# 一种基于潜在语义索引和卷积神经网络的智能阅读模型

**摘要：**

人们在日常生活中也需要阅读各式各样的电子文档，有时候他们希望不用通读文本就能获得自己想要的信息。而近年来，自然语言处理(NLP)作为人工智能的一个重要领域得到了飞速发展，因此，本文通过比较不同的方法，构建基于自然语言处理技术的智能阅读模型，以解决此类问题。

整个解题过程分为以下几个步骤：

第一步对智能阅读模型中的阅读材料以及问答训练集进行数据预处理，对训练集中数据的特征有一个清晰的了解，并对训练集进行去噪处理，除去空回答、重复回答等无效回答，防止干扰训练。

第二步选取经典文本“射雕英雄传”进行实验，通过使用词频-逆文件频率(TF-IDF)模型以及基于奇异值矩阵分解(SVD)的潜在语义索引模型(LSI)进行关键词匹配，得出较佳答案。

第三步根据经典的文本分类卷积神经网络模型，我们设计了一个更深更复杂的卷积神经网络模型。通过词嵌入后，分别对问题和关键词匹配结果中的回答进行两次卷积核大小为 3、4、5 的卷积操作，经过最大池化层后，将池化的向量连接起来。并通过使用 ReLU 激活函数，防止反向传播过程中的梯度问题（梯度消失和梯度爆炸）以及使用 Batch Normalization 批规范化，加速收敛，最终选取置信度前 15 的答案作为候选答案。随后计算得到准确率为 77.0492%， F1-score 为 0.5767，以此来评价模型的优劣，并设计测试用例查看模型运行结果。

实验最后分析并评估了该智能阅读系统的泛化能力，并简要介绍了未来的计划：通过改进损失函数，构建基于 web 开放域的问答系统以及研究基于众包的智能阅读数据服务来完善该智能阅读模型。

**关键词： TF-IDF,LSI,智能阅读模型，卷积神经网络，自然语言处理**

# An Intelligent Reading Model Based on Latent Semantic Index and Convolutional Neural Network

**Abstract:**

People need to read a variety of electronic documents in their daily lives. But sometimes they want to obtain the information they want without reading through the text. In recent years, natural language processing (NLP) has been rapidly developed as an important area of artificial intelligence. Therefore, this article compares different methods to construct an intelligent reading model based on natural language processing technology to solve such problems.

The entire problem-solving process is divided into the following steps:
The first step is to preprocess the reading materials and the question and answer training set in the intelligent reading model, to have a clear understanding of the characteristics of the training centralized data. Then we denoise the training set by eliminating null answers, repeated answers and other invalid answers to prevent interference with training.
In the second step, the classical text "Legends of the Condor Heroes" was chosen to conduct experiments. The key words were matched by using the word frequency-inverse document frequency(TF-IDF) model and the latent semantic index(LSI) model based on singular value matrix decomposition to obtain a better answer.

The third step is to design a deeper and more complex convolutional neural network model based on the classic text classification convolutional neural network model. After word embedding, the convolution operation with the size of 3, 4, and 5 is performed twice for the answers in the problem and keyword matching results. After the largest pooled layer, the pooled vectors are connected. And by using the ReLU activation function to prevent gradient problems (gradient disappearance and gradient explosion) in the back propagation process and batch normalization using Batch Normalization, accelerating convergence, the final confidence level 15 answers were selected as candidate answers. Afterwards, the accuracy rate is 77.0492% and F1-score is 0.5767. Thus the merits and demerits of the model are evaluated, and then the test cases are designed to see the results of the model.

Finally, the experiment analyzes and evaluates the generalization ability of the intelligent reading system, and introduces our future work: to improve the intelligent reading model by improving the loss function, constructing a question-answering system based on the open domain of the web, and researching intelligent reading data services based on crowdsourcing.

**Keywords: TF-IDF, LSI, intelligent reading model, natural language process, convolutional neural network**

# 目录

# 一、引言

随着互联网的高速发展以及智能设备的普及，数字阅读以方便、快捷的优势，越来越被大众所接受和认可。据中国数字阅读大会上的调研数据显示，2017 年全国数字阅读用户近 4 亿，人均电子书阅读量为 10.1 本，而纸质书阅读量仅 7.5 本。除电子书籍外，人们在日常生活中也需要阅读各式各样的电子文档，如说明书、教程、文集以及词典等。然而，在传统的数字阅读中存在用户无法精准定位关键信息的问题，即无法满足用户仅需查找文档中某些片段以获取关键信息的需求。例如，当用户需要查找法律文献中的一些段落来解决法律疑惑时，只需要理解关键部分而无需精读整个法律文献；同样，对于小说阅读，如果用户仅需了解其中的特殊细节，也不需要对整部小说进行精细化阅读。

智能交互在电子书阅读中的应用为上述问题提供了解决方案。近年来，自然语言处理(NLP)作为人工智能的一个重要领域得到了飞速发展，构建基于自然语言处理技术的智能阅读模型，通过端到端的处理技术辅助快速阅读，直接对用户的问题进行处理，无需基于关键词搜索即可直接定位文档中的相关段落，并将答案直接反馈至用户。

基于对智能阅读系统的理解和认识，本文将立足于以上背景和问题，构建基于潜在语义索引(LSI)及卷积神经网络(CNN)的智能阅读模型，完成基于限定文本的阅读问答智能交互操作。在完成对题目所给问题集的数据分析以及预处理工作后，该模型与其他主流方案相比，在 F1-Score、准确率以及泛化能力上都表现出优越的效果。本文包括引言、系统模型、实验方案、实验结果、总结与展望五个部分。

# 二、模型框架

为了使智能阅读模型能正确理解用户的问题，并跳转到答案所在文本所在行，我们提出了一种基于潜在语义索引(LSI)及卷积神经网络(CNN)的智能阅读模型，该模型主要包括三个部分：数据分析与预处理、关键词匹配以及精准匹配。模型架构图如图 1 所示。

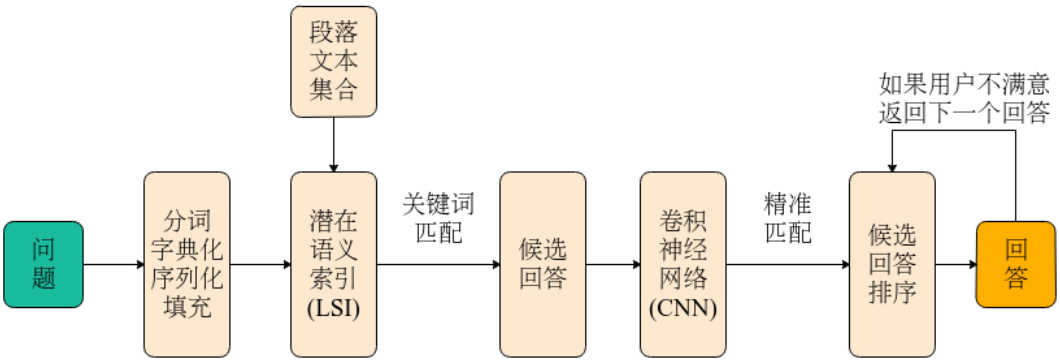


**图 1 智能阅读模型框架**

**第一步：数据分析与预处理。**我们对问题给出的数据集进行统计分析，提出该数据集进行处理时的关键挑战，并给出相应的预处理步骤；

**第二步：关键词匹配。**首先对用户提出的问题进 1 行分词，并将需要在其中寻找答案的文本构建成问答数据库。进而使用词频-逆向文件词频(TF-IDF)计算出问题以及段落的词频矩阵，再利用基于奇异值分解(SVD)的 LSI 方法将其转化为奇异矩阵，计算相似度，将相似度较大的若干个可能答案段落作为问题的粗匹配结果；

**第三步：精准匹配。**我们在经典的 TextCNN 模型上进行优化，提出一个新的 CNN 模型在粗匹配结果上进行二次优化达到精确匹配的目的。在这个过程中首先进行词嵌入，然后分别对问题和粗匹配结果中的回答进行两次卷积核大小为 3、4、5 的卷积操作，经过最大池化层后，将池化的向量连接起来。并通过使用 ReLU 激活函数，防止反向传播过程中的梯度问题（梯度消失和梯度爆炸）以及使用 Batch Normalization 批规范化，加速收敛，最终输出排序后的较佳结果。

# 三、方案介绍

## 3.1 数据分析与预处理

### 3.1.1 数据分析

高质量的数据集是模型匹配和优化的基础，对整个数据集进行分析处理可以促进对数据集的全面认知，从而更好地对数据进行特征工程编码表示，进一步提

高数据集的质量。根据分析结果，更容易选择预处理阶段的相关参数，减少重复摸索的概率。根据问题所给的数据集，我们完成了数据集的分析工作，如图 2 及表 1 所示。

表 1. 问答训练集统计表

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 分词前 | 问题集 | 问题数量/个 | 最长问题/字 | 最短问题/字 | 平均长度/字 | | |
| | | 30000 | 243 | 4 | 13 | | |
| | 答案集 | 回答数量/个 | 正确数量/个 | 错误数量/个 | 最长回答/字 | 最短回答/字 | 平均长度/字 |
| | | 477019 | 127328 | 349691 | 6425 | 0 | 95 |
| 分词后 | 问题集 | 问题数量/个 | 最长问题/词 | 最短问题/词 | 平均长度/词 | | |
| | | 30000 | 148 | 2 | 8 | | |
| | 答案集 | 回答数量/个 | 正确数量/个 | 错误数量/个 | 最长回答/词 | 最短回答/词 | 平均长度/词 |
| | | 477019 | 127328 | 349691 | 3545 | 0 | 60 |

