1. 问题描述及调研

## 问题描述

近年来，在自然语言处理（NLP）领域中，神经网络和机器学习已经成为问答任务的流行方法。这些方法的一个问题是为特定域创建合适的数据集的成本。机器学习在训练和测试来自相同分布的假设下工作得很好。因此，机器学习所能解决的任务高度依赖于数据集。因此，大多数神经QA研究主要使用现有的数据集，而针对特定领域的QA所做的工作并不多。

本文主要研究回答句的选取问题。此任务将问题和上下文段落作为输入，并从上下文中请求包含给定问题答案的句子。在我们的例子中，上下文将由《圣经》中的段落组成。

《圣经》不仅是一部有影响的文学作品，也是基督教徒最重要的宗教文献。到目前为止，在QA或其他NLP任务中使用该语料库的工作还不多。甚至在今天，它仍然在宗教界和世俗团体中被广泛阅读和研究。圣经常常被视为智慧的源泉，人们转向寻找生活中重大问题的答案。一个能够使用圣经中的段落回答问题的QA系统对用户非常有益。

《圣经》的QA体系对于非基督徒，寻求更多地了解圣经可能是有用的。他们可能会问这样的问题：“耶稣是谁？”、“我死后会发生什么？”。然后系统可以输出一系列相关的诗句作为答案。同样的系统对于那些试图将《圣经》用作历史或考古文献的学者也是有用的。他们可能对诸如“巴比伦什么时候摧毁了耶路撒冷神庙？”之类的问题感兴趣。耶利哥城在哪里？“。这些问题可以从描述圣经历史方面的相关段落中得到回答。最后，一个最广泛的用例可能来自基督教团体，他们拥护圣经作为他们信仰和生活的最终权威。他们可能对从神学到实践的各种各样的问题感兴趣。例如，“救恩是因信得救，还是因行为得救？”、“我应该如何对待那些冤枉过我的人？”、“我该如何祈祷？”。虽然许多这些问题也可以通过搜索引擎来回答，但搜索引擎的结果质量常常是值得怀疑的，从基督教的角度来看，直接使用圣经中的段落来回答这些问题是很有价值的资源。

## 贡献

本研究旨在探讨以神经为基础的方法，藉由诗歌选择来解答圣经问题。

(1)由于使用神经网络方法进行有效学习需要大规模数据集，因此我们的第一个贡献涉及创建新的数据集BibleQA。数据集由圣经问题和从因特网上现有的一组问题中得到的相应诗句组成。

(2)针对《圣经》中的句子选择，设计了三种基于不同神经网络结构的答案选择模型。这些模型中的每一个都以问题和答案诗作为输入，并输出包含问题答案的诗的预测概率。

(3)我们利用迁移学习技术，通过在更大的QA数据集上预训练模型，并深入了解QA任务中使用的领域适应的效果。我们的实验还揭示了上下文长度的变化如何影响答案选择的性能，并揭示了关于各种圣经翻译的新见解。

1. 相关工作

从文本检索的角度来看，QA体现了寻找包含答案的相关文本片段并随后提取答案的任务[1]。这种观点导致了开放领域的QA，它涵盖了今天的大多数QA系统。近年来，机器学习开始应用于QA，IBM Watson是最著名的系统之一[2]。Watson背后的主要方法是广泛的数据，统计和机器学习分析。还探讨了其他几种神经网络方法。Iyyer等人。使用神经网络回答问答碗型问题，其中给定描述，任务是识别正在讨论的主题[3]。Kumar等人，扩展了具有注意机制的简单RNN模型以实现传递式推理，并逐步实现基于推理的QA[4]。Malinowski提出了一种利用CNN和LSTM结合图像识别和QA的模型[5]。

答案句子选择是一项QA任务，它涉及选择最可能包含答案的句子。早期的方法主要是基于句法的，使用疑问句和回答句应该通过句法转换松散地相互关联的思想。Wang等，提出了一个生成模型，用于转换问题的答案[6]。Wang和Manning引入了一个概率模型，该模型利用复杂的语言学特征对依赖解析树的树编辑操作进行建模[7]。其他类似的模型包括使用动态规划来找到最优的树编辑序列[8]。这些方法的主要缺点是它们需要太多的特征工程，并且难以适应新的领域。直到最近，研究人员才开始应用神经网络模型。Yu等，使用CNN模型在TREC基准测试中选择答案句子[9]。Feng等，针对应答句子选择任务，提出了几种CNN模型。Wang和Nyberg利用LSTM模型构造了一个基于问题和答案的联合向量模型[10]。

