ for i in {-2,-1,0,1,2}; $f[t+i]\|f[t+i+1]$ for i in {-1,0}, f in $[w, chk, pos, type, shape]$ Prefixes and suffixes features: $s[t+i]$, for i in {-2,-1,0,1,2}	e_{t+i} 表示句子中第 $t+i$ 词的词向量。d 的取值范围为词向量的维度大小。 $w, chk, pos, type, shape$ 分别表示单词，组块类型，词性，词的类标和词的形状类标特征。 s 表示在句子 t 位置的词的前缀和后缀特征。

对于基于条件随机场的命名实体识别系统，常用的基本特征如表2.5.4 所示。这些特征是通过 Naoaki Okazaki 提供的脚本程序 (ner.py[4]) 抽取获得。在这些常用特征的基础上，增加词向量作为新的特征，通过其对整个识别系统的性能提高幅度，判断词向量的质量。为了更好的发挥词向量的能力，Turain 等人通过实验发现，对词向量进行尺度归一化能够显著的提高基于条件随机场的命名实体识别和组块分析任务的性能 [21]。其尺度归一化公式见2-12。本文对所有的词向量同样做了尺度化（Scaling）处理。通过实验发现当我们选择词向量的标准方差为 σ=1.0 时，所有词向量的性能对于系统提高的幅度最高。

$$E \leftarrow \sigma \times E / stddev(E) \tag{2-12}$$

式中 E———词向量;

stddev(E)———词向量的标准方差;

σ———控制了词向量归一化之后新的词向量的标准方差;

[4]http://www.chokkan.org/software/crfsuite/

表2-4列出了基于条件随机场的命名实体识别系统当增加了不同的词向量作为特征后的性能。表2-5列出了 CDNV-I、CDNV-II、CDNV-III 与其他模型的显著性检验的结果，表2-5中每个单元格中的数字表示对应的行模型与对应的列中的模型的性能显著性差异对比结果。其中“1”表示对应的行中模型显著的好于列中的模型，“-1”表示对应的行模型显著的劣于列中的模型，“0”表示行中模型与列中模型的性能没有显著性差别。结果显示，CDNV 与其他模型的词向量一样，都可以显著地提高基于 CRF 的命名实体识别系统的性能。结合表2-4和2-5，当增加了 CDNV-I、CDNV-II、CDNV-III 的词向量后，系统显著地优于基准系统 CBOW 和 CCWM 的词向量以及其他公开的词向量的性能。显著性检验同时显示 CDNV-I、CDNV-II 和 CDNV-III 的性能并没有显著性差异。这些结果表明三种类型的动名分离方式对命名实体系统的性能影响差异性较小。

表 2-4 基于条件随机场的命名实体识别系统的 F1 性能结果（%）

Table2-4 F1-measure Performance of CRF-based NER systems(%)

Data	Baseline	C&W	Huang	Turian	HLBL	CBOW	Skip-gram	CCWM	**CDNV-I**	**CDNV-II**	**CDNV-III**
Dev	89.18	92.12	89.09	91.24	90.80	91.58	91.90	91.85	**92.01**	**92.13**	**91.78**
Test	84.62	87.61	84.68	87.21	86.65	87.56	87.42	87.83	**88.44**	**88.39**	**88.33**

表 2-5 基于条件随机场的命名实体识别系统的 F1 统计显著性检验结果

Table2-5 Performance comparisons of F1 scores regarding the statistical significance

Model	Base	C&W	Huang	Turian	HLBL	CBOW	Skip-gram	CCWM	**CDNV-I**	**CDNV-II**	**CDNV-III**
CDNV-I	1	1	1	1	1	1	1	1	——	0	0
CDNV-II	1	1	1	1	1	1	1	1	0	——	0
CDNV-III	1	1	1	1	1	1	1	1	0	0	——

表2-6列出了 CDNV 模型以及其基准模型（CBOW 和 CCWM）在命名实体识别各个类别上具体的性能。结果显示，CDNV 模型在命名实体识别的所有类别上都超过了 CBOW 和 CCWM 模型。通过分析词的词性类别，可以发现大多数命名实体的组成单词是名词（NN）。统计结果显示，大约 88% 的词语是专有

名词，大约 5% 的词语是名词，只有不到 1% 词语是动词，其他的词是形容词或者副词。这些统计数据表明，对名词得到更好的词向量将对命名实体识别系统的性能起到主要的作用。从实例分析部分中的例子中，我们可以得知 CDNV 模型对于名词可以学到更好的词向量表示。在本节试验中，通过以下例子进行说明，给定句子 S，

S :*International trade union leaders on Friday expressed outrage that the head of the* ***International Labour Organization (ILO)*** *had been barred from speaking at next week's WTO meeting in Singapore.*

在该句话中，“***International Labour Organization (ILO)***”的正确标注为“***International/B-ORG Labour/I-ORG Organization/I-ORG (ILO/B-ORG)***”。然而，只有 CDNV 模型将他标注正确，CBOW 和 CCWM 将其标注为“***International/B-LOC Labour/I-LOC Organization/I-LOC (ILO/B-LOC)***”。增加 CDNV 词向量使得系统可以正确识别该命名实体，可能归因于动名分离特性使得模型对名词的表示质量较高，在本例中主要体现在单词“organization”的向量上。

表 2-6 在命名实体识别任务上各个类别上的三种模型的 F1 值（%）

Table2-6 Performance of F1 scores on all categories of NER(%)

Model	CBOW	CCWM	CDNV-I	CDNV-II	CDNV-III
LOC	89.81	89.90	90.92	90.04	89.90
MISC	78.34	78.72	79.23	79.29	79.60
ORG	83.50	83.61	84.61	84.56	84.48
PER	93.37	93.94	94.34	94.61	94.38

命名实体识别任务不仅要识别出名词与其他词之间的区别，同时需要对名词做进一步的识别（例如，Hilton 既可以是人名也可以是酒店名）。在维基百科和路透社 RCV1 上的统计数据显示，1）数据集词典中大约 90% 的词拥有名词性（“NN”）类别；2）大约 40% 的名词同时有其他词性（“OT”）类别；3）不到 10% 的名词同时拥有动词性（“VB”）类别。从 CDNV 三种动名分离方案的原理图（2-3 （b）、（c）和（d））可以发现，三种变形都对“NN”与“OT”，“NN”与“VB”做了分离，从而可以对名词学到更好的词向量（例如，“organization”的例子）。虽然同时拥有“OT”类别的“NN”的比重较大，从实验结果可以发现，对“NN”和“OT”做进一步的分离（如 CDNV-II）对性能的影响并不显著，由于同时拥有“NN”和“VB”两个类别的词语在“NN”

中占比较低，进一步对“NN”和“VB”做分离，对整个“NN”的词向量质量影响有限。这就解释了为什么 CDNV 的三种不同变形 CDNV-I，CNDV-II 和 CDNV-III 的表现并没有显著性的差别。

2.5.5 组块分析任务

组块分析又被称为浅层句法分析，其目标是将句子根据句法功能分为不同的部分。组块分析是一种典型的序列标注问题。CoNLL2000 公开任务[5]为组块分析提供了标准的数据集。该数据来自于华尔街报纸（Wall Street Journal）。数据集将 15-18 部分作为训练集，第 20 部分作为测试集[64]。与命名实体识别任务不同，在组块分析任务中除了标点符号，所有的词语都要被标上不同的组块类别。其数据样例如 S，

S :*Rockwell/B-NP International/I-NP Corp./I-NP 's/B-NP Tulsa/I-NP unit/I-NP said/B-VP it/B-NP signed/B-VP a/B-NP tentative/I-NP agreement/I-NP extending/B-VP its/B-NP contract/I-NP with/B-PP Boeing/B-NP Co./I-NP to/B-VP provide/I-VP structural/B-NP parts/I-NP for/B-PP Boeing/B-NP 's/B-NP 747/I-NP jetliners/I-NP ./O*

从数据中可以看出，除了最后的标点符号，句子中的所有词分别属于对应的某一个具体的组块类。本文使用 CRFsuite 和 L-BFGS 优化算法构建组块分析系统。由于该数据集缺少开发集，我们使用 10-fold 交叉验证的方法选择模型参数。我们将模型的超参数 $c1$ 固定为 $c1 = 0$，对模型超参数 $c2$ 从 0.1 到 1.0 以步长 $step = 0.1$ 进行试验选择得到最好的模型参数。

表 2-7 基于条件随机场的组块分析系统用到的特征

Table2-7 Features used in the CRF-based Chunking systems

特征	特征描述
Embedding features[if applicable]: $e_{t+i}[d]$, for i in {-2,-1,0,1,2} features:features: $f[t+i]$ for i in {-2,-1,0,1,2}; $f[t+i]\|f[t+i+1]$ for i in {-1,0}, f in $[w, pos]$	e_{t+i} 表示句子中第 $t+i$ 词的词向量。d 的取值范围为词向量的维度大小。w, pos 分别表示单词，词性特征。

同样对于组块分析任务，其也有一些常用的特征如表2-7所示，这些特征通

[5]http://www.cnts.ua.ac.be/conll2000/chunking/

过 Naoaki Okazaki 提供的脚本[6]抽取得到，在此基础上增加词向量特征，从而通过观察词向量对标注系统性能的影响判断词向量的质量。实验结果表明，对基于条件随机场的组块分析任务，当对所有的词向量以超参赛数 $\sigma = 0.1$ 使用公式2-12尺度归一化处理后，对标注系统的提升最大。

表 2-8 基于条件随机场的组块分析系统的 F1 性能结果（%）

Table2-8 F1-measure Performance of CRF-based Chunking systems(%)

Data	Base	C&W	Huang	Turian	HLBL	CBOW	Skip-gram	CCWM	**CDNV-I**	**CDNV-II**	**CDNV-III**
Test	93.65	94.00	93.96	94.04	93.94	94.07	93.84	94.02	**94.14**	**94.12**	**94.18**

表2-8和表2-9分别列出了不同词向量的性能，以及 CDNV-I、CDNV-II 和 CDNV-III 与其他模型的性能的显著性检验结果。从表中结果可以看出，CDNV 模型得到的词向量与其他词向量一样，都对基于 CRF 的组块分析系统的性能有所提高。结合表2-8和表2-9，CDNV-I、CDNV-II 和 CDNV-III 的性能显著的优于基准模型（CBOW 和 CCWM）以及其他公开的词向量。

表 2-9 基于条件随机场的组块分析系统的 F1 统计显著性检验结果

Table2-9 Performance comparisons of F1 scores regarding the statistical significance

Model	Base	C&W	Huang	Turian	HLBL	CBOW	Skip-gram	CCWM	**CDNV-I**	**CDNV-II**	**CDNV-III**
CDNV-I	1	1	1	1	1	1	1	1	——	0	0
CDNV-II	1	1	1	1	1	1	1	1	0	——	-1
CDNV-III	1	1	1	1	1	1	1	1	0	1	——

为了进一步分析的方便，我们将组块分析中除了动词性组块（“VP” Chunk）和名词性组块（“NP” Chunk）的其他类型组块（即“ADJP”，“CONJP”，“PP”等）归一化到一类 OT 组块。表2-10列出了 CDNV 模型与其两个基准模型（CBOW 和 CCWM）在各个细分的组块类上的实验结果。从结果显示，CDNV-I、CDNV-II 和 CDNV-III 在三个类别“NP”、“VP”和“OT”上，超过了基准系统的性能。例如，给定句子 S，

[6]http://www.chokkan.org/software/crfsuite/

S :*He said the Chaos Computer Club, of West Germany, once managed to invade SPAN and do such things as* **change** *the value of pi, messing up some calculations.*

词语***“change”***的正确类别为***“change/B-VP”***。***“change”***在不同的语境中，既可以作为名词（NN）也可以作为动词（VB）使用。然而，在这个例子中，只有增加了 CDNV 词向量的模型，才能得出正确的标注结果。CBOW 和 CCWM 都将其标注为***“change/B-NP”***。从这个例子可以看出，CDNV 模型对***“change”***学习到了更好的词向量。

与命名实体识别任务上的性能不同，在 CDNV 三种变形中，CDNV-III 显著优于 CDNV-I 和 CDNV-II。从三种模型的词的编码生成方案（2-3 （b）,（c）和（d））可以看出，相比于 CDNV-I 和 CDNV-II，CDNV-III 对名词和动词做了更为深入的分离。组块分析需要对不同的词，区分出它们的词性并决定它们属的组块。在实验数据集上的统计结果显示，大约 76% 的英文单词属于“NP”和“VP”这两个主要组块。因此，动词和名词的词向量的好坏对整个系统性能的影响起着主要作用。在维基百科和路透社 RCV1 上的统计结果显示，1）词典中大约 20% 的词语可以用作动词（“VB”），2）拥有“VB”类别的词语中大约 65% 的词同时有“OT”类别，3）拥有“VB”类别的词语中大约 85% 同时有“NN”类别。这些统计结果表明不同的动名分离方式对 CDNV 模型学习动词的词向量的表示影响较大，由于 CDNV-III 的分离方式使得模型将动词（“VB”）与名词（“NN”）和其他类（“OT”）完全分离，从而对动词（“VB”）的词向量学习更为有利，因此，CDNV-III 的词向量对标注系统的提升显著优于 CDNV-I 和 CDNV-II。

表 2-10 F1 性能结果在各个组块类别上的结果（%）。

Table2-10 Performance of F1 scores on all categories of Chunk(%)

Model	CBOW	CCWM	CDNV-I	CDNV-II	CDNV-III
NP	94.44	94.45	94.54	94.52	94.58
VP	94.48	94.30	94.54	94.49	94.60
OT	95.72	95.73	95.75	95.73	95.80

2.5.6 讨论

本文提出的基于动名分离的词向量学习模型，不仅在输出层考虑了动名分离，同时在映射层考虑了局部窗口内的词序信息。那么还有两个问题我们需要

回答：1）是不是所有的词性分离方式都能够提高词向量的质量？2）如果不考虑局部窗口内的词序特征，动名分离是否依然能够提升词向量的质量？下面我们分别通过实验对上述两个问题进行讨论分析。

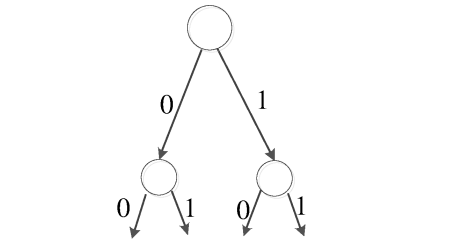


图 2-4 一种对所有词的词性进行分离的编码生成方案

Fig. 2-4 A method to generate codes of words with all POS tags

对词典的编码存在大量的潜在分离方案。为了便于研究第一个问题，本文只对一种较为代表性的分离方案进行探讨，也即对所有词性进行分离，所有词性的分离方案的具体编码二叉树的构造方式如图2-4 所示。该分离方案将词语的所有词性都进行分离，需要注意的是，这里所指的词性是词性标注工具普遍使用的词性类别，而不是上文中所使用的“NN”、“VB”和“OT”方案。然后，根据 (w_i, p_i) 在数据集中的出现频数，构造哈夫曼树，这里 p_i 的取值为所有的词性，从而得到其对应的编码。表2-11 列出了 CBOW 和 CCWM 模型在顶层分别使用该分离方式后，得到的词向量（CBOW-POS 和 CCWM-POS）在命名实体识别和组块分析任务上的性能。从实验结果可以看出，在顶层使用对所有词性进行分离的方案学习得到的词向量虽然对命名实体识别和组块分析任务都有提高，但不能稳定的提升 CBOW 和 CCWM 模型的性能。因此，从这样一种特殊的分离方案中我们可以得出的结论是，并不是所有的分离方案都可以提升模型的性能。

表 2-11 命名实体和组块分析系统使用所有词性分离得到的词向量作为特征后的结果（%）

Table2-11 Performance NER and Chunking systems using the dissociation of all POSs(%)

Model	**NER**	**Chunking**
Base	84.62	93.65
CBOW	87.56	94.07
CCWM	87.83	94.02
CBOW-POS	87.40	93.86
CCWM-POS	87.77	94,07

为了回答第二个问题，我们将 CBOW 的输出层分别替换为本文提出三种动名分离方案后的层次 Softmax，分别命名为 CBOW-DNV-I,CBOW-DNV-II 和 CBOW-DNV-III。表 Table2-12 列出了 CBOW 模型使用本文提出的三种动名分离方案后的实验结果。从结果可以看出，相比于不加词向量特征的系统，三种模型的词向量加入到基于条件随机场的命名实体系统中，对系统的性能都有提高。相比于 CBOW，CBOW-DNV-I 和 CBOW-DNV-III 的词向量对系统性能的提升都有所增加，但使用 DNV-II 系统的性能反而降低。对于基于条件随机场的组块分析系统，在 CBOW 模型上无论使用哪种分离方式所学到的词向量，与 CBOW 的词向量对比，对标注系统的性能提升都有所下降。这些实验结果表明忽略了词序但使用动名分离的模型的词向量性能非常不稳定。

表 2-12 CBOW 模型使用不同的动名分离方案在命名实体和组块分析任务上的性能（%）
Table2-12 Performance of CBOW using different DNVs on NER and Chunking(%)

Model	**NER**	**Chunking**
Base	84.62	93.65
CBOW	87.56	94.07
CBOW-DNV-I	87.83	93.97
CBOW-DNV-II	87.32	93.95
CBOW-DNV-III	87.86	93.96

结合上节中的实验分析，可以看出词序和动名分离都非常重要，这两种信息共同使用能对词向量学习模型具有较为稳定的提升，单独引入词序特征或者动名分离无法对词向量的质量进行稳定的提高。本文提出的 CDNV 模型充分考虑了词序信息以及动名分离特性，在命名实体识别和组块分析任务上获得了稳定的显著地提高。

2.6 本章小结

本章提出了一种基于动名分离的词向量学习模型。该模型受人类学习语言过程中动名分离特性的启发，创新性地将动名分离特性引入到了词向量的学习过程中，同时保持了局部上下文的词序信息。本章详细分析了其时间复杂度、在常用词上的实例、以及其在命名实体识别和组块分析任务上的性能。实验与分析表明，基于动名分离的词向量学习模型，不仅具有较高的学习效率，同时在命名实体识别和组块分析任务上，显著地优于其对比的其它词向量。

第 3 章 基于深度卷积神经网络的语句匹配架构

3.1 引言

上一章，我们介绍了通过神经网络学习语言中基本单位词的表示方法，词向量的表示包含了词与词之间的语义关系，使得词与词之间不再是相互独立的表示。那么，能否对语句也表示成类似词向量的稠密而低维的实值向量表示？自然语言处理领域中，有大量语句级的任务。例如，语句的情感分类[65]、问答系统中的问题分类[66]、以及语句级的复述检测[67]等。对于这些任务，首先需要解决的问题是语句的表示。传统的方法对语句进行表示，通常使用向量空间模型 (vector space model)。虽然向量空间模型在信息检索，以及多个自然语言处理任务都表现出了有效性，但是向量空间模型不仅忽略了语句中词与词之间的顺序，同时向量空间模型假设语句中的词与词之间是相互独立的，这种表示方法完全忽略了词与词之间的依赖关系。通过向量空间模型对语句表示也存在维数灾难和表示稀疏等严重问题。近年来，随着深度学习在自然语言处理领域的发展，已有一些工作通过深度模型学习语句表示，并在大量的自然语言处理任务上取得了突破性进展，例如，Socher 等人提出的递归神经网络在情感分析[17]和复述检测[28]任务上显著超过了传统方法。本章在词向量的表示基础上，提出基于深度卷积神经网络语句表示模型。基于深度卷积神经网络的语句表示模型通过逐层的卷积操作对相邻词进行组合表示，通过局部最大池化操作选取合理的邻近组合，从而通过多层的操作得到固定维度的语句向量表示。

自然语言处理的很多任务都可以归纳为语句匹配问题，例如，问答系统中问句与答案的匹配、统计机器翻译中的源端与目标端短语的匹配、以及复述检测等。传统的相似性匹配（例如，复述检测中判断两个语句是否表达同一个语义）和相关性匹配（例如，信息检索中关键词检索）方法，使用的传统向量空间模型表示文本只能捕捉到匹配对在较浅层次上的匹配关系。复杂的语言匹配问题需要将语言匹配对的不同特性，在不同的抽象层次上进行匹配，例如，对话系统中，判断一句话是否是对说话者的合适回复，不仅需要模型将两者的语义进行很好的匹配，也需要对对话细节能够进行很好的捕捉。传统的方法大多是通过观察大规模的匹配数据集或者根据人类对语言的先验知识，人工发现这种匹配关系，进而提取匹配特征[68, 69]。这种方式很难覆盖语言中多种多样的复杂匹配关系。基于此，本章从深度神经网络模型的角度出发，对自然语言中

的语句级的语义匹配进行深入研究。借鉴图像处理领域的卷积神经网络模型，本文提出了基于深度卷积神经网络的语句匹配架构，不仅可以很好的表示语句的层次结构，而且可以获取语句在不同抽象层次上的匹配模式。由于不需要任何先验知识，深度卷积神经网络语句匹配架构具有很好的通用性，可以用到不同性质，不同语言的匹配任务上。通过实验验证，深度卷积神经网络语句匹配架构在多个语义匹配任务上都有突出的表现，显著优于其他可对比模型。

本章的组织结构如下：章节 3.2、首先详细介绍了本文提出的基于深度卷积神经网络的语句表示模型，对模型中两类基本操作卷积计算和局部最大池化操作进行了详细描述，对模型如何处理语句长度不一致的情况做了详细介绍。章节 3.3、对基于深度卷积神经网络的语句表示模型的特点进行了深入分析。章节 3.4、详细介绍了提出的两种基于深度卷积神经网络的语句匹配架构，并分析了两种架构之间的关系。章节 3.5、描述了本章提出的两种语句匹配架构的训练方法。章节 3.6、将本章提出的两种基于深度卷积神经网络的语句匹配架构以及主要的对比模型在三种不同类型、不同性质的语句匹配任务上进行了深入的实验对比。最后给出了本章的结论。

3.2 基于深度卷积神经网络的语句表示模型

本章提出的基于深度卷积神经网络的语句表示模型的整体架构如图3-1所示，图中虚线的框图表示对语句长度进行补全得到最大长度的全 0 向量，该部分向量的影响可以通过“门函数”逐步消除后续的影响。该模型的输入是词向量矩阵。词向量矩阵是通过将语句中的词转换为对应的词向量，然后按照词的顺序排列得到。词向量可以通过相应的无监督词向量学习方法（如 word2vec[11]，CDNV 模型等）在大规模的纯文本语料上训练获得。该模型通过多层交叠的卷积和最大池化操作，最终将语句表示为一个固定长度的向量。该架构可以通过在语句向量顶层增加一个 Softmax 分类器用于多种有监督的自然语言处理任务上。从而有监督的针对具体的任务学习得到语句的表示。与大多数的卷积模型一样[10] [32]，本文采用共享参数的局部卷积操作，为了能够充分对语句中丰富的结构以及词的组合方式进行建模，对每个卷积操作设置较大维度的卷积输出。为了能够对卷积操作得到的相邻词的组合表示进行合理选择，基于深度卷积神经网络的语句表示模型采取了局部最大池化操作。

卷积操作 卷积神经网络在图像和语音处理领域上得到了广泛的应用。传统卷积神经网络的卷积操作通常是在输入图像的局部区域进行。因为对于图像

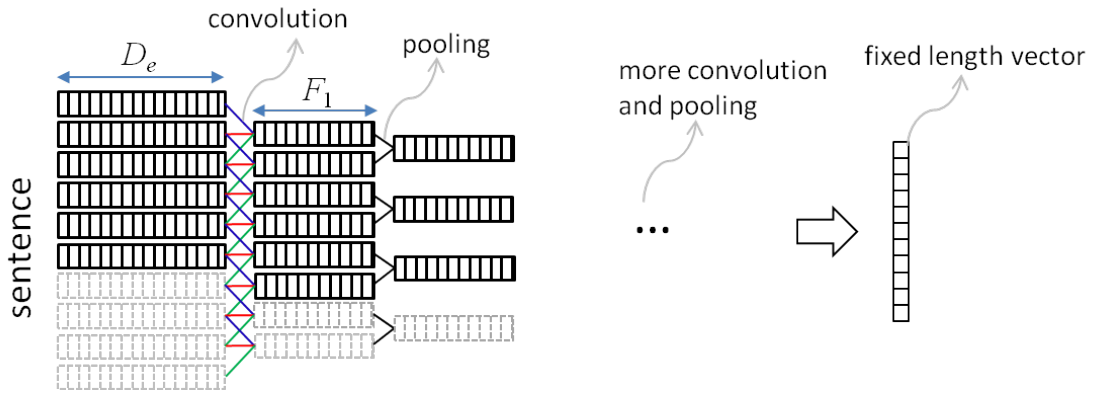


图 3-1 基于深度卷积神经网络的语句表示模型的整体架构图

Fig.3-1 The overall architecture of the Deep Convolutional Neural Network Sentence Model

的输入来说，图像像素中的每个点都有其具体物理意义，点与点之间具有相对的独立性。然而，语句通过将词向量顺序排列得到的矩阵中的每个点的值没有其具体明确的物理意义。如图3-1所示，语句输入矩阵的每一行的多个点的值在一起才有明确的物理意义，其代表语句中对应的一个词。因此，与图像上的卷积操作不同，我们选择的卷积操作是在每一层相邻的若干行向量上进行，其等价于对相邻的词通过卷积操作进行组合，以得到其组合的表示向量。假设第一层的卷积操作在大小为 k_1 的滑动窗口上进行，更深的卷积层的计算与第一层类似。一般地，给定语句输入 $\mathbf{x}$，卷积操作在第 l 层上生成第 f 个特征映射（该类特征是整个 F_ℓ 特征层的一部分）的计算见公式3-1：

$$z_i^{(\ell,f)} \stackrel{\text{def}}{=} z_i^{(\ell,f)}(\mathbf{x}) = \sigma(\mathbf{w}^{(\ell,f)}\hat{\mathbf{z}}_i^{(\ell-1)} + b^{(\ell,f)}),\quad f = 1,2,\cdots,F_\ell \tag{3-1}$$

