# 1. 选题依据与价值

## 1.1 课题背景

  互联网时代，随着自由文本的数据量逐渐增加，以及计算机硬件配置的逐渐加强，基于大规模文本数据的机器学习技术在自然语言处理（Natural Language Processing，NLP）中的应用也逐渐成熟。其中文本蕴含识别（Recognizing Textual Entailment，RTE）作为一项重要的自然语言处理基础任务，在问答系统、信息检索和信息抽取等诸多领域都有很多应用[1]。

  文本蕴含识别通常是指识别两句话的蕴含关系，在给定前提文本P和假设文本H的情况下，如果我们可以通过文本P的内容推断出文本H的内容是正确的，那么我们便可以说前提文本P蕴含假设文本H，文本P和文本H之间便是蕴含关系。目前，通过机器学习等方法可以从一定程度上实现自动识别文本蕴含关系，该任务目标的实现对问答系统、语义检索、自动摘要和机器翻译评估等应用具有重要的意义。

## 1.2 应用价值

  文本间的蕴含关系普遍存在于自然语言文本之间，所以文本蕴含关系识别在许多自然语言处理领域都存在着实际应用价值。本节将列出一些文本蕴含关系识别的应用[2]。

(1)问答系统

   一个理想的文本蕴含系统可以作为问答系统的答案验证模块，用来判断一个从知识库种抽取的候选答案是否能够推断出目标问题或是目标问题的陈述句形式。例如，某人提问“港珠澳大桥的总工程师是谁？”，通过信息检索技术从知识库中检索出候选答案“林鸣，自2010年12月起，担任港珠澳大桥岛隧工程项目总经理、总工程师”，利用文本蕴含系统可以验证是否可以从候选答案推断出目标问题。为检验文本蕴含系统的这种能力，欧洲跨语言评测平台(Cross Language Evaluation Forum, CLEF)在2006-2008举办了三届答案验证评测(Answer Validation Exercise, AVE)。在评测中，每一条数据由一个问题、一个候选答案和一篇支持文本组成，文本蕴含模型的目标就是在给定支持文本的情景下，针对目标问题，判断候选答案是否正确。

(2)语义检索

  语义检索的目标是基于检索语句的语义从大量的文档库中检索出目标文档的目标语句。如果将文本蕴含系统运用在语义检索中，它可以用来识别目标问题与目标源文档中各个语句的语义相似性。例如，用户搜索“人们示威反对自由贸易”，现存的基于关键字的搜索引擎将会返回包含“示威”、“自由贸易”等关键字的文档，如果文档中仅存在像“游行人员高举‘反对解除贸易壁垒协议’的标语”这样不含关键字的语句，那将不会被检索到，而造成信息遗漏，利用文本蕴含识别可以避免上述问题。

(3)自动摘要

  在自动摘要任务中，一个主要的挑战是怎样消除冗余。冗余现象在多文档摘要中特别明显，多文档摘要是融合了多篇源文档的内容，而多篇文档之间的内容可能存在语义相似的，从而造成自动生成的摘要中存在多条语句表达相似语义的问题。文本蕴含识别系统用来判断是否存在语句所表达的语义可以由其他摘要语句推理得到。自动摘要的另一个挑战就是正确性，也就是摘要应该准确地反映单个源文档或多个源文档的内容。文本蕴含识别系统可以通过判断源文档是否可以推理出摘要，以确保摘要的正确性。

(4)机器翻译的自动评估

  文本蕴含识别的一个相对比较新的应用就是自动评估机器翻译的性能。目前，机器翻译研究者主要利用BLEU作为翻译好坏的指标，BLEU是通过衡量机器翻译语句与人类翻译语句之间的重合程度来，并没有考虑语义层次的相似性。而文本蕴含识别可以判断机器翻译的语句是否可以推断出人类翻译的语句，若可以推断出，则可以认为这是一个好的翻译，即使机器翻译的句子与人类翻译的句子之间没有一个字是一样的。