表 1 给出了问答数据集中的统计结果，由表中可知问答训练集中的问题数量为 30000 个，最长的问题有 243 个字符，最短的问题只有 4 个字符，平均长度为 13 个音符；而分词后最长的问题有 148 个词，最短的问题只有 8 个词，句子平均长度为 8 个词。同理，问答训练集中答案的数量为 477019 个，其中正确答案为 127328 个，错误答案有 349691 个，正确答案与错误答案的比值约为 1：3，最长的回答有 6425 个字符，最短的回答是 0（空回答），回答的平均长度 35 个词；进行分词之后，最长的回答有 3545 个词，最短的回答依然只是空回答，答案的平均长度为 60 个词。

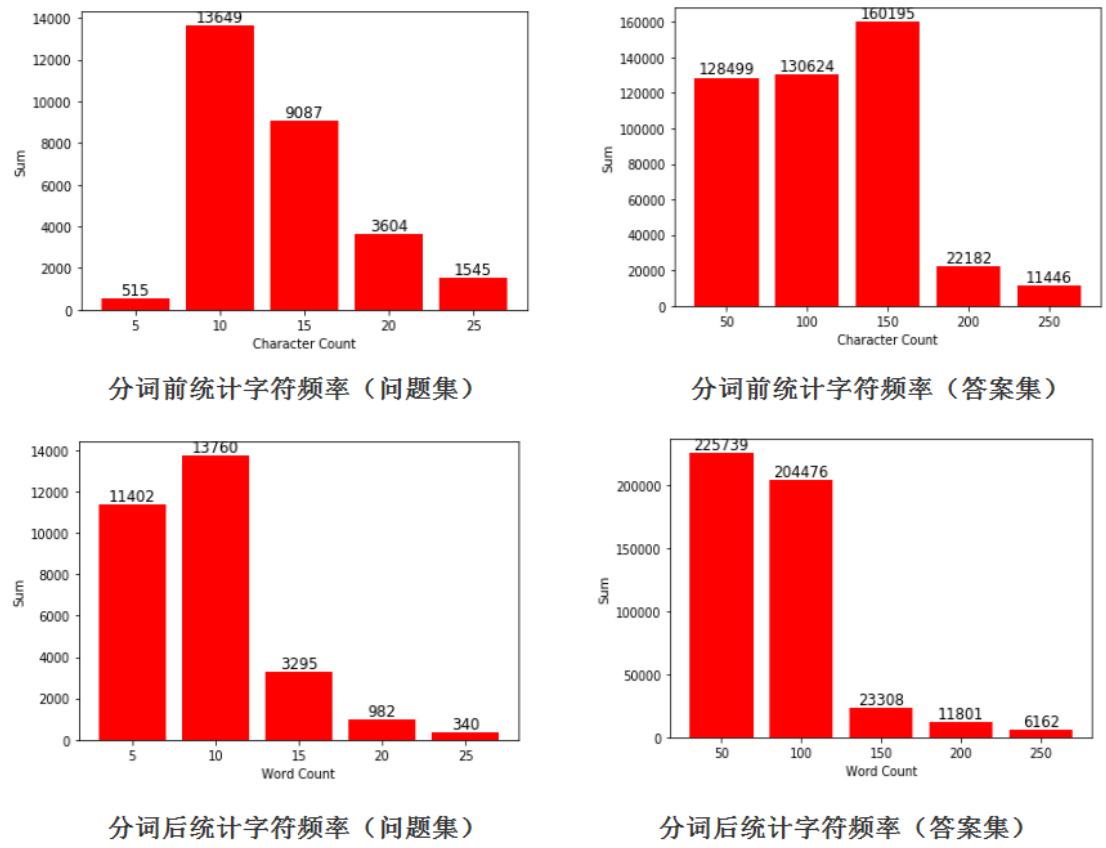


**图 2. 问答训练集统计示意图**

从图 2 中可以看出，分词前每个问题的长度大部分集中在 10~20 个字符以内，每个答案的长度大部分在 200 个字符以内。而在分词后，问题的长度集中在 15 个词以内，答案的长度大部分在 150 个词以内。

根据上述的统计分析，数据预处理的相关参数选择可以从中参考。由于模型的输入长度是固定的，因为需要选择一个输入序列长度作为参数，参数 200、400 等都是合理选择。实际上，我们进行了相关实验，并得出结果：当长度为 200 时，平均训练耗时 280 秒；当扩大长度到 400 时，平均训练耗时 400 秒。然而这两种选择的最终准确率基本相同，意味着扩大的那一部分并没有给模型带来提升的效果。我们最终也在综合了准确率和训练效率后，决定选择 200 作为模型输入序列长度。

### 3.1.2 数据预处理

在本模型中，我们主要采用将自然语言处理的问题要转化为机器学习的方式来进行学习，首先需要将自然语音进行数字化表示。例如在语音处理中，需要将音频文件转化为音频信号向量；在图像处理中，需要将图片文件转化为图片像素

矩阵。但是这两种应用场景中，音频数据和图像数据都可以采用连续数字的方式进行表示，而由于自然语言本身具有多样性的特点，利用连续数字可以完成英文字母的 ASCII 码序列表示，但是这种方法无法应对其他国家的语言，如中文、日文、韩文等。此外，自然语言的文字类型也多种多样，包括形意文字、意音文字以及拼音文字，都具有高度抽象的特征。特别地，在自然语言处理中任意两个互为近义词或者反义词的词语，也可能在拼写上毫无关系但是语义上高度相关的情况。

为了解决这个问题，我们可以采用独热表示（One-Hot Representation）和分布式表示（Distributed Representation）来完成自然语言处理数字化表示，如图 3 所示。

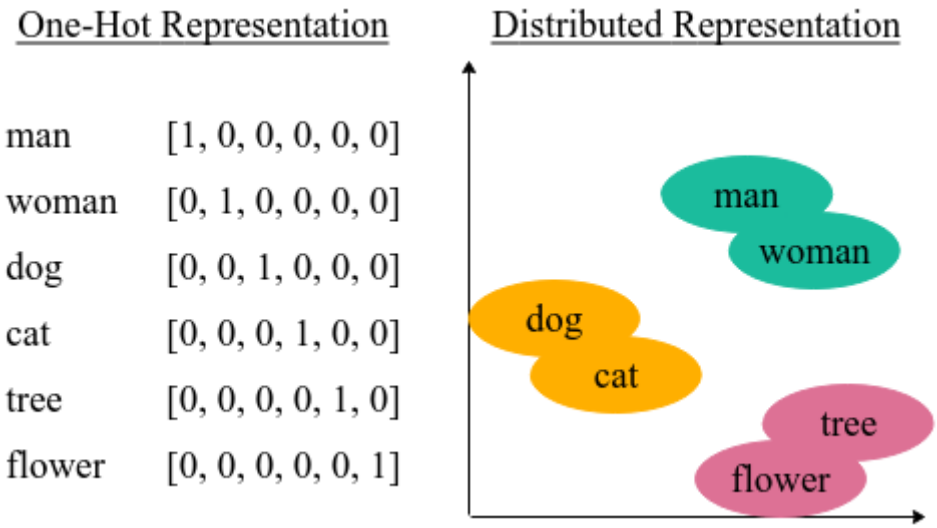


**图 3. 语言数字化示意图**

其中，独热编码是将每个词用 0 和 1 构成的稀疏向量来进行表示，其向量维度是词典大小，所有维度中只有一个元素为 1。然而这种表示方法容易主要存在两个问题，一是容易导致“维度灾难”的发生，当维度增加时，所需存储空间呈指数增长。另一个重要问题就是“词汇鸿沟”，也就是说任意两个词之间都是孤立的，光从这两个向量看不出两个词是否存在关系。

分布式表示是一类将词的语义映射到向量空间中的自然语言处理技术，每一个词用特定的向量来表示，向量之间的距离一定程度上表征了词之间的语义关系，即两个词语义相近，在向量空间的位置也相近。

例如，“小狗”和“小猫”都是动物的一种，所以它们的词向量在向量空间的距离会很相近。而“男人”和“大树”是语义上完全不同的词，所以它们的词向量在向量空间的距离会相对远。理想的情况下，我们甚至可以使用这样的关系向量来回答某些问题，例如：用“学习” + “发生的地点” = “学校”来回答“学习发生在哪里？”的问题。

分布式表示中典型的代表是词嵌入(Word Embedding)，通过神经网络或者矩阵分解等降维技术，表征文本中单词的共现信息。我们采用这种分布式表示的方法来进行智能阅读模型中的单词嵌入问题，从而避免传统语言模型中的“维度灾难”和“词汇鸿沟”情况。

在对数据集分布式表示前，我们需要对数据集进行预处理，将问题和回答转换为数字化的词向量。主要包括分词、字典化、序列化以及填充字符。在进行步骤描述时，我们同时给出一个例子对步骤进行具象呈现，即：

**例子**

**问题：“'射雕英雄传中谁的武功天下第一”**

**回答：“王重阳武功天下第一”**

**步骤一：分词。**分词采用 Python 自然语言处理工具 jieba。开发者可以指定自己自定义的词典，以便包含 jieba 词库里没有的词。虽然 jieba 有新词识别能力，但是自行添加新词可以保证更高的正确率。

**输出：['射雕 英雄传 中 谁 的 武功 天下第一', '王重阳 武功 天下第一']**

**步骤二：字典化。**将分词后的词语编号，映射到一个数字，以标识这个词语。

**输出：{'中': 5, '天下第一': 2, '射雕': 3, '武功': 1, '王重阳': 8, '的': 7, '英雄传': 4, '谁': 6}**

**步骤三：序列化。**将一个句子中的词语序列化成词向量列表。

**输出： [[3, 4, 5, 6, 7, 1, 2]], [[8, 1, 2]]**

**步骤四：填充字符。**神经网络的输入数据为固定长度，因此需要对序列进行填充或截断操作。小于固定长度的序列用 0 填充，大于固定长度的序列被截断，以便符合所需的长度。

**输出：[[0 0 0 3 4 5 6 7 1 2]], [[0 0 0 0 0 0 0 8 1 2]]**

经过上述预处理流程，我们构建了训练数据集和测试数据集：tokenizer.pkl (语料字典)、train_a.npy (训练问题集)、test_a.npy (测试问题集)、train_q.npy (训练回答集)、test_q.npy (测试回答集)、train_y.npy (训练标签集)以及 test_y.npy (测试标签集)。这些词向量数据集，将用于后续的模型“嵌入层”中进行训练，训练出来的词向量可以更好的适应自然语言处理任务。