最近，使用NLP进行宗教文本挖掘的作品开始出现。Hu运用无监督学习分析谚语和诗篇。他们按内容对诗篇进行聚类，并观察结果如何与诗篇的文学形式相匹配[11]。Tschuggnall和Specht在《圣经》中探索了基于语法的文本分析用于作者归属[12]。Faigenbaum等，使用新的图像处理和机器学习算法进行作者身份检测[13]。虽然NLP无疑为圣经学者提供了一种新的圣经分析方法，但圣经本身也是计算语言学研究的宝贵语料。Buchler等人，使用圣经的七个英文译本来检查释义检测的算法[14]。《圣经》为释义检测提供了一个很好的试验平台，因为存在几个不同的译本，它们都来自同一个来源。Agi´c等人，使用圣经来学习低资源语言（如Akawaio、Aukan或Cakchiquel）的词性标记（POS），其中圣经只被部分翻译[15]。在跨域机器学习中，迁移学习和域适应已经非常成功，尤其是当我们在一个域中拥有大量数据但是相似的感兴趣域没有足够的数据用于学习时。计算机视觉主要受益于迁移学习[16]、[17]。迁移学习最近开始在质量保证中应用。Glorot等，使用迁移学习来检查一个被训练来回答来自一个知识库的问题的系统如何回答来自另一个知识库的问题[18]。Yang等人，使用迁移学习进行问题生成[19]。对于域自适应在神经模型中的应用，最常见和最直接的方法是在源数据上预训练模型，然后从目标域[20]、[21]微调数据上的参数。

1. 方法与系统结构

在我们的工作中，我们对比了三个主要的神经网络模型：一个使用递归神经网络（RNN），一个使用卷积神经网络（CNN），另一个使用由Seo等人初次提出并经过修改的双向注意流模型（BiDAF）。这三个模型都遵循相同的通用架构，后续部分将更详细地描述架构：

1. 嵌入：首先对输入的问题和答案进行预处理并将其转换为单词向量。
2. 编码：然后处理和编码嵌入的句子，以获得捕获句子的单个矢量表示。
3. 答案选择：根据编码的问题和答案，选择答案作为预测输出。

## 词嵌入

词嵌入使用分布式词向量捕获单词上下文。这里直觉得认为类似环境中的词语往往具有相似的含义[26]。这里我们使用GloVe向量和word2vec。word2vec被建模为一个浅的，双层的神经网络，它使用随机梯度下降和反向传播来迭代地使一个词嵌入更接近于其相邻的单词。该模型成功地降低了非线性隐藏层的复杂性，并使得可以在大量数据上学习高维词向量。GloVe是word2vec向量的替代无监督学习算法，它也用于获得单词的向量表示[27]。GloVe拥有预先训练的在线训练载体，这些载体由来自维基百科和各种新闻媒体的60亿个词例进行训练，使得它非常适合在SQuAD上进行训练。

由于圣经由许多单词和名字组成，这些单词和名字在其背景下是独特的，我们也为圣经训练了我们自己的单词向量。我们将所有四种前述英语翻译的组合用于单词向量训练，其中包括大约300万个单词。在向量训练中，我们使用大小为5的上下文窗口和连续的Bag-of-Words算法来训练维度为200的向量。得到的单词向量能够捕获特定于圣经的词汇表的一般语义。下面是使用导出的单词向量，为几个选定单词的一些最相似的单词进行相似性的降序排列。从这些例子中，我们看到与“上帝”最相似的这些词语涵盖了神在整本圣经中所具有的许多品质和角色。与“David”最相似的词是与他密切相关的其他名字：Saul，他的主要对手;Absalom和Solomon，他的儿子们;他的军队指挥官Joab和他最好的朋友Jonathan。

上帝：主，救世主，圣洁，怜悯，慈爱，宣誓，重新认识，拯救，耶和华，忍耐

罪：侵犯，罪恶，内疚，内疚，犯罪，原谅，犯罪，罪恶，原谅，无知

david：saul，absalom，joab，abimelech，solomon，abner，jonathan，abraham，achish，samuel

训练后的单词向量在迁移学习过程中与GloVe向量连接，以使得对BibleQA的训练更有意义。

## QA系统模型

1. baseline模型。

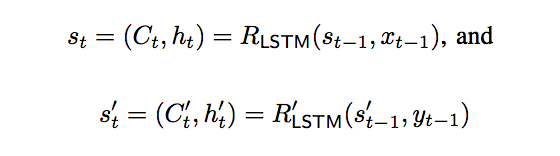
该模型作为比较所有其他结果的基础。该模型使用随机函数为每个数据点均匀随机生成[0,1]范围内的输出。baseline为我们提供了一个模型，该模型在不涉及任何学习的水平上执行，并简单地为每个问答配对分配随机预测。然后将baseline与我们的模型进行比较，以评估更复杂模型所做的改进。