式中 $z_i^{(\ell,f)}(\mathbf{x})$———第 ℓ 层位置 i 的第 f 个类型的特征表示；

$\mathbf{w}^{(\ell,f)}$———f 在第 ℓ 层的参数，该层参数的矩阵表示为：$\mathbf{W}^{(\ell)} \stackrel{\text{def}}{=} [\mathbf{w}^{(\ell,1)},\cdots,\mathbf{w}^{(\ell,F_\ell)}]$；

$\sigma(\cdot)$———非线性激活函数（例如，Sigmoid 或 Relu [70]）；

$\hat{\mathbf{z}}_i^{(\ell-1)}$———第 $\ell-1$ 层上位置 i 的待卷积块。

$\hat{\mathbf{z}}_i^{(\ell-1)}$ 的计算以 $\hat{\mathbf{z}}_i^{(0)}$ 为例，其计算公式见3-2，其意义为在给定的语句 $\mathbf{x}$ 上将 k_1（滑动窗口的大小）个词的向量连接成一个向量表示，其他层的操作与之类似。其矩阵形式为 $\mathbf{z}_i^{(\ell)} \stackrel{\text{def}}{=} \mathbf{z}_i^{(\ell)}(\mathbf{x}) = \sigma(\mathbf{W}^{(\ell)}\hat{\mathbf{z}}_i^{(\ell-1)} + b^{(\ell)})$

$$\hat{\mathbf{z}}_i^{(0)} = \mathbf{x}_{i:i+k_1-1} \stackrel{\text{def}}{=} [\mathbf{x}_i^\top,\ \mathbf{x}_{i+1}^\top,\ \cdots,\ \mathbf{x}_{i+k_1-1}^\top]^\top \tag{3-2}$$

最大池化操作 在传统的处理图像的卷积神经网络中，当完成一层的卷积操作后，其紧接的是最大池化操作。最大池化操作在卷积层输出的特征映射上窗口大小为 $m \times n$ 的区域上进行。与图像上的最大池化操作不同，语句上的

最大池化操作是对相邻的两个滑动窗口通过卷积操作得到的不同组合的向量 $z_{2i-1}^{(\ell-1)}$ 和 $z_{2i}^{(\ell-1)}$，在第 f 个特征映射上进行局部最大池化操作，其公式见3-3：

$$z_i^{(\ell,f)} = \max(z_{2i-1}^{(\ell-1,f)}, z_{2i}^{(\ell-1,f)}), \ \ \ell = 2, 4, \cdots . \tag{3-3}$$

通过以上的局部最大池化操作，可以起到以下两方面作用：1）可以将卷积操作输出的表示向量的数量减少一半，这样可以快速地消除语句长度的差异性给语句表示学习带来的影响；2）可以将语句中不理想的词与词的组合通过最大池化操作过滤掉，从而保留较为合理的词的组合表示。

语句长度不一致性解决方法 通过以上对基于深度卷积神经网络的语句表示模型中的两类主要操作的描述，可以发现该模型需要输入的语句长度一致。然而在实际情况中，语句的长度差异性较大。解决语句长度不一致问题较为直接的方式是使用“占位符”将给定的语句从末尾补全到预先设定的语句最大长度。但是这样的方式对于一些较短的语句的过多的填充，会给语句的表示带来大量噪音。基于此，本文提出通过以下方式解决语句长度的不一致问题。首先，我们对语句的长度预设一个最大值。对于长度小于最大值的语句，在语句末尾添加“占位符”直到语句最大长度为止。“占位符”的词向量为所有元素为“0”值的向量。其次，为了消除“占位符”对语句表示带来的影响，在卷积操作的基础上，我们设置了“门函数”的操作，也即，当进入卷积运算的滑动窗口内的词都是“占位符”时，将卷积操作的输出强置为全“0”向量，否则，不对卷积操作的输出做任何处理。给定语句 $\mathbf{x}$，第 ℓ 层上，位置 i 的第 f 个特征层的输出加上“门函数”操作后，其数学描述为公式见3-4。

$$z_i^{(\ell,f)} \stackrel{\text{def}}{=} z_i^{(\ell,f)}(\mathbf{x}) = g(\hat{\mathbf{z}}_i^{(\ell-1)}) \cdot \sigma(\mathbf{w}^{(\ell,f)}\hat{\mathbf{z}}_i^{(\ell-1)} + b^{(\ell,f)}), \tag{3-4}$$

式中 $g(\mathbf{v})$——“门函数” $g(\mathbf{v}) = 0$ 如果向量 $\mathbf{v}$ 的所有元素都为 0，否则 $g(\mathbf{v}) = 1$；

考虑到模型的激活函数的输出值大于“0”，“门函数”与局部最大池化操作共同作用，可以有效的保证在每次最大池化操作时，将语句中词的信息保留，而将占位符部分的卷积输出过滤掉。每经过一次最大池化层，“占位符”部分的比例都降低 1/2 直到占位符全部被过滤掉为止。

3.3 基于深度卷积神经网络语句表示模型的分析

我们通过具体实例分析基于深度卷积神经网络语句表示模型的建模能力。模型中的卷积操作是在语句的滑动窗口上进行的，该卷积操作通过对滑动窗

口内的词做非线性组合（Composition）得到其表示。理想情况下，如果滑动窗口内的词是短语或者常用组合，通过卷积操作得到表示较为理想（组合向量内的各个元素的值较大），反之窗口内的组合表示较差（组合向量内的各个元素值较小）。模型中的最大池化层由于对相邻的两个滑动窗口的组合表示在各个维度上进行对比，然后保留较大的值。因此，其作用类似于对相邻两个滑动窗口的组合进行选择。通过卷积操作与局部最大池化操作的一起配合，深度卷积神经网络语句表示模型等价为带有局部选择功能的“组合算子”，模型的层次越深，模型得到的表示输出能够覆盖的语句内的词的范围越广，最后通过多层的运算得到语句的固定维度的向量。从上述描述可以看出，该过程的功能与“递归自动编码”的递归机制，具有一定的功能类似[27]。如图3-2所示，给定语句“The cat sat on the mat”，如果我们选择卷积操作的滑动窗口大小为3，通过一层卷积操作后得到的卷积输出可能覆盖的语句中的组合情况如图3-2中所示。为了更清楚的说明模型的能力，这里给出一种可能的组合的示例，图中灰色部分表示不太理想的组合。对所有可能的组合表示中，有些组合比较理想，例如，在第一个卷积窗口“the cat sat”中，由于“the cat”是一个名词性短语，其组合表示相对较为理想。图中的灰色框图表示组合的置信度较低，白色表示置信度较高。由于卷积操作是在滑动窗口上进行的，所以相邻的两个窗口存在较大的部分交叉信息，例如，第二个滑动窗口与第一个滑动窗口都包含了单词“cat sat”。在第二个滑动窗口中，由于“sat on”是较为常见的短语，因此窗口的表示向量中，“sat on”的置信度较高。由于在我们的模型结构中，采用了相互不重叠的窗口为2的局部最大池化操作，通过在两个窗口上的卷积操作的输出的每一维度上选择较大的值，将语句中较为理想的组合表示保留，将较差的组合表示过滤掉。通过这样多层交叠的卷积和最大池化操作，可以得到语句中不同层次的组合表示，最终得到整个语句的固定维度的表示向量。

从以上的分析可以看出通过大规模数据集训练，该模型具有递归神经网络语句表示模型类似的特性，但是基于深度卷积神经网络的语句表示模型与递归神经网络[71]和递归自动编码[27]具有以下几个重要的不同。首先，递归自动编码模型[27]需要通过句法分析树预先给出语句的结构，然后在给定的结构上对语句中的相邻词进行组合得到其表示，然后通过同样的参数，在句法分析树上递归得到整个语句的表示。这种方式需要依赖于句法分析树，因此句法分析的性能好坏是语句表示质量的关键。但是由于对于不太规范的文本，句法分析的质量往往不太理想，从而限制了该模型的使用范围。另一类递归自动编码模型则通过无监督的方式对语句中所有相邻的词都计算得到组合。然后，通过模型

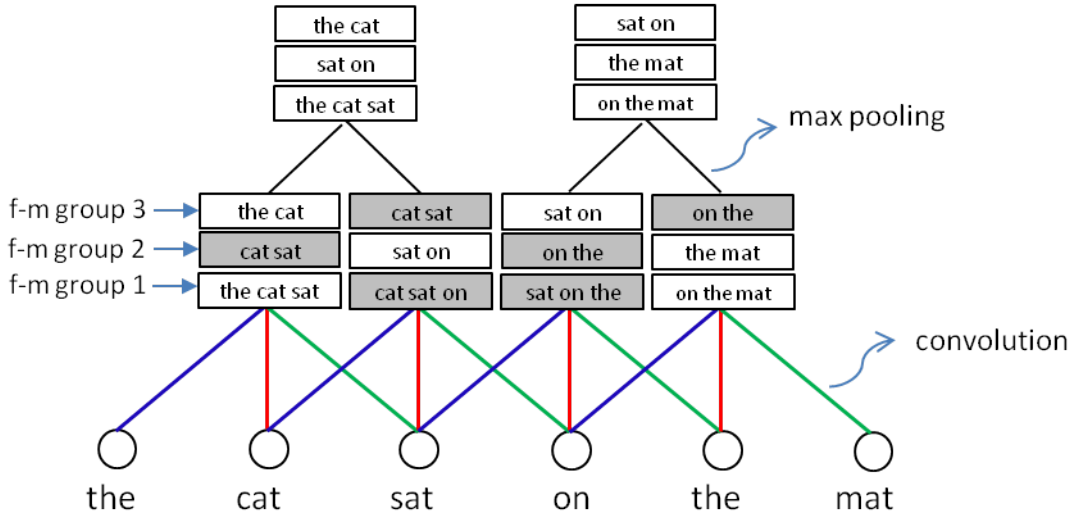


图 3-2 基于深度卷积神经网络语句表示模型的能力示意图

Fig.3-2 The illustration of the Deep Convolutional Neural Network Sentence Model

生成的置信度选择最佳的组合，由于该方式需要对所有可能的组合都计算得到表示，并且计算所有组合的置信度，这种方式的计算复杂度相对较高。本文提出的基于深度卷积神经网络的语句表示模型不需要预先定义的单一路径、单独的选择函数以及外部的句法分析器。该模型一方面使用较高维度的卷积输出，使得窗口内的卷积表示包含窗口中的多种组合情况，另一方面，模型使用局部最大池化机制将较为合理的组合表示保留下来。在任意宽度大于 3 的滑动窗口上进行卷积操作，该模型得到的词组合比递归自动编码机制更为丰富。该模型对于文本不规范的数据，可以通过具体任务的有监督信号，学习得到对特定任务比较重要的局部组合方式，并通过最大池化层将其保留到语句的最后表示向量中。其次，本章提出的基于深度卷积神经网络的语句表示模型可以针对具体的任务，动态的调整模型参数，这种特性对于本章研究的有监督的语句匹配问题尤其重要。尽管该模型相比于递归神经网络模型有这些优点，这种模型同时也有其局限性。基于深度卷积神经网络语句表示模型需要预先设定的固定深度，并且对语句长度需要预先设置一个最大的长度。因此相比递归神经网络模型，对不同长度的语句的处理方式显得没有那么自然。但对于类似于语句匹配的自然语言处理任务，这种局限性可以通过大规模训练数据对模型参数的不断调整并配合“门函数”机制可以得到较大程度的弥补。

Collobert 等人提出通过将词使用词向量表示，将语句表示成词向量矩阵，并对每个词加上其额外的人工特征 [10]，然后通过一层卷积操作对滑动窗口内的词的组合得到表示，最后，通过一层全局最大池化操作得到整个语句

的表示。在得到的语句表示的基础上，通过多层感知机完成了语义角色标注（Semantic Role Labeling）任务，该模型的结构如 3-3所示。由于该模型只使用了一层的卷积操作和一层的全局最大池化操作，因此我们称之为浅层卷积神经网络模型。该模型被广泛应用于自然语言处理中语句级分类任务上，如语句分类 [32]，关系分类 [33] 上。但浅层的卷积神经网络模型，一方面不能对语句中复杂的局部语义关系进行建模，不能对语句中深层次的语义组合进行很好的表示，另一方面全局的最大池化操作丢失了语句中的词序特征。不难看出，本文提出基于深度卷积神经网络语句表示模型，当设置一层卷积层，并将池化层的窗口设置为整个语句长度时，该模型便会退化为浅层的卷积神经网络模型。针对自然语言处理领域的复杂语义匹配任务，浅层的卷积网络模型只能对语句间的局部特征匹配进行建模。不能对语句间复杂多样的匹配关系进行建模。因此，本章接下来介绍的基于深度卷积神经网络的语句匹配架构是以本文提出的基于深度卷积神经网络语句表示模型为基础的。

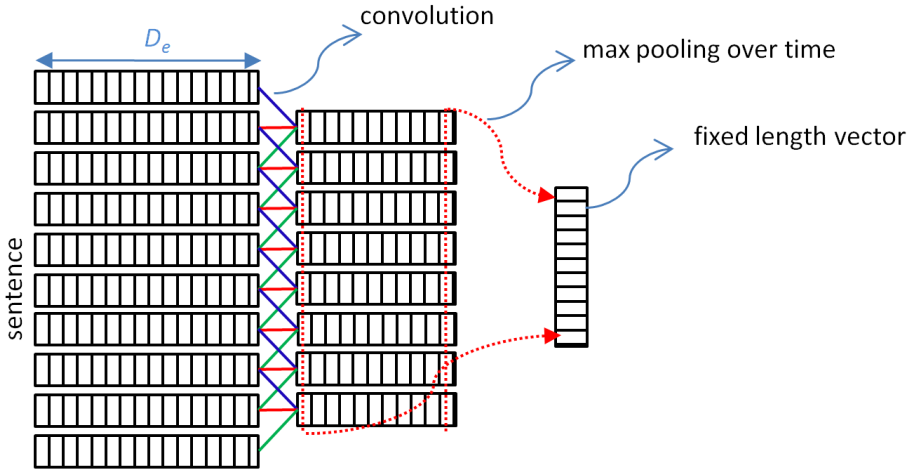


图 3-3 浅层卷积神经网络语句表示模型架构图

Fig.3-3 The overall of the shallow convolutional neural network sentence model

3.4 基于深度卷积神经网络的语句匹配架构

我们分别从两个不同思路针对语句匹配问题，提出了两种基于深度卷积神经网络的语句匹配架构，为了后文方便描述，下文分别称之为架构一（ARC-I）和架构二（ARC-II）。架构一对两个语句分别使用深度卷积神经网络语句表示模型对它们建模，以得到它们固定维度的向量表示。然后，对两个语句向量使用多层感知机进行匹配，该架构的性能受限于深度卷积神经网络语句表示模型对语句学习到的表示的质量。架构二则从输入层开始，就通过深度卷积神经网

络对两个语句的匹配学习多层表示。最后对两个语句的匹配得到一个固定维度的表示向量，然后通过多层感知机对该匹配进行打分。以下分别对这两种匹配架构进行详细介绍。

3.4.1 基于深度卷积神经网络的语句匹配架构一

架构一的基本原理如图3-4所示。该模型与 Siamese 神经网络模型类似。Siamese 神经网络模型最早是由 Bromley 等人于上世纪九十年代为了判断两个签名是否是同一个人所写而提出的[72]。Siamese 网络在底层包含两部分神经网络，分别对输入的数据对的两部分进行表示，最后，在顶层通过使用势能函数（Energy Function）对数据两部分的深层表示进行匹配。

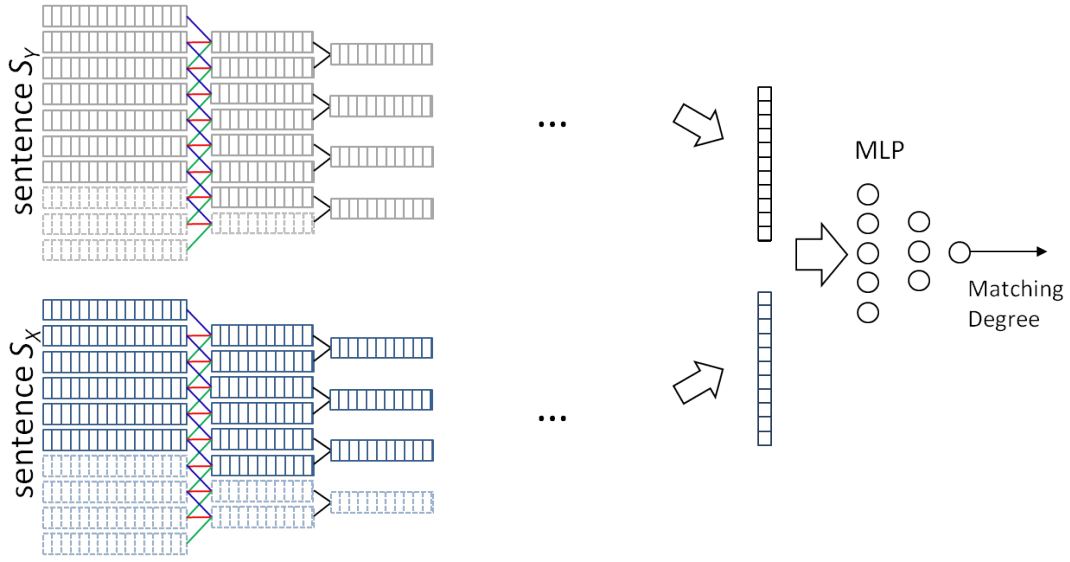


图 3-4 基于卷积深度神经网络的语句匹配架构一的结构图

Fig. 3-4 The overview of the ARC-I for matching two sentences

基于深度卷积神经网络的语句匹配架构一在底层分别包括两个深度卷积神经网络语句表示模型 $DCNN_x(\cdot)$ 和 $DCNN_y(\cdot)$。给定语句对 (S_x, S_y)，

首先对 S_x 和 S_y 分别通过 $DCNN_x(\cdot)$ 和 $DCNN_y(\cdot)$ 得到其表示 h_x 和 h_y。

$$h_x = DCNN_x(S_x) \tag{3-5}$$

$$h_y = DCNN_y(S_y) \tag{3-6}$$

然后，将 S_x 和 S_y 的表示向量 h_x 和 h_y 拼接成为一个向量 h。

$$h = [h_x, h_y] \tag{3-7}$$

之后，将 h 输入给一层的非线性全连接神经网络得到 h_1。

$$h_1 = \sigma(W_1 \cdot h + b_1) \tag{3-8}$$

式中 σ———非线性激活函数（如 Sigmoid）；

W_l———$W_1 \in \mathbb{R}^{m \times n}$;

b_l———$b_1 \in \mathbb{R}^{m \times 1}$;

n———输入向量 h 的维度;

m———输出向量 h_1 的维度;

最后，我们通过线性打分器对语句对 (S_x, S_y) 的匹配程度进行打分，得到其打分值 o，$W_2 \in \mathbb{R}^{1 \times m}$

$$o = W_2 \cdot h_1 \tag{3-9}$$

从该架构的结构可以看出，该架构的特点是，两个语句的表示分别通过两个独立的深度卷积神经网络语句表示模型得到，在得到它们各自的表示之前，两个语句间的信息互不影响。该架构是对两个需要匹配的语句从全局语义上进行匹配，该架构忽略了两个语句间更为精细的局部匹配特征。而在语句匹配的相关问题中，两个待匹配的语句中往往存在丰富的局部匹配，例如，在对话系统中的对话对 (S_x, S_y)，

S_x: “好饿啊，今天去哪里吃饭呢。”

S_y: “听说肯德基最近出了新品，要不要去尝尝呢？”

在这一对话对中，“吃饭”和“肯德基”之间具有较强的相关性匹配关系，而架构一则是对两个语句的全局表示进行匹配，在得到整个语句的表示之前，“吃饭”和“肯德基”之间并不会互相影响，然而，随着深度卷积神经网络语句表示模型对语句的表示层次不断深入，语句中的细节信息会部分丢失，而更关注整个语句的整体语义信息匹配。

3.4.2 基于深度卷积神经网络的语句匹配架构二

考虑到匹配架构一的缺点，本文提出基于深度卷积神经网络的语句匹配架构二。与架构一不同，架构二的思路是对两个语句的匹配学习表示，在模型的不同深度对两个语句间不同粒度的局部之间进行交互匹配，以学习得到语句匹配在不同层次上的表示，最终得到语句对固定维度的匹配表示，并对匹配表示进行打分。架构二的基本结构图如3-5所示。

首先，将语句 S_X 和 S_Y 中的词都转化为其对应的词向量，并按照词的顺序排列。设置 S_X 和 S_Y 的各自最大长度，并对语句使用全“0”向量“占位符”进行补全。在两个语句上分别使用窗口大小为 k_1 和 k_2 的滑动窗口。对 S_X 上的第

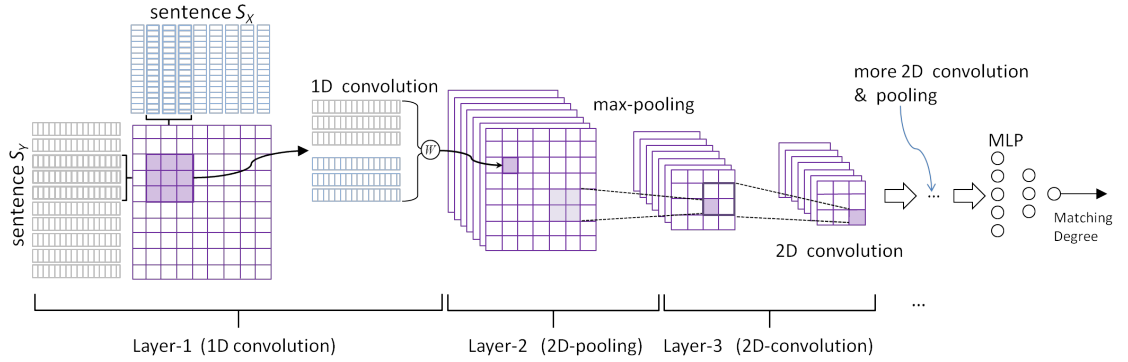


图 3-5 基于卷积深度神经网络的语句匹配架构二的结构图

Fig.3-5 The overview of the ARC-II for matching two sentences

i 个滑动窗口 $\mathbf{x}_{i:i+k_1-1}^{\top}$ 和 S_Y 上的第 j 个滑动窗口 $\mathbf{y}_{j:j+k_2-1}^{\top}$ 拼接为向量 $\hat{\mathbf{z}}_{i,j}^{(0)}$。

$$\hat{\mathbf{z}}_{i,j}^{(0)} = [\mathbf{x}_{i:i+k_1-1}^{\top},\ \mathbf{y}_{j:j+k_1-1}^{\top}]^{\top}. \tag{3-10}$$

因此，$\hat{\mathbf{z}}_{i,j}^{(0)}$ 是维度为 $(k_1+k_2)\times D_e$ 的列向量，其中 D_e 表示词向量的维度。然后对 $\hat{\mathbf{z}}_{i,j}^{(0)}$ 做如公式 3-11的运算，从而得到语句 S_X 的第 i 个窗口与 S_Y 的第 j 个滑动窗口的匹配表示向量。

$$z_{i,j}^{(1,f)} \overset{\text{def}}{=} z_{i,j}^{(1,f)}(\mathbf{x},\mathbf{y}) = g(\hat{\mathbf{z}}_{i,j}^{(0)}) \cdot \sigma(\mathbf{w}^{(1,f)}\hat{\mathbf{z}}_{i,j}^{(0)} + b^{(1,f)}), \tag{3-11}$$

式中 $\mathbf{w}^{(1,f)}$———$\mathbf{w}^{(1,f)} \in \mathbb{R}^{(1\times k_1+k_2)D_e}$；

$\mathbf{b}^{(1,f)}$———$\mathbf{b}^{(1,f)} \in \mathbb{R}^{1\times 1}$；

$g(\cdot)$———为上节描述的“门函数”；

那么，该层的所有参数包括 $\mathbf{W}^{(1)}$ 和 $\mathbf{B}^{(1)}$:

$$\mathbf{W}^{(1)} \overset{\text{def}}{=} [\mathbf{w}^{(1,1)},\cdots,\mathbf{w}^{(1,F_1)}]^{\top} \tag{3-12}$$

$$\mathbf{B}^{(1)} \overset{\text{def}}{=} [\mathbf{b}^{(1,1)},\cdots,\mathbf{b}^{(1,F_1)}]^{\top} \tag{3-13}$$

式中 F_1———S_X 上的第 i 个滑动窗口与 S_Y 上的第 j 个滑动窗口的匹配表示向量的维度；

从以上的计算可以看出，架构二从 S_X 和 S_Y 进入模型之初就对两者既做了表示学习，也对两者的局部之间做了匹配学习，并得到了它们局部之间的匹配表示。通过这种方式使得架构二既能对两个语句的局部之间的匹配关系进行丰富建模，也可以使得模型对每个语句各自内的信息进行建模。很显然，通过以上的计算操作后，得到的表示保持了来自两个语句的滑动窗口的位置信息，同时包含了两个滑动窗口的匹配表示。在接下一层，我们在非重叠的 2×2 的窗口上进行二维局部最大池化操作（参考图3-6），采用二维局部最大池化的目的是综合考虑各个语句的滑动窗口内的词的组合表示是否合理，以及两个滑动窗口