## 3.2 关键词匹配

在本文所提出的智能阅读模型中，我们根据词频-逆向文件频率模型(TF-IDF)以及其相应的优化模型潜在语义索引(LSI)进行关键词匹配，下面我们将对两个模型做详细介绍。

### 3.2.1 词频-逆向文件频率模型

词频-逆向文件频率模型(TF-IDF)的主要思想是指在一篇文章中，某个词语的重要性与该词语在这篇文章中出现的次数成正相关，同时与整个语料库中出现该词语的文章数成负相关。其中：TF(term frequency)：词频，表示一个词语与一篇文章的相关性。计算时用该词在一篇文章中出现的次数除以文章的总词数。IDF(inverse document frequency)：逆向文件频率，表示一个词语的出现的普遍程度。可以表示为 log（总文章数/出现该词语的文章数）。

一篇文章中某个词语的重要程度，可以标记为词频和逆向文件词频的乘积。基于词频-逆向文件频率模型的主要思想，我们构建了这样一个智能阅读模型：

$$\mathrm{TF}(\text{词频}) = \frac{\text{某个词在段落中的出现次数}}{\text{文章出现次数最多的词语的次数}}$$

$$\mathrm{IDF}(\text{逆文档频率}) = \log\left(\frac{\text{语料库中段落总数}}{\text{包含该词的段落数}}\right)$$

计算得出：

$$\mathrm{TFIDF} = \mathrm{TF} * \mathrm{IDF}$$

在基于文章的词语重要性与词语在文章中出现的位置不相关的假设下，根据上述公式我们可以获取文档中每个词语的 TF-IDF 值，再结合余弦定理，我们就可以计算问题与段落的相似度了。

首先，把语料库中每个短路表示成向量空间模型。为了方便阐述，做以下符号定义：文章 D 中出现所有词语的集合标记为$W = (w_1, w_2 \dots w_M)$。通过 TFIDF 算法，可以得到包含句子 d 中每个词语 TF-IDF 值的向量，记做$t = (t_1, t_2 \dots t_M)$，其中$t_1$表示$w_1$在 d 中的 TF-IDF 值。

于是可以将要比较的问题 d1 与段落 d2 表示为 TF-IDF 值的向量：

$$d1 = (t_{11}, t_{12} \dots t_{1M})$$

$$d2 = (t_{21}, t_{22} \dots t_{2M})$$

最后利用余弦定理计算相似度:

$$\cos\theta = \frac{\sum_{i=1}^{n} t_{1i} * t_{2i}}{\sqrt{\sum_{i=1}^{n}(t_{1i})^2} * \sqrt{\sum_{i=1}^{n}(t_{2i})^2}}$$

当余弦值越接近 1 时，表明问题 d1 与段落 d2 越相似。

### 3.2.2 潜在语义索引模型

像上述所提到的词频-逆向文件频率模型中，主要是利用单词的词频作为特征，但是没有考虑到语义，无法处理同义词等情况。为了解决一词多义以及一义多词的现象，有学者提出基于词频-逆向文件频率模型的优化模型，即潜在语义索引模型(LSI 模型)。例如，“苹果”不仅仅指的是可食用的水果，也可能指代苹果公司或者苹果手机；而“快乐”与“高兴”，表达的都是喜悦的心情。

特别地，在检索的过程中仅仅按照余弦相似性，并不能很好的处理一词多义以及一义多词的问题，因此 LSI 模型采用了基于奇异值分解（SVD）的方法，利用 SVD，将使用 TF-IDF 方法计算得出的词频矩阵转化为奇异矩阵，再将词语和文本映射到一个新的空间进行降维。因此在单词-文档矩阵中不太相似的两个句子，可能会在语义空间中比较相似。

SVD 是将一个$m * n$的词语矩阵W分解为三个矩阵的过程，其中 $\sum$ 是一个非负对角矩阵，对角线上元素为W的奇异值。

$$W_{m*n} = U_{m*m} * \sum_{m*n} * V_{m*n}^{T}$$

为了降低矩阵的维度到k，SVD 的分解可以近似的写为：

$$W_{m*n} = U_{m*k} * \sum_{k*k} * V^{T}_{k*n}$$

对于智能阅读模型，在搜索答案段落时，可以这样来描述 SVD：输入的有m个问题，每个问题有n个词。而$W_{ij}$则对应第i个段落的第j个词的特征值，使用的是基于预处理后的标准化 TF-IDF 值。k是我们假设的主题数，一般要比文本数少。SVD 分解后，$U_{il}$对应第i个段落和第l个问句的相关度。$\sum_{j*m}$ 对应第j个词和第m个词义的相关度。$\sum_{l*m}$ 对应第l个主题和第m个词义的相关度。

这样我们通过一次 SVD，就可以得到段落和问句的相关度，词和词义的相关度以及词义和主题的相关度。

## 3.3 精准匹配

在关键词匹配之后得到次优集的回答集合，我们引入了一个基于卷积神经网络的模型进行二次优化精准匹配从而选出精选回答，并对候选回答的排序。在介绍本文提出的模型之前我们首先对面向自然语言处理的通用卷积神经网络模型进行介绍。

### 3.3.1 卷积神经网络

CNN 在图像处理方面已经取得瞩目的成绩，各种模型层出不穷：VGG、Inception、ResNet、DesNet 等，而在应用也是遍地开花：图像识别、图像分割、图像检索等。然而 CNN 不仅在图像领域表现优秀，在自然语言处理方面也是大有用武之地：情感分析、文本分类、问答系统等。

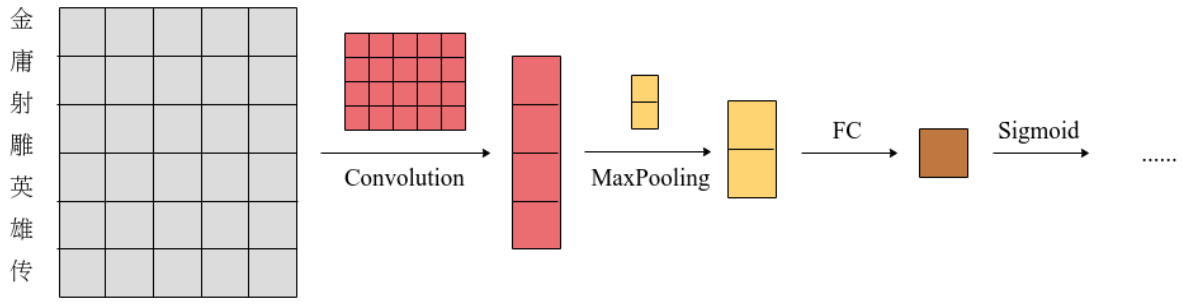


**图 4. 通用的 CNN-NLP 模型示意图**

如图 4 所示，一个通用的面向自然语言处理的 CNN-NLP 模型结构主要包括

卷积层、池化层、全连接层以及激活函数。网络结构包含：

**卷积层 (Convolution Layer)**是 CNN 的重要组成部分，利用词嵌入处理技术，我们可以将 CNN 运用在自然语言处理的各种任务。主要区别为，在图像中，卷积核通常是对图像的一小块像素区域进行计算；而在自然语言处理中，卷积核通常是对文本所构成的词向量进行计算。

**池化层 (Pooling Layer)**主要置于卷积层之后，对卷积后的输出做降采样。由于一个区域有用的特征极有可能在另一个区域同样适用，池化的过程实际上是对卷积后的某块特征区域求最大值或者求最平均值。这些统计特征不仅起到降维作用，同时还可以防止过拟合。

**全连接层 (Fully Connected Layer)**在整个卷积神经网络中起到“分类器”的作用。

**激活函数 (Activation Function)**主要是用做非线性变换。由于神经网络的目标是用于实现非线性的复杂函数，因此引入非线性激活函数可以使神经网络尽可能地逼近复杂函数。没有激活函数带来的非线性，多层神经网络和单层无异，甚至无法实现异或门等简单函数。常见的激活函数有：Sigmoid,Tanh,ReLU 等。

### 3.3.2 模型设计

我们提出了一个基于 CNN 的精准匹配模型，该模型拥有多输入（即成对输入的问题 Q 和回答 A）以及单输出（输出 0 到 1 之间的浮点数，其中 0 代表问答毫无关系，1 代表问答完全匹配），其模型架构如图 5 所示。