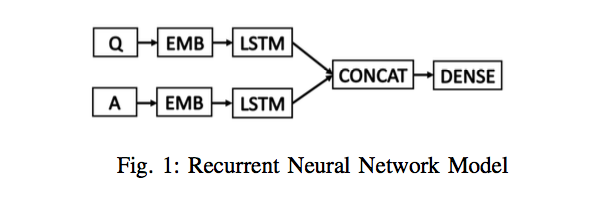
1. RNN模型。

循环网络旨在对序列进行建模，允许用户在保留结构信息的同时处理序列。由于语言的顺序性，它们在NLP任务中特别有用。循环网络由循环组成，其中特定层的输出作为输入传递回同一层。这允许信息持久化并捕获长期依赖性，例如序列中出现的依赖性。RNN最受欢迎的实现之一是长短期记忆（LSTM），它是为了减轻消失的梯度问题而引入的[28]。随着序列变长，当前单词与依赖上下文之间的距离变长。然而，这意味着序列后面步骤中的误差梯度在反向传播中迅速减小，并且没有达到较早的输入信号，因此梯度“消失”。这使得捕获相关信息变得非常困难。LSTM引入了一个充当存储单元的向量，可以随时间保留渐变。通过门控可以被认为是逻辑门的组件来控制对存储器单元的访问。

我们的RNN模型利用LSTM层来生成问答短语的向量表示。获得输出作为[0,1]之间的概率，其表示问题向量和答案向量之间的相似性。这是基于直觉，即具有更接近的向量的句子应该更相似，因此答案应该与问题更相关。

单词嵌入层将每个单词转换为单词向量。一个问题是一个单词向量序列⃗x=（x1，x2，...，xt），答案是另一个单词向量序列⃗y=（y1，y2，...，yt'）。编码程序应用两个LSTM，一个用于问题Q，另一个用于答案A：对于每个示例（⃗x，⃗y），它为每个t设置





其中st（s't）是第t个状态，Ct（Ct'）是存储单元状态，h（h'）是输出状态，R（R'）t t是LSTM LSTM

是问题（答案）的LSTM网络。输出

将是m =（Qe，Ae）其中Qe =（h1，h2，...，ht），

Ae =（h'1，h'2，...，h't）。最后，我们将问题连接起来

并回答向量，将它们传递给最后一层

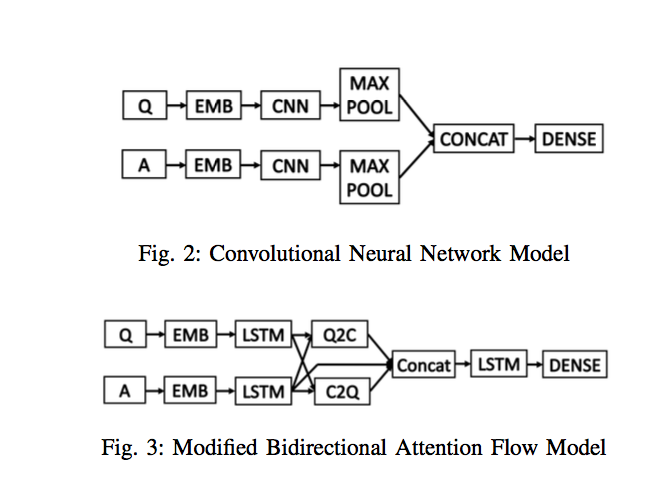
使用sigmoid激活函数σ= 1，得到1 + e-x

预测答案的可能性是问题的正确答案。图1显示了各层的概况。

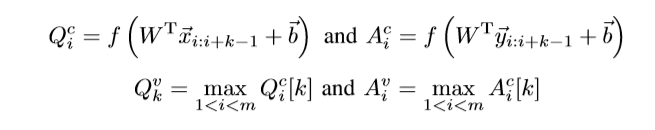
1. CNN模型。

CNN是特殊的前馈神经网络，具有完全连接的层，由卷积层和池化层组成。无论信号的位置如何，这些专用层都可用于查找输入中存在的强大的本地信息，例如，在QA任务中，句子可能包含强烈指示它作为问题答案的关键短语。在NLP中，CNN网络首先采用一系列单词并对通过k个单词的滑动窗口获得的每个n-gram序列应用滤波器。过滤器将n-gram转换为d维向量，从而捕获单词的重要属性。最后，池化层通过对矢量的每个维度进行最大或平均操作，将所有d维矢量组合成一个单独的d维矢量。然后将该最终矢量用于神经网络中的进一步处理，因为它现在包含整个序列中的一些最重要的本地信息。