是否存在较强的匹配关系。局部最大池化操作的数学描述见3-14：

$$z_{i,j}^{(2,f)} = \max(\{z_{2i-1,2j-1}^{(1,f)}, z_{2i-1,2j}^{(1,f)}, z_{2i,2j-1}^{(1,f)}, z_{2i,2j}^{(1,f)}\}) \tag{3-14}$$

接着，第三层在池化层的的输出结果上对大小为 $k_3 \times k_3$ 的滑动窗口继续进行二维卷积操作。一方面可以得到两个语句各自的更高层次的表示，另一方面，可以得到它们更高层次表示的匹配向量，使得新得到的匹配向量表示看到的周围词信息更广。其卷积操作运算如公式见3-15：

$$z_{i,j}^{(3,f)} = g(\hat{\mathbf{z}}_{i,j}^{(2)}) \cdot \sigma(\mathbf{W}^{(3,f)}\hat{\mathbf{z}}_{i,j}^{(2)} + b^{(3,f)}) \tag{3-15}$$

式中 M———训练语料的大小；

C———词向量;

θ———除词向量外的其他参数

$\hat{\mathbf{z}}_{i,j}^{(2)}$ 表示在第二层的输出结果上以位置 (i,j) 为中心以 $k_3 \times k_3$ 为窗口大小的区域内的向量按顺序进行拼接得到的向量。通过这样的操作，卷积输入窗口内包含了分别来自 S_X 和 S_Y 的 $k_3 \times k_3$ 个卷积表示，从而对两个语句中各个局部的匹配模式进行了建模，具体表示见公式 3-16：

$$\hat{\mathbf{z}}_{i,j}^{(2)} = [z_{i,j+1}^{(2)}\top, ..., z_{i,j+k_3}^{(2)}\top, ..., z_{i+k_3,j+k_3}^{(2)}\top]\top \tag{3-16}$$

式中 $\mathbf{w}^{(3,f)}$———$\mathbf{w}^{(3,f)} \in \mathbb{R}^{(1\times k_1+k_2)F_1}$；

$\mathbf{b}^{(3,f)}$———$\mathbf{b}^{(3,f)} \in \mathbb{R}^{1\times 1}$；

σ———非线性激活函数；

那么该层的所有参数包括 $\mathbf{W}^{(3)}$ 和 $\mathbf{B}^{(3)}$，F_3 为 S_X 上的第 i 个滑动窗口与 S_Y 上的第 j 个滑动窗口的匹配表示向量的维度。

$$\mathbf{W}^{(3)} \stackrel{\text{def}}{=} [\mathbf{w}^{(3,1)}, \cdots, \mathbf{w}^{(3,F_3)}]^\top \tag{3-17}$$

$$\mathbf{B}^{(3)} \stackrel{\text{def}}{=} [\mathbf{b}^{(3,1)}, \cdots, \mathbf{b}^{(1,F_3)}]^\top \tag{3-18}$$

式中 F_3———S_X 上的第 i 个滑动窗口与 S_Y 上的第 j 个滑动窗口的匹配表示向量的维度；

随后，同样在非重叠的 2×2 的窗口上进行二维局部最大池化操作3-19.

$$z_{i,j}^{(4,f)} = \max(\{z_{2i-1,2j-1}^{(3,f)}, z_{2i-1,2j}^{(3,f)}, z_{2i,2j-1}^{(3,f)}, z_{2i,2j}^{(3,f)}\}) \tag{3-19}$$

类似的，二维卷积操作和二维局部最大池化操作可以依次进行多层，直到对两个语句的匹配得到固定维度为 l 的向量表示 $\mathbf{z}$ 为止。然后，将 $\mathbf{z}$ 输入给一层的非线性全连接神经网络，得到 h_1。

$$h_1 = \sigma(W_1 \cdot h + b_1) \tag{3-20}$$

式中 σ———非线性激活函数；

W_l———$W_l \in \mathbb{R}^{l \times n}$；

b_l———$b_l \in \mathbb{R}^{m \times 1}$；

m———h_1 的维度

最后，我们通过线性打分器，对语句对 (S_x, S_y) 的匹配表示进行打分，得到其打分值 o，$W_2 \in \mathbb{R}^{1 \times m}$

$$o = W_2 \cdot h_1 \tag{3-21}$$

式中 W_2———$W_2 \in \mathbb{R}^{1 \times m}$；

从架构二的卷积结构可以看出，通过第一层对两个语句间的滑动窗口进行直接的卷积匹配操作，得到了两个语句间较为底层的局部匹配表示。之后，通过二维卷积操作以及局部最大池化操作得到两个语句间更高层的匹配表示，其中从第 $\ell - 1$ 层得到 $\mathbf{z}_{i,j}^{(\ell)}$ 的二维卷积操作的一般形式可以表示为3-22。

$$\mathbf{z}_{i,j}^{(\ell)} = g(\hat{\mathbf{z}}_{i,j}^{(\ell-1)}) \cdot \sigma(\mathbf{W}^{(\ell)} \hat{\mathbf{z}}_{i,j}^{(\ell-1)} + \mathbf{b}^{(\ell,f)}), \ \ \ell = 3, 5, \cdots \tag{3-22}$$

这里，$\hat{\mathbf{z}}_{i,j}^{(\ell-1)}$ 表示将 ℓ–1 层滑动窗口中的向量按顺序拼接得到的向量。从架构二的描述可以看出，该模型通过充分考虑两个语句间的局部匹配关系，并通过二维的卷积与二维局部最大池化操作，最终得到两个语句的匹配表示向量。整个过程中，两个语句的内部窗口间相互影响，因此架构二更为关注语句间的匹配关系，可以对两个语句进行更为细致的匹配。同时，架构二的局部最大池化操作与架构一截然不同。架构一的最大池化操作，是对单个语句中连续的若干滑动窗口的卷积表示选择最为合理的组合，最终通过多层的卷积与池化操作得到语句的表示向量。而架构二不仅需要考虑到单个语句中滑动窗口内的词的组合质量，同时对分别来自两个语句的组合间的匹配关系的质量也进行考虑。虽然匹配架构二的局部最大池化操作与 Socher 等提出的在二维相似度矩阵上进行动态最大池化（Dynamic Pooling）[27] 的操作有一定程度的类似，但是它们有两个重要的不同：1）架构二的最大池化操作发生在固定的结构上，因为语句匹配对的输入长度以及模型的整体架构都是固定的；2）架构二上进行最大池化操作，比单纯对两个语句间的局部相似度，动态选择最大值更为复杂。Socher 等人提出的在二维相似度矩阵上进行动态最大池化针对的是复述检测任务，其每个元素点对应的是两个语句间的不同层次上表示的相似度，而本文的匹配架构二并没有定义局部最大池化的值的含义，另外两个语句间的局部匹配关系也不使用简单的相似度标量值表示，而是将它们表示成一个多维向量，其具体含义可以通过具体的任务进行优化，因此可以推广到多种不同性质的匹配任务上。

3.4.3 基于深度卷积神经网络的语句匹配架构二的特点分析

1. 词序保持能力：对于架构一，由于其分别对两个语句在顺序的滑动窗口上进行建模，因此，架构一可以很好的保持两个语句各自的词序信息。架构二是对两个语句的匹配学习表示过，程中存在两个语句的局部信息的交互。但该模型仍然能够保持 S_X 和 S_Y 上的词序信息。对于架构二，卷积和局部最大池化操作都不改变两个语句的局部匹配表示的整体顺序。一般地，虽然由于二维局部最大池化的影响使得 $\mathbf{z}_{i,j}^{(\ell)}$ 和 $\mathbf{z}_{i+1,j}^{(\ell)}$ 可能包含来自 S_Y 上的不同词的信息。但是如图3-6 所示，对于 S_X 上的词，$\mathbf{z}_{i,j}^{(\ell)}$ 比 $\mathbf{z}_{i+1,j}^{(\ell)}$ 包含更靠前位置的词信息。同样，最大池化操作的影响使得 $\mathbf{z}_{i,j}^{(\ell)}$ 和 $\mathbf{z}_{i,j+1}^{(\ell)}$ 可能包含来自 S_X 上的不同的词的信息，但对于 S_Y，$\mathbf{z}_{i,j}^{(\ell)}$ 比 $\mathbf{z}_{i,j+1}^{(\ell)}$ 包含更靠前位置的词信息。相关实验表明，当我们构造训练数据集 $(S_X, S_Y, \tilde{S}_Y)$，其中，$\tilde{S}_Y$ 是将 S_Y 中的词的顺序随机打乱而得到的负例，并在其上训练架构二之后，架构二可以较为稳定的在我们构造的较为困难的（见语句补全任务和微博与回复的匹配任务）数据集上，找出正确的 S_Y。而这种现象在架构一上不存在。这说明架构二通过对语句 S_X 和 S_Y 的匹配进行建模，可以学习到两个语句间的局部匹配模式，并且这种匹配模式需要在正常顺序的语句中才有意义，对于随机无序的语句，架构二则忽略这种局部匹配。因此，架构二不仅可以保持 S_X 和 S_Y 上的词序的信息，而且可以有效利用这些词序信息学习到有意义的匹配模式。

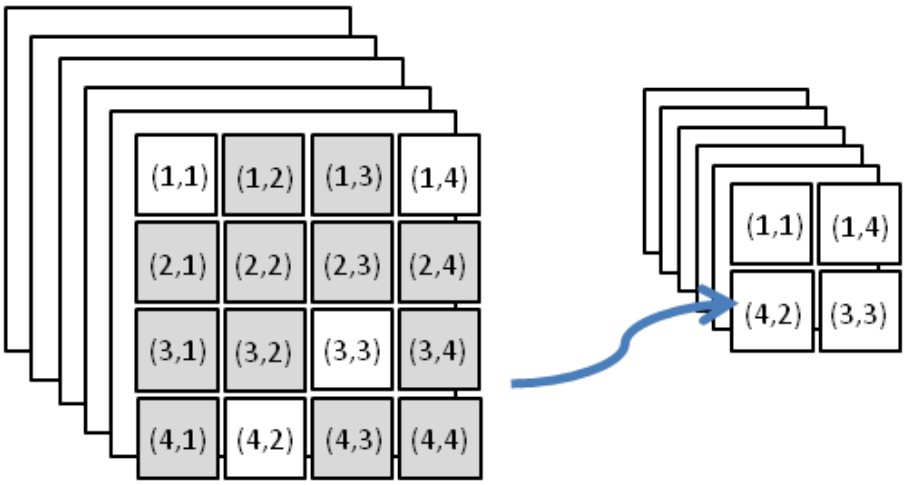


图 3-6 基于卷积神经网络模型的语句匹配架构二词序保持能力的示例图

Fig.3-6 The illustration of word order preserving ability in ARC-II

2. 架构二与架构一之间的关系：虽然架构一与架构二的设计思路以及网络结构完全不同，但是不难发现，在一些特殊情况下，架构二可以退化为架构

一。事实上，在架构二中，保持两个语句的局部卷积操作得到的表示之间不相互影响（通过对卷积参数 $\mathbf{W}^{(\ell)}$ 的部分元素强行置为 0），架构二的操作则与架构一的相同，如图3-7所示。具体地，如果在第一层卷积操作中，我们通过设置卷积参数使得该层部分参数只对来自 S_X 进行计算，部分参数只对 S_Y 的数据进行计算，而不是对它们作为一个整体进行计算。那么第一层的输出就可以截然分成两部分的表示，其中一部分表示来自 S_X，另一部分表示来自 S_Y。这样对每个卷积层上的输出 $\mathbf{z}_{1:n,1:n}^{(1,f)}$（$n$ 表示滑动窗口的数量）的秩为 1，其上的信息对应于架构一分别对每个语句中的滑动窗口的卷积信息相同。同时，二维最大池化操作便退化为一维最大池化操作的相同功能，因为同一列或者同一行的数值完全相同。同样，如果对第二层的卷积操作的参数（$\mathbf{w}^{(2,f)}$）进一步做特殊设置，我们可以继续使得两个语句更高层的表示完全分开，其表现能力与匹配架构一完全一样。但匹配架构二的时间复杂度比匹配架构一的时间复杂度要高很多。因为这种情况下，匹配架构二做了大量的重复性计算。

通过对架构二保持词序的特性及其与架构一的关系进行分析可以看出，尽管架构二是对两个语句的匹配进行建模，并学习提取两个语句间的局部匹配模式。但是架构二具有对两个语句单独抽象表示的能力。因此架构二可以将以下两种过程进行充分融合：1）单个语句中的连续的词与词的组合得到更高层次的表示向量，2）抽取两个语句间的多层的局部匹配模式并进行融合得到更高层次的匹配模式。而架构一只有第一个过程，最后对两个语句从全局表示上进行匹配，对两个语句间的局部匹配模式的学习能力显然不足。因此，架构二具有对语句对之间丰富的结构进行匹配并挖掘的能力。这一点也得到了实验结果的验证。

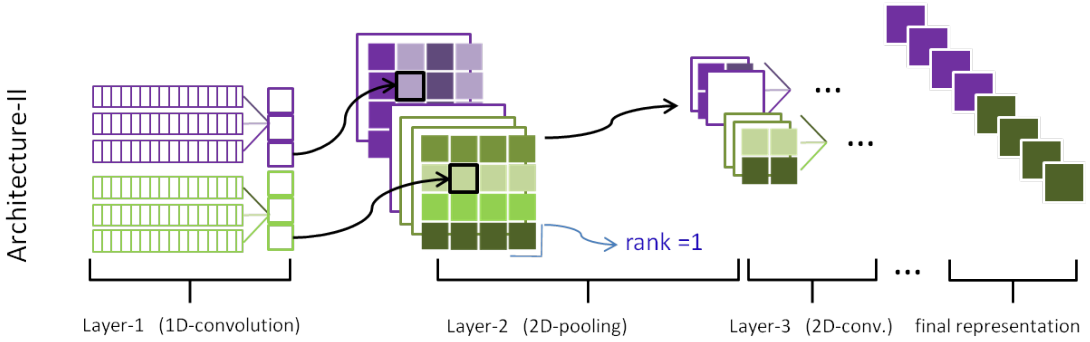


图 3-7 深度卷积神经网络语句匹配架构一是架构二的一种特殊情况的示例图

Fig.3-7 Arc-I as a special case of Arc-II

3.5 基于深度卷积神经网络的语句匹配架构的训练

由于在实验部分，任务一（语句补全）和任务二（微博与评论匹配）是对给定的语句对进行排序，因此，我们采用成对输入（Pairwise）的带有差距（margin）的学习排序（Learn to rank）训练策略。具体地，每次迭代需要输入给模型一个三元组 $(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-)$，$\mathbf{x}$ 表示一个语句，$\mathbf{y}^+$ 表示对 $\mathbf{x}$ 的一个正确匹配的语句，$\mathbf{y}^-$ 表示对语句 $\mathbf{x}$ 的一个错误匹配语句（也即负例）。对每次迭代的前向计算，首先输入给匹配模型 $(\mathbf{x}, \mathbf{y}^+)$ 计算出其匹配分数 $\mathsf{s}(\mathbf{x}, \mathbf{y}^+)$，然后，输入给模型 $(\mathbf{x}, \mathbf{y}^-)$ 计算出其匹配分数 $\mathsf{s}(\mathbf{x}, \mathbf{y}^-)$，最后给出其基于排序的损失函数 $e(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-; \Theta)$ 见公式 3-23：。

$$e(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-; \Theta) = \max(0, \alpha + \mathsf{s}(\mathbf{x}, \mathbf{y}^-) - \mathsf{s}(\mathbf{x}, \mathbf{y}^+)), \tag{3-23}$$

式中 Θ———卷积架构与多层感知机的模型参数；

α———正确的匹配对 $\mathsf{s}(\mathbf{x}, \mathbf{y}^+)$ 的得分比错误的匹配对得分 $\mathsf{s}(\mathbf{x}, \mathbf{y}^-)$ 至少大的差值，其在实验部分被设置为 1;

对于实验部分的任务三（复述检测），由于该任务是判断给定的语句对 $(\mathbf{x}, \mathbf{y})$ 是否是复述关系，因此，该任务是典型的分类任务其类别是“1”或者“0”。因此，对于给定的语句对 $(\mathbf{x}, \mathbf{y})$，其输出的匹配分数为 $\mathsf{s}(\mathbf{x}, \mathbf{y})$。然后，在其上使用逻辑回归分类器（logistic regression），那么其类别“1”的概率见公式 3-24：

$$P((\mathbf{x}, \mathbf{y})) = \frac{1}{1 + e^{-\mathsf{s}(\mathbf{x}, \mathbf{y})}} \tag{3-24}$$

对于给定的语句对 $(\mathbf{x}, \mathbf{y})$ 的损失函数见公式 3-25:

$$J(\theta, \mathbf{x}, \mathbf{y}, \ell) = -(\ell log P((\mathbf{x}, \mathbf{y})) + (1 - \ell) log(1 - P((\mathbf{x}, \mathbf{y})))) \tag{3-25}$$

这里 ℓ 表示语句对 $(\mathbf{x}, \mathbf{y})$ 是否是复述关系，取值为“0”或“1”。对于架构一和架构二的参数更新，采用了常见的基于随机批处理的后向传播算法。通过关掉来自于“占位符”的卷积输出，“门函数”可以非常容易的引入到梯度传播过程中。所有模型的单个批量的数据大小在 100-200 之间时达到最好的性能。模型可以非常容易地在具有多核的单个机器上进行并行化。对于模型的正则化，通过实验发现对于架构一和架构二由于训练数据集足够大，模型在训练集上可以有效避免过拟合，因此对于任务一和任务二，我选择在整个训练集上的目标函数使用早期停止技术（Early Stoping）[73] 确定模型的终止。而对于较小的数据集（数据规模小于 10k），如实验部分的任务三，本文综合使用早期停止技术和 Dropout[74] 避免可能出现的过拟合问题。Dropout 是 Hintion 提出的在训练集

较小的情况下，防止模型过拟合的技术，所谓 Dropout 指在模型训练时，随机让网络中某些隐藏层节点的权重不工作，也即按照一定的概率 p 将该层网络的输出结点强行置为“0”，这样在后向传播时，网络被强行置为“0”的输出结点对应的参数不更新。在测试阶段，模型的参数设置为 $(1-p)W$。

3.6 实验

为了验证本章提出的基于深度卷积神经网络的语句匹配架构一和架构二的性能。本文将两种匹配架构在三个语句匹配任务上进行试验。并与相关的匹配模型作对比。这三种任务分别是英文语句补全任务、中文微博与回复之间的匹配和英文复述检测任务。三种任务分别具有不同的特点。其中，在英文语句补全和中文微博与回复之间的匹配任务中，语句对之间的意义是不相同的，语句间的匹配关系是从一个语义空间到另一个语义空间的关系匹配。因此，在这两个任务上，可以考察模型在非同质的语义匹配任务上的性能。复述检测任务是考察两个语句是否是表达同一种语义，因此，该任务可以检验模型在同质语义匹配任务上的性能。另外，任务二与任务一和任务三的区别是语言类型不同，可以考察模型对不同语言的适应能力。

3.6.1 实验设置

试验中，词向量通过 Word2vec[11] 训练得到，维度为 50。英文的词向量（语句补全任务和复述检测任务）是在文本规模为 10 亿词左右的维基百科数据集上学习得到的。中文词向量（微博与回复匹配任务）在文本规模为 3 亿词的中文微博数据集上学习得到的。最大语句长度的选择根据不同的任务设置不同。同时对于所有的实验，模型的卷积层的滑动窗口大小选择为 3，对卷积输出层的维度（范围为 200-500 之间）进行优化选择。架构二的结构设置一共八层，3 层卷积层，3 层最大采样层，2 层全连接层，架构一在较浅的网络结构，较大的隐藏层结点数时，表现的性能相对较好（两层卷积层，两层最大采样层，两层全连接隐藏层）。所有模型的激活函数选择 ReLu，其性能表现与 Sigmoid 函数相当，但收敛速度更快。需要注意的是，为了模型的效率，这里没有更新词向量的参数。相关的实验表明，进一步更新匹配架构一和匹配架构二的词向量，其性能会有进一步的提升，但是其训练时间需要更长。

为了与架构一和架构二进行实验对比，本文选择与以下模型进行对比:

- WordEmbed: 该模型的基本架构是首先对语句对中的所有词的词向量相

加得到每个语句的表示，然后将得到的两个语句的表示输入给具有多个隐藏层的全连接神经网络，最后对两个匹配进行打分。

• DeepMatch: Lu 等提出的短文本匹配模型[75]，该模型具有三个隐藏层，我们选择第一个隐藏成的结点个数为 1000，在相关任务上进行训练。

• uRAE+MLP: 首先通过展开的循环自动标码 (unfolding Recursive Autoencoder)[28] 对每个语句得到 100 维的向量表示，然后通过多层感知机对语句对进行匹配打分。

• SENNA+MLP:该模型采用 SENNA 类型的语句模型对语句进行建模。

• SenMLP: 将语句中所有的词的词向量拼接起来得到一个向量，将其视作语句的向量表示，然后通过多层感知机对语句对进行匹配打分。

3.6.2 评价指标

由于语句补全任务和微博与回复匹配任务是学习排序任务，本文使用 $P@1$ 指标评价算法性能，其计算见公式 3-26：

$$P@1 = \frac{Count_{top1}(TestSet)}{Count(TestSet)} \tag{3-26}$$

式中 $Count_{top1}(TestSet)$———系统标注的结果中排序最高的匹配是最好的匹配的数量；

$Count(TestSet)$———测试集中的序列个数;

对于复述检测任务我们使用的评价指标是复述关系数据的 F_1 值评价算法性能。其计算见公式 3-27：

$$F = \frac{2PR}{(R+P)} \tag{3-27}$$

式中 P———算法标注的结果中复述关系的数据的准确率；

R———算法标注的结果中复述关系的数据的召回率;

3.6.3 语句补全任务

语句补全任务是本章设计的一种匹配任务，该任务的设计动机，是为了阐释各个模型对同一个语句中的不同部分的语义匹配的建模能力。数据集的构建方法如下，首先，在路透社 RCV1 数据集上抽取满足如下要求的长句：语句由两个以逗号分开的平衡半句组成。每个半句的长度在 8-28 之间。这样将一个语句的前半句看做是匹配问题中的 S_X，后半句是 S_Y。该问题的目的是给定 S_X

和其多个候选后半句，从中选择最好的 S_Y 对语句进行补全。该匹配任务可以被看作是英语语言的异质匹配任务，因为要匹配的语句前半句和后半句在词汇和语义上是不对称的，前半句和后半句通常没有语义相似的关系。为了使得问题更为困难以检验各个模型的潜力，在构造数据集时，我们使用语句中的词向量加和求平均后的向量表示语句，对于一个语句的真实后半句，我们从大规模数据中选择与后半句的余弦相似度在 0.7~0.8 之间的半句作为负例。数据集样例如下：

S_X: *Although the state has only four votes in the Electoral College,*

S_Y^+: *its loss would be a symbolic blow to republican presidential candidate Bob Dole.*

S_Y^-: *but it failed to garner enough votes to override an expected veto by*

我们构造了一个具有 3 百万规模的三元组训练集，所有的模型在该数据上进行训练，其中，正例为 60 万个匹配对，对每个正例生成 5 个负例。测试数据集包含 5 万个正例，每个正例对应 4 个负例，模型需要将它们一起进行排序，并将正例排在最前面。

表 3-1 语句补全任务的性能

Table3-1 Performance of Sentence Completion

Model	P@1(%)
Random Guess	20.00
DeepMatch	32.50
WordEmbed	37.63
SenMLP	36.14
SENNA+MLP	41.56
uRAE+MLP	25.76
Arc-I	47.51
Arc-II	**49.62**

实验结果如表3-1所示。从表中可以看出，本章提出的两种匹配架构的正确率接近百分之五十，同时两种架构显著超过了所有对比模型的性能。架构二显著了超过了架构一，从而说明了架构二通过对语句匹配进行建模的优势。从结果中还可以看出，由于 SENNA+MLP 的模型较浅，其不能充分的对语句进行表示，与本文提出的架构一和架构二的性能有较大差距，但是其表现相对较好，说明卷积与最大池化操作能够有效的学习到两个语句间的基本匹配关系。uRAE 表现较差可能有两部分原因导致：1）uRAE 的模型对语句进行表示时的词向量，并不是从路透社语料 RCV1 上进行训练的，由于数据的不一致性导致

对语句表示的质量降低，2）uRAE 对语句进行表示时需要首先对语句进行句法分析，然而本任务的 S_Y 是一个语句的半句而不是整个语句，数据集的这种特性一定程度上降低了句法分析的性能，从而导致 uRAE 对语句的表示质量较差。

3.6.4 微博与回复匹配任务

该任务的数据集来源于中文微博社交平台新浪微博。其训练数据集包含 450 万的原始的（微博，回复）对。与语句补全数据集不同，微博数据的语句表达更为随意和不规范。对任意一条微博，我们通过对 450 万条回复中，随机选取的回复作为该条微博的负例。在训练中，对每条微博，我们产生 10 条负例，这样我们的所有模型在 4500 万的三元组上进行训练。以下是其中之一的一个三元组的例子，其中 S_X 表示微博，S_Y^+ 表示 S_X 对应的正确回复，S_Y^- 表示通过在所有评论中随机产生的负例。

S_X: 好饿啊，今天去哪里吃饭呢。

S_Y^+: 听说肯德基最近出了新品，要不要去尝尝呢？

S_Y^-: 修理费太贵还是另造合算。

测试集则由 30 万对的原始微博与回复组成，测试时，对每条微博随机产生 4 个负例。也即，在测试集上，我们测试模型在 5 个回复中，找出最为合适的语句作为对应微博回复的能力。即使该任务中的负例的选择是通过随机产生的，但该任务也并不容易。原因是其一，微博的回复相对较为随意，其是一种较为松散的匹配关系，其二，由于微博的书写以及表达不规范为语句的表示建模带来了一定的难度。