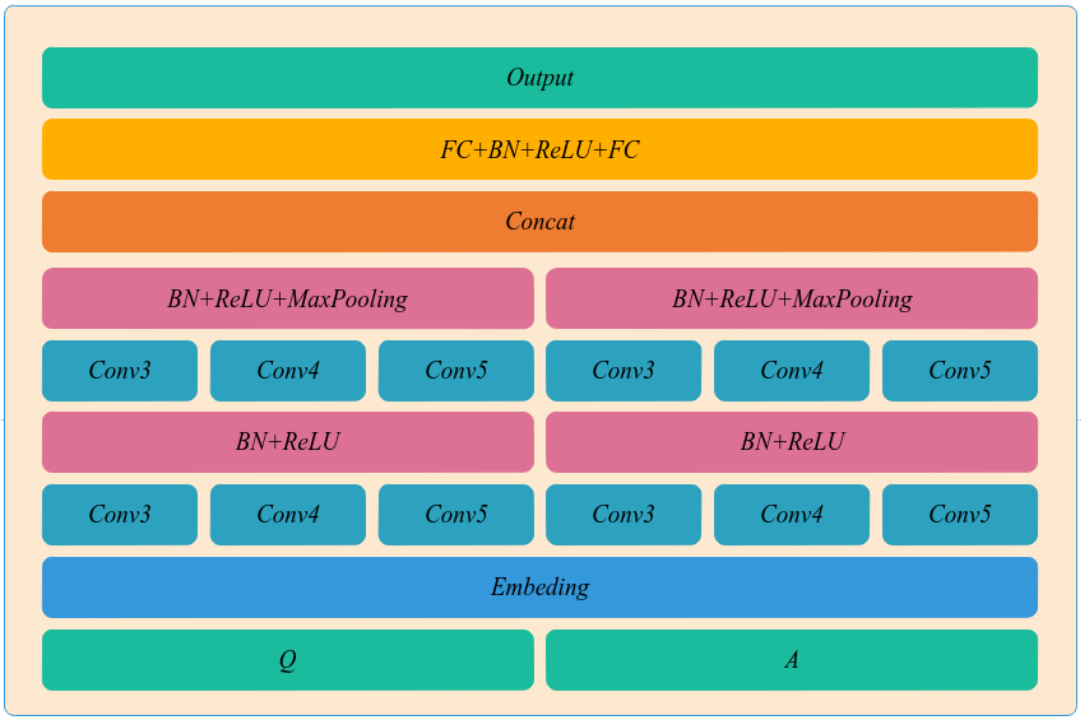


**图 5. 所提出的模型示意图**

该模型分为 8 层，如下为每层的详细介绍：

①Q/A：模型分别读入问题和回答。

②Embedding：分别对问题和回答进行词嵌入，该层会在每次迭代中训练词向量，训练出来的词向量可以更好的适应自然语言处理任务。

③Conv3/Conv4/Conv5：分别对问题和回答进行两次卷积核大小为 3、4、5 的卷积操作，提取问答特征。

④BN+ReLU：使用批规范化(Batch Normalization，简称 BN)，加速收敛；同时使用线性整流函数(Rectified Linear Unit，简称 ReLU)激活函数，防止反向传播过程中的梯度问题（梯度消失和梯度爆炸）。

⑤Conv3/Conv4/Conv5：再次进行同③的卷积操作，进一步提取问答特征。

⑥BN+ReLU+MaxPooling：再次进行同④的批规范化(BN)和激活函数(ReLU)操作，紧接着经过最大池化层(MaxPooling)，对数据进行降维，降低后续全连接层的复杂度。

⑦Concat：将池化的向量连接起来。

⑧FC+BN+ReLU+FC：最后一步的 FC 起到“分类器”的作用，而第一步的 FC 则是起到降维的作用，如果直接进入最后分类阶段，神经元参数过多，容易导致模型过拟合。

上述流程的本质是文本二分类，即输出代表问题和回答的匹配程度。对问题的多个候选回答进行匹配预测，将匹配置信度排序后，即可得到智能阅读系统对该问题的输出选择。

# 四、实验结果

## 4.1 实验环境

在我们的模型验证过程中，我们主要基 Ubuntu 16.04 的操作系统，实验环境为 128 G 的内存容量，8T 的固态硬盘容量，GTX 1080Ti * 4 的 GPU，主要以 Python 为开发语言，并基于 Keras / Tensorflow 开发框架完成模型的构建。

## 4.2 评价指标

本模型采用的评价指标主要包括准确率以及 F1-Score。准确率（Accuracy）即对于给定的测试数据集，模型正确分类的样本数与总样本数之比。准确率在某些场合下确实可以有效地评价一个模型的好坏，然而在一些极端情况下，却显得不是那么重要了。例如：某个地区的总人数为 100 万人，而人群中患有某种病的人数只有 100 人。一个负责检测行人是否患有该病的模型只需要持续将行人归为“不患病”一类，即可获得超过 99% 的概率。

$$\mathbf{Accuracy} = \frac{\text{“预测正确”的样本数}}{\text{总样本数}}$$

显然单纯使用准确率是不够的，我们引入 F1-Score 来解决上述现象，作为我们判断模型好坏的另一个指标。F1-Score 是精确率（precision）和召回率（recall）的调和函数。为了符合赛题评分标准，作出如下修改：

$$\mathbf{F1-score} = \frac{\text{“预测标签为1且真实标签也为1”的样本数}}{\text{“标签为1”的样本数} + \text{“真实标签为1”的样本数}}$$

## 4.3 实验结果

首先我们先给出本模型的实验结果：

**测试问题 1：丐帮帮主是谁？**

1. 5953 0.24029719829559326 奉立帮主是丐帮中的第一等大事，丐帮的兴衰成败，倒有一大半决定于帮主是否有德有能。当年第十七代钱帮主昏喑懦弱，武功虽高，但处事不当，净衣派与污衣派纷争不休，丐帮声势大衰。直至洪七公接任帮主，强行镇压两派不许内奸.丐帮方得在江湖上重振雄风。这些旧事此日与会群丐尽皆知晓，是以一听到要奉立帮主，人人全神贯注，屏息无声。
2. 6099 0.2300946315129598 鲁有脚道：“自来打狗棒法，非丐帮帮主不传，简长老难道不知这个规矩？”
3. 6009 0.19652195771535239 丐帮自洪七公接掌帮主以来，在江湖上从未失过半点威风，现下洪七公一死，新帮主竟如此软弱，群丐听了他这几句言语，无不愤恨难平。
4. 5961 0.1742761731147766 鲁有脚侧目斜睨杨康，心道：“凭你这小子也配作本帮帮主，统率天下各路丐帮？”伸手接过竹杖，见那杖碧绿晶莹，果是本帮帮主世代相传之物，心想，“必是洪帮主感念相救之德，是以传他。老帮主既有遗命，我辈岂敢不遵？我当赤胆忠心的辅他，莫要堕了洪帮主建下的基业。”于是双手举杖过顶，恭恭敬敬的将竹杖递还给杨康，朗声说道：“我等遵从老帮主遗命，奉杨相公为本帮第十九代帮主。”众丐齐声欢呼。
5. 6876 0.15333682298660278 “到了岳州后，丐帮大会君山。他事先悄悄对我说道：洪恩师曾有遗命，着他接任丐帮的帮主，我又惊又喜，实在难以相信，但见丐帮中连辈份最高的众长老对他也是十分敬重，却又不由得我不信。我不是丐帮的人，不能去参预大会，便在岳州城里等他，心里想着，他一旦领袖丐帮群雄，必能为国为民，做一番轰轰烈烈的大事出来，将来也必能手刃大寇，为义父义母报仇。

在以上列出的置信度前 5 的答案中，答案 1 中“自洪七公接任帮主以来...”显然给出了解答，说明丐帮帮主为“洪七公”，所在段落对应 txt 文本中的 5953 行。

**测试问题 2：江南七怪分别是哪几位**

1. 944 0.4063337246576945 江南六怪这时已知那男子并非她丈夫，只是一个被她捉来喂招练功的活靶子，这女子自必是铁尸梅超风了。
2. 3373 0.37404680252075195 江南六怪面面相觑，都是又惊又喜：“靖儿从哪里学来这样高的武功？”
3. 7584 0.34434278806050617 欧阳伯伯拦在墓门，那江南六怪如何能再逃脱毒手？这是个瓮中捉鳖之计啊。”
4. 1137 0.3165022134780884 “全真教下弟子丘处机沐手稽首，谨拜上江南六侠柯公、朱公、韩公、南公、全公、韩女侠尊前：江南一别，忽忽十有六载。七侠千金一诺，间关万里，云天高义，海内同钦，识与不识，皆相顾击掌而言曰：不意古人仁侠之风，复见之于今日也。”
5. 1574 0.29010117053985596 江南六怪与郭靖晓行夜宿，向东南进发，在路非止一日，过了大漠草原。

在以上列出的置信度前 5 的答案中，答案 4 中“谨拜上江南六侠柯公、朱公、

韩公、南公、全公、韩女侠…”显然给出了解答，说明江南七怪分别为“柯公、朱公、韩公、南公、全公、韩女侠”，至于为什么只有 6 人，是因为还有一人在小说中去世过早，而七人感情过深，故剩余 6 人依然自称江南七怪。所在段落对应 txt 文本中的 1137 行。

其次，为了验证本文所提出的模型优势，我们将分别在模型的三个处理阶段引入其他主流的模型进行比较。

在数据预处理阶段，我们对比了独热表示和分布式表示，由于分布式表示较好地解决了“维度灾难”和“词汇鸿沟”现象，我们在这个阶段选择了分布式表示作为语言数值化方案。

在关键词匹配阶段，我们对比了词频-逆向文件频率模型 、潜在语义索引模型、Doc2Vec-DM、Doc2Vec-DOW 模型，由于潜在语义索引模型在词频-逆向文件频率模型的基础上，采用了奇异值分解，保留了一定的语义信息，进一步优化了搜索结果。而 Doc2Vec-DM、Doc2Vec-DOW 根据文本单词的空间向量匹配，面对全文搜索显得有些吃力。我们在这个阶段选择了 LSI 模型作为关键词匹配方案。

在精准匹配阶段，我们与 FastText, CNN with Word2Vec, Bi-LSTM 以及 Attention 四个模型做了对比，综合训练时间、训练效率、准确率、F1-Score 以及泛化能力的评估，本文提出的模型均得到优秀的表现。

### 4.3.1 数据预处理阶段实验结果

经过数据预处理流程，我们构建了训练数据集和测试数据集：tokenizer.pkl（语料字典）、train_a.npy （训练问题集）、test_a.npy （测试问题集）、train_q.npy（训练回答集）、test_q.npy（测试回答集）、train_y.npy（训练标签集）以及 test_y.npy（测试标签集）。

### 4.3.2 关键词匹配阶段实验结果

**测试问题："射雕英雄传中谁的武功天下第一？"**

以下分别为词频-逆向文件频率模型、潜在语义检索模型、Doc2Vec-DM、Doc2Vec-DBOW4 个模型的匹配结果，以词频-逆向文件频率模型匹配结果中的答案 4 为例：