我们的CNN模型使用卷积和池化层来表示问题和答案短语。我们还使用过滤层来调节重量并避免过度拟合。对于每个问题和答案序列，卷积层在序列上应用内核，使用过滤器对其进行变换，通过最大池层以获得元素最大值，最后通过返回预测的输出层。



对于每个问题X和答案Y的卷积和最大池化层，



其中k是窗口大小，f是relu激活函数f（x）= max {0，x}，W是执行线性变换的滤波器矢量，⃗b是网络的偏置参数。最后，输出层与RNN模型相同。图2显示了各层的概况。

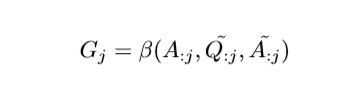
1. BiDAF模型。

针对跨度级QA任务的SQuAD数据集提出了BiDAF模型[25]。原始模型用于在上下文段落中找到问题答案的开始和结束索引：问题和上下文段落首先使用单词和字符嵌入转换为向量，然后组合使用LSTM形成短语嵌入。然后组合问题和上下文段落，为上下文中的每个单词生成一组查询感知向量。最后，另一个LSTM用于扫描上下文段落，输出层生成问题的开始和结束索引。

对于我们的句子层面的任务，我们稍微修改了原始的BiDAF模型。我们使用候选句子作为原始模型中的上下文，消除字符嵌入的使用，并仅输出答案句子作为问题的正确答案的概率。图3显示了各层的概况。

单词和短语嵌入层在使用LSTM时类似于RNN模型，其结果包含问题q和候选答案a的矩阵表示。在此之后，Q2C和C2Q是两层计算问题和答案的“注意力”，这主要是问题和答案之间的相互作用。更正式地，我们使用U：i来指示任何矩阵U的第i列向量。双向关注由第j个答案词和第k个问题词之间的相似性矩阵Sj，k确定，定义为Sj，k =α（ a：j，q：k）其中α（⃗a，⃗q）是可训练的标量函数，表示向量⃗a，⃗q之间的相似性。Q2C或查询层的上下文确定哪个答案单词与其中一个问题单词具有最接近的相似性。C2Q或查询层的上下文确定哪个问题单词与每个答案单词最相关。Sj，k和Q2C，C2Q的确切定义可以在[25]中找到。

然后将Q2C和C2Q与答案嵌入连接，以输出上下文字的查询感知矢量表示;它采用矩阵G的形式，其第j列是



其中β是一种可训练的函数，它结合了三种表示。然后将结果传递给另一个LSTM层以输出表示矩阵m，最后传递到输出层，该输出层预测作为q的答案的概率。

1. 源使用

BIBLEQA：圣经问题-回答数据集

1. 回答句子选择。

QA任务按上下文中的结构级别和任务类型进行分类。通过从上下文中选择相关句子，答案可以是句子级别;跨级，就是通过从上下文中选择文本的范围作为答案;或者可以使用谓词值和句子生成模型来生成答案。本文基于BibleQA数据集从有限的非结构化数据中执行句子选择。具体来说，输入的（Q，A）包含M个问题的列表Q =（q1，q2，... qM），以及候选答案列表A =（a1,1，a1,2，... ，a1，N1; ...; aM，1，aM，2，...，aM，NM），其中aj，k是第j个问题的第k个候选者。对于每个aj，将评估输出似然性pj，k∈[0,1]，将k作为问题qj的正确答案句子。最后，对应于qt的输出答案aj，k \*是具有最高似然性的那个，即k \* = arg maxk {pj，k | 1≤k≤Nj}。

1. SQuAD.

斯坦福问答数据集（SQUAD）是目前最大的基于跨度的数据集，包含来自500多篇维基百科文章的超过10,000个QA对[22]。数据集是基于跨度的，这意味着给定上下文段落和问题，数据集输出最可能是问题答案的文本范围。由于我们对句子级任务感兴趣，因此我们将SQuAD转换为新的句子级数据集。对于每个原始上下文段落，我们将段落分为句子。然后，我们根据原始给定的跨级答案是否在句子内来标记每个句子。如果是，则将该句子标记为1，并将同一段落中的所有其他句子标记为0。

1. BibleQA1。

虽然圣经是由40个不同的人在1000年的时间内写成的，但它讲述了关于上帝在过去，现在和将来的救赎工作的总体主题的统一故事。圣经包含旧约中的39本书和新约中的27本书。旧约包含各种文学体裁，包括历史叙事，智慧书，诗歌和预言。新约的第一部分包含有关耶稣生平和教义的故事。新约圣经的其余部分包括今天广泛阅读的指示信，这些信是基督教教义的基础。圣经是古代宗教文献的集合，对文化和社会具有无可否认的影响，影响着从语言，文学到法律和科学的各个领域[23]。通过一项研究发现，三分之二的美国人认为圣经能够解决生活中所有或大部分基本问题[24]。通过这些，我们可以看到在QA系统中使用圣经的多方面潜力。