表 3-2 微博与回复任务的实验结果

Table 3-2 The result of Tweet Matching

Model	P@1(%)
Random Guess	20.00
DeepMatch	49.85
WordEmbed	54,31
SenMLP	52.22
SENNA+MLP	56.48
Arc-I	59.18
Arc-II	**61.95**

表3-2列出了各个模型在该任务上的性能。从表中可以看出，基于深度卷积神经网络的语句匹配架构可以对语句进行较好的表示，本文提出的架构一和架构二，显著地超过了其他模型的性能。同时，微博与回复的匹配需要对局部的细节匹配充分的建模，例如，上面例子中“吃饭”和“肯德基”就有很好的匹配关系，由于架构二对语句间的局部匹配做了充分建模，架构二以较大的差距超过了架构一。

3.6.5 复述检测任务

复述检测是自然语言处理领域的重要任务，其可以被广泛应用于问答系统，信息检索等应用中。所谓复述检测是判断给定的两个语句是否表达了同一个意义，该任务一直被认为是一种较为困难的语义匹配问题。本文将所有模型在复述检测任务上进行检验，以验证该模型在同质性的语义匹配任务上的性能。这里采用的是较为常用的标准复述检测数据集 MSRP [76]。该数据集包含 4076 个人工标注的语句（0 或 1）对作为训练集，1725 个数据对作为测试集。将训练集中的所有数据对（S_X，S_Y）置换顺序输入后得到的（S_Y，S_X）加入到模型的训练过程中作为对训练集的扩展。由于该数据集的规模较小，对于深度模型来说远远不够，为了防止在训练过程中出现的严重过拟合问题，本实验在每一个卷积层的输出上使用了 Dropout[74]，也即通过对一定比例结点随机置为 0 防止过拟合。

表 3-3 复述检测任务的性能

Table3-3 The results on Paraphrase

Model	Acc. (%)	F1(%)
Baseline	66.50	79.90
Rus et al. (2008) [76]	70.60	80.50
Socher et al.(2011) [28]	76.80	83.60
WordEmbed	68.70	80.49
SENNA+MLP	68.40	79.70
SenMLP	68.40	79.50
Arc-I	69.60	80.27
Arc-II	69.90	80.91

表3-3列出了所有模型的相关性能。通过结果可以看出，架构一和架构二虽然在性能上超过了 WordEmbed 和 SenMLP 相对简单的语句表示模型，但其离最

好水平的性能还有较大差距。架构一和架构二都是完全有监督的深度模型，而且模型结构复杂，参数庞大。仅仅 4076 个大小的训练数据集，不能对架构一和架构二进行充足的训练。然而即使在数据严重不足的情况下，表3-3显示架构二的水平依然达到了 Rus [76] 等基于人工特征的系统性能。该结果证明了架构一和架构二在同质的语义匹配任务上的有效性。

从以上三种匹配任务的实验结果可以看出架构二在数据集较为充分（语句补全和微博与回复的匹配任务）的情况下，可以充分发挥其优越性。当两个语句之间的匹配关系对它们之间的局部匹配依赖性不强时，架构二对架构一的优越性相对减弱。如与微博与回复的匹配相比，语句补全任务的局部匹配关系相对较弱。因此，架构二在微博与回复匹配的任务上的优越性，表现的更为明显。从实验结果我们还可以发现，卷积模型（包括架构一，架构二和 SENNA+MLP）的性能超过了 WordEmbed 和 SenMLP 结果，证实了卷积操作可以有效的将语句中词与词的组合进行表示，其性能显著优于其它模型。

3.7 本章小结

本章在词向量的基础上，提出一种基于深度卷积神经网络的语句表示模型，该模型不依赖句法分析、预先定义的语句结构，而是通过多层的卷积操作对语句中相邻的词进行组合，通过局部最大池化操作选出最为合理的组合方式。其次，我们针对自然语言处理领域广泛存在的语句级的语义匹配问题，提出了两种基于深度卷积神经网络的语句匹配架构。架构一从一种较为直接的方式出发分别对两个语句使用深度卷积神经网络语句表示模型进行建模，然后通过多层感知机对它们的语句向量进行匹配。架构二则从模型输入层就对两个语句间的匹配进行建模，通过深度卷积神经网络得到它们的匹配表示，然后计算它们的匹配分数。最后，我们通过实验设计，在多种类型的任务上，对两种匹配架构进行了验证。分析与实验表明我们提出的两种基于深度卷积神经网络的语句匹配架构显著地优于其它对比模型，特别是架构二，由于其直接可以对两个语句的匹配进行建模，能够对语句间细粒度的语义差异以及匹配模式有效捕捉，其性能达到了最优。

第 4 章 上下文依赖的卷积神经网络短语匹配模型

4.1 引言

机器翻译是自然语言处理领域重要的研究问题，使机器能够自动将一种语言的句子转换成流畅、意义对等的目标语言句子，被认为是极具挑战性的任务。机器翻译领域的研究工作大多集中在基于统计的方法，也称之为统计机器翻译技术。统计机器翻译（Statistical Machine Translation，SMT）需要在大规模的平行语料上，通过统计分析构建源端语言与目标端语言的概率翻译模型，然后通过翻译模型进行翻译[77]。构建一个统计机器翻译系统是一个复杂的工程，其主要包括三个阶段，建模、训练和解码。其中，建模对系统的性能起着极其重要的作用。翻译模型的构建一般分为两个阶段[78]。首先，通过对平行语料进行词对齐后，抽取双语短语对。其次，通过统计每个短语对在平行语料上出现的频率，对每个短语对分配一个置信度分数。在翻译的过程中，需要利用短语对的置信度分数，在给定源端短语的情况下，找出置信度最高的目标端短语，很显然这是一个典型的语言匹配问题。理想的目标端短语不仅需要在语义上与源端短语一致，同时在语法上要符合目标语言的习惯。仅仅依靠双语短语对的统计信息对短语对进行选择，很难找到语法和语义上相近的双语短语对。为了解决上述问题，相关工作提出通过在连续空间中学习源端短语与目标端短语的表示，然后利用源端短语与目标端短语的表示的语义相似度特征，提高统计机器翻译系统的性能[41, 79, 80]。这类方法的基本假设是语义相似的双语短语对，在连续空间中应该有相近的向量表示。源端短语与目标端短语的匹配得分，通过计算它们在连续空间中的向量相似度得到，最后，将它们的匹配得分加入到传统机器翻译系统中提升翻译性能。

现有方法在计算双语短语对的匹配得分时，大多忽略了它们的上下文信息。也即对于一个源端短语得到其在连续空间上的表示后，无论其出现在何种上下文环境中，其表示都是固定的，同样对于目标端短语来说其表示也是固定的。这种做法是非常不合理的，图4-1描述了短语“错误”的目标端短语选择的例子，图中“Src”表示源端句子，“Ref”表示“错误”对应的理想的目标翻译，最后一行表示忽略源端短语所在的上下文时，通过频率对“错误”的所有候选翻译进行的排序。在中文到英文的翻译问题中，如果忽略短语“错误”的上下文，其可以有多种翻译候选。那么，统计机器翻译系统会根据它们的频率

Src: 伊拉克 拥有 大 杀伤力 武器 的 **错误** 情报
Ref: incorrect, faulty, wrong, erroneous

Src: 在 确定 关塔那摩 囚犯 的 身份 方面 犯了 **错误**
Ref: a mistake, mistakes

错误：wrong (1143), mistakes (361), mistake (314)

图 4-1 短语对选择的例子

Fig.4-1 A example of phrase pair selection

大小进行排序选择，其顺序为“wrong, mistakes, mistake”。根据它的上下文信息得知，“错误”在不同的上下文语境中的正确翻译截然不同。在第一个句子中，“错误”显然是作为形容词使用的，其正确翻译应该为“wrong”。而在第二个句子中，“错误”是被用作名词使用，其正确翻译应为“mistake”。现有方法由于忽略了源端短语的上下文，给出的源端短语的翻译候选只能反应出源端短语的基本语义信息，不能对源端短语进行深层理解以得到其正确匹配的目标端短语。相关研究表明，短语的上下文信息对翻译候选进行消歧起着重要作用[81, 82]。

本章提出一种上下文依赖的卷积神经网络短语匹配模型 (Context-Dependent Convolutional Neural Network Phrases Matching Model,CDCM)，该模型是上一章提出的基于深度卷积神经网络语句匹配架构一 (ARC-I) 的演进形式，这里没有采取上一章的提出的匹配架构二 (ARC-II) 的原因包括三方面，其一，架构二虽然相对于架构一可以捕捉到语句间丰富的局部匹配模式，但是对于翻译中的短语匹配，这种局部之间的匹配模式并不强烈，因为翻译中短语较长的匹配对并不多见，并且对于较长的源端短语，其候选一般相对较少，通过传统的统计信息就能较大程度的将大部分的短语对分开，机器翻译目标端短语选择的主要困难，集中在源端短语相对较短（1-10 个字）时的情形，这些源端短语的目标端短语候选较多（一般在 50 个翻译候选以上）。其二，通过对架构一进行改进，将源端短语与它的上下文信息区分，本章提出的上下文依赖的卷积神经网络短语匹配模型可以有效的捕捉到目标端短语与源端短语以及其上下文之间的关系。其三，相对于架构一，架构二的计算复杂度明显高很多，而统计机器翻译的过程需要对源端短语与大量的目标端短语进行匹配，ARC-I 的效率更加适合。因此，本章在 ARC-I 的基础上进行改进。

统计机器翻译中的源端短语与目标端短语的匹配可以分为不同的层级。传统基于频率统计的方法给出了源端短语所有可能的翻译候选是较为低层的语义匹配，本章提出的上下文依赖的卷积神经网络短语匹配模型则针对源端短语所在的上下文，从其所有候选翻译中找出较为合理目标端短语，而不是简单的给出其所有可能的翻译，因此是一种更为深层的语义匹配。为了有效地训练本章提出的上下文依赖的卷积神经网络短语匹配模型，首先，我们使用基于上下文依赖的双语词向量初始化模型，使模型更有效的捕捉到两种语言的基本词级语义对应关系。其次，我们设计了“课程式”学习的策略[83] 对模型进行训练。该策略通过对训练集的合理组织对模型进行循序渐进的训练。最终，在大规模的翻译任务上的实验结果表明，上下文依赖的卷积神经网络短语匹配模型在一个较强的基于短语的统计翻译系统上 BELU 值提高了 1.0 个百分点。

本章的组织结构如下：章节 4.2、介绍了目标端短语选择方法的相关研究工作；章节 4.3 、详细介绍了本章提出的上下文依赖的卷积神经网络短语匹配模型；章节 4.4、介绍了上下文依赖的卷积神经网络短语对选择模型的目标函数，词向量初始化方法以及设计的“课程式”学习算法。章节 4.5、详细介绍了实验设置以及实验结果。最后给出了本章的结论。

4.2 相关研究工作

本章的研究与基于上下文的源端短语与目标端短语匹配，以及基于神经网络的双语短语的表示学习研究工作相关，以下分别对这些工作做简单介绍：

一些工作基于词和短语的离散表示方法，通过利用短语或词的局部上下文，提高源端短语与目标端短语的匹配。例如，[81]、[84] 和 [82] 利用离散的上下文表示指导源端短语与目标端短语的语义匹配。然而，一方面，这种方式通常会有比较严重的数据稀疏问题。另一方面，由于这些方法对词的表示是基于传统的离散向量，而不是词向量的方式，使得其不能有效的利用词与词之间的语义信息。Wu 等 [85] 提出利用离散的上下文信息（即词和词性信息）学习双语词向量，提高统计机器翻译系统的性能。然而，Wu 等只对经常出现的短语对进行研究，得到短语匹配得分的方法也非常简单，即通过对源端短语中的词与目标端短语中的词的相似度求和得到源端短语与目标短语的匹配得分。虽然考虑了上下文信息，但是在短语对匹配时，忽略了短语内的词序信息。与之不同，本章提出的模型不仅考虑了句子中所有词的信息，并且直接通过卷积操作对源端短语与目标端短语进行建模得到它们的表示向量。然后，对它们进行

匹配。另外一些研究工作尝试通过文本的连续表示（Distributed Representation）得到文档级别的语境信息，然后将其引入到短语对的匹配计算过程中，例如，[86] 和 [87] 引进文档级别的主题信息选择语义更相关的双语对。虽然同一个文档中的大多数语句具有同一个主题，但是，依然有较大比例的句子主题与文档主题并不一致 [88]。因此，整个文档级主题对于特定的句子并不一定准确。与之不同，本章提出的方法对每个句子的不同短语学习得到不同的向量表示，因此对短语对的语义匹配更为准确。[86] 和 [87] 对文档学习向量表示，从而导出相应的短语表示，这样的做法会导致一定的信息损失，对句子和短语的表示向量质量有一定的局限性。

近些年，研究人员对双语短语的表示学习也产生了浓厚的兴趣。首先对源端短语与目标端短语联合学习向量表示。然后，将不同语言中具有相似语义的短语聚在一起。基于双语短语对具有同样的语义的假设，它们可以互相作为监督信号指导模型对源端短语与目标端短语在连续向量空间中学习表示。例如，[79] 将目标端短语和源端短语的向量表示映射到同一个与语言类型无关的连续空间，从而得到源端短语与其目标端短语相近的连续向量。Zhang [80] 等则将两种语言的短语分别映射到两个不同的向量空间上，然后，在两种语言的向量空间之间建立了一种变换关系。然而，这些工作仍然是根据短语对在平行语料中的统计信息，在连续空间中学习得到它们的语义相似度。因此其主要优势来自于短语对的表示方法，而不是考虑短语的特定上下文信息。与之不同，上下文依赖的卷积神经网络短语匹配模型利用短语特定的上下文信息，通过对源端短语和目标端短语分别得到其连续向量表示，然后计算得到源端短语与目标端短语的语义匹配。

4.3 上下文依赖的卷积神经网络短语匹配模型

上下文依赖的卷积神经网络短语匹配模型的基本架构如图4-2所示，图中虚线下部表示通过两个深度卷积神经网络语句表示模型，对源端句子和目标端短语分别得到它们的表示，虚线上部是通过多层感知机对源端句子表示和目标端短语表示进行匹配。图中符号“/”表示通过全 0 向量对句子进行填充，其影响可以通过本文设计的门函数进行逐步消除，该模型包括以下两部分。

- 深度卷积神经网络语句表示模型 分别对源端短语和其所在的句子上下文，以及目标端短语建模得到其向量表示。
- 匹配模型 通过多层感知机对得到的两个表示向量进行匹配 [89]。

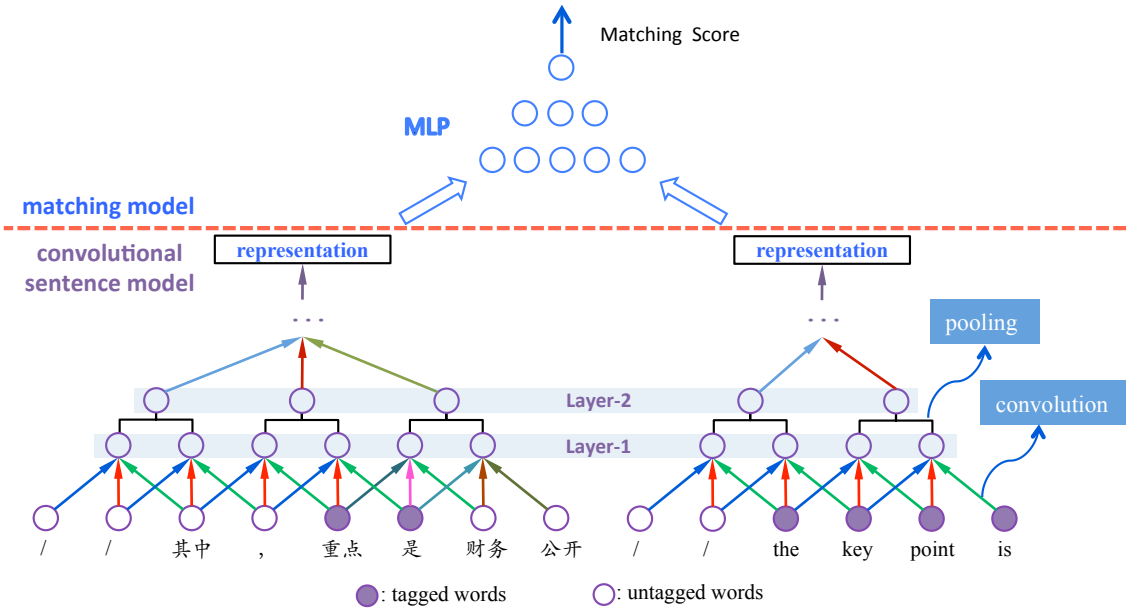


图 4-2 上下文依赖的卷积神经网络短语匹配模型的架构图

Fig.4-2 Architecture of Context-Dependent Convolutional Neural Network Phrases Matching Model

令 $\hat{e}$ 表示一个目标端的短语，$\mathbf{f}$ 表示源端句子，$\mathbf{f}$ 包含与 $\hat{e}$ 对应的源端短语。首先，我们通过两个不同的深度卷积神经网络语句表示模型分别将 $\mathbf{f}$ 和 $\hat{e}$ 表示成向量 $\mathbf{x}$ 和 $\mathbf{y}$，然后通过多层感知机计算它们的匹配得分 $s(\mathbf{x},\mathbf{y})$。最后，将 $s(\mathbf{x},\mathbf{y})$ 作为一种特征加入到统计机器翻译系统中，从而提升统计机器翻译系统的性能。

4.3.1 深度卷积神经网络语句表示模型

如图4-2所示，模型首先将 $\mathbf{f}$ 与 $\hat{e}$ 中的词转换为其对应的词向量，并按照词的顺序排列。然后经过多层的卷积和最大池化操作分别得到它们固定维度的向量表示。同样，每一层内不同窗口内的卷积操作共享同一套参数。在第一层，卷积层分别对 $\mathbf{f}$ 和 $\hat{e}$ 上所有的滑动窗口做卷积操作，从而对 $\mathbf{f}$ 和 $\hat{e}$ 中所有滑动窗口内的词的组合进行建模。给定 $\mathbf{f}$ 和 $\hat{e}$ 上大小为 k 的滑动窗口 i, 其卷积输出的第 j 维为 $\mathbf{c}_i^{(1,j)}$，其计算见公式 4-1：

$$\mathbf{c}_i^{(1,j)} = g(\hat{\mathbf{c}}_i^{(0)}) \cdot \phi(\mathbf{w}^{(1,j)} \cdot \hat{\mathbf{c}}_i^{(0)} + \mathbf{b}^{(1,j)}) \tag{4-1}$$

式中 $g(\cdot)$——“门函数”，其决定 $\phi(\cdot)$ 是否被计算；

$\phi(\cdot)$——非线性激活函数，在模型中选择函数 ReLu [70]；

$\mathbf{w}^{(1,j)}$——第一层上第 j 个激活单元的参数, 其矩阵形式为 $\mathbf{W}^{(1)} = [\mathbf{w}^{(1,1)}, \ldots, \mathbf{w}^{(1,J)}]$；$J$ 表示卷积操作的输出维度；

$\hat{\mathbf{c}}_i^{(0)}$——通过将大小为 k 的滑动窗口 i 中词的词向量按顺序拼接之后得到的向量；

$\mathbf{b}^{(1,j)}$——偏置项, 其向量形式为 $\mathbf{B}^{(1)} = [\mathbf{b}^{(1,1)}, \ldots, \mathbf{b}^{(1,J)}]$；

为了将源端短语与它所在句子的上下文进行区分，我们在词向量的基础上增加一个标记位，其值取 1 或者 0, 1 表示该词是源端短语中的一部分，0 表示该词属于源端短语的上下文。需要注意的是卷积操作可以越过源端短语的边界，这样设置的目的是将源端短语与其上下文信息同时包含进来。为了解决源端句子以及目标端短语的长度不一致性的问题，我们在源端句子和目标端短语的末尾增加向量元素为全 0 的“占位符”，为了消除“占位符”的影响，我们使用了“门函数”$g(\cdot)$，该函数当卷积窗口内的词全为“占位符”时，取值为“0”，否则取值为“1”。

深度卷积神经网络语句表示模型的第二层是局部最大池化层。局部最大池化操作在非重叠的大小为 1×2 的窗口上进行，其输出见公式 4-2：

$$\mathbf{c}_i^{(2,j)} = \max\{\mathbf{c}_{2i}^{(1,j)}, \mathbf{c}_{2i+1}^{(1,j)}\} \tag{4-2}$$

第三层继续进行卷积操作，以得到源端句子或目标端短语的更高层次的表示，其计算见公式 4-3：

$$\mathbf{c}_i^{(3,j)} = g(\hat{\mathbf{c}}_i^{(2)}) \cdot \phi(\mathbf{w}^{(3,j)} \cdot \hat{\mathbf{c}}_i^{(2)} + \mathbf{b}^{(3,j)}) \tag{4-3}$$

以此类推，经过多层的卷积和最大池化操作，最后分别对源端短语及其上下文、目标端短语得到固定长度的向量表示 $\mathbf{x}$ 和 $\mathbf{y}$。需要注意的是，由于源端句子与目标端短语的差异性，它们的卷积神经网络句子模型的架构设置不同，同时它们的参数也不相同。

4.3.2 匹配模型

对源端句子和目标端短语的表示进行匹配，我们使用使用多层感知机。首先，使用一层带有非线性函数的全连接层，对源端句子向量 $\mathbf{x}_{\bar{f}_i}$ 和目标端短语向量 $\mathbf{y}_{\bar{e}_j}$ 进行综合，得到它们的联合表示 h_c.

$$h_c = \phi(w_c \cdot [\mathbf{x}_{\bar{f}_i} : \mathbf{y}_{\bar{e}_j}] + b_c) \tag{4-4}$$

然后，通过一层线性运算得到它们的匹配打分 $s(\mathbf{x},\mathbf{y})$：

$$s(\mathbf{x},\mathbf{y}) = w_s \cdot h_c \tag{4-5}$$

4.4 模型训练

4.4.1 目标函数

上下文依赖的卷积神经网络短语匹配模型的目的是对带有上下文的源端短语 **f** 与其正确的目标端短语 $\hat{e}^+$ 匹配得到较高的分，与其不正确的目标端短语 $\hat{e}^-$ 匹配得到较低的分。因此我们采用成对输入（Pairwise）的带有差距（Margin）的学习排序（Learn to rank）训练策略。假设给定如下三元组 $(\mathbf{x},\mathbf{y}^+,\mathbf{y}^-)$，这里 $\mathbf{x},\mathbf{y}^+,\mathbf{y}^-$ 分别表示通过深度卷积神经网络语句表示模型对 **f**，$\hat{e}^+$ 和 $\hat{e}^-$ 得到的向量表示。我们可以得到如公式 4-6的目标函数。我们采用基于批处理的梯度下降算法优化模型参数 Θ。

$$L_\Theta(\mathbf{x},\mathbf{y}^+,\mathbf{y}^-) = \max(0, 1 + s(\mathbf{x},\mathbf{y}^-) - s(\mathbf{x},\mathbf{y}^+)) \tag{4-6}$$

式中 $s(\mathbf{x},\mathbf{y})$———公式4-5定义的匹配打分函数；

Θ———词向量;

θ———包含了卷积句子模型部分、匹配打分部分的参数、以及源端语言与目标端语言的词向量

为了更有效的训练模型。首先，我们通过基于上下文依赖的双语词向量初始化源端语言与目标语言的词向量。这样，使得模型在底层部分就能包含进词级别的上下文和语义匹配信息。其次，我们设计了一种基于“课程学习”的策略对模型的整个架构由易到难、循序渐进的进行训练。

4.4.2 基于上下文依赖的双语词向量模型

模型的初始化对非凸优化问题起着非常重要的作用。对 CDCM 模型初始化中非常重要的一部分是对两种语言的词向量进行初始化。典型的词向量是在大规模的单语语料上训练得到的[11]。然而，对于机器翻译问题来说，通过这种方式训练得到的源语言和目标语言词向量空间相互独立，不能够捕捉到两种语言中词之间的语义关系。相关研究表明，双语词向量在捕捉词级别的语义相关性上起到重要的作用[85, 90]，因此，我们通过双语词向量初始化上下文依赖的卷积神经网络短语匹配模型。这样能够使得模型从初始化阶段就具有了较强的词

级别的双语语义关联信息。Zou 等[90]通过利用机器翻译上词对齐之后的平行语料指导模型对常见的短语对学习得到相似的词向量，同时 Wu 等[85]人通过离散的上下文信息进一步提升双语词向量质量。Yang 等通过利用上下文依赖的深度神经网络提升词对齐的质量[91]，其设计了基于隐马可夫模型的词对齐模型。该模型引入了双语词向量。其双语词向量的训练利用了词对齐信息以及词对周围的上下文信息。