1. 2880 0.18251502513885498 武功天下第一的王真人已经逝世，剩下我们四个大家半斤八两，各有所忌。

其中 2880 指的是该句子所在 txt 文本的行数，0.1825150…则表示对的是该句子的置信度。

**词频-逆向文件频率模型匹配结果**

2. 1061 0.25669920444488525 第五回　弯弓射雕(1)
3. 1172 0.25669920444488525 第五回　弯弓射雕(2)
4. 3880 0.20211602747440338 郭靖涨红了脸，答道：“我想，王真人的武功既已天下第一，他再练得更强，仍也不过是天下第一。我还想，他到华山论剑，倒不是为了争天下第一的名头，而是要得这部《九阴真经》。他要得到经书，也不是为了要练其中的功夫，却是相救普天下的英雄豪杰，教他们免得互相所杀，大家不得好死。”
5. 2880 0.18251502513885498 武功天下第一的王真人已经逝世，剩下我们四个大家半斤八两，各有所忌。
6. 8377 0.18021109700202942 你上得华山来，妄想争那武功天下第一的荣号，莫说你武功未必能独魁群雄，纵然是当世无敌，天下英雄能服你这卖国好徒么？”

**潜在语义索引模型匹配结果**

1. 2880 0.6837087869644165 武功天下第一的王真人已经逝世，剩下我们四个大家半斤八两，各有所忌。
2. 7965 0.6464896798133385 两人回到帐中，这番当真研习起《九阴真经》上的武功来，谈论之下，均觉对方一年来武功大有长进，均感欣慰。
3. 8377 0.6456590890884399 你上得华山来，妄想争那武功天下第一的荣号，莫说你武功未必能独魁群雄，纵然是当世无敌，天下英雄能服你这卖国好徒么？”
4. 2626 0.5858802199363708 丘处机道：“韩女侠，天下武学之士，肩上受了这样的一扳，若是抵挡不住，必向后跌，只有九指神丐的独家武功，却是向前俯跌。只因他的武功刚猛绝伦，遇强愈强。穆姑娘受教时日虽短，却已习得洪老前辈这派武功的要旨。她抵不住王师弟的一扳，但决不随势屈服，就算跌倒，也要跌得与敌人用力的方向相反。”
5. 5277 0.5856705904006958 武术中有言道：”未学打人，先学挨打。”初练粗浅功夫，却须由师父传授怎生挨打而不受重伤，到了武功精深之时，就得研习护身保命、解穴救伤、接骨疗毒诸般法门。须知强中更有强中手，任你武功盖世，也难保没失手的日子。这《九阴真经》中的“疗伤篇”，讲的是若为高手以气功击伤，如何以气功调理真元，治疗内伤。至于折骨、金创等外伤的治疗，研习真经之人自也不用再学。

**Doc2Vec-DM 匹配结果**

1. 1158 0.8241669535636902 正自怔怔出神，突然听到华筝的声音在后叫道：”郭靖，快来，快来！”
2. 2608 0.8154236674308777 郭靖在赵王府中见过包惜弱的居所，听到这里，心下已是恍然。

3. 2441 0.8090466260910034 郭靖听得语声，心中大喜，叫道：“师父，快救弟子！”
4. 7378 0.8052639365196228 第三十五回　铁枪庙中(1)
5. 6354 0.8042024970054626 两人走入林中，郭靖将黄蓉背起，仍由她指点路径，一步步的向外走去。

**Doc2Vec-DBOW 匹配结果**

1. 5177 0.9147704839706421 宫内带刀护卫巡逻严紧，但周、郭、黄轻身功夫何等了得，岂能让护卫发见？洪七公识得御厨房的所在，低声指路，片刻间来到了六部山后的御厨。
2. 3014 0.9035613536834717 这词黄蓉曾由父亲教过，知道是岳飞所作的《小重山》，又见下款写着“五湖废人病中涂鸦”八字，想来这“五湖废人”必是那庄主的别号了。但见书法与图画中的笔致波磔森森，如剑如戟，岂但力透纸背，直欲破纸飞出一般。
3. 4405 0.9006547331809998 他武功既强，眼力又高，搜罗的奇珍异宝不计其数，这时都供在亡妻的圹室之中。黄蓉见那些明珠美玉、翡翠玛瑙之属在灯光下发出淡淡光芒，心想：
4. 1240 0.8996094465255737 那道人道：“睡觉之前，必须脑中空明澄澈，没一丝思虑。然后敛身侧卧，鼻息绵绵，魂不内荡，神不外游。”当下传授了呼吸运气之法、静坐敛虑之术。
5. 2428 0.8995554447174072 郭靖道：“眼不视而魂在肝、耳不闻而精在肾、舌不吟而神在心、鼻不香而魄在肺、四肢不动而意在脾，是为五气朝元。”

由以上匹配结果可以看出，潜在语义索引模型效果最好,匹配的段落含有问题“谁的武功天下天下第一”的答案，可以回答“王真人是武功天下第一的人”，词频-逆向文件频率模型次之，虽然匹配出包含正确答案“王真人”的段落，但是其对应的置信度较低，置信度较高的答案又无法正确回答问题。而 Doc2Vec 模型无法正确提取关键信息,匹配得出的段落与问题无关，答非所问，匹配效果较差。我们在使用足够多的问题进行关键词匹配后，结果依然显示 LSI 的效果最好。

### 4.3.3 精准匹配阶段实验结果

在该部分，我们主要比较了训练效率、训练效果以及泛化能力。将对比模型及本文提出的模型统一训练十次后对模型的训练时间取平均值得出模型训练效率如表 2 所示。

表 2. 模型训练效率对比

| 模型 | FastText | CNN with Word2Vec | Bi-LSTM | Attention | Ours |
|---|---|---|---|---|---|
| 时间/s | 94 | 455 | 2888 | 3889 | 300 |

从表 2 我们可以看出，实验结果表明：FastText 模型所需的训练时间最少，主要是因为该模型在词嵌入后，只有一个简单的平均池化层在隐藏层中，模型架

构较为简单，除了考察训练效率之外还需要考虑其表现的正确率（如图 6 所示，其准确率远远低于我们所提出的模型）。本文所提出的的模型训练效率除 FastText 模型外最优，表明我们在构建复杂模型提高准确率的同时还降低了算法的复杂度，保持较高训练效率。

CNN with Word2Vec 模型虽然使用预训练词向量，却比我们不使用预训练词向量的训练效率低，可能是由于 Word2Vec 的向量较为稠密，在计算上产生较高的复杂度。Bi-LSTM 模型由于需要维持过去和未来所有时间步的隐藏状态，所以训练起来比较慢。Attention 模型基于 Bi-LSTM 模型，在训练过程中还要计算单词注意力权重，最后通过加权得到一个表示问题或回答的向量，进一步增加了训练时间。

训练效果如图 6 所示，从图中可以看出 FastText、CNN with Word2Vec、Attention 模型尽管在训练集上获得较高准确率，但是在验证集上的效果却出现越训练越差的情况，即所谓的过拟合现象。而我们的模型和 Bi-LSTM 模型在训练集和验证集上的准确率均波动不大，总体趋向稳定状态。

准确率以及 F1-Score 的对比数据如图 7 所示，从图中可以看出 FastText 模型由于较为简单，所以在两种指标上均表现较差，而我们的模型以及其他对比模型都表现出较高的准确率。特别地，我们的模型相较于其他模型获得了最高的 F1-Score 分值，也就是说我们的模型在不仅能够处理好匹配情况，还能在面对正面样本少于负面样本的情况下，准确地筛选出所需的正面样本。


图 6. 精准匹配阶段训练效果示意图

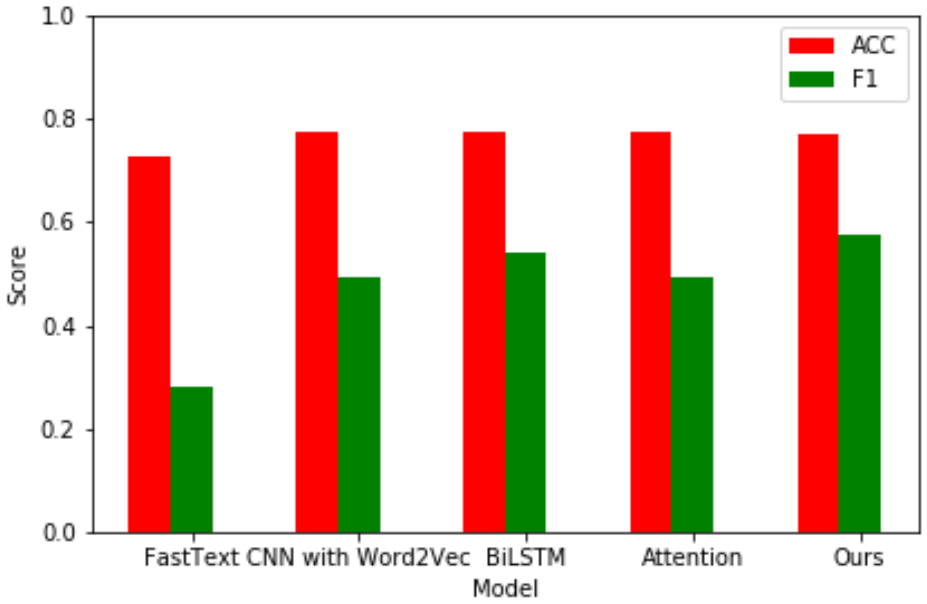


**图 7. 精准匹配阶段结果示意图**

泛化能力是指机器学习算法对新鲜样本的适应能力，模型学习的目的是学到隐含在数据对背后的规律，对于具有同一规律的学习集以外的数据，经过训练的网络也能给出合适的输出，该能力称为泛化能力。

模型好坏还需要从泛化能力上考虑，验证模型对新鲜样本的适应能力。通常期望经训练样本训练的网络具有较强的泛化能力，也就是对新输入给出合理响应的能力，应当指出并非训练的次数越多越能得到正确的输入输出映射关系。网络的性能主要用它的泛化能力来衡量。