没有现有数据集直接适用于圣经的句子级QA任务，我们开始使用一些现有的可用问题创建我们的数据集。我们使用了来自Bible2的一组免费提供的1001个琐事问题作为数据集的基础。琐事问题集包括问题，答案和圣经中与答案相关的相应诗句。

使用这个资源，我们得到了一个句子级数据集，我们将其命名为BibleQA。我们提取了围绕目标经文的经文作为候选答案。实际的经文用1标记，所有其他经文用0标记。原始列表中也有一些问题不适合用圣经来回答，特别是我们无法直接从圣经中找到答案的问题。例如“哪本圣经中的章节最多？”。因此，我们过滤掉了这些问题并手动添加了更多问题。我们在BibleQA数据集中总共得到了886个问题。下面给出一个例子：

问题：耶稣母亲的名字是什么？

第1节：[马太福音1:17]所以从亚伯拉罕到大卫的所有世代

都是十四代;从大卫到流放到巴比伦十四代;从被带到巴比伦到基督，十四代。

第2节：[马太福音1:18]现在耶稣基督的诞生就像这样;因为他的母亲玛丽与约瑟订婚，在他们走到一起之前，她被圣灵怀孕了。

第3节：[马太福音1:19]约瑟夫，她的丈夫，是一个正直的男人，不愿意让她成为一个公开的榜样，打算秘密地把她带走。

答案：[0,1,0]

值得注意的是，英语有很多不同圣经的翻译，每个都使用不同的翻译哲学，并产生略有包括相同的想法的不同的经文。我们决定使用在GitHub上找到的圣经的数字化版本。我们从四个公共领域翻译中检索经文：King James Version（KJV），Young's Literal Translation（YLT），American Standard Version（ASV）和World English Bible（WEB）。詹姆斯国王版本于1611年出版，仍然是最广泛使用的圣经翻译之一，并被许多人认为是最有价值的。 Young的文字版本，从1862年开始，遵循严格的字面翻译哲学，几乎逐字翻译成从希腊语和希伯来语的英语。1901年的美国标准版在20世纪被圣经学者使用时非常流行。世界英语圣经是ASV的更新版本，是2000年出版的四种翻译中最现代的翻译。我们选择了这四种翻译，因为它们在使用英语和翻译方法方面各不相同。翻译的比较如下所示。从比较中我们可以看出每个翻译都有其微妙的差异，但却截然不同。对于BibleQA，我们对每个问题和答案对使用了四个翻译，这意味着数据集的总大小最终会有3544个问答对。

KJV：现在耶稣基督的诞生正是这样明智的：当他的母亲玛利亚被约瑟夫聚集在一起时，在他们聚集之前，她被发现有圣灵的孩子。 （马太福音1:18）

YLT：耶稣基督的诞生就是这样的：因为他的母亲玛利亚已经与约瑟订婚，在他们聚会之前，她被发现是从圣灵中受孕的。 （马太福音1:18）

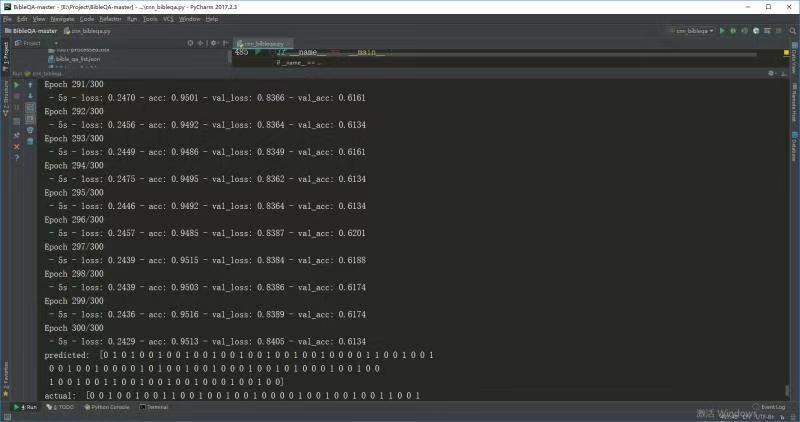
ASV：现在耶稣基督的诞生就是这样明智的：当他的母亲玛丽与约瑟订婚时，在他们走到一起之前，她被发现有圣灵的孩子。 （马太福音1:18）

WEB：现在耶稣基督的诞生就像这样;因为他的母亲玛s丽与约瑟订婚，在他们走到一起之前，她被圣灵怀孕了。 （马太福音1:18）

1. 任务分工

|  |  |  |
| --- | --- | --- |
| 学号 | 姓名 | 任务 |
| 3220180768 | 张心阳 | Report ppt 项目 文档 |
| 3220180729 | 任建国 | Report 项目 |
| 3220180798 | 巩卫参 | Report 项目 |