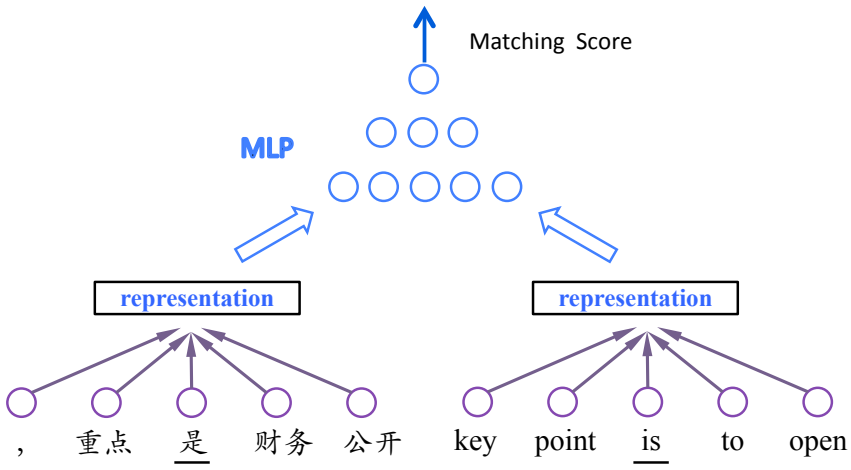


图 4-3 基于上下文依赖的双语词向量模型架构图

Fig.4-3 Architecture of the CDCM bilingual word embedding model

受以上研究工作的启发，本文利用词对齐信息和对齐词对的上下文信息训练得到双语词向量。其结构图如4-3所示。给定一个词对 (f_i, e_j)，f_i 和 e_j 分别来自平行语料的两种不同语言。分别从 f_i 和 e_j 所在的句子中抽取他们的局部上下文 $\bar{f}_i = f_{i-2}, f_{i-1}, f_i, f_{i+1}, f_{i+2}$ 和 $\bar{e}_j = e_{j-2}, e_{j-1}, e_j, e_{j+1}, e_{j+2}$，分别构造 $\widetilde{f}_i = f_{i-2}, f_{i-1}, f_i^{'}, f_{i+1}, f_{i+2}$ 和 $\widetilde{e}_j = e_{j-2}, e_{j-1}, e_j^{'}, e_{j+1}, e_{j+2}$，其中 $f_i^{'}$ 和 $e_i^{'}$ 表示分别从源语言和目标语言的词典中随机选择的词，也即 $\widetilde{f}_i$ 和 $\widetilde{e}_i$ 分别表示 $\bar{f}_i$ 和 $\bar{e}_j$ 对应的负例，然后分别对它们做如下操作。

$$\mathbf{x}_{\bar{f}_i} = \phi(\mathbf{w}_f \cdot Le(\bar{f}_i) + \mathbf{b}_f) \tag{4-7}$$

$$\mathbf{x}_{\widetilde{f}_i} = \phi(\mathbf{w}_f \cdot Le(\bar{f}_i) + \mathbf{b}_f) \tag{4-8}$$

$$\mathbf{y}_{\bar{e}_j} = \phi(\mathbf{w}_e \cdot Le(\bar{e}_j) + \mathbf{b}_e) \tag{4-9}$$

$$\mathbf{y}_{\widetilde{e}_j} = \phi(\mathbf{w}_e \cdot Le(\bar{e}_j) + \mathbf{b}_e) \tag{4-10}$$

$Le(\cdot)$ 表示将一个序列中的所有词转换为其对应的词向量，并返回所有词向量按照顺序拼接一起后的向量。为了训练双语词向量，我们选择使用基于 Pairwise 排序策略的的目标函数[10]，其目标函数见公式 4-11：

$$L_\Theta(\bar{f}_i, \bar{e}_i, \widetilde{f}_i, \widetilde{e}_i) = \max(0, 1 + 2 \cdot s(\mathbf{x}_{\bar{f}_i}, \mathbf{y}_{\bar{e}_j}) - s(\mathbf{x}_{\bar{f}_i}, \mathbf{y}_{\widetilde{e}_j}) - s(\mathbf{x}_{\widetilde{f}_i}, \mathbf{y}_{\widetilde{e}_j})) \quad (4\text{-}11)$$

$s(\cdot)$ 表示对两个向量的匹配打分。对于给定的的向量 h_x 和 h_y，其匹配得分的计算形式见公式 4-12：

$$s(h_x, h_y) = w_m \cdot (\phi(w_c \cdot [h_x : h_y] + b_c)) \quad (4\text{-}12)$$

我们使用基于批处理的随机梯度后向传播算法对模型中的参数 $\mathbf{w}_f, \mathbf{w}_e, \mathbf{w}_c, \mathbf{b}_f, \mathbf{b}_e, \mathbf{b}_c$ 进行更新，同时更新源端语言和目标端语言的词向量 ($\mathbf{E_e}, \mathbf{E_f}$)，在大规模的平行语料上多次随机替换源端短语中心词 f_i 和目标端短语中心词 e_j，每次迭代分别随机产生 50 个源端负例和目标端负例。通过这样的策略使得基于上下文依赖的双语词向量模型，不仅具有传统的基于 Pairwise 排序策略的词向量学习模型的特点，同时利用了词之间的对齐信息，使得模型将两种语言词之间的对应关系蕴含到两种语言的词向量表示中。

4.4.3 课程式训练

人类在学习的过程中会对学习的内容的难度，有一个比较清晰的概念。在学习的过程中，会对学习的顺序有一个具体的安排。通常这种安排是由易到难，由简到繁，循序渐进进行的，例如，学生几乎不可能在没掌握四则运算的情况下，学习微积分的课程。一个合理的课程安排能够有效地提高学习的效率。受人类学习特点的启发，针对复杂的深度神经网络模型，Bengio 等人于 2009 年提出了课程学习的思想[83]。所谓课程学习（Curriculum Learning）是指在训练机器学习系统的过程中，首先从小规模简单的训练样例开始，然后逐渐增加训练样例的难度，课程学习的思想可以在非凸优化问题上提高模型的泛化能力以及收敛速度。使用课程学习思想的关键是对对训练样本进行合理组织，然后逐步的将样本输入给机器学习系统进行训练，而不是随机的选择训练样本。

根据课程学习的思想，针对上下文依赖的卷积神经网络短语匹配模型，本文设计了一种“课程式”的训练策略。短语对的匹配可以分为不同层次的语义匹配。对于一个训练样例 ($\mathbf{f}, \hat{e}^+, \hat{e}^+$) 其难度根据将正确的目标端短语 $\hat{e}^+$ 与错误的目标端短语 $\hat{e}^-$ 正确分开的难度定义。当错误的目标端短语与正确的目标端短语意义相差较大时，模型只需要捕捉到源端短语与目标端短语的部分浅层的匹配信息，就可以将正确的目标端短语选择出来。当正确的目标端短语与错误的目标端短语只有一些细微的差异，并且这些差异对于特殊的上下文非常敏感时，模型需要学习到深层的语义匹配信息，并有效的利用短语的上下文信息，才能将错误的目标端短语与正确的目标端短语分开。本文根据双语短语对匹配

问题的特点，将数据集中的样本划分为三种不同困难程度等级。

• 容易: 错误目标端短语是随机从所有目标语言的短语中随机选取的，这些错误短语与正确的目标端短语差异性较大，例如，“错误”的候选短语可以有“false，mistake，error，wrong 等”，而“beautiful”相对于其候选短语就是一种较为容易的负例，因为，模型只要能够捕捉到源端短语与目标端短语的基本语义关系，不需要利用源端短语的任何上下文信息，就能将正确的目标端短语选择出来。我们将其定义为最为“容易”的负例。

• 中等难度: 给定源端短语及其所在的句子 $s = (\mathbf{f_1}, ..., \mathbf{f_k}, ..., \mathbf{f_l}, , ..., \mathbf{f_n})$，该句子所对应的包含正确目标端短语的目标语言句子 $t = (\mathbf{e_1}, ..., \mathbf{e_i}, ..., \mathbf{e_j}, , ..., \mathbf{e_n})$，对于源端短语 $\mathbf{f_k}, ..., \mathbf{f_l}$ 对应的正确目标端短语为 $\mathbf{e_i}, ..., \mathbf{e_j}$，其错误的负例为除去 $\mathbf{e_i}, ..., \mathbf{e_j}$ 之外 t 中的其它短语。从这些负例可以看出，它们与正确的目标端短语在同一个句子中出现，并且它们对应的源端短语也在同一个源端语言句子中，从这些错误的目标端短语中将正确的目标端短语选出，需要模型能够区分出源端短语中的待匹配短语与其上下文信息，同时使得模型能够有能力对其基本语义进行匹配。因此在“容易”的负例基础上，增加了部分难度。我们将其定义为“中等难度”的负例。

• 困难: 一个源端语言短语在不同的上下文环境中对应不同的目标端短语。我们选择源端短语在当前句子中对应的目标端短语为正例，在其他上下文环境下对应的目标端短语为当前上下文环境下的目标端短语的负例。这种情形下，模型不仅需要能够区分源端短语与其上下文，并且能够利用其上下文的差异正确区分其所有的候选目标端短语。这些目标端短语在语义上相关，因为其都是源端短语的候选，只是不适合当前上下文环境。因此这些负例较为困难，也是当前统计机器翻译系统没有考虑到的匹配特征。我们将其定义为最为“困难”的负例。

从以上的描述可以发现，这样设计的不同难度的负例，可以代表匹配模型可以学到的三种不同层次的匹配关系:

• 基本语义匹配：通过“容易”的负例可以学到基本的语义对应。

• 一般的语义对应：通过“中等难度”的负例使得模型根据源端句子中要匹配的短语的基本语义，利用其上下文与目标端中的短语正确对应。

• 上下文依赖对应：通过“困难”的负例的学习，使得模型根据源端短语以及其所在的上下文，得到源端短语的深层语义信息，从其所有候选的目标端短语中选出最契合当前上下文的目标端短语。

算法 4-1 CDCM 的课程训练算法
Algo. 4-1 Curriculum training algorithm

Input: 训练集 $\mathcal{T}$
Output: 模型参数 Θ, 以及词向量 W

$N_1 \leftarrow$ *easy_negative*($\mathcal{T}$) ; // 产生‘容易’的训练负例
$N_2 \leftarrow$ *medium_negative*($\mathcal{T}$) ; // 产生‘中等难度’的训练负例
$N_3 \leftarrow$ *difficult_negative*($\mathcal{T}$) ; // 产生‘难度’的训练负例
$N \leftarrow [N_1, N_2, N_3]$
/* 课程学习第一阶段 */
while 损失函数收敛或者达到预先设定的迭代次数 s_1 **do**
  $n \leftarrow$ sample($[1,0,0]$,N) ; // 以概率$[1,0,0]$ 对不同难度的训练样例采样
  $\Theta = \Theta - \eta \cdot \frac{\partial L_\Theta}{\partial \Theta}$; // 更新模型参数 Θ
  $W = W - \eta \cdot 0.01 \cdot \frac{\partial L_\Theta}{\partial W}$; // 微调词向量
/* 课程学习第二阶段 */
while 损失函数收敛或者达到预先设定的迭代次数 s_2 **do**
  $n \leftarrow$ sample($[\frac{1}{s+2}, \frac{s+1}{s+2}, 0]$,$N$)
  $\Theta = \Theta - \eta \cdot \frac{\partial L_\Theta}{\partial \Theta}$; // 更新模型参数 Θ
  $W = W - \eta \cdot 0.01 \cdot \frac{\partial L_\Theta}{\partial W}$; // 微调词向量
/* 课程学习第三阶段 */
while 损失函数收敛或者达到预先设定的迭代次数 s_3 **do**
  $n \leftarrow$ sample($[\frac{1}{s+2}, \frac{1}{s+2}, \frac{s}{s+2}]$,$N$)
  $\Theta = \Theta - \eta \cdot \frac{\partial L_\Theta}{\partial \Theta}$; // 更新模型参数 Θ
  $W = W - \eta \cdot 0.01 \cdot \frac{\partial L_\Theta}{\partial W}$; // 微调词向量

基于对数据集的样本的划分，本文提出了针对上下文依赖的卷积神经网络短语匹配模型“课程式”训练算法，如算法 4–1 所示，其中 $\mathcal{T}$ 表示训练样例集合、W 表示初始化的词向量、η 是随机梯度下降算法的学习率、t 表示训练样例的个数。根据构造的三种不同难度的负例在每次迭代中所占的比例，对模型的训练划分了三个不同阶段。

在不同阶段对模型参数的更新采用基于批处理的随机梯度下降方法。需要

注意的是，我们模型的参数在更新的过程中，同时对双语词向量进行更新。对于不同阶段，当模型收敛到一个局部最优解或者达到了预先设置的训练轮数时，终止当前阶段的训练进入下一阶段的训练。例如，第一阶段（算法中 5–9 行）模型见到的负例只包括最为“容易”的负例，通过第一阶段的训练使得模型能够学习到源端短语与目标端短语间的基本语义匹配。第二阶段（算法中 10–14 行）将输入给模型的训练样例中“容易”和“中等难度”的负例的比例设置为 $[\frac{1}{s+2}, \frac{s+1}{s+2}, 0]$，这样，随着第二阶段训练的不断深入，“中等难度”的负例占的比例逐渐增加，“容易”的负例的比例逐渐减小。需要注意的是，在这个过程中“容易”的负例始终占有一定的比例。这样做的目的是防止模型对之前的“容易”的负例遗忘，而过多的偏向于“中等难度”负例的区分。在第三阶段 (算法中 15–19 行)，随着训练进程的不断深入，模型一次迭代见到的数据中三种不同难度的负例的比例按照 $[\frac{1}{s+2}, \frac{1}{s+2}, \frac{s}{s+2}]$ 不断变化，s 表示第三阶段迭代的次数，s 初始值为 1。一般地，第一阶段和第二阶段，模型能较快的收敛，第三阶段在第一、二阶段的基础上，需要运行较长的时间才能够收敛。

4.5 实验

我们通过实验回答以下几个问题：1）本章所提出的上下文依赖的卷积神经网络短语匹配模型对翻译候选的选择是否有效，通过将源端短语与目标端短语的匹配得分作为特征添加到统计机器翻译系统中，能否提高翻译系统性能？2）从翻译性能上能否体现出本文设计的上下文依赖的双语词向量的优势？3）本章提出的“课程式”学习算法能否对模型的训练以及性能进行提高？

4.5.1 实验设置

数据集 通过 NIST 中英翻译任务评价本章提出的模型。NIST 是当前国际上最具权威的机器翻译评测，它于 2002 年由美国 TIDES 的项目资助设立，一般每年举行一次。其为机器翻译领域的相关研究提供公共的测试基准。为了有效的提取源端短语与目标端短语的翻译对，并充分训练统计机器翻译系统，我们选择 LDC 数据集[1]作为训练数据，其中包括 LDC2002E18、LDC2003E07、LDC2003E14 以及 LDC2004T07、LDC2004T08 和 LDC2005T06 的 Hansards 部分。该数据包括约 150 万对中英句子对。为了提高双语短语对的质量，本文采用强制解码的策略（Forced Decoding）。其主要流程是先从 LDC 150 万中英文句子中

[1]https://catalog.ldc.upenn.edu/LDC2011T07

抽取出中英短语对，然后利用这些短语对，对翻译训练语料中的中文句子强制得到标准答案，通过这样的过程过滤掉其它不常用的中英短语对，保留那些能得到标准答案的常用中英短语对，最终得到大约 240 万不同的中英短语对（长度在 1-7 个词之间），考虑到源端短语上下文的不同，我们得到大约 2 千万不同的匹配对，将只有一个目标端短语候选的短语对去除，得到大约 1350 万不同的匹配对，也即训练数据集中的所有正例。本文使用 NIST2002 测试集作为开发集，NIST2004、NIST2005 数据集作为测试集。

基准翻译系统 本文选择开源的基于短语的机器翻译系统“摩西”（Moses）[2] [92] 作为基准翻译系统。该系统是由爱丁堡大学、德国亚琛工业大学等 8 家单位联合开发的一个开源的基于短语的统计机器翻译系统。其是当前统计机器翻译研究领域最有影响力的开源系统。代码经过大量的优化，性能很高，已成为统计机器翻译领域最主要的基准系统。本文使用了基于短语的统计机器翻译系统中的一系列的常用特征，包括短语惩罚（Phrase Penalties）、词化调序（Lexicalized Reordering Model）、语言模型（Language Model）以及线性扭曲代价（Linear Distortion Model）等。为了充分训练统计机器翻译系统的英文语言模型，本文采用 SRI 语言工具包[93]，其被广泛用于机器翻译系统中的语言模型的构建。在 GIGAWORD 数据集的 Xinhua 部分训练 4-gram 语言模型。GIGAWORD 数据集的 Xinhua 数据集包括 1,744,025 文档，大约 3.6 亿英文单词，其语料非常规范。通过该数据集能够训练出一个较强的语言模型。使用最小错误率训练（Minimum Error Rate Training） [94] 算法优化统计机器翻译系统中的特征权重。

4.5.2 评价指标

机器翻译系统中译文质量的评价是最为困难的任务之一。目前，通常采用 BLEU 对译文质量进行评价，BLEU（Bilingual Evaluation Understudy）是由 IBM 于 2002 年提出的评价标准 [95]，其基本思想是通过计算机器翻译系统给出的译文与人工书写的参考译文间字面相似度，评价机器翻译系统给出的译文的质量。NIST 在 BLEU 的基础上进行了改进，其对参考译文中出现次数较少的词给定更高的权重，以体现其这些词的重要性，另外使用算术平均取代 BLEU 中的几何平均。NIST BLEU 是目前较好的一种评价标准，与人工给出的质量评定具有较高的一致性 [96]，因此本文采用 NIST BLEU 对译文质量进行评测。

[2] http://www.statmt.org/moses/

4.5.3 翻译性能对比

首先，为了验证目标端短语选择中源端短语上下文的作用，本文构造了一个忽略源端短语上下文的卷积神经网络匹配模型 CICM（Context Independent Convolutional Matching Model）。该模型参考以前的工作 [41, 79, 80] 的基础上，使用上下文无关的卷积神经网络匹配模型对源端和目标端短语进行匹配打分，对从短语表中随机选择的其他英文短语作为负例训练模型。然后将中文短语与英文短语的匹配得分，作为额外特征加入到“摩西”基准翻译系统中。

表 4-1 翻译质量的评价结果

Table4-1 Evaluation of translation quality

Model	**MT04(%)**	**MT05(%)**	**All(%)**
Baseline	34.86	33.18	34.40
CICM	35.82^{α}	33.51^{α}	34.95^{α}
CDCM_1	35.87^{α}	33.58	35.01^{α}
CDCM_2	35.97^{α}	33.80^{α}	35.21^{α}
CDCM_3	$36.26^{\alpha\beta}$	$33.94^{\alpha\beta}$	$35.40^{\alpha\beta}$

表4-1列出了“摩西”基准（Baseline）系统，将 CDCM 训练的各个阶段的特征加到“摩西”系统，以及将 CICM 模型的特征加到“摩西”系统之后的性能。图中 CDCM_k 表示 CDCM 模型通过“课程式”训练算法的第 k 阶段时的结果，CICM 表示忽略源端短语的上下文的匹配模型的结果，“All”表示将所有测试集整合到一起的结果，上标 α 和 β 分别表示对应的模型与 Baseline 的结果、加入 CICM 的结果的显著性差异检验结果（$p < 0.05$）[97]。通过结果可以看出，CDCM 模型在训练的各个阶段，都可以显著地提升“摩西”的性能，CDCM 经过三个阶段的训练后，可以在“摩西”的基础上整体提升 1.0 个 BELU 的性能。在 MT04 测试集上最高可以提升 1.4 个 BLEU。CDCM 随着三个不同阶段的训练可以逐渐的增加模型的性能。从而验证了本文提出的“课程式”训练算法能够逐步提升模型的性能，随着课程训练的不断进行，CDCM 模型可以逐步学习到中英短语更困难的匹配。从而证明了“课程式”训练算法使得 CDCM 从易到难的逐步学习到双语短语各个层次的语义匹配信息。

与 CICM 相比，CDCM 在整个测试集上都显著优于其性能，从而验证了源端短语上下文信息对选择其候选翻译的重要性。CDCM_1 的性能与 CICM 的性能相当，这是因为此时两个模型都只能捕捉到短语匹配最为基本的语义信息。虽然，CDCM_1 的训练也包含了短语的上下文信息，但是由于其负例是从整个

目标端短语集中随机选择的，其对于正例来说过于简单，模型并不需要利用上下文信息，就能通过训练将负例和正例分开，因此其并没有充分的利用上下文信息。$CDCM_3$ 阶段如果不考虑上下文信息，绝大多数（“困难”的负例）都是源端短语的正确候选，因此 $CDCM_3$ 的训练就使得模型可以学习到上下文信息将负例与正例正确区分。

sentence	伊拉克 拥有 大 杀伤力 武器 的 错误 情报	在 确定 关塔那摩 囚犯 的 身份 方面 犯了 错误
references	incorrect, faulty, wrong, erroneous	a mistake, mistakes
TM	wrong (1143), mistakes (361), mistake (314)	
CICM	was wrong (7), is wrong (8), the wrong (134)	
$CDCM_1$	a wrong (44), wrong (1143), its mistake (12)	a mistake (16), by mistake (5), the mistake (30)
$CDCM_2$	wrong (1143), the mistake (30), a wrong (44)	the erroneous (31), a mistake (16), the mistake (30)
$CDCM_3$	false (42), wrong (1143), faulty (16)	mistake (314), error (162), fault (14)

sentence	其中 , 重点 是 财务 公开	我们 另一个 政策 重点 是 市场 发展
references	the key point is	focus is
TM	focus is (10), focus on (8), focuses on (6)	
CICM	the key point is (3), key point is (3), where the focus is (2)	
$CDCM_1$	the focus is (4), focus was (2), where the focus is (2)	focus was (2), the focus is (4), focuses on a (2)
$CDCM_2$	where the focus is (2), is mainly (2), priority is (2)	focus was (2), focus is (10), priority is (2)
$CDCM_3$	the key point is (3), the focus is (4), main point is that (2)	focus is (10), priority is (2), focus of (2)

图 4-4 不同模型对源端短语给出的目标端短语的示例图

Fig.4-4 The top ranked target phrases according to different methods

图4-4通过展示一些例子说明 CDCM 模型为什么可以提高机器翻译系统的性能，图中“reference”表示源端短语参考的正确目标端短语。“TM”、“CDCM”和“CICM”分别表示通过机器翻译模型、CDCM 模型和 CICM 模型给出的目标端短语结果，$CDCM_1$，$CDCM_2$ 和 $CDCM_3$ 分别表示 CDCM 在课程学习训练的第 1、2 和 3 阶段的结果，图中括号中的数字表示源端与目标端短语的共现频数。例如，如果忽略“错误”的上下文信息，其对应的翻译候选有多种，而机器翻译系统按照短语对出现的频数统计进行排序。因此，无论“错误”的上下文是什么，其给出的翻译候选都是按照“wrong,mistakes,mistake...”进行排序，CICM 由于也没有考虑到上下文信息，其排序完全按照其得分情况，并没有根据“错误”的上下文不同，给出不同的排序。从图中可以看出 $CDCM_3$ 能够根据源端短语的上下文不同，对候选短语进行排序，虽然“false”的统计频数在“mistake”之后，对于第一个句子 $CDCM_3$ 依然将其排在首位。不难判断“false”对于第一个句子中的“错误”才是一个

较为恰当的翻译。CDCM 模型对短语的匹配打分，综合考虑了源端短语的基础语义信息及其所在的上下文，而不是根据源端短语与目标端短语共线的次数。因此，CDCM 的与统计机器翻译系统中的其他特征具有很好的互补性。随着“课程式”训练算法的不断进行，CDCM 可以逐步学习到双语短语对的基于上下文的匹配信息。事实上，如图4-4中所示，无论短语对出现频率高或低，通过 $CDCM_3$ 得到的目标端短语的候选与正确的目标端短语基本一致。CICM 忽略了源端短语的上下文信息，对于源端短语在不同上下文的情形，不能做到合理的适应。相反，CDCM 模型可以根据源端短语的上下文的不同，选择正确的目标端短语与之对应。

4.5.4 双语词向量性能对比

中英的单语词向量是通过 Word2vec 中 CBOW 模型[11] 分别在 150 万的中文句子和英文句子上训练得到的。双语词向量是通过本文设计的基于上下文依赖的双语词向量学习模型，在带有词对齐信息的 150 万对中英句子对上训练得到的。单语和双语词向量的维度都为 50。表4-2列出了单语词向量和双语词向量对 CDCM 模型的性能影响。从结果可以看出，双语词向量在课程训练的各个阶段都比单语词向量的性能高。虽然 CDCM 在训练的过程中，仍然会微调词向量，但双语词向量的初始化能够一定程度上降低模型的训练难度与时间，从而使模型更容易收敛到更好的结果。另一方面，双语词向量比单语词向量在词级别上捕获到更多的不同语言之间的匹配关系。相关工作也表明双语词向量可以在一定程度上学习到语言间词级别的语义匹配信息，其可以有效地提高机器翻译系统的性能 [85, 90, 91]。

图4-5列出了分别通过单语与双语词向量初始化模型，对源端短语与目标端短语匹配得分按照大小进行排序的例子。从图中可以看出，通过双语词向

表 4-2 单语词向量和双语词向量对翻译质量的影响对比结果

Table4-2 Comparison of the monolingual word embeddings and the bilingual word embeddings

Models	Monolingual			Bilingual		
	MT04(%)	MT05(%)	All	MT04(%)	MT05(%)	All(%)
$CDCM_1$	35.74	33.38	34.85	35.87	33.58	35.01
$CDCM_2$	35.80	33.59	35.04	35.97	33.80	35.21
$CDCM_3$	35.95	33.65	35.14	36.26	33.94	35.40

量初始化的模型，在词的层面比单语词向量初始化的模型好。例如，模型训练到第三阶段 $CDCM_3$，对“重点是”的翻译，单语词向量初始化的模型给出的结果，虽然在整体短语语义上已非常正确，但针对个别词的翻译，并不非常理想，例如，对单词“main”的翻译。而经过双语词向量初始化的模型给出的结果在词层面上较为理想，并且这种词级别上的较好对应，通过卷积操作的组合操作，同样得到更好的短语间的匹配，例如，“the key”。这是因为双语词向量的训练过程能够包含进词层面的语义匹配信息，例如，“重点”与“the key point”在词对齐级别上经常出现。