# 2. 国内外研究现状[3]

  文本蕴含识别作为自然语言处理领域里一项重要研究内容，是许多NLP 应用的基础，吸引着世界各地研究者的广泛关注。为了促进文本蕴含识别理论方法的实用化，自2004年开始，每年都有一些相关评测比赛举行。正是这些语义评测比赛的举办 推动着文本蕴含识别的发展，但早期的方法主要是在一些小规模的评测语料上进行实验。2015年，Bowman 等人[4]发布 SNLI 数据集，使得文本蕴含识别的语料库规模增加了几个量级。2017年，Williams 等人[5]在SNLI数据集的基础上建立了MultiNLI数据集，其数据内容来源更加丰富，语言形式更加复杂。算法方面，越来越多的研究者开始利用深度学习方法来解决文本蕴含问题。总体上说，可以将文本蕴含识别方法分为两类：传统文本蕴含识别方法和基于深度学习的文本蕴含识别方法。下面对这两类RTE方法进行介绍。

## 2.1 传统文本蕴含识别方法研究现状

  近十年来，有许多RTE相关评测比赛举行，主要的评测比赛列举于表2-1中。这些评测通常有这样一些特点：评测中的例子是多来源的，包括新闻素材 和图片标题等等，因此开发出的系统必须是领域独立开放的；同时，所需要做出的推断都是从人类思考的角度进行的，并不需要非常严密的逻辑推理，也不包含多阶段的“长链”的推理。参赛者在这些评测语料库上提出了许多有效的文本蕴含识别方法，这些方法主要可以分为三类：基于相似度的方法、基于对齐的方法和基于逻辑演算的方法。下面对这三类主流方法分别进行介绍。

表2-1：文本蕴含书别相关评测赛事表

### 2.1.1 基于相似度的方法

  构成蕴含关系的P-H对往往比较相似，所以有研究者提出利用P-H对的相似程度来判断其是否构成蕴含关系。但这类方法非常依赖于词性标注，共指消解，命名实体识别等自然语言处理的基础技术，通常会引起错误的后向传播，也非常依赖于外部资源[6][7][8][9]。随着研究的不断发展，现在多是把相似度作为判别模型的一个特征[10][11][12][13]。

### 2.1.2 基于对齐的方法

  在基于相似度的识别方法的基础上，研究者们演化出了基于对齐的识别方法。这类方法并不是直接使用相似度来判断蕴含关系，而是先把前提序列和假设序列中的相似部分找出来进行对齐，然后把对齐的方式和程度作为判断是否构成蕴含的依据。在对齐过程中一般会遍历所有可能的组合，利用一个打分函数来判断对齐的质量。[14]与传统基于相似度的对齐打分函数不同，文本蕴含中的打分函数需要重点考虑一些对判断蕴含关系起到重要的作用的细节（例如：动词的否定、反义词和主被动关系词等等）。有些工作专门将研究重点集中在学习这样的对齐算法上[15][16][17][18]。另外，基于对齐的方法产生的结果也常用作基于分类器算法的特征。

### 2.1.3 基于逻辑推理的方法

  文本蕴含识别实际是一种语义推理模型，而数学中对命题的逻辑证明问题已经有了比较成熟的方法与工具。这类方法会使用一些现有的逻辑推理工具，将句子转换成一个正式的逻辑表示（例如：一阶逻辑表达式等），然后基于知识库中的事实运用相关推理规则判断两个逻辑表示之间的蕴含关系[19][20][21][22]。在进行推理时，所使用的知识通常是领域相关的，在开放域文本蕴含任务上有一定的局限性。同时，将文字命题转化为逻辑表达式的过程不够鲁棒，错误率较高。但由于逻辑演算方法的数学严密性，可以用来处理一些特殊的文本蕴含问题。例如：数量蕴含问题，即需要对前提和假设中所提到的数字进行运算才能判断其蕴含关系的特定蕴含问题，Roy 等人[23]通过数学演算的方法解决了数量蕴含问题。