在本文验证过程中，泛化实验使用百度开源数据集 WebQA，该数据集含有 44 万条问答记录。实验前需要将 WebQA 数据集转换为赛题数据集格式：answer 对应赛题数据集的 label，其中 no_answer 代表 0，其余回答代表 1。

在泛化数据集上各个模型的准确率和 F1-Score 如图 8 所示，从图中我们可以看出，单个模型泛化效果最好的是依然是我们所提出的模型，准确率和 F1-Score 甚至高于其他模型的平均水平大约 5%。

与其他模型的泛化效果差距这么大，可能的原因首先表现为数据集数量较少，过拟合的情况严重。其次数据集质量较低，由于 WebQA 数据集来源于百度知道等问答社区，社区存在大量复制粘贴现象，以及答非所问的现象。再者数据集标记中存在一定的噪音，可能打标的时候工作人员自身水平有限，导致数据错误分

类现象。而我们的模型在应对上述情况时，较好地适应了数据集，不会因为数据集的情况出现较大波动。

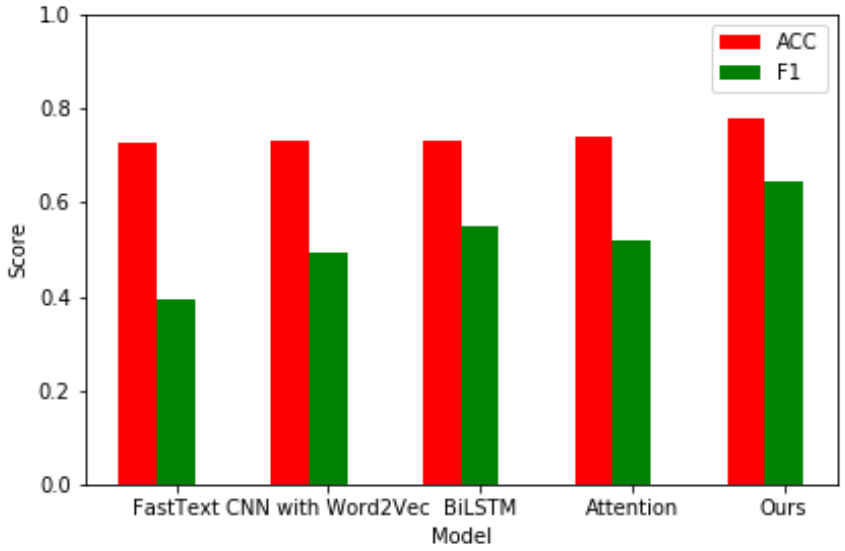


**图 8. 模型泛化能力示意图**

除了对以上指标或方案进行对比之外，我们还研究了模型融合对智能阅读的影响。在机器学习研究领域，使用单一模型的风险是很大，泛化能力表现较弱模型表现不稳定。具体表现为模型可能对数据集存在片面性，不能考虑到所有影响因素。特别地，模型可能对数据集存在依赖性，容易造成过拟合。如果采用集成学习，通过将多个模型进行结合，理论上可以获得比单一模型显著优越的泛化性能。

为了验证这种方法是否适合于我们研究的问题背景，我们将所提出的方案以及其余四种主流方案进行融合，并测试在将 FastText, CNN, CNN with Word2Vec, Bi-LSTM 以及 Attention 融合后，是否比我们原本单独的 CNN 模型优秀。融合方案如表 3 所示，模型融合后的结果见图 9。

**表 3. 模型训练效率对比**

| 方案 | 投票 | 平均(1:1:1:1:1) | 权重 (1:2:2:2:3) |
|---|---|---|---|
| 说明 | 即少数服从多数原则，分类得票数超过一半的作为预测结果。 | 将所有预测结果相加取平均值。 | 将所有预测结果按照权重计算。 |

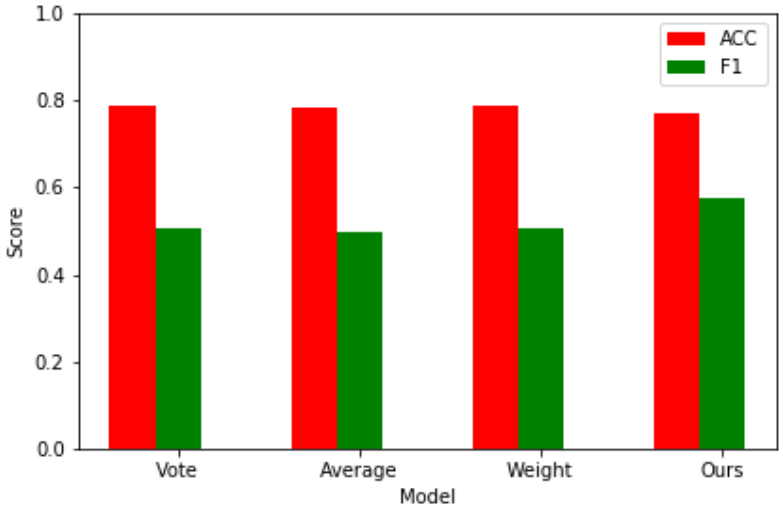


**图 9. 模型融合泛化能力示意图**

从图 9 可以看出，采用这三种模型融合方案后，模型的准确率均比我们的 CNN 模型提高大约 1%，然而在 F1-Score 指标上，我们的模型获取了最高的分值，提高 7%左右。从而我们得出模型融合策略并不适用于我们研究问题背景的结论。

# 五、总结与展望

## 5.1 总结

为了满足人们日益增长的可交互式阅读需求，本文基于人工智能的相关理论和实验，构建了一个智能阅读模型，让用户“更自然”、“更低成本”地实现人与机器的交流。模型主要在分为数据预处理、关键词匹配以及精准匹配三个阶段，在对赛题研究的基础上，我们根据研究思路撰写本论文，通过实验验证了本模型的可行性，基本实现本赛题设立的目标。

## 5.2 展望

### 5.2.1 改进损失函数——Triplet Loss

深度学习模型有固定的输入和输出，并且损失函数为真实值与预测值的某种函数关系，然而有些模型并非如此，比如 Triplet Loss。

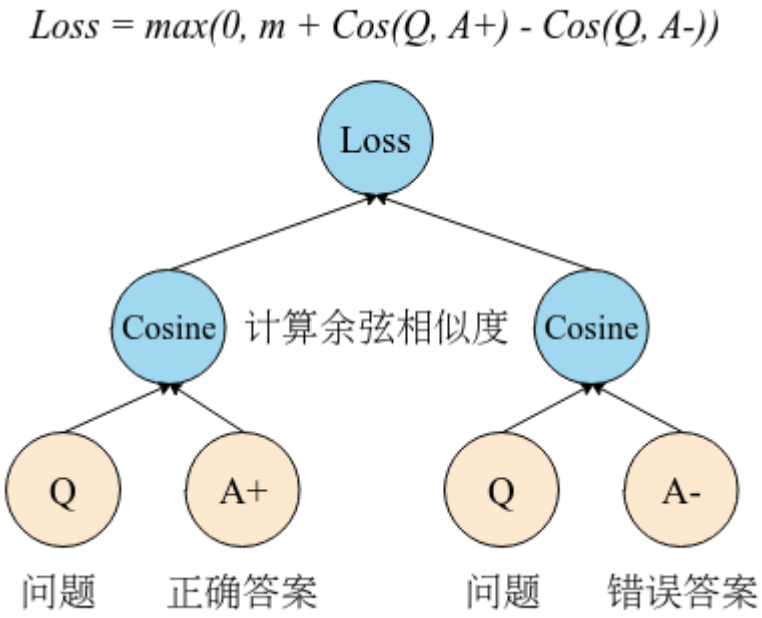


图 10. Triplet Loss 结构

如图 10 所示，在 Triplet Loss 结构中，(Q, A+)为一对正样本对，(Q, A-)为一对负样本对。Loss 计算公式中的 m 是一个大于零的正数，我们希望正样本分数越高越好，负样本分数越低越好，但二者得分之差最多到 m 就足够了，差距增大并不会有任何奖励。使用 Triplet Loss 有助于细节区分，在众多候选回答中更容易找出正确回答。

### 5.2.2 构建基于 Web 的问答系统

基于 Web 的问答系统 (Web-based Question Answering, WQA) 以开放的互联网上的 Web 文档作为问答系统的知识来源，从搜索引擎返回的相关网页片段从中抽取出用户所提问题的答案。如图 11 所示为 WQA 的基本流程：

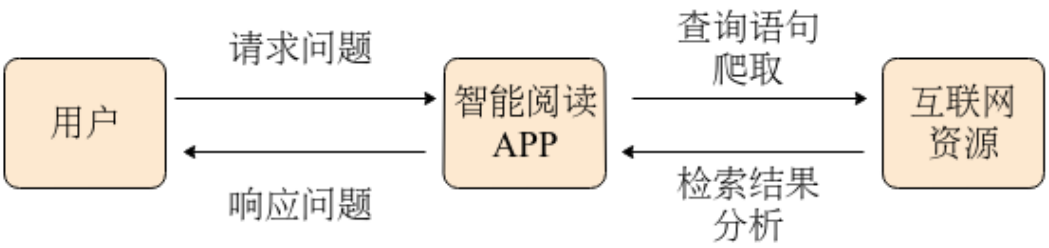


图 11. WQA 基本流程

WQA 系统同时具有搜索引擎和问答系统的优点：

① 能通过现有成熟的搜索引擎来获取整个互联网上的各种相关信息，这些信息无所不包，不受领域的限制，且与时俱进，不断更新。

② 单纯使用关键词匹配候选回答，使用卷积神经网络对候选回答精准匹配，是很难处理基于现实世界的事实性问答，对于问题“景区随地吐痰罚款多少钱”，有的回答“50 元”，也有的回答“60 元”。如果知识库没有相关信息，模型是无法正确匹配的，而 WQA 一般能够较好地处理这种事实型问题。