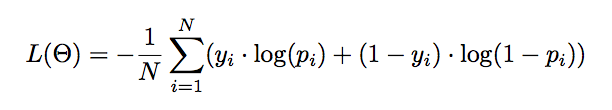
1. 程序实现



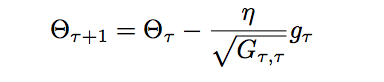
1. 实验结果及分析
2. 实验设置

我们对每个baseline，RNN，CNN和BiDAF模型进行了实验。对于任何数据集，我们使用70％的样本进行训练，30％进行测试。在70％的训练数据中，10％将用作验证集，用来防止过拟合。每当模型在训练集上训练一次，就计算关于训练和验证集的Loss和准确率。如果验证集Loss远远高于训练损失并且验证集正确率远低于训练的，那么我们可以得出结论，该模型在训练集上过拟合并且需要相应地修改模型。

我们使用GloVe词向量（维度d = 100）来训练SQuAD模型，而且同时使用GloVe和word2vec向量训练BibleQA模型，其中word2vec向量（d = 200）在4个版本的圣经上进行训练。通过二元交叉熵来将损失函数最小化，定义如下：



其中Θ是参数集，N代表训练样本的数量，pi是标签为1的概率，-pi是标签为0的概率，yi∈{0,1}是样本i的真正的标签。pi的值是使用sigmoid激活函数得到激励值，来计算参数向量的权重和输入向量。我们使用自适应梯度（AdaGrad）优化器来训练神经网络。这是一个改进过的随机梯度，具有单个参数的学习率。模型的学习速率决定了每次反向传播迭代之间的更新速率。 AdaGrad允许学习率根据参数进行调整。它针对更改频率低的参数进行更大的更新，相应的更改频率高的参数进行较小的更新，并且通常能够改进任务中稀疏的数据的收敛，例如NLP和图像识别。设gτ=∇L（Θ）为迭代τ的梯度。 AdaGrad的每个参数更新使用以下公式：



其中η是学习率，产生参数Θτ的缩放因子。通过学习过程的自适应和缩放因数，来得到每个参数的不同学习速率。对于我们的实验，我们将使用两个指标进行评估他们的表现：（1）F1得分，它是一种广泛使用的准确性指标，定义为其中P和R分别是精度和回归。 （2）平均倒数排名（MRR）：通常测量经排序的输出的准确性，以及在排名每个问题的候选句子时是实用的。 MRR定义为其中n是问题的数量，ranki是第i个问题的正确答案排序。

1. 实验1：迁移学习参数的调整
2. 目标和方法。

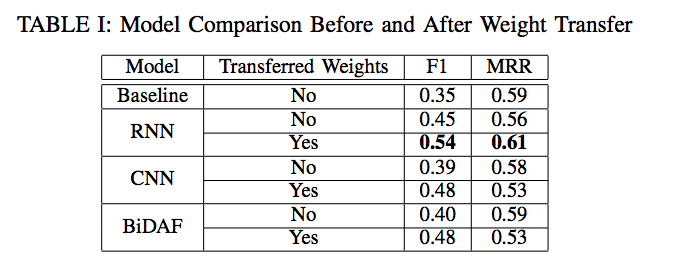
第一个实验研究了迁移学习对模型准确性的影响。我们在SQuAD数据集上预先训练模型，并将其与仅在BibleQA上与训练结果进行比较。对于每个模型，

1. 我们首先在BibleQA上运行模型以获得一组结果。
2. 然后，我们在SQuAD上训练相同的模型以获得训练的权重。
3. 最后，我们再次使用来自SQUAD的训练重量在BibleQA上运行模型，并再次进行重量微调。

然后比较每个模型的结果，看看在使用迁移学习的权重之前是否有改进。我们调整学习率和步数等参数，以找到表现最佳的模型。我们发现，对于RNN和BiDAF模型，学习率η= 0.001最佳，而CNN模型的学习率η= 0.0001最佳。我们使用早期停止机制来确定训练的最佳时的epoch，监督每个时期的验证集loss，并在模型停止改进后停止训练。我们将耐心设置为10，这意味着模型将在终止训练之前等待10个epoch。大约在20到30个epoch时会实现最佳结果。