	Monolingual Word Embedding	Bilingual Word Embedding
sentence	其中，重点是 财务 公开	
references	the key point is	
TM	focus is (10), focus on (8), focuses on (6)	
$CDCM_1$	focus (3), focus is (10), focus was (2)	the focus is (4), focus was (2), where the focus is (2)
$CDCM_2$	is mainly (2), emphasis is (4), important thing is (2)	where the focus is (2), is mainly (2), priority is (2)
$CDCM_3$	main point is that the (2), main point is that (2), main point is (2)	the key point is (3), the focus is (4), main point is that (2)

图 4-5 单语词向量和双语词向量初始化模型后的目标端短语选择例子

Fig.4-5 The top ranked target phrases according to the CDCM models with different initializations

4.6 本章小结

本章是第三章提出的语句级表示以及匹配架构的扩展和应用。针对统计机器翻译中的目标端短语的选择问题，我们提出了上下文依赖的卷积神经网络短语匹配模型。该模型可以有效利用源端短语的上下文信息对目标端短语进行选择。为了有效的训练模型，首先，我们通过设计的基于上下文依赖的双语词向量学习模型学习源语言与目标语言的双语词向量，这样在词级别就建立了两种语言的关系，其次，我们提出了“课程式”训练算法，通过将不同难度的训练样例合理组织后输入给模型，对模型由易到难，由简到繁，循序渐进地训练。将短语对的匹配得分作为特征加入到一个较强的统计机器翻译系统中，本文提出的模型对系统的 BLEU 提升了 1.0%。

第 5 章 基于循环神经网络的摘要生成学习

5.1 引言

在自然语言处理领域中，段落或篇章表示是大量任务的基础，例如，主题发现、文本分类、自动文摘等。其中，自动文摘尤其是生成式自动文摘被普遍认为是极具挑战性的问题。本章将从深度神经网络模型的角度研究自动文摘生成。自动文摘的概念是由 Luhn 于上世纪 50 年代首次提出的[98]。所谓自动文摘是指从一个或者多个文档中，自动生成高度浓缩、通顺流畅并能表达原始文本重要信息的摘要[99, 100]。近年来，随着互联网信息的急剧增长，自动文摘显得越来越重要。一个优秀的文摘生成系统，一方面要能够对原始文本进行深入理解，并对其核心语义信息进行合理表示，另一方面需要具有根据原始文本表示，生成信息丰富、简明扼要、通顺流畅的摘要的能力。

自上世纪 50 年代以来，尽管自动文摘已经被深入研究多年，并且大量的方法被提出，但基本上是集中在抽取式文摘和相对规模较小的数据集上。其基本流程通常分为两步[101]，首先，通过语言学知识或者统计分析抽取文档中的关键词、短语、句子或者段落；然后，将抽取的文本进行重组作为文摘。虽然抽取式摘要一定程度上避免了人们必须通过阅读全文了解重要信息的方式。然而，通过抽取式方法得到的摘要，在内容和语言质量方面都不能令人满意，很多时候人们不得不再重新阅读原文。一旦文档的句子被抽取，句子的所有内容都被包含在摘要当中，这些句子不仅包含大量的冗余信息，而且句子与句子之间通常没有一定关联性，仅仅是文档中一些重要句子的简单拼凑，从而导致了信息的片段化，歧义性以及用户理解的不确定性。这种文摘与人类通过阅读文档得到的摘要差异较大。例如，在 DUC2005 的评测中，超过一半的摘要具有句法结构不合理，表达不通顺，对原文的重点信息不突出等问题[100]。

为了克服抽取式文摘的缺点，生成式自动文摘应运而生。生成式文摘的重要特点是摘要内容通顺流畅，语句之间具有高度的关联性（例如，句子与句子之间的转折），生成式摘要很多时候是对文档中的若干句子高度浓缩后的重写与复述。无论是对长文文档的准确理解与吸收（文档语义表示），还是生成通顺流畅的简要文本（语言生成），在过去的研究中都极富挑战性。一方面，是由于生成式的文本摘要数据集规模非常小（例如，DUC、TEC 的文摘数据集只

有数百篇），限制了一些机器学习算法的探索；另一方面，由于以往的计算设备的性能不能满足复杂模型的训练。基于以上问题，以往对生成式摘要的研究并不多，以往的生成式摘要的方法的研究思路，通常选择抽取式和生成式的折中方案。例如，通过对抽取式的原始文本的句子进行压缩处理来解决抽取式摘要中信息冗余的现象[102]；通过对抽取式摘要中的多个句子进行复述解决抽取式摘要中句子与句子不连贯性的问题[103]，以及通过对抽取的关键词和关键短语使用语言模型进行重组得到摘要。虽然这些方法在一定程度上解决了抽取式摘要的一些缺点，但并没有从根本上解决抽取式摘要的问题，而是刻意绕开了文档语义表示和语言生成这两个核心问题。

与自动文摘几乎是同时起步，在机器学习领域，神经网络也已经被研究数十年，特别是近些年图形处理器 (GPU) 以及高性能计算设备的出现，使得神经网络以深度学习的全新概念再一次兴起，并在语音识别、图像处理以及自然语言处理的一些任务上取得了较大的进展。深度神经网络模型无论是在词、句和篇的表示学习方面，还是在语言生成方面都取得了令人振奋的效果，例如，在自然语言处理领域的对话自动回复、机器翻译等任务上取得了重大突破。然而在过去的研究中，鲜有看到深度神经网络被用到自动文摘的生成上面。

基于此，我们首先基于互联网社交平台，利用自然标注信息构造了一个大规模的中文短文本摘要数据集。该数据集规模大，包含了超过 240 多万对的短文本以及对应的摘要，质量高，涵盖领域广。数据集来源于微博上具有较大影响力的官方微博，例如，“人民日报”、“经济观察报”、“国防部”等，因此，非常适合深度学习模型的研究。为了验证数据的质量，我们通过人工标注了 10,666 个样本。为了便于相关研究在该数据集上评测模型的性能，我们构建了多人交叉标注后的测试集。同时，将深度神经网络模型应用到短文本摘要自动生成任务中，提出使用基于循环神经网络的编码 -解码架构解决文摘自动生成，并构建了两类基于循环神经网络模型的端到端的短文本摘要生成模型。该类方法首先需要使用循环神经网络对原文进行表示，然后从表示中通过循环神经网络解码生成摘要。两种模型都无须依赖人工先验知识，如词性标注，句法分析，篇章结构分析等。

本章的组织结构如下：章节 5.2、首先介绍了我们从互联网上构建大规模中文短文本摘要数据集的过程，以及该数据集的相关特点。章节 5.3、介绍了在该数据集上本文构建的两种基于循环神经网络的短文本摘要生成模型。章节 5.4、介绍了两类模型的训练。章节 5.5、对构建的两类模型在本文构造的数据集上进行了实验。最后给出了本章小结。

5.2 大规模中文短文本摘要数据集

文摘自动生成被认为是自然语言处理中极其困难的任务之一，多年来其研究并没有较大进展，部分原因是缺少高质量的大规模生成式文摘数据集。构造生成式文摘数据集需要人工详细阅读整个文档并书写出其摘要。因此，仅仅靠人工的方式构造大规模的文档级生成式文摘数据集，显然不现实，以往的生成式摘要数据集规模都非常小，例如，比较流行的文摘数据集DUC [1],TAC [2],TREC [3] 包含的文摘数量通常在2,000篇英文文摘以内，对于中文生成式摘要数据来说，情况更为严重。摘要数据集的匮乏已经成为深度模型在文摘生成领域研究的瓶颈。

近年来，一些研究工作尝试通过利用用户使用互联网时，无意识产生的自然标注数据，解决自然语言处理领域一些较为困难任务的数据匮乏问题。自然标注数据的概念最早由清华大学孙茂松教授提出[104]。所谓自然标注数据是指互联网用户为了交流目的而产生的诸如，博客、网页、微博等数据。通常用户在产生这些数据时，会无意识的添加一些标签。研究人员可以利用这些标签信息，挖掘对相关研究有用的资源。由于获得篇章级的生成式文摘数据集存在较大的困难性。基于自然标注数据的思想，本文构造了一个大规模段落级的中文短文本摘要数据集。

5.2.1 数据集的构建

随着互联网的发展，个人和机构在互联网社交平台上发布信息变的越来越方便。特别地，一些具有影响力的机构或者媒体通常会在微博上注册账号并实名验证，这些账号通常称之为大“V”。这些大“V”会频繁地向公众发布信息。以人民日报为例，每天在其官方微博上推送数十条微博。由于其具有的严肃的新闻媒体性质，其在微博平台上发布的微博具有信息量丰富，语言规整等特点。由于微博发布的内容最长是140个字，其通常有若干个完整的语句组成。为了让其粉丝能够快速获得其发布的信息的核心内容，有时会配上人工书写的对微博内容的一句话总结。图5-1给出了由人民日报发出的一条微博例子。从图中可以看出，其通过一段文本描述了一个新闻事件，然后通过使用“【】”标记，新闻编辑人员对这个事件进行了高度概括。如果能将诸如这样的高质量

[1]http://duc.nist.gov/data.html

[2]http://www.nist.gov/tac/2015/KBP/

[3]http://trec.nist.gov/

的微博从互联网上海量数据中提取出来，便可以得到大规模的中文短文本摘要数据集。这种数据集将对文摘的生成以及段落表示的研究具有重要的意义。

图 5-1 一条由人民日报发出的微博

Fig. 5-1 One Weibo posted by the People's Daily

图 5-2描述了本章通过微博平台构造大规模中文短文本摘要数据的具体流程。其具体描述如下: 首先，为了保证爬取的微博质量，我们人工选择一些知名度高的新闻媒体官方微博作为种子账号。这些账号是具有较高权威性或者知名度的官方媒体，它们发布的微博内容通常比较规范，并且发布微博的频率较高。同时，为了爬取数据主题的多样性，尽可能的使种子账户涵盖的领域较广，例如，政治、新闻、军事、教育、经济、科技、体育等。一些代表性的账号如，人民日报、经济观察报、中国国防部等。通常这些机构关注的微博账号也具有较高的质量。因此，我们通过爬取它们关注的微博账户，并通过人工制定的大量规则过滤掉一些质量不是太高的微博账户。比较有代表性的过滤的规则如，其粉丝数量是否较高、其每天发布微博是否频繁、是否是微博认证的官方账户等。然后将符合规则要求的微博账户加入到微博账号数据库里。通过这种方法，我们收集了一个规模较大、影响力较高、发送微博较频繁的微博账号库。

然后，通过网络爬虫爬取微博账号库中的账号的历史微博。由于微博平台的随意性，爬取的微博里面充斥着大量的噪音数据，例如，虽然图 5-1中微博的文本是一条很好的摘要数据，但是其中的图片并不是我们需要的内容，需要过滤掉。有一些微博虽然比较干净，但是并不满足摘要的性质，如图 5-3虽然

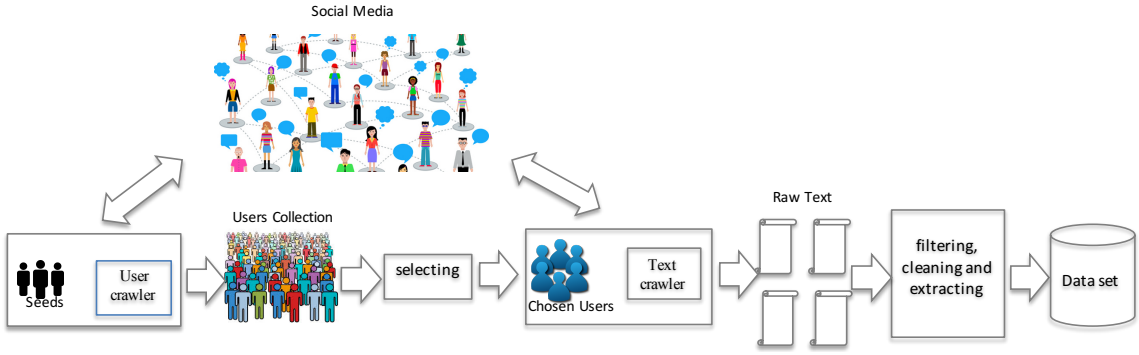


图 5-2 构造大规模中文短文本摘要数据集的流程图

Fig. 5-2 Diagram of the process for constructing the LCSTS

该条微博较为规整，但是从微博的内容并不能完全推出“【】”里面的摘要，其不符合摘要的信息与原文一致性的要求，其更像是评论性的内容。因此，也需要对这些数据进行过滤。

最终，通过观察这些微博数据的特点，我们人工书写了大量的规则过滤这些不符合摘要特点的微博。规则数量大约为 100 多条。通过这些规则的过滤，本文构造了一个大规模的中文短文本摘要数据集（Large Scale Chinese Short Text Summarization Dataset, LCSTS）。

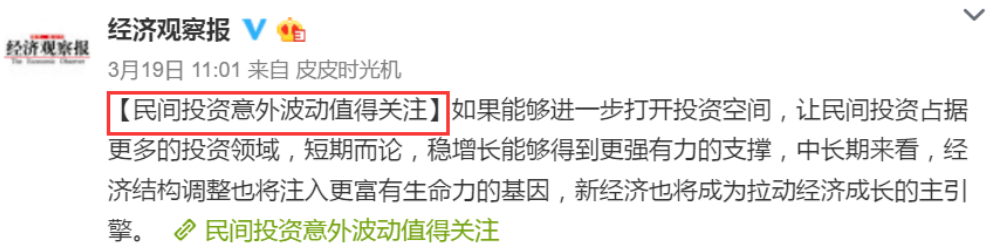


图 5-3 一条由经济观察报发出的微博，该微博中的数据不符合摘要的性质

Fig. 5-3 One Weibo posted by the Economic Observer, which can not be regarded as summarization

5.2.2 数据集特性

LCSTS 数据集包括三个部分，其数据集的统计信息如表 5-1所示，其各部分具体描述如下：

Part I 是数据集的主要部分，该部分包含 2,400,591 个短文本摘要数据，这部分数据可以作为训练集，训练深度神经网络模型生成摘要。该数据集去除了微博中的其他标签以及图片等噪音，对短文本和其对应的摘要的长度做了限制，短文本的长度超过 80 个字，摘要的长度需要在 10-30 个字之间。短文本和摘要的文本长度盒图如图 5-4所示，图中“ST”表示以字为单位的

短文本，“Segmented ST”表示分词后的短文本，“SUM”表示以字为单位的摘要，“Segmented SUM”表示分词后的摘要，红线代表中位数，盒图的上下边界代表了四分位数。从图中可以看出，如果以字为单位，多数短文本的长度大于100，与之对应的摘要长度通常小于 20；如果对短文本和摘要进行分词处理，文本长度变为 60 左右，与其对应的摘要长度大约为 10。

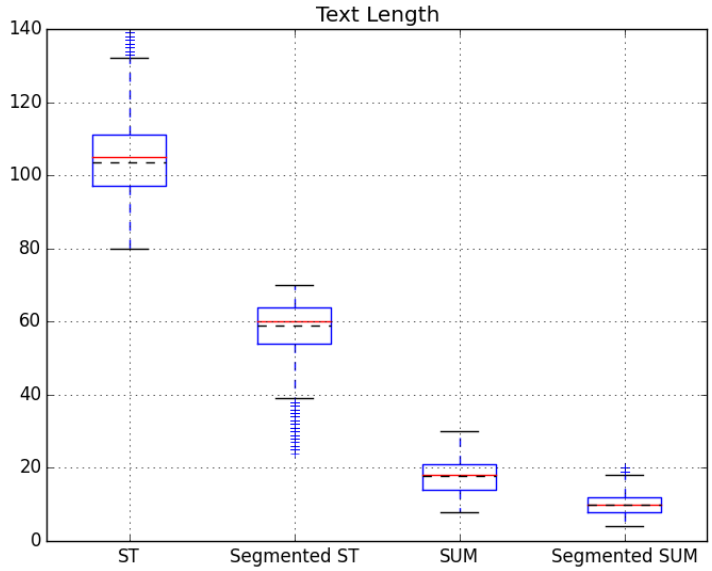


图 5-4 LCSTS 中短文本以及摘要的长度分布盒图

Fig. 5-4 Box plot of lengths for short text and summary in LCSTS

Part II 的构建，一方面，是为了检验 **Part I**数据的质量，以更细致的了解 **Part I**的特性，另一方面，是为了其他研究的需要。我们从 **Part I**中随机采样了 10,666 条数据，对这些数据分成五份，招募 5 位志愿者分别对对每份数据进行标注，标注的标准是短文本与摘要的相关性程度，“5”表示最为相关，“1”表示相关程度较低。

图 5-5给出了不同分数的数据样例。从这些样例中可以看出，人工标注分数为“3”、“4”、“5”的数据非常符合文摘的特点，这些数据的摘要，相比于原文来说非常简短，而且传达了短文本中的主要信息，表达也非常规范。同时，可以看出对于这类数据，摘要中的字符有较大部分是在短文本中没有出现的，这表明了本文构建的 LCSTS 是一种生成式摘要数据集。而得分为“1”和“2”的数据相对来说质量不高，这些数据的摘要和短文本的信息有一定的差异，从摘要信息中并不能很好地反映出短文本中的重要信息，例如，得分为“1”的数据的摘要更像是对一个事件的评论。表 5-1同时列出了 **Part II**中不同

分数的数据数量。从表中可以看出，虽然分数“1”和“2”的数据占有比率在20%以下，在一定程度上影响了数据的质量。

Short Text:商务部数据显示，中国7月实际利用外资同比大幅下降16.95%至78.1亿美元。外界有分析与近期官方对外资企业的密集反垄断调查有关。沈丹阳回应指出，“不能与对外资的反垄断调查挂钩，或者做其他没有根据的联想”。

Summarization:商务部表态反垄断：几个案子不会把外商吓回去

Human Score: 1

Short Text:记者梳理发现，2009年至今有8起福彩开奖延迟事件，至少延迟2小时，2014年5月6日第2014050期延迟开奖达4小时。8起事件中福彩中心对其中3起给出了回应，理由有通讯故障及暴雨导致的数据上传延迟。另5起均未解释原因。

Summarization:三问双色球开奖延迟：开奖为何要等数据汇总？

Human Score: 2

Short Text:7月百城住宅新建住宅平均价格为10347元/平方米，环比上涨0.87%，自去年6月以来连续14个月环比上涨。其中，广州、北京、深圳、南京涨幅均超过10%。中原张大伟认为，一二线城市因为集聚了过多资源，房价易涨难跌。

Summarization:百城房价环比“14连涨”一二线城市涨幅扩大

Human Score:3

Short Text:各团购网站移动端销售额占比均在30%以下，用户通过PC端购物习惯短时间内难以转变。未来中国餐饮O2O市场，移动端将成为餐饮O2O的战略性发展方向，也将由线上驱动转变为线下驱动。一二线城市面临增长窘境，三四线城市O2O市场蕴含机会。

Summarization:移动端成餐饮O2O的战略性发展方向

Human Score: 4

Short Text:水利部水资源司司长陈明忠今日在新闻发布会上透露，根据刚刚完成的水资源管理制度的考核，有部分省接近了红线的指标，有部分省超过红线的指标。在一些超过红线的地方，将对一些取用水项目进行区域的限批，严格地进行水资源论证和取水许可的批准。

Summarization:部分省超过年度用水红线指标 取水项目将被限批

Human Score: 5

图 5-5 PART II 中人工标记为不同分数的五个数据样例

Fig.5-5 Five examples of different scores

Part III 为了方便在该数据集上测试相关摘要生成模型的性能，我们构建了 **Part III**。为了保证测试数据集的质量，我们招募三名志愿者同时对 2000 条数据进行标注，从中挑选出三个标注人员标注一致的数据，这样得到了 1,106 条数据。在后续实验中，选择得分为“3”，“4”和“5”的数据作为测试集，评价摘要生成系统的性能。这部分的样本数据不包括在 **Part I**和 **Part II**中。

表 5-1 数据集的统计数字

Table5-1 Data Statistics

Part I	2,400,591	
Part II	Number of Pairs	10,666
	Human Score 1	942
	Human Score 2	1,039
	Human Score 3	2,019
	Human Score 4	3,128
	Human Score 5	3,538
Part III	Number of Pairs	1,106
	Human Score 1	165
	Human Score 2	216
	Human Score 3	227
	Human Score 4	301
	Human Score 5	197

5.3 基于循环神经网络的短文本摘要生成模型

短文本摘要的自动生成需要解决两个重要的问题，分别为短文本表示和语言生成。短文本可以看作是一个包含若干个句子的段落，因此本文第 3 章提出的基于深度卷积神经网络的语句表示模型并不适用于短文本。近年来，循环神经网络在语言的表示以及生成方面都表现出了优异的性能。基于循环神经网络的编码 -解码架构在多个自然语言处理任务上取得了突破，例如，机器翻译 [15, 16]、对话生成 [105] 等。基于此，本文提出基于循环神经网络的编码 - 解码框架解决短文本文摘自动生成问题。本节内容的组织安排如下，首先介绍两种常用的循环神经网络，其次，引入基于循环神经网络的短文本摘要生成模型一和二。

5.3.1 循环神经网络

理论上，循环神经网络可以对任意长度的时间序列进行建模，以记住其历史信息。然而，实际实验中，由于经典的循环神经网络模型随着序列长度的增加，在训练的过程中存在梯度消失（Vanishing Gradient Problem）的问题 [106]。换言之，当序列较长时，梯度传递到较早时刻时趋向于 0，对模型参数的更新

非常弱，使得模型对于相隔较长的历史信息的输入不能有效的训练。为了解决上述问题，研究人员对循环神经网络中的循环计算单元进行设计，提出了不同的变形。目前较为流行的循环神经网络模型是长短记忆（Long Short Term Memory）和门控循环单元（Gated Recurrent Unit,GRU）。

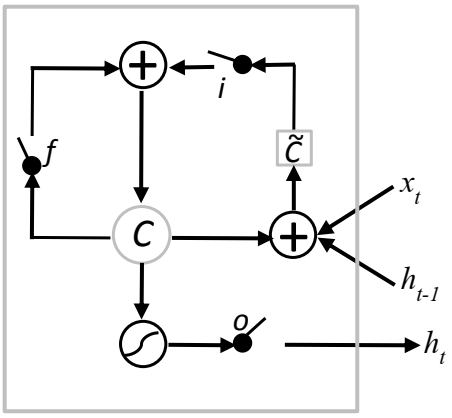


图 5-6 LSTM 的计算单元的示意图

Fig.5-6 Illustration of LSTM

LSTM 最早是由 Hochreiter 和 Schmidhuber 提出的一种循环计算单元[107]。随后人们提出了不同的变种，这里简单介绍一种由 Grave 等提出的比较常见的实现形式[108]。人们将朴素的循环神经网络中的循环计算单元替换为 LSTM 后得到的模型，称之为 LSTM 循环神经网络。LSTM 计算单元定义了记忆单元（cell）c_t 以及输入门（Input Gate）i_t、遗忘门 f_t（Forget Gate）和输出门 o_t（Output Gate）。LSTM 计算单元的基本结构如图 5-6 所示。对于 t 时刻 LSTM 计算单元的运算如下所示：

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{5-1}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{5-2}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{5-3}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{5-4}$$

$$h_t = o_t tanh(c_t) \tag{5-5}$$

受 LSTM 循环计算单元中的“门”机制的启发，Chung 等提出了一种更为简化的循环计算单元，即门循环计算单元（Gated Recurrent Unit, GRU）[109]。与 LSTM 相比，GRU 并没有单独的记忆单元，同时也不存在遗忘门。而是通过两种机制保证模型对较长句子信息的记忆能力，分别是重置门 r_t（Reset Gate）和更新门 z_t（Update Gate）。该模型结构如图 5-7所示。

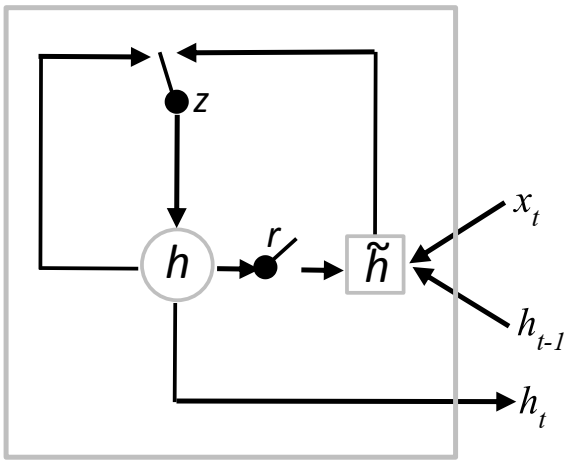


图 5-7 GRU 的计算单元的示意图

Fig. 5-7 Illustration of GRU

该模型的计算过程为如下所示，其中 $\odot$ 表示将两个向量中的对应元素相乘得到新的向量。

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{5-6}$$

$$\tilde{h}_t = tanh(W x_t + U(r_t \odot h_{t-1})) \tag{5-7}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{5-8}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{5-9}$$

由于 GRU 的结构较为简单、易于实现 [109]、收敛速度快，同时在相关任务上与 LSTM 的性能相当 [110]，在我们构建的基于循环神经网络的摘要生成模型中，选择 GRU 作为计算单元。

5.3.2 基于循环神经网络的摘要生成模型一

首先，我们构建一种基于循环神经网络的编码 -解码模型，其基本结构如图 5-8所示，该模型与 [16] 在机器翻译上的结构类似。给定一个短文本 $S = (x_0, ..., x_n)$ 以及其对应的摘要 $T = (y_0, .., y_m)$。首先通过一个循环神经网络对短文本进行建模，得到其每个词对应时刻的状态 $(h_0, ..., h_n)$。该模型使用词向量对短文本中的每个词进行表示。对第 i 个词生成状态 h_i 的过程如下，为了更为方便的描述，公式里省略了偏置项 b。

$$z_i = \sigma(W_z E_S(x_i) + U_z h_{i-1}) \tag{5-10}$$

$$r_i = \sigma(W_r E_S(x_i) + U_r h_{i-1}) \tag{5-11}$$

$$\tilde{h}_i = \tanh(W E_S(x_i) + U[r_i \odot h_{i-1}]) \tag{5-12}$$

$$h_i = (1 - z_i) \odot h_{i-1} + z_i \odot \tilde{h}_i \tag{5-13}$$

式中 $E_S(\cdot)$———从词向量矩阵中返回对应词的词向量；

h_{-1}———初始化为全 0 向量;