③ 用户更倾向于使用人类自然语言的表达方式提问，而 WQA 能够利用搜索引擎本身优势进行人性化的交互。

目前我们实现的智能阅读模型仅仅实现了定位回答问题的行，而要想进一步给出明确的答案，还需要定制不同粒度的方案，加大了模型的设计难度。而作为一种特殊的问答系统，WQA 就能够很好地应对这种问题。


**图 12. 百度知道搜索问题示例**

如图 12 所示，在百度知道上搜索问题“‘江南七怪’分别是谁？”，我们仅仅需要爬取这种搜索结果，并经过简单的过滤提取，即可返回给用户。在应对用户提出的事实性问题情景，往往能起到很好的效果。

### 5.2.3 研究基于众包的智能阅读数据处理服务

数据众包服务使用低成本高效率的众包模式满足客户对数据的需求，可采集到大量的原始数据，通过数据标注对原始数据进行加工，最终提供计算机可以识别的高质量数据，帮助数据科学家更精准地训练算法模型、开展机器学习工作，提高在 AI 领域的竞争力。目前世界上最大的图像识别数据集 ImageNet 就是通

过亚马逊众包平台 Mechanical Turk 进行人工标注，这项服务能够让世界各地坐在电脑前的人在线完成雇主的工作任务，并获得有偿报酬。

机器学习的好坏往往取决于数据集的质量，目前构建智能阅读的问答数据集大都爬取自互联网问答社区，存在数量有限和参差不齐的现象。因此，进行大规模人工数据集众包采集、清洗和标注显得很有必要。

## 参考文献

[1]邹俊杰. 受限域问答系统文本检索研究[D].昆明理工大学,2011.
[2]李欢. 问答系统中的文本信息抽取研究与应用[D].中国科学技术大学,2009.
[3]《分布式单词表示综述》http://www.bigdatalab.ac.cn/~gjf/papers/2017/JCIP1.pdf
[4]《Bag of Tricks for Efficient Text Classification》https://arxiv.org/abs/1607.01759
[5]《A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification》
[6]https://arxiv.org/abs/1510.03820
[7]http://www.52nlp.cn/中英文维基百科语料上的 Word2Vec 实验
[8]《LONG SHORT-TERM MEMORY》http://www.bioinf.jku.at/publications/older/2604.pdf
[9]《Attention Is All You Need》
https://arxiv.org/abs/1706.03762
https://en.wikipedia.org/wiki/Information_retrieval
[10]《机器学习》，周志华，清华大学出版社
[11]https://baike.baidu.com/item/泛化能力
[12]https://www.spaces.ac.cn/archives/4338
[13]《基于 Web 的问答系统综述》
http://www.jsjkx.com/jsjkx/ch/reader/create_pdf.aspx?file_no=20170601&flag=&journal_id=jsjkx&year_id=2017

# 附录

## （一）FastText 模型

FastTex 模型将整篇文档的词或 N-gram 向量叠加平均得到文档向量，然后使用文档向量分类。适合海量数据和高速训练，能将训练时间由几小时缩短到几分钟。

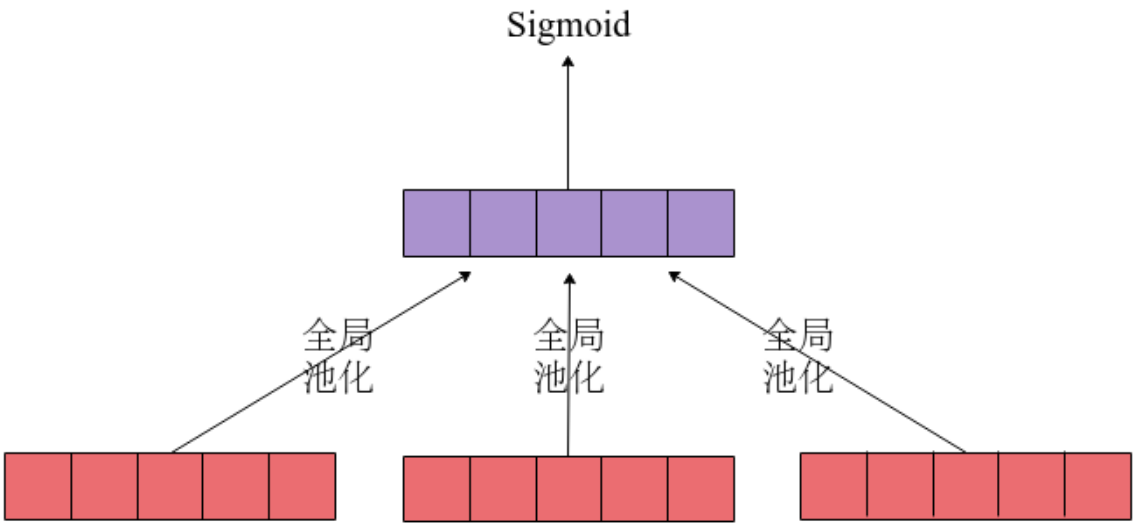


图 13. FastText 思想

模型架构：词嵌入后，隐藏层只是一个简单的全局平均池化层，然后连接经过池化的问题和回答向量，最后的全连接层作为分类器使用。注意这里的输入可以是单词，也可以是 N-gram 组合。

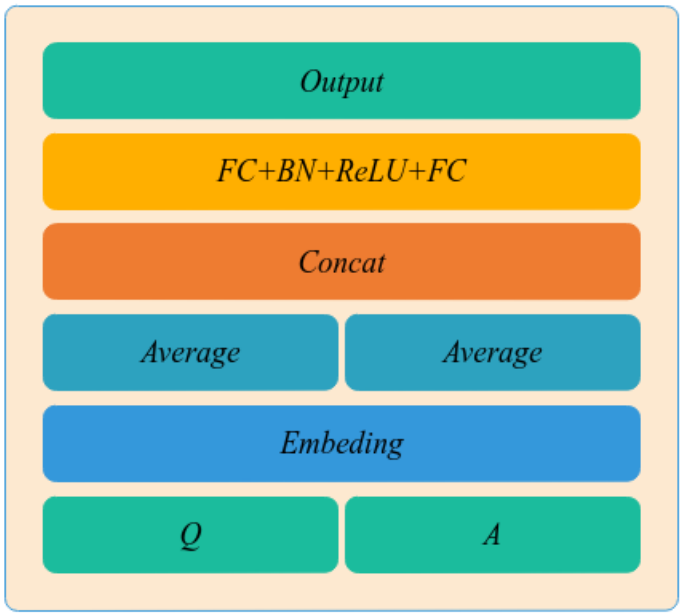


图 14. FastText 模型

## （二）CNN with Word2Vec 模型

CNN with Word2Vec 主要是将自然语言处理中词向量模型 Word2Vec 与卷积神经网络进行结合。Word2Vec 是用来产生词向量的相关模型，训练完成之后，可用来映射每个词到一个向量，可用来表示词对词之间的关系。Word2Vec 采用 CBOW 和 Skip-Gram 来建立神经网络词嵌入。CBOW 是已知当前词的上下文，预测当前词。而 Skip-Gram 相反，是在已知当前词，预测当前词的上下文。

实验使用维基百科中文语料生成 Word2Vec 模型，最终可以从 1.5 多 G 的原始语料中，提取到 900 多 M 的词向量模型，将近 80 万条词向量，每条词向量 100 维。

具体流程如下：

① 下载语料

https://dumps.wikimedia.org/zhwiki/latest/zhwiki-latest-pages-articles.xml.bz2

② 提取正文

将 xml 格式的 wiki 数据转换为 text 格式。

③ 繁简转换

如果抽取中文的话需要将繁体转化为简体(维基百科的中文数据是繁简混杂

的，里面包含大陆简体、台湾繁体、港澳繁体等多种不同的数据)。可以使用 opencc 进行转换，也可以使用其它繁简转换工具。

④ 编码转换

由于后续的分词需要使用 utf-8 格式的字符，而上述简体字中可能存在非 utf-8 的字符集，避免在分词时候进行到一半而出现错误，因此先进行字符格式转换。使用 iconv 命令将文件转换成 utf-8 编码。

⑤ 分词处理

使用 jieba 分词工具。

⑥ 训练 Word2Vec

```
In [ ]: from gensim.models import word2vec
        import logging

        input = './wiki.zh.text.jian.utf8.seg'
        output = './wiki.vector'
        logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO)
        sentences = word2vec.LineSentence(input)
        model = word2vec.Word2Vec(sentences, size=100, window=5, min_count=5, workers=4)
        model.wv.save_word2vec_format(output, binary=False)
```

图 14. Word2Vec 训练代码示例

⑦ 测试 Word2Vec

```
In [1]: from gensim.models.keyedvectors import KeyedVectors
        model = KeyedVectors.load_word2vec_format('./wiki.vector', binary=False)

In [2]: # model['女人'] + model['国王'] - model['男人'] = model['皇后']
        model.most_similar(positive=['woman', 'king'], negative=['man'])

Out[2]: [('queen', 0.7293198108673096),
         ('bride', 0.683803915977478),
         ('mistress', 0.6707652807235718),
         ('prince', 0.6648019552230835),
         ('wives', 0.6588137149810791),
         ('princess', 0.6529775857925415),
         ('queens', 0.6459839344024658),
         ('daughters', 0.6443694829940796),
         ('mother', 0.6377485990524292),
         ('godmother', 0.6289682388305664)]

In [3]: model.similarity('woman', 'man')

Out[3]: 0.6361447854063201

In [4]: model.similarity('queen', 'king')

Out[4]: 0.6681771014772736
```

图 15. Word2Vec 测试代码示例

可以看到，在 Word2Vec 词向量模型中，出现一些有趣的现象：

① “女人” + “国王” - “男人” ≈ “皇后”。

② “女人”和“男人”（或“皇后”和“国王”）在空间向量上接近。

也成功验证了“词嵌入模型可用来映射每个词到一个向量，可用来表示词对词之间的关系”这一结论。

### （三）Bi-LSTM 模型

近两年深度学习在自然语言处理领域取得了非常好的效果。深度学习模型可以直接进行端到端的训练，而无须进行传统的特征工程过程。在自然语言处理方面，主要的深度学习模型是 RNN，以及在 RNN 之上扩展出来的 LSTM。