1. 结果和分析。

表I包含权重迁移之前和之后获得的结果。我们可以看到，使用迁移的权重可以将F1结果提高0.08到0.09（大约提高20％-30％）。这表明，正如我们假设的那样，预训练对训练准确性有积极影响。然而，虽然F1分数随着权重的增加而增加，但CNN和BiDAF模型的模型的MRR下降了0.05和0.06，只有RNN模型增加了0.05。这是令人惊讶的，因为通常认为不同的评估措施之间存在相关性 - 较高的F1也会导致更高的MRR。通过考虑MRR测量的内容，似乎这些模型对正确输出的平均排名更高。然而，由于这些模型的F1也较低，因此他们不太可能选择正确答案作为排名靠前的答案。因此，虽然模型改进了F1分数，它更经常地选择正确的输出，但在模型错误地预测结果的情况下，它也将正确的答案排在较低的位置。这是一个有趣的现象，将来值得研究。在三种模型中，RNN模型在权重迁移之前和之后都具有最高的F1得分，并且在权重迁移之后具有最高的MRR。



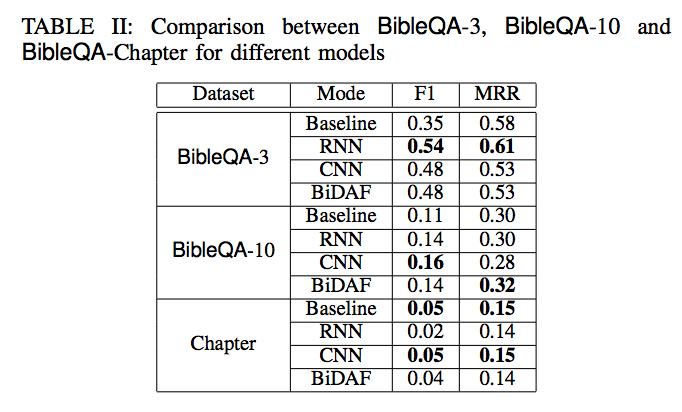
1. 实验2：回答上下文长度
2. 目标和方法。

第二个实验旨在通过改变答案上下文的长度，或者换句话说，每个问题的候选句子的数量，来找到预测结果的改变。在数据集构建阶段，在我们确定对应于问题的正确答案的经文之后，我们在正确答案周围添加不同数量的上下文经文。我们用这种方式创建了三种类型的数据集：BibleQA-3，BibleQA-10和BibleQA-chapter。对于BibleQA-3和BibleQA-10，我们分别将3节和10节经文包括在真正的经文中作为候选回答。对于章节版本，我们包括了同一章节中通常为10到60节经文的所有经文。每个RNN，CNN和BiDAF模型都在BibleQA-3上调整以获得最大准确度结果，随后，相同的模型用于预测BibleQA-10和BibleQA-chapter。对于所有数据集，我们包括所有四种圣经翻译：KJV，ASV，YLT和WEB。

1. 结果和分析。

我们使用上一次实验中的微调模型来训练每个数据集，比较更改上下文长度对模型精度的影响。表II描述了具有不同答案上下文长度的三个数据集之间的结果。在所有三个数据集中，CNN模型使用F1测量表现最佳，而BiDAF通常具有最佳MRR评分。这与上面提到的有趣现象相呼应，为什么某些模型具有更高的F1分数但是MRR低于其他模型。这需要进行更多的研究。

对于具有三节经文的最短上下文，所有模型显着改善0.13至0.19 F1，在baseline上RNN模型具有F1和MRR最佳结果。随着上下文长度增加到10节，模型准确度显着降低，比如baseline模型F1得分仅从改善到0.03到0.05。与其他模型相比，RNN的性能下降，CNN上升为具有最高F1的模型，但BiDAF成为具有最高MRR的模型。最后，在使用整个章节的最长上下文长度中，模型的表现与baseline模型大致相同，如果不是更差的话。