由于通过循环神经网络对短文本进行建模，其时刻 t 的状态理论上包含了其 $(x_0, ..., x_t)$ 的所有信息，因此，可以认为其最后一个词的状态 h_n 包含了整个句子的所有信息，也即将 h_n 看作是短文本的表示向量。

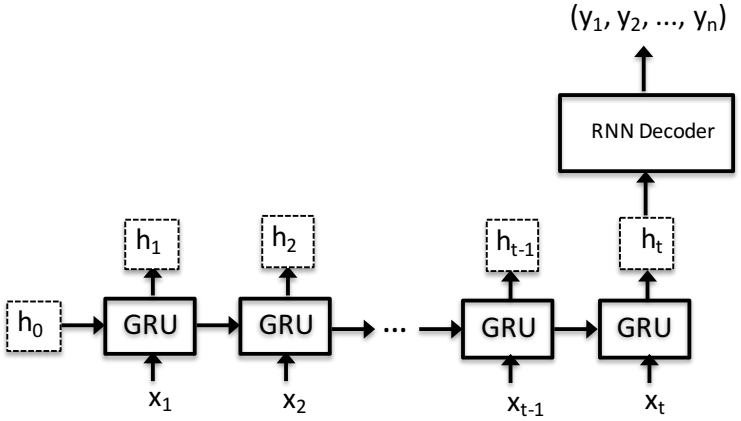


图 5-8 基于循环神经网络的摘要生成模型一的结构示意图

Fig. 5-8 The graphical depiction of Recurrent Neural Network based Summary Generation Model I

为了生成短文本 S 的摘要，我们使用另一个循环神经网络从 h_n 中，解码生成摘要 T。摘要是序列化生成的，也即在历史生成的序列 $(y_0, ..., y_{i-1})$ 的基础进行。首先，我们使用循环神经网络对已生成的序列 $(y_0, ..., y_{i-1})$ 进行建模，时刻 $i-1$ 的状态 s_{i-1} 的计算过程如下。其中，$s_{-1} = W_s h_n$，$E_T(\cdot)$ 表示从摘要端的词向量矩阵中返回对应词的词向量。

$$z_{i-1} = \sigma(W_z^{'} E_T(y_{i-1}) + U_z^{'} s_{i-2} + H_z h_n) \tag{5-14}$$

$$r_{i-1} = \sigma(W_r^{'} E_T(y_{i-1}) + U_r^{'} s_{i-2} + H_r h_n) \tag{5-15}$$

$$\tilde{s}_{i-2} = \tanh(W^{'} E_T(y_{i-1}) + U^{'}[r_{i-1} \odot s_{i-2}] + H h_n) \tag{5-16}$$

$$s_{i-1} = (1 - z_{i-1}) \odot s_{i-2} + z_{i-1} \odot \tilde{s}_{i-1}, \tag{5-17}$$

s_{i-1} 包含了已生成的 $(y_0, ..., y_{i-1})$ 的信息，同时 h_n 包含了短文本的所有信息。在此基础上，生成下一个词 y_i。同时着重强调已生成的词 y_{i-1} 对要生成词 y_i 的作用，因此，我们对 s_{i-1}, h_n, y_{i-1} 的信息进行整合得到 $\tilde{t}_i$

$$\tilde{t}_i = U_o s_{i-1} + V_o E_T(y_{i-1}) + H_o h_n \tag{5-18}$$

在 $\tilde{t}_i$ 上，使用 maxout 激活函数 [111] 得到 t_i，其中，$2l$ 为 $\tilde{t}_i$ 的维度:

$$t_i = [\max(\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j})]^{\top}_{j=1,\ldots,l} \tag{5-19}$$

最后，将 t_i 输入给一个 Softmax 分类器，首先我们得到输出 o。

$$o = W_o t_i \tag{5-20}$$

从而得到下一个词 y_i 为词典中第 k 个词的条件概率 $p(y_i = k|h_n, s_{i-1}, y_{i-1})$。最终选择词典中概率最大的词作为当前词。

$$p(y_i = k|h_n, s_{i-1}, y_{i-1}) = \frac{\exp(e^{o_k})}{\sum_{l=1}^{|V|}(e^{o_l})}, \tag{5-21}$$

5.3.3 基于循环神经网络的摘要生成模型二

模型一将 h_n 看作短文本的整体表示，同时使用这个整体表示解码生成摘要。在解码的过程中对每个时刻 h_n 都保持不变。然而，这种做法存在两个问题，其一，h_n 理论上包含了短文本的整体信息，但短文本的文字长度通常包含几句话，从数据集的特性可以看出，大部分短文本的长度在 100-140 个字之间，因此 h_n 对于历史较长的信息依然有较大的损失，同时用 h_n 表示短文本的整体信息，可能导致如数字、日期等细节性信息的丢失。其二，摘要生成的过程中，一些关键词来源于短文本中的部分片段信息，例如，图 5-1中的例子摘要中，“毅然推开为他准备的椅子”就是从短文本中直接提取的。生成“毅然推开为他准备的椅子”，只需要合理对应到原文中的片段。我们将类似这种对于生成摘要中当前词比较有意义的原文表示，称之为对应摘要关键词的上下文（Context）表示。这种思想，最早是由 Bahdanau 等提出并成功应用到了机器翻译的自动生成上 [15]。

基于此，我们在模型一的基础上引进模型二。模型二主要的特点是在生成摘要的过程中，根据已生成的历史信息的综合 s_{i-1}，对短文本中的所有时刻的状态，动态选择生成对下一个词更为有效的上下文表示 c_i，而不是固定的选择短文本的全局表示 h_n。该模型的整体架构如图 5-9所示。给定解码端第 $i-1$ 个时刻的状态 s_{i-1}，生成下一个词 y_i 时，首先，通过 s_{i-1} 对短文本编码端的状态 h_j 进行打分。

$$a(s_{i-1}, h_j) = v_a^{\top} \tanh(W_a s_{i-1} + U_a h_j) \tag{5-22}$$

然后, 对短文本编码端所有状态的打分进行归一化得到 h_j 的打分。

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{5-23}$$

最后，通过对短文本编码端的所有状态进行综合，得到预测当前词 y_i 的上下文

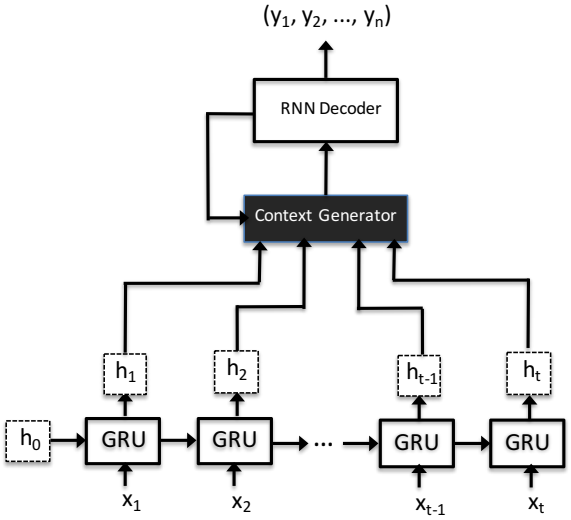


图 5-9 基于循环神经网络的摘要生成模型二的结构示意图

Fig.5-9 The graphical depiction of Recurrent Neural Network based Summary Generation Model II

表示 c_i:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{5-24}$$

因此，模型二与模型一的编码阶段完全相同，所不同的是将解码阶段公式 5-14，5-15, 5-16 和 5-18 的中的 h_n 替换为 c_i。模型二根据当前的解码状态，通过对编码端所有的状态进行选择综合，得到解码下一个词较为合理的表示。

5.4 模型训练

对于模型一其模型参数 θ_1 包括编码端参数 $(E_S, W_z, U_z, W_r, U_r, W, U)$ 和解码端参数 $(E_T, W_z', U_z', H_z, W_r', U_r', H_r, W', U', H, U_o, V_o, H_o, W_o)$ 以及模型中必要的偏置项。对于模型二，其参数 θ_2 在模型一的基础上，增加了解码阶段预测下一个词进行上下文动态选择的模型参数 (v_a, W_a, U_a)。给定的训练集中的短文本 S，以及其对应的人工书写摘要 $T = (y_0, .., y_i, ..., y_m)$，预测第 i 个词时，得到正确的 y_i 的概率如公式 5-21所示，我们的训练目标是对每个时刻，都最大化正确词的对数似然概率（log likelihood） 5-25：

$$J(\theta) = logP(y_i) \tag{5-25}$$

对于模型的训练，我们使用基于批处理的随机梯度下降算法。通过 Adadelta [112] 对学习率进行动态更新。

5.5 实验

5.5.1 实验设置

我们使用 LCSTS 的 **PART I**作为模型的训练集，将 **PART III**中评分为“3”，“4”和“5”的部分作为模型的测试集。我们对集数据进行两种方式的处理，1）以词为单位的文本表示，通过分词工具 jieba [4]对文本进行分词，从中选取 50,000 个常用词构建为词典。对于数据中不在词典中的词，统一使用字符“UNK”表示；2）以字为单位的文本表示，对原始文本进行简单的以字为单位分割，从中选取 4,000 个常用字作为词典。

模型一和模型二的编码端和解码端的词向量都是通过随机初始化的，本文没有使用常用的与任务无关的词向量学习方法初始化它们，编码端和解码端的词向量是各自单独更新并不共享。对编码循环神经网络中 GRU 计算单元中的参数 U_z, U_r, U 和解码循环神经网络的参数 (U_z', U_r', U') 随机初始化为正交矩阵。在模型二中，上下文计算部分的参数 (W_a, U_a) 是从均值为 0、标准差为 0.001^2 的高斯分布中随机采样得到的。V_a 以及其他偏置项的元素初始化为 0，其它参数矩阵以及词向量从均值为 0、标准差为 0.01^2 的高斯分布中随机采样得到。词向量的维度为 400。batch 的大小为 80，模型中的所有隐藏层的节点数为 500，学习率初始化为 1.0。Adadelta 算法中的 $\epsilon = 10^{-6}$、$\rho = 0.95$。测试时，为了得到最符合语言模型的摘要，本文选择使用集束搜索（Beam Search）算法，束的的大小（Beam Size）选择为 10，当生成“EOL”时，生成结束。

5.5.2 评价指标

为了评价基于循环神经网络的摘要生成模型的性能，我们选择使用的 ROUGE。ROUGE 评价指标是由 Lin 等提出的 [113]。ROUGE 被广泛用来评价摘要的质量，其与人工评价的结果具有高度的一致性。ROUGE 的基本原理是通过计算系统生成的摘要与人工书写的摘要之间的字符重叠数来计算，其常用的评价包括 ROUGE-N 以及 ROUGE-L。与人工书写的文摘相比，ROUGE-N 表示系统生成的文摘的 n-gram 召回率。其计算公式见 5-26：

$$ROUGE - N = \frac{\sum_{s \in R} \sum_{gram_n \in s} Count_{match}(gram_n)}{\sum_{s \in R} \sum_{gram_n \in s} Count(gram_n)} \tag{5-26}$$

[4]https://pypi.python.org/pypi/jieba/

式中 R———对于一个文本，人工写的几种不同的摘要集合；

$Count_{match}(gram_n)$———系统输出的摘要与人工书写的摘要重叠的 n-gram 的个数；

ROUGE-L 表示的是系统生成的摘要与人工书写的摘要的最大公共子序列的相似度。在我们的实验部分，我们选用由 Lin 提供的标准工具包[5]，并选用 ROUGE-1（R-1）、ROUGE-2（R-2）以及 ROUGE-L（R-L）评价系统性能。由于标准的 ROUGE 工具包，通常只能用来评价英文，我们将人工书写的摘要和系统生成的摘要中的中文字符，通过编码转换为英文字母组成的 ID。对系统的评价基于“字”为单位进行的。考虑到中文的特点，我们没有去除任何的停用词。一方面，有些出现频率较高的停用词，在部分摘要文本中占有重要的地位，例如，“了”是词“了凡四训”的一部分，虽然“了”在大部分时候可以看作停用词，然而“了凡四训”对于特定文本来说是一个非常重要的词。另一方面，该数据集中的测试集的摘要是高度浓缩的信息，其停用词非常稀少。

5.5.3 实验结果

表 5-2列出了基于循环神经网络的短文本摘要生成模型一（RNN）和模型二（RNN Contex），分别在以字（Char）为单位的输入和以词（Word）为单位的输入时的 Rouge 结果。从结果我们可以看出，由于模型二在解码阶段动态的选择生成上下文表示，其结果明显优于模型一。

表 5-2 实验结果

Table5-2 The experiment result

model	data	R-1	R-2	R-L
RNN	Word	0.177	0.085	0.158
	Char	0.215	0.089	0.186
RNN context	Word	0.268	0.161	0.241
	Char	**0.299**	**0.174**	**0.272**

为了更为直观的说明本文构建的两种模型生成的摘要质量，图 5-10给出了模型一和二分别在以字和词为单位的短文本输入时，生成的摘要实例。从该例子可以看出，模型一生成的摘要，虽然在主要信息上与短文本的内容非常相关，但是其生成的摘要的可读性较差。而模型二生成的摘要其可读性较好，并

[5]http://www.berouge.com/Pages/default.aspx

且其语义信息基本概括了原文的主要信息。特别地，模型二在以字为单位的输入上，其生成的摘要与人工书写的摘要质量具有可比性，从而验证了模型二可以通过大规模的数据训练，能够对输入文本进行很好的表示，并从中提取重要信息进行合理组织。而模型一由于只用了编码阶段的循环网络的最后一个状态进行解码，其生成的摘要的信息量以及对原文的细节的捕捉能力都有一定的丢失，例如，RNN + Char 中生成了“26”这个数字，而在输入文本中“26”根本没有出现，这种数字信息对于表达原文本的语义准确性非常重要，而模型一显然丢失了这部分信息。模型二由于对编码端的所有状态进行动态选择，以得到解码阶段的上下文表示。因此，能够很好的将这些敏感信息进行捕捉，例如，“排名 28”以及“连续六年”这些表达重要信息的数字并没有缺失。

Short Text: 9月3日，总部位于日内瓦的世界经济论坛发布了《2014-2015年全球竞争力报告》，瑞士连续六年位居榜首，成为全球最具竞争力的国家，新加坡和美国分列第二位和第三位。中国排名第28位，在金砖国家中排名最高。

Human: 全球竞争力排行榜 中国居2 8 位 居金砖国家首位

RNN+Char: 全球竞争力报告：瑞士居第六位居第26位第三位居第26位第六位居第26位第六位居第26位第六位居第26位

RNN+Word: 2014年全球竞争力排名：瑞士第一北京第第第第名单第第第名单第68位世界第第第名单第68位

RNN+Context+Char: 全球竞争力排名：中国排名第28位金砖国最高

RNN+Context+Word: 2014-2015年全球竞争力报告：瑞士连续6年居榜首中国居28位(首/3 — — — 访榜首)中国排名第28位

图 5-10 摘要生成的例子

Fig.5-10 An example of the generated summaries

从表 5-2还可以看出，模型一和模型二在以字为输入时，明显优于以词为输入时的结果。导致这种结果的原因有两个：其一，以词为输入将会极大的增加解码阶段的输出节点个数，从而导致模型训练较为困难，例如，我们通过观察发现，4,000 个字可以覆盖原文以及摘要中超过 98% 的字符。而如果以词为单位，即使我们将词典的大小从 4000 增加到了 50,000，其对数据集中的字符覆盖率仍低于 93%，进一步增加词典仍不能有效的提高字符的覆盖率。因为，其它词出现的频率非常少，分词造成了字典的稀疏性。这样反而会导致其词典规模的快速增加，从而显著增加模型的时间复杂度。其二，分词工具的不准确性，将会给数据集增加大量的噪音，虽然 LCSTS 的数据书写非常规范，但由于其是从社交平台上构建的，其书写格式存在一定独特性，并且里面存在一定量的新词，通用的分词工具在其上的性能有一定的下降。这两方面原因直接导致

了分词后数据集中，出现了大量的“UNK”字符，不仅对语言的连贯性造成一定的破坏，也导致了一定量的信息损失。

Short Text: 工厂， 大门紧锁， 约20名工人散坐在树荫下。“我们就是普通工人，在这里等工资。”其中一人说道。7月4日上午，记者抵达深圳龙华区清湖路上的深圳愿景光电子有限公司。正如传言一般，愿景光电子倒闭了，大股东刑毅不知所踪。

Human: 深圳亿元级LED企业倒闭烈日下工人苦等老板

RNN+Context+Char: 深圳愿景光电子倒闭了(图)（组图）

RNN+Context+Word: 深圳“UNK”:深圳UNK，UNK，UNK，UNK

图 5-11 含有大量’UNK’ 字符的摘要生成结果的例子

Fig.5-11 An example of the generated summaries with UNKs

如图 5-11所示，模型二以字为单位输入时，生成的摘要相对较好，而以词为单位输入时，生成的摘要中存在大量的“UNK”字符。例如，虽然，“愿景光电子”对于原文来说非常重要，然而，由于分词后其并不在词典中，在原文中其就被“UNK”字符代替。类似的问题在基于编码 -解码的神经网络机器翻译模型中也广泛存在[114]，如何减少“UNK”词对系统的性能影响，还值得进一步的深入探索。

5.6 本章小结

本章利用自然标注信息构建了一个大规模的中文短文本摘要数据集。该数据集规模大、涵盖领域广、质量较高。在此基础上，提出使用循环神经网络表示短文本，然后通过循环神经网络从短文本表示中生成摘要的架构，并构建了两种基于循环神经网络的短文本摘要生成模型。模型一将编码端循环神经网络的最后一个状态看作短文本的表示，在解码端始终利用该表示进行解码。与模型一不同的是，模型二在解码阶段通过对编码端的所有隐状态进行综合，动态的生成不同时刻的上下文表示。在测试集上进行的评测以及样例分析表明，基于循环神经网络的短文本摘要成模型一与模型二能够根据短文本的输入自动生成摘要，特别地，模型二生成的摘要无论是从信息准确性，还是语言流畅性上，都具有较高质量。

结 论

近年来，研究人员对深度模型在自然语言处理领域的应用研究表现出了强烈的兴趣。其核心是针对具体问题，设计深度模型学习数据的深层表示，从而提高相关任务的性能。本文以深度神经网络为研究手段，以文本表示及其在相关任务上的应用为研究主题，进行了深入研究。

本文主要贡献可以具体归纳为以下几点：

1. 提出了一种基于动名分离的词向量学习模型。该模型将词性信息引入到了词向量的学习过程，同时保持了文本的词序信息。受大脑动名分离结构的启发，该模型将现存的词性分为三个大类别动词、名词和其他。在预测中心词时，对同一词的动词和名词情形分别动态的选择不同的神经连接。对模型的时间复杂度分析表明，该模型的时间复杂度与 CBOW 和 Skip-gram 相当。通过对常见词的相近词的实例分析表明，该模型学习得到的词向量更为合理。在典型的命名实体识别和组块分析任务上，本文提出的模型显著地超过了对比的词向量方法。从而验证了本文提出的模型的优越性。

2. 提出了一种基于深度卷积神经网络的语句表示模型。该模型不依赖句法分析树，通过多层交叠的卷积与局部最大池化操作对语句进行建模。对语句级匹配问题进行了深入研究，提出了两种基于深度卷积神经网络的语句匹配架构。两种架构不需要依赖任何先验知识，可以广泛的应用于不同语言，不同性质的匹配任务上。特别地，架构二对两个句子的匹配表示直接进行建模，可以捕捉到句子间丰富的匹配模式。将两个匹配架构应用到三种不同语言、不同性质的匹配任务上，实验结果表明，两种架构显著地超过了其它对比模型和方法。一方面，证明了本文提出的基于深度卷积神经网络的语句表示模型的有效性，另一方面，验证了本文提出的两种基于深度卷积神经网络的语句匹配架构具有广泛的适用性。

3. 提出了一种上下文依赖的卷积神经网络短语匹配模型。该模型不仅考虑到了两个短语各自的语义信息，并且将源端短语的上下文融入到了模型中，可以有效的解决上下文敏感的源端短语的目标短语的选择问题。提出了基于上下文依赖的双语词向量学习模型，对上下文依赖的卷积神经网络短语匹配模型的词向量初始化，设计了一种“课程式”训练算法训练模型。按照训练样例语义匹配的不同层次，将训练数据分为容易、中等、困难三个等级，对数据进行合

理组织安排。对模型进行由易到难、循序渐进的训练。通过将该模型对短语对的匹配打分，加入到统计机器翻译系统中显著地提高了模型的翻译性能。

4. 构建了一个大规模的中文短文本摘要数据集。该数据集为文摘自动生成以及段落表示研究等提供了一个重要的数据。该数据集目前收到来自清华大学，中科院，台湾中央研究院，香港大学，卡内基梅隆大学等十余家研究机构的申请使用。针对摘要自动生成，提出基于循环神经网络的编码 -解码架构。同时构建了两种基于循环神经网络摘要生成模型。两种模型不需要先验知识，可以自动的从大规模数据集中学习生成摘要。其实验结果表明两种模型，特别是模型二生成的摘要，不仅信息上具有较高的准确性，语言上具有较高的流畅性。

基于以上的研究结果，本文认为以后可能存在的研究方向或改进思路包括：

1.**词表示研究** 提出的基于动名分离的词向量学习模型，只对具有代表性的三种动名分离结构进行了探讨，从实验结果上并不能得出哪一种分离结构最为合理。因此，可以结合神经心理学的研究，设计更加合理的分离结构。另外，对词向量内部蕴含的一些语言现象的深入研究，也是本人认为较为有意思的研究点。

2.**语句表示研究** 提出的基于深度卷积神经网络的语句表示模型是根据具体任务进行训练的。能否对模型进行改进从而通过大规模的无监督数据对模型进行预训练，提高数据量匮乏的任务的性能是一个值得深入研究的问题。

3.**语句级语义匹配** 提出的基于深度卷积神经网络的语句匹配架构二虽然能够对句子间的局部匹配关系进行建模，但是从结构上可以看出其具有较高的时间复杂度。而架构一虽然时间复杂度较低，但是其不能有效的对语句间的局部匹配关系进行建模。因此，未来存在两个思路，其一，对架构一进行改进，利用目前较为流行的 attention 机制，在两个句子进行建模的过程中，通过一个句子的信息辅助对另一个句子进行建模。其二，针对架构二能否设计出合理的机制，在架构二的较早阶段就大幅度减少无意义的匹配计算从而提高架构二的效率。

4. **摘要自动生成** 构建的两种基于循环神经网络的短文本摘要生成模型对短文本的建模是通过一个循环神经网络进行的，然而短文本包括了多个句子，因此，通过层次循环神经网络对短文本进行建模是未来的一个潜在研究方向。

参考文献

[1] Utgoff P E, Stracuzzi D J. Many-Layered Learning.[J]. Neural Computation, 2002, 14(10): 2497-2529.

[2] Bengio Y. Learning Deep Architectures for AI[J]. Found. Trends Mach. Learn., 2009, 2(1): 1–127.

[3] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507.

[4] Serre T, Kreiman G, Kouh M, et al. A quantitative theory of immediate visual recognition[J]. Progress in Brain Research, 2007: 33–56.

[5] Hinton G, Deng L, Yu D, et al. Deep Neural Networks for Acoustic Modeling in Speech Recognition[J]. IEEE Signal Processing Magazine, 2012(6): 82-97.

[6] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[C]. Advances in Neural Information Processing Systems 25. South Lake Tahoe, Nevada, United States: Curran Associates, Inc., 2012: 1097–1105.

[7] Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529: 484–503.

[8] Joachims T. Text Categorization with Suport Vector Machines: Learning with Many Relevant Features[C]. Proceedings of the 10th European Conference on Machine Learning. London, UK, UK: Springer-Verlag, 1998: 137–142.

[9] Lafferty J D, McCallum A, Pereira F C N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data[C]. Proceedings of the Eighteenth International Conference on Machine Learning. San Francisco, USA: Morgan Kaufmann Publishers Inc., 2001: 282–289.

[10] Collobert R, Weston J, Bottou L, et al. Natural Language Processing (almost) from Scratch[J]. Journal of Machine Learning Research, 2011, 12: 2493-2537.

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

[12] Mnih A, Hinton G E. A Scalable Hierarchical Distributed Language Model[C].

Advances in Neural Information Processing Systems 21(NIPS 2008). Vancouver, British Columbia, Canada: Curran Associates, Inc., 2008: 1081–1088.

[13] Morin F, Bengio Y. Hierarchical probabilistic neural network language model[C]. Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics. Bridgetown, Barbados: Society for Artificial Intelligence and Statistics, 2005: 246–252.

[14] Huang E H, Socher R, Manning C D, et al. Improving Word Representations via Global Context and Multiple Word Prototypes[C]. Proceedings of Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012.

[15] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. CoRR, 2014, abs/1409.0473.

[16] Sutskever I, Vinyals O, Le Q V V. Sequence to Sequence Learning with Neural Networks[C]. Advances in Neural Information Processing Systems 27. Montreal, Canada: Curran Associates, Inc., 2014: 3104–3112.

[17] Richard S, Alex P, Jean W, et al. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank[C]. Proceddings of Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, 2013: 1631–1642.