LSTM 是一种带有选择性记忆功能的 RNN,它可以有效地解决长时间依赖问题，并能学习到之前的关键信息。它增加了一条状态线，以记住从之前的输入学到的信息，另外增加三个门(gate)来控制该状态，分别为忘记门、输入门和输出门：

忘记门的作用是选择性地将之前不重要的信息丢掉，以便存储新信息。

输入门的作用是根据当前输入学习到新信息，更新当前状态。

输出门的作用是根据当前输入和当前状态得到一个输出，该输出除了作为基本的输出外，还会作为下一个时刻的输入。

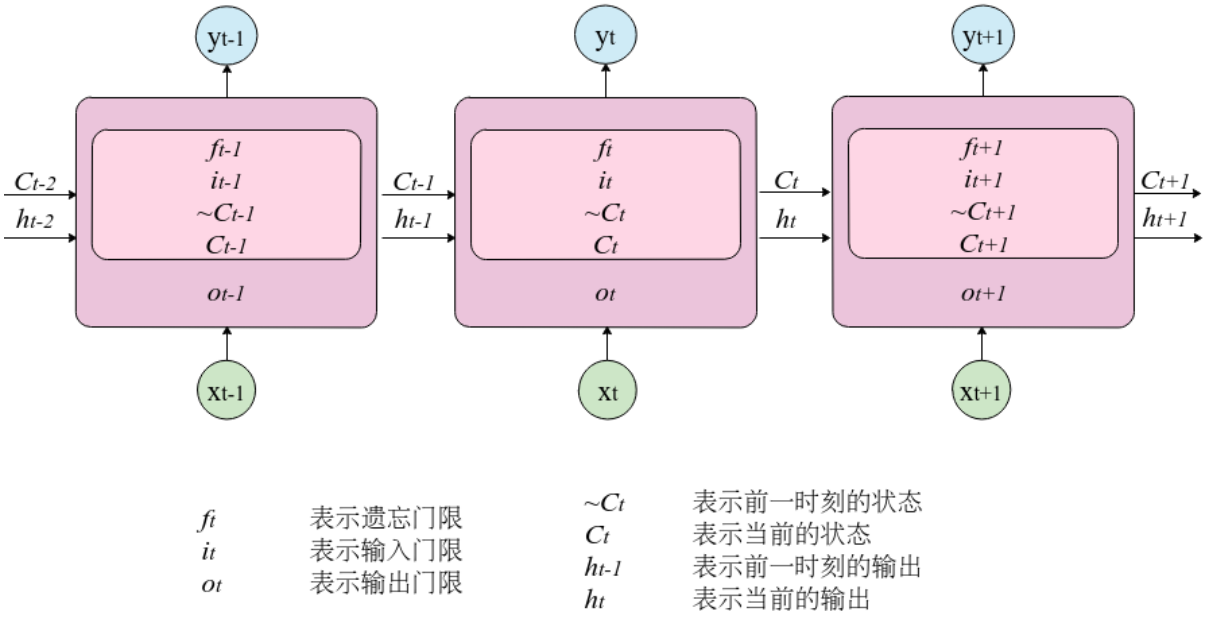


**图 16. LSTM 内部结构**

单向 LSTM 根据前面的信息推出后面的信息，但有时候只看前面的信息是不够的。例如：今天天气__，风刮在脸上仿佛刀割一样。

如果根据“天气”，可能推出“晴朗”、“暖和”、“寒冷”等。但是如果

加上后面的形容，能选择的范围就变小了，“晴朗”、“暖和”不可能选，而“寒冷”被选择的概率更大。

LSTM 虽然解决了长期依赖问题，但是无法利用文本的下文信息。双向 LSTM 同时考虑文本的上下文信息，将时序相反的两个 LSTM 网络连接到同一个输出。前向 LSTM 可以获取输入序列的上文信息（历史数据），后向 LSTM 可以获取输入序列的下文信息（未来数据）。两个方向有各自的隐藏层，相互之间没有直接连接，只是最后一起连接到输出节点上，模型准确率得到大大提升。

模型架构：首先经过词嵌入，隐藏层是双向 LSTM 网络，LSTM 结束后连接问题和回答向量，最后的全连接层作为分类器使用。

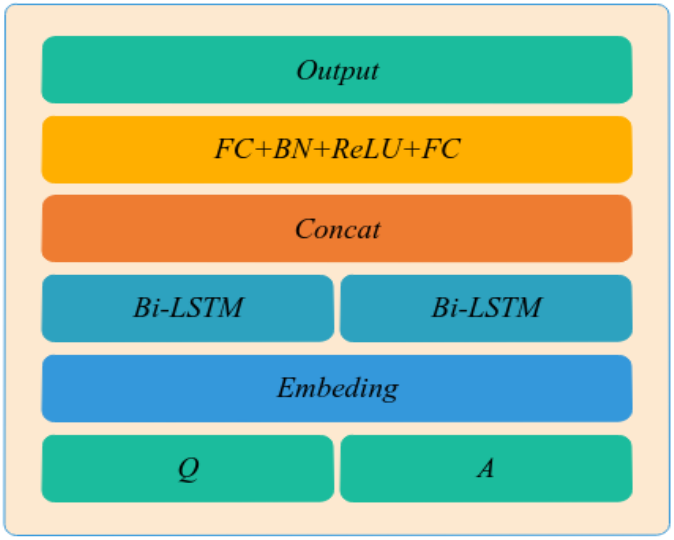


图 17. Bi-LSTM 模型

### （四）基于 Attention 模型

基于深度学习的 NLP 研究方法，基本上都是对数据集进行分词，然后进行分布式表示，将每个句子转化为对应的词向量序列。模型普遍使用 CNN 和 RNN 等成熟方案，而 Google 的大作《Attention Is All You Need》的出现，提供了另一种方案：Attention，注意力机制。

注意力机制来源于人脑，我们在阅读的时候，注意力通常不会平均分配在文本中的每个词。如果直接将每个时刻的输出向量相加再平均，就等于认为每个输入词对于文本表示的贡献是相等的。但实际情况往往不是这样，比如在情感分析中，文本中地名、人名这些词应该占有更小的权重，而情感类词汇应该享有更大

的权重。所以在合并这些输出向量时，希望可以将注意力集中在那些对当前任务更重要的向量上。也就是给他们都分配一个权值，将所有的输出向量加权平均。

注意力机制的思路是：原先都是相同的中间语义表示 C 会替换成根据当前生成单词而不断变化的 Ci，每个 Ci 对应着不同单词的注意力分配概率分布。f 函数代表 Encoder 对单词的某种变换函数，g 函数代表 Encoder 根据单词的中间表示合成整个句子中间语义表示的变换函数。一般来说，g 函数是 f 函数加权求和。

如图 18 所示，当阅读句子“老鼠爱大米”的注意力在“爱”时，对应的中间语义计算将会分配给“爱”比较大的权重，而“老鼠”和“大米”的权重相对较轻。

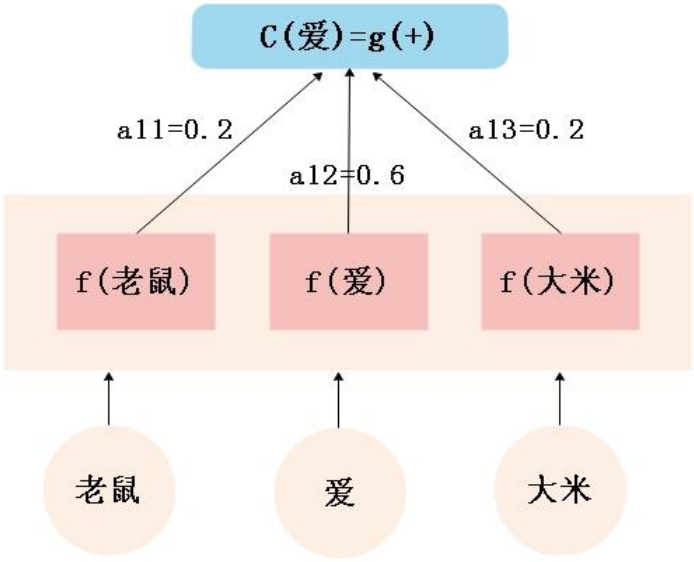


**图 18. Attention 思想**

模型架构：首先经过词嵌入，隐藏层是 Attention 网络，Attention 结束后连接问题和回答向量，最后的全连接层作为分类器使用。

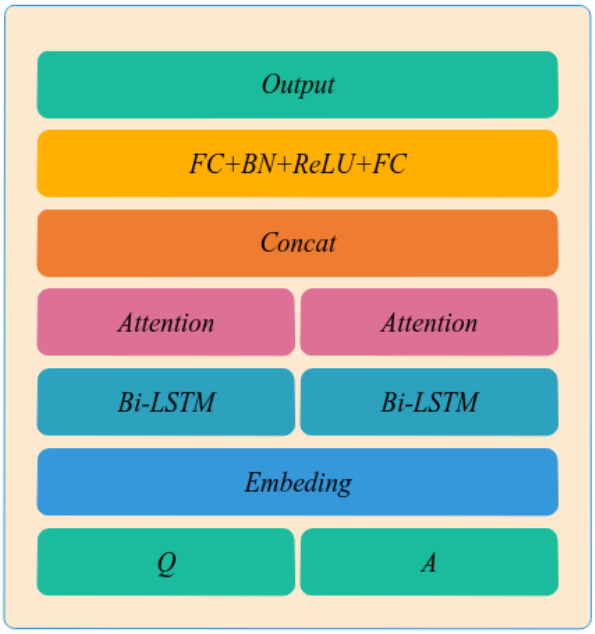


图 19. Attention 模型