这表明模型尚不能用于更长的上下文。可以使用更大的数据集和更长的训练时间来训练更强大的模型，该模型可以在将来处理更长的上下文长度。

1. 实验3：翻译版本
2. 目标和方法。

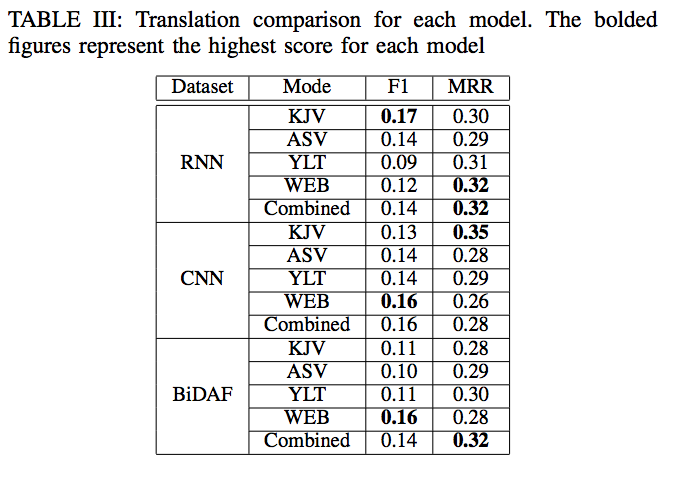
第三个实验侧重于圣经的各种英语翻译之间的差异。如上所述，我们使用4种英语翻译来训练创建数据集的单词向量。这些翻译中的每一种都有不同的翻译哲学，以及他们语言的现代性。 YLT是原始语言的最直接的英文翻译。其他三种翻译在翻译哲学中大致相同，因为它们不像YLT那样字面，而是努力真实地捕捉原始意义。按现代性排序，KJV是400年前出版的最早的翻译，而WEB是2000年最新的翻译。翻译的字面意义可能影响句子表示，因为他们可以选择使用某些单词直接翻译预期的含义。语言的现代性也可能影响单词向量，因为较旧的翻译可能包含不再使用的过时单词，因此单词向量可能不一定获得准确的表示。我们想要比较模型预测是基于字面翻译的水平还是基于语言的现代性来改变的。对于此实验，我们创建了另外四个数据集，每个数据集仅使用一个特定的转换。我们使用BibleQA-10的格式，并为每个问题选择10个候选经文。我们使用相同的三个模型，并分别比较每个翻译的结果。

1. 结果和分析。

表III比较了每个模型中每个翻译的结果。通过比较F1和MRR得分，WEB获得最佳表现，拥有RNN的最高MRR结果和CNN / BiDAF的最高F1得分。这表明使用具有更现代语言的翻译可能对QA过程有益，并且可能的情况是所使用的单词向量能够捕获更准确的含义。

KJV翻译遵循WEB翻译，并且使用F1测量以及MRR下的CNN模型具有最高的RNN模型性能。正如我们所期望的那样，对于像KJV这样的翻译，翻译中使用的一些古老语言可能是学习有用单词向量的一种威胁。事实证明，尽管选择了英语单词，KJV仍然可以在NLP任务中表现得相对较好。

对于BiDAF模型，YLT具有最高的MRR结果。然而，方差不足以使结果显着，并且YLT在任何其他模型中表现不佳。由此，我们得出结论，翻译哲学和文字水平不一定在培养QA系统中起主导作用。



1. 分析与展望

在本文中，我们利用迁移学习技术，使用BibleQA数据集研究QA任务中的域适应。从更大的SQuAD数据集中传递权重可以显着提高模型精度。这显示了使用迁移权重进行此特定任务的可能性。我们还发现RNN是性能最佳的模型，而尽管是最复杂的模型，但BiDAF的表现并不如预期。这表明更简单的架构有时仍可获得相对较好的结果。

当增加可供选择的候选句子的数量作为问题的答案时，不出所料，模型性能恶化。比较不同的圣经翻译在翻译的文字程度以及语言的现代性方面，我们发现世界英语圣经给出了最好的结果，其次是詹姆斯国王版本。语言的现代性可能归功于WEB的良好表现。与此同时，虽然KJV是几个世纪前写的，并且使用的语言与现在不同，但它仍然能够产生有用的结果。此外，杨的字面翻译是最直接的翻译并不是特别好。我们得出结论，翻译哲学以及翻译的字面意义并不一定能改善结果。

我们的系统有一定的局限性，在这里我们将为未来的研究提出一些潜在的改进和方向。 BiDAF模型的实施完全取决于最近被弃用的DeepQA库。 DeepQA背后的研究人员已将该库移植到PyTorch4，他们发现这对于NLP研究更为有效。将来，使用PyTorch而不是Keras来考虑实现并使用可靠和稳定的软件框架是值得的。我们模型中使用的句子编码方法仍然相对简单，特别是RNN和CNN模型。我们当前的编码方法的主要问题是它主要只考虑句子的语义，而不是语法。虽然它是一种已经表现得相对较好的简单方法，但为了获得更好的准确性，我们可以考虑采用兼顾语法和语义的编码方案，例如treeLSTM [30]或不同编码方案的集合。在我们的系统中使用的域适应方法也是一种简单的方法，其仅涉及预训练权重和传递权重。对改进转学方法的更多探索可能是有益的。例如，仅传输某些层的权重或以不同的学习速率调整它们。随着迁移学习在NLP研究中的应用越来越广泛，我们期望能够出现更有效的方法来改善系统。模型的准确性高度依赖于数据集的质量。我们创建的BibleQA数据集只有886个不同的问题。扩展数据集的大小在未来也是一项值得的任务，可以通过手动添加更多问题，结合圣经问题的其他来源，或可能利用可以基于文本自动生成问题的技术来完成。