[18] Tang D, Wei F, Yang N, et al. Learning Sentiment-Specific Word Embedding for Twitter Sentiment Classification[C]. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Baltimore, Maryland: Association for Computational Linguistics, 2014: 1555–1565.

[19] Bellman R E. Dynamic Programming[M].[S.l.]: Dover Publications, Incorporated, 2003.

[20] Yoshua B, Réjean D, Pascal V, et al. A Neural Probabilistic Language Model[J]. Journal of Machine Learning Research, 2003, 3: 1137–1155.

[21] Turian J, Ratinov L, Bengio Y. Word representations: A simple and general method for semi-supervised learning[C]. Proceedings of Annual Meeting of the Association

for Computational Linguistics. Uppsala, Sweden: Association for Computational Linguistics, 2010: 384–394.

[22] van der Maaten L, Hinton G E. Visualizing High-Dimensional Data Using t-SNE[J]. Journal of Machine Learning Research, 2008, 9: 2579-2605.

[23] Miller G A. WordNet: A Lexical Database for English[J]. Communications of the ACM, 1995, 38(11): 39-41.

[24] Lev F, Gabrilovich E, Matias Y, et al. Placing search in context: the concept revisited[C]. Proceedings of the Tenth International World Wide Web Conference. Hong Kong, Hong Kong: ACM, 2001: 406–414.

[25] Socher R, Manning C D, Ng A Y. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks[C]. Proceedings of Deep Learning and Unsupervised Feature Learning Workshop-NIPS2010. Vancouver, British Columbia, Canada: Curran Associates, Inc., 2010: 1–9.

[26] Socher R, Lin C C, Manning C, et al. Parsing natural scenes and natural language with recursive neural networks[C]. Proceddings of International Conference on Machine Learning. Haifa, Israel: Omnipress, 2011: 129-136.

[27] Socher R, Pennington J, Huang E H, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions[C]. EMNLP 2011. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011: 151–161.

[28] Socher R, Huang E H, Ng A Y. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection[C]. Advances in Neural Information Processing Systems 24. Granada, Spain, EU: Curran Associates, Inc., 2011: 801–809.

[29] Cao Z, Wei F, Dong L, et al. Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization[C]. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. Austin, Texas: Association for the Advancement of Artificial Intelligence, 2015: 2153–2159.

[30] Su J, Xiong D, Zhang B, et al. Bilingual Correspondence Recursive Autoencoder for Statistical Machine Translation[C]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 1248–1258.

[31] LeCun Y, Bottou L, Bengio Y, et al. Gradient-Based Learning Applied to Document

Recognition[C]. Proceedings of the IEEE. New York, NY, USA: IEEE Press, 1998, 86: 2278-2324.

[32] Kim Y. Convolutional Neural Networks for Sentence Classification[C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1746–1751.

[33] Zeng D, Liu K, Lai S, et al. Relation Classification via Convolutional Deep Neural Network[C]. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. Dublin, Ireland: Association for Computational Linguistics, 2014: 2335–2344.

[34] Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences[C]. Proceedings of ACL. Baltimore and USA: Association for Computational Linguistics, 2014: 655–665.

[35] Yin W, Schütze H. Convolutional Neural Network for Paraphrase Identification[C]. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver, Colorado: Association for Computational Linguistics, 2015: 901–911.

[36] Yin W, Schütze H. Multichannel Variable-Size Convolution for Sentence Classification[C]. Proceedings of the Nineteenth Conference on Computational Natural Language Learning. Beijing, China: Association for Computational Linguistics, 2015: 204–214.

[37] Johnson R, 0001 T Z. Effective Use of Word Order for Text Categorization with Convolutional Neural Networks.[J]. CoRR, 2014, abs/1412.1058.

[38] Zhao H, Lu Z, Poupart P. Self-Adaptive Hierarchical Sentence Model[C]. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. Buenos Aires, Argentina: AAAI Press, 2015: 4069–4076.

[39] Meng F, Lu Z, Wang M, et al. Encoding Source Language with Convolutional Neural Network for Machine Translation[C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Beijing, China: Association for Computational Linguistics, 2015: 20–30.

[40] Zhang J, Zhang D, Hao J. Local Translation Prediction with Global Sentence Repre-

sentation[C]. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence. Buenos Aires, Argentina: AAAI Press, 2015: 1398–1404.

[41] Cho K, van Merrienboer B, Gulcehre C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[C]. Proceddings of Conference on Empirical Methods in Natural Language Processing. Doha, Qatar: Association for Computational Linguistics, 2014: 1724–1734.

[42] Kiros R, Zhu Y, Salakhutdinov R R, et al. Skip-Thought Vectors[M]. . Cortes C, Lawrence N D, Lee D D, et al. Advances in Neural Information Processing Systems 28. Montreal, Canada: Curran Associates, Inc., 2015: 3294–3302, http://papers.nips.cc/paper/5950-skip-thought-vectors.pdf.

[43] Le Q, Mikolov T. Distributed Representations of Sentences and Documents[C]. Proceedings of the 31st International Conference on Machine Learning (ICML-14). Beijing, China: JMLR.org, 2014: 1188-1196.

[44] Li J, Luong M T, Jurafsky D. A Hierarchical Neural Autoencoder for Paragraphs and Documents[C]. Proceedings of Annual Meeting of the Association for Computational Linguistics. Beijing, China: Association for Computational Linguistics, 2015: 1106–1115.

[45] Tang D, Qin B, Liu T. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification[C]. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015: 1422–1432.

[46] Denil M, Demiraj A, Kalchbrenner N, et al. Modelling, Visualising and Summarising Documents with a Single Convolutional Neural Network[J]. CoRR, 2014, abs/1406.3830.

[47] Charles Carpenter F. The structure of English: an introduction to the construction of English sentences[M]. New York, NY, USA: Harcourt, Brace, 1952.

[48] Sylvia C, Elizabeth B. The dissociation between nouns and verbs in Broca's and Wernicke's aphasia: Findings from Chinese[J]. Aphasiology, 1998, 12: 5-36.

[49] Roger W B. Linguistic determinism and the part of speech[J]. Journal of Abnormal and Social Psychology, 1973, 55: 1-5.

[50] Denes G, Dalla Barba G. G.B. Vico, precursor of cognitive neuropsychology? The

first reported case of noun-verb dissociation following brain damage[J]. Brain and Language, 1998, 62: 29-33.

[51] Pulvermúller F, Lutzenberger W, Preissl H. Nouns and verbs in the intact brain: evidence from event-related potentials and high-frequency cortical responses[J]. Cereb Cortex, 1999, 9: 497-506.

[52] G. V, D. V, J. D, et al. Nouns and verbs in the brain: a review of behavioural, electrophysiological, neuropsychological and imaging studies[J]. Neuroscience and Biobehavioral Reviews, 2011, 35: 407-426.

[53] Mikolov T, Kombrink S, Burget L, et al. Extensions of recurrent neural network language model[C]. Proceddings of International Conference on Acoustics, Speech and Signal Processing. Prague, Czech Republic: IEEE, 2011: 5528-5531.

[54] Rumelhart D E, Hinton G E, Williams R J. Neurocomputing: Foundations of Research[J]. 1988: 696–699.

[55] Lewis D. D. Y, Yang, et al. RCV1: A New Benchmark Collection for Text Categorization Research[J]. Journal of Machine Learning Research, 2004, 5: 361-397.

[56] Yanqing C, Bryan P, Rami A, et al. The Expressive Power of Word Embeddings[J]. CoRR, 2013, abs/1301.3226.

[57] Okazaki N. CRFsuite: a fast implementation of Conditional Random Fields (CRFs)[R]. http://www.chokkan.org/software/crfsuite/: [s.n.] , 2007.

[58] Fei S, Fernando P. Shallow Parsing with Conditional Random Fields[C]. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology. Edmonton, Canada: Association for Computational Linguistics, 2003: 134-141.

[59] Yeh A. More Accurate Tests for the Statistical Significance of Result Differences[C]. Proceedings of the 18th Conference on Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000: 947–953.

[60] Wilcoxon F. Individual Comparisons by Ranking Methods[J]. Biometrics Bulletin, 1945, 1: 80-83.

[61] Buzhou T, Hongxin C, Yonghui W, et al. Recognizing clinical entities in hospital discharge summaries using structural support vector machines with word representation features[J]. BMC Medical Informatics and Decision Making, 2013, 13.

[62] Demsar J. Statistical comparisons of classifiers over multiple data sets[J]. Journal of Machine Learning Research, 2006, 7: 1-30.

[63] Tjong Kim Sang E F, De Meulder F. Introduction to the conll-2003 shared task:

Language-independent named entity recognition[C]. Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4. Edmonton, Canada: Association for Computational Linguistics, 2003: 142-147.

[64] Tjong Kim Sang E F, Buchholz S. Introduction to the CoNLL-2000 Shared Task: Chunking[C]. Proceedings of the 2nd Workshop on Learning Language in Logic and the 4th Conference on Computational Natural Language Learning. Lisbon, Portugal: Association for Computational Linguistics, 2000: 127-132.

[65] Liu B, California,USA: Morgan & Claypool Publishers.

[66] Zhang D, Lee W S. Question Classification Using Support Vector Machines[C]. Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. 2003. New York, NY, USA: ACM, SIGIR '03, http://doi.acm.org/10.1145/860435.860443.

[67] Dolan B, Quirk C, Brockett C. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources[C]. Proceedings of the 20th International Conference on Computational Linguistics. Stroudsburg, PA, USA: Association for Computational Linguistics, 2004: 350–356.

[68] Kang L, Hu B, Wu X, et al. A Short Texts Matching Method Using Shallow Features and Deep Features.[C]. . Zong C, Nie J Y, Zhao D, et al. NLPCC. Shenzhen, China: Springer, 2014, 496: 150-159.

[69] Wang H, Lu Z, Li H, et al. A Dataset for Research on Short-Text Conversations[C]. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, 2013: 935–945.

[70] Dahl G E, Sainath T N, Hinton G E. Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout[C]. Proceedings of ICASSP. Vancouver: IEEE Press, 2013: 8609-8613.

[71] Mikolov T, Karafiát M. Recurrent Neural Network based Language Model[C]. Proceedings of International Conference on Spoken Language Processing(INTERSPEECH). Makuhari, Japan: IEEE Press, 2010.

[72] Bromley J, Guyon I, LeCun Y, et al. Signature Verification using a "Siamese" Time Delay Neural Network[M]. . Cowan J D, Tesauro G, Alspector J. Advances in Neu-

ral Information Processing Systems 6. Burlington, Massachusetts,USA: Morgan-Kaufmann, 1994: 737–744.

[73] Rich C, Steve L, Lee G. Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early stopping[C]. Advances in Neural Information Processing Systems. Denver, Colorado, United States: Nips Foundation, 2000: 402–408.

[74] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting[J]. Journal of Machine Learning Research, 2012, 15: 1929-1958.

[75] Lu Z, Li H. A Deep Architecture for Matching Short Texts[C]. Advances in Neural Information Processing Systems 26. South Lake Tahoe, Nevada, US: Curran Associates, Inc., 2013: 1367–1375.

[76] Rus V, McCarthy P M, Lintean M C, et al. Paraphrase Identification with Lexico-Syntactic Graph Subsumption[C]. Proceddings of Florida Artificial Intelligence Research Society Conference. Coconut Grove, Florida,USA: AAAI Press, 2008: 201-206.

[77] Brown P F, Pietra V J D, Pietra S A D, et al. The Mathematics of Statistical Machine Translation: Parameter Estimation[J]. Computational Linguistics, 1993, 19(2): 263–311.

[78] Koehn P, Och F J, Marcu D. Statistical Phrase-based Translation[C]. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1. Edmonton and Canada: Association for Computational Linguistics, 2003: 48-54.

[79] Gao J, He X, Yih W t, et al. Learning continuous phrase representations for translation modeling[C]. Proceedings of Annual Meeting of the Association for Computational Linguistics. Baltimore, Maryland: Association for Computational Linguistics, 2014: 699–709.

[80] Zhang J, Liu S, Li M, et al. Bilingually-constrained phrase embeddings for machine translation[C]. Proceedings of Annual Meeting of the Association for Computational Linguistics. Baltimore, Maryland: Association for Computational Linguistics, 2014: 111–121.

[81] He Z, Liu Q, Lin S. Improving statistical machine translation using lexicalized rule selection[C]. Proceedings of the 22nd International Conference on Computa-

tional Linguistics. Manchester, United Kingdom: Association for Computational Linguistics, 2008: 321–328.

[82] Marton Y, Resnik P. Soft Syntactic Constraints for Hierarchical Phrased-Based Translation[C]. Proceddings of Annual Meeting of the Association for Computational Linguistic: Human Language Technologies. Columbus, Ohio: Association for Computational Linguistics, 2008: 1003–1011.

[83] Bengio Y, Louradour J, Collobert R, et al. Curriculum learning[C]. Proceedings of the 26th Annual International Conference on Machine Learning. Montreal, Canada: ACM, 2009: 41–48.

[84] Liu Q, He Z, Liu Y, et al. Maximum entropy based rule selection model for syntax-based statistical machine translation[C]. Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing. Honolulu, Hawaii: Association for Computational Linguistics, 2008: 89–97.

[85] Wu H, Dong D, Hu X, et al. Improve statistical machine translation with context-sensitive bilingual semantic embedding model[C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 142–146.

[86] Xiao X, Xiong D, Zhang M, et al. A Topic Similarity Model for Hierarchical Phrase-based Translation[C]. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1. Jeju Island, Korea: Association for Computational Linguistics, 2012: 750–758.

[87] Cui L, Zhang D, Liu S, et al. Learning topic representation for smt with neural networks[C]. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore, Maryland, USA,: Association for Computational Linguistics, 2014: 133–143.

[88] Xiong D, Zhang M. A topic-based coherence model for statistical machine translation[C]. Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence. Bellevue, Washington, USA: AAAI Press, 2013: 977–983.

[89] Bengio Y. Learning deep architectures for AI[J]. Foundations and Trends® in Machine Learning, 2009, 2(1): 1–127.

[90] Zou W Y, Socher R, Cer D, et al. Bilingual word embeddings for phrase-based machine translation[C]. Proceedings of the 2013 Conference on Empirical Meth-

ods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, 2013: 1393–1398.

[91] Yang N, Liu S, Li M, et al. Word Alignment Modeling with Context Dependent Deep Neural Network[C]. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, Bulgaria: Association for Computational Linguistics, 2013: 166–175.

[92] Koehn P, Hoang H, Birch A, et al. Moses: open source toolkit for statistical machine translation[C]. Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. Prague, Czech Republic: Association for Computational Linguistics, 2007: 177–180.

[93] Stolcke A. SRILM-an extensible language modeling toolkit[C]. Proceedings of Seventh International Conference on Spoken Language Processing. Denver, Colorado, USA: International Speech Communication Association, 2002, 3: 901–904.

[94] Och F J. Minimum Error Rate Training in Statistical Machine Translation[C]. Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1. Sapporo, Japan: Association for Computational Linguistics, 2003: 160–167.

[95] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]. Proceedings of 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2002: 311-318.

[96] Doddington G. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics[C]. Proceedings of the Second International Conference on Human Language Technology Research. San Diego, California: Morgan Kaufmann Publishers Inc., 2002: 138–145.

[97] Collins M, Koehn P, Kučerová I. Clause restructuring for statistical machine translation[C]. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Ann Arbor, Michigan: Association for Computational Linguistics, 2005: 531-540.

[98] Luhn H P. The Automatic Creation of Literature Abstracts[J]. IBM J. Res. Dev., 1958, 2(2): 159–165. http://dx.doi.org/10.1147/rd.22.0159.

[99] Hovy E, Lin C Y. Automated Text Summarization and the SUMMARIST System[C]. Proceedings of a Workshop on Held at Baltimore, Maryland: October 13-

15, 1998. Baltimore, Maryland: Association for Computational Linguistics, 1998: 197–214.

[100] Martins D D A F. A survey on automatic text summarization[R].[S.l.]: CMU, 2007.

[101] Bing L, Li P, Liao Y, et al. Abstractive Multi-Document Summarization via Phrase Selection and Merging[C]. Proceedings of the ACL-IJCNLP. Beijing, China: Association for Computational Linguistics, 2015: 1587–1597.

[102] Li C, Liu F, Weng F, et al. Document Summarization via Guided Sentence Compression[C]. Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics, 2013: 490–500.

[103] Colmenares C A, Litvak M, Mantrach A, et al. HEADS: Headline Generation as Sequence Prediction Using an Abstract Feature-Rich Space[C]. Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Denver, Colorado: Association for Computational Linguistics, 2015: 133–142.

[104] Sun M S. Natural Language Procesing Based on Naturaly Annotated Web Resources[J]. Journal of Chinese Information Processing, 2011, 25(6): 26–32.

[105] Shang L, Lu Z, Li H. Neural Responding Machine for Short-Text Conversation[C]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing, China: Association for Computational Linguistics, 2015: 1577–1586.

[106] Hochreiter S, Bengio Y, Frasconi P, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies[M]. A Field Guide to Dynamical Recurrent Neural Networks. New York, NY, USA: IEEE Press, 2001.

[107] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Comput., 1997, 9(8): 1735–1780.

[108] Graves A. Generating Sequences With Recurrent Neural Networks[J]. CoRR, 2013, abs/1308.0850.

[109] Chung J, Gülçehre Ç, Cho K, et al. Gated Feedback Recurrent Neural Networks[C]. Proceedings of the 32nd International Conference on Machine Learning (ICML-

15). Lille, France: JMLR Workshop and Conference Proceedings, 2015: 2067-2075.

[110] Chung J, Gülçehre Ç, Cho K, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. CoRR, 2014, abs/1412.3555.

[111] Goodfellow I J, Warde-Farley D, Mirza M, et al. Maxout Networks.[C]. Proceddings of 30th International Conference on Machine Learning. Atlanta, USA: JMLR Workshop and Conference Proceedings, 2013, 28: 1319-1327.

[112] Zeiler M D. ADADELTA: An Adaptive Learning Rate Method[J]. CoRR, 2012, abs/1212.5701.

[113] Lin C Y, Hovy E. Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics[C]. Proceedings of 2003 Language Technology Conference (HLT-NAACL 2003). Edmonton, Canada: Association for Computational Linguistics, 2003: 138-145.

[114] Luong T, Sutskever I, Le Q V, et al. Addressing the Rare Word Problem in Neural Machine Translation[J]. CoRR, 2014, abs/1410.8206.

攻读博士学位期间发表的论文及其他成果

（一）发表的学术论文

[1] Baotian Hu, Buzhou Tang, Qingcai Chen, Longbiao Kang. A novel word embedding learning model using the dissociation between nouns and verbs[J]. Neurocomputing, 2016(171):1108-1117.（SCI，DOI:10.1016/j.neucom.2015.07.046，IF=2.392）

[2] Baotian Hu, Zhengdong Lu, Hang Li, Qingcai Chen.Convolutional neural network architectures for matching natural language sentences[C]. Advances in Neural Information Processing Systems（NIPS）, Montreal, Canada, 2014: 2042-2050.（EI 收录号：201531010811757, CCF A，谷歌引用统计 81 次）

[3] Baotian Hu, Zhaopeng Tu, Zhengdong Lu, Hang Li, Qingcai Chen.Context-Dependent Translation Selection Using Convolutional Neural[C]. The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing(ACL-IJCNLP), Beijing, China, 2015: 536-541.（EI 收录号：20154201394786, CCF A，谷歌引用统计 4 次）

[4] Baotian Hu，Qingcai Chen, Fangze Zhu. LCSTS: a large scale chinese short text summarization dataset[C]. Conference on Empirical Methods in Natural Language Processing(EMNLP), Lisbon,Portugal, 2015: 1967-1972.（EI 收录号：20161102087057，CCF B，谷歌引用统计 8 次）

[5] Xiaoqiang Zhou, Baotian Hu, Qingcai Chen, and Xiaolong Wang. Answer Sequence Learning with Neural Networks for Answer Selection in Community Question Answering[C]. The joint conference of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing(ACL-IJCNLP), Beijing, China, 2015: 536-541.（EI 收录号：20154201394865, CCF A，谷歌引用统计 8 次）

[6] Longbiao Kang, Baotian Hu, Qingcai Chen. A Short Texts Matching Method Using Shallow features and Deep features[C]. Conference on Natural Language Processing & Chinese Computing(NLPCC2014), Shenzhen, China, 2014: 150-159.（EI 收录号：20150200416759，谷歌引用统计 2 次）

[7] XiaoqiangZhou, Baotian Hu, Qingcai Chen, Xiaolong Wang. An Auto-Encoder for Learning Conversation Representation Using LSTM[C], In Proceddings of Interna-

tional Conference on Neural Information Processing (ICONIP). Istanbul, Turkey, 2015:310-317, 2015. （EI 收录号：20160101767811, CCF C）

[8] 侯永帅，张耀允，王晓龙，陈清财，王宇亮，户保田. 中文问答系统中时间敏感问句的识别和检索, 计算机研究与发展 [J]. 2612-2620, 2013.

[9] Xiaoqiang Zhou, Baotian Hu, Qingcai Chen, and Xiaolong Wang. ICRC-HIT: A Deep Learning based Comment Sequence Label System for Answer Selection Challenge[C]. In Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, Colorado, 2015:210-214.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于深度神经网络的文本表示及其应用》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：卢绪田　　日期：2016年09月25日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：卢绪田　　日期：2016 年09月25日

导师签名：　　日期：2016 年09月25日

致　谢

值此论文完成之际，谨向培养我成长的哈尔滨工业大学、指导我不断进步的老师、给予我支持的家人和朋友们致以最诚挚的感谢！

感谢我的导师陈清财教授。陈老师严谨的治学态度、对学术研究追求完美的精神，深深地影响和鞭策着我，使我受益终身。陈老师对学术问题深刻的认识，独特的见解和深入浅出的讲解给了我很多学术上的启发、指导与鼓励。本研究的定题就是在陈老师的悉心指导下展开的。在博士生学习生活期间，陈老师给我创造了良好的学习和科研环境，资助我多次参加国际会议，让我有机会能够向国际上的专业人士学习，获益颇多。每当遇到心情低落以及生活困惑的时候，总能第一时间想到陈老师，陈老师总能像朋友一样与我聊天谈心，并给出切实的帮助。能有幸成为陈老师的学生，我深感骄傲，永存感激。

感谢王晓龙教授。王老师严谨细致的工作态度，对科研工作的热情和为之付出的精力，为我在科研道路的行进树立了榜样。在博士生学习生活期间，王老师给我们创造了良好的学习和科研环境，始终把握着中心的发展方向，硕士期间王老师对我的指导，奠定了我博士期间踏实的工作心态，受益于此，我才能在博士期间脚踏实地的做项目、做科研、发论文。感谢汤步洲老师对我学习上和生活上的帮助，每每遇到问题，汤师兄总能给出富有建设性的意见。此外，感谢徐睿峰副教授、刘滨教授和丁宇新副教授对我的学习和生活的帮助和支持，博士四年有了你们生活多了很多快乐。

感谢华为诺亚方舟实验室的李航博士，吕正东博士对我在诺亚方舟实习时的指导。李航老师“论文不在多，在精，差的工作不仅不会给你加分，反而会影响你的学术声誉”的告诫，对我影响深远。吕正东博士思维的深度与大胆的想法使我的研究思路大开，并与你们合作发表了多篇高质量的文章。与你们在一起搞科研的日子是我博士期间一段非常难忘而富有成果的日子，使我受益终生。同时感谢诺亚方舟的马林博士，尚利锋博士，蒋欣博士，涂兆鹏博士，蔡涛工程师等，每天工作之余和你们在吐露港散步聊天，让我增长了大量知识。感谢和我一起实习过的中科院计算所王明轩，孟凡东，王书鑫，北航的陈燕，北大的尹珺，清华的赵晗，香港中文的彭宝霖同学。和你们的学术交流让我受益匪浅。

其次，感谢给予我无私帮助和支持的智能计算中心的各位同仁。感谢张超，张耀允、孙彬彬、黄冬、侯永帅、王丹丹、刘胜宇、周小强、相洋、桂林、

刘增建、陈毅，陈俊杰、陈涛、周继云、潘囿丞、刘羽朦，陈凯，刘欣，吴湘平等博士生。多年来，在与他们的交流中我得到了很多宝贵的意见。特别感谢周小强同学，与之合作发表了多篇论文，建立了深厚的友谊，并在我论文撰写阶段给了大量的宝贵意见。同时，我也要感谢与我合作过的师弟康龙彪、祝芳泽和师妹余丽、陈静对我研究工作的大力支持与配合。

最后，我要深深地感谢我的父母和妻子蒋丽平，感谢您们在我漫长的求学生涯中给予我无微不至的关心、支持、理解和帮助，尊重我的每一个选择，让我快乐的学习成长，您们是我最坚强的后盾，让我有勇气面对各种挑战和困难。特别要感谢我刚刚出生的女儿户睿婕，你的到来给我带来无限的欢乐和前进的动力。再次感谢所有给我帮助理解和关心爱护我的人，有您们的支持，我会更加努力！

个人简历

户保田，男，汉族，1987 年 04 月 21 日出生于山东省菏泽市。

学习经历

[1] 2012 年 09 月——至今 哈尔滨工业大学 计算机应用技术 攻读博士学位。

[2] 2010 年 09 月——2012 年 07 月 哈尔滨工业大学 计算机科学与技术 工学硕士学位。

[3] 2006 年 09 月——2010 年 07 月 山东科技大学 信息与计算科学（辅修英语） 理学学士学位和文学学士学位。

研究方向

人工智能，深度学习，自然语言处理

学术论文

在领域内高水平会议或期刊发表论文多篇，如 NIPS2014，ACL2015，EMNP2015，Neurocomputing 等。谷歌引用统计 100 余次。

获奖情况：

2015 年博士生国家奖学金。