  此外，还有一些基于变换的方法、基于混合模型的方法提出。

## 2.2 基于深度学习的文本蕴含识别方法研究现状

  深度学习通过神经网络可以自动的学习多层次的复杂抽象特征。常见的应用于自然语言处理领域中的神经网络有递归神经网络和卷积神经网络。深度学习在语言模型、中文分析、知识挖掘、情感计算、机器翻译、信息检索、命名实体识别、语义角色标注等任务上都取得了非常好的结果，在大多数任务中都超越了传统的机器学习方法的最好水平。

  由于早期文本蕴含识别的语料库规模较小，利用传统方法会带来错误传播和工作量大等问题，而神经网络模型能有效避免这些问题。随着SNLI和MultiNLI的发布，许多研究者都开始探索利用深度学习技术来解决文本蕴含识别任务。目前，针对英文文本蕴含识别任务，国内外的研究者已经探索尝试出了一些深度学习方法。这些方法主要可以分为两大类：基于句子表示的方法和基于句子匹配的方法。下面对这两类方法分别进行介绍。

### 2.2.1 基于句子表示的方法

  基于句子表示的方法通常通过某种神经网络将前提句和假设句分别编码成定长向量，然后对这一对向量进行匹配，最后将得到的匹配向量送入后续神经网络中完成分类任务。Bowman 等人[4]使用长短期记忆网络(Long Short Term Memory, LSTM)分别对前提句和假设句进行编码，得到两个100维的句子表示向量，然后将两个句子表示向量连接后送入三层包含200个隐含节点的tanh层中，最后通过Softmax层进行分类。他们的SNLI语料库上取得77.6%的准确率。将句子向量维度更改为300后准确率提高到80.6%。Vendrov 等人[24]利用预先编码好的ordered-embedding输入到GRU中来训练得到句子向量，将两个句子向量和两个向量相减后的绝对值向量相连，然后进行分类，他们在SNLI语料库上取得了81.4%的准确率。Mou 等人[25]利用基于依存树的卷积神经网络来分别编码两个句子，将编码后的向量进行相连、按位相减和按位相乘操作，最后相连后在结果向量的基础上分类，它们在SNLI的准确率为82.1%。Bowman 等人[26]又提出了SPINN(Stack-augmented Parse-Interpreter Neural Network)编码器，该编码器利用一个树形结构从左到右顺序扫描输入词串，它的设计参考了shift-reduce分析器的思想，既能接受句法分析的输入，也能接受原始词串输入，其实验结果达到了83.2%的准确率。

### 2.2.2 基于句子匹配的方法

  基于句子匹配的文本蕴含识别方法不生成句子向量，直接针对任务产生匹配向量进行分类。这类方法都引入了在多个任务中都表现优良的Attention机制。实际上，他们的主要工作就是针对文本蕴含识别任务对Attention机制进行改进。Attention机制类似于人类的注意力机制，能够关注于句子中相对重要的部分。Attention机制在文本蕴含识别中的作用就是关注对判断句子蕴含关系起到重要作用的词，给予相对较高的权重，然后根据Attention权重产生匹配向量来完成分类任务，这类方法都取得了非常不错的效果。Rocktäschel 等人[27]提出了在文本蕴含关系识别中的Attention和Word-by-Word Attention分别取得了82.3%和83.5%的好成绩。Wang[28]和 Chang[29]在word-by-word attention的基础上进行改进，分别提出了mLSTM和融合memory network 的LSTMN，结果上都取得了不错的提升。

  综上所述，传统文本蕴含识别方法通常需要基础NLP技术和外部语义资源的支持，比较费时费力，方法相对比较复杂。基于深度学习的文本蕴含识别方法能够有效避免上述两类问题，并且取得了很不错的效果。因此，本文的研究重点是基于深度学习的文本蕴含识别方法，并针对深度学习无法有效解决的问题（数量蕴含问题），利用逻辑推理的方法进行解决。

# 3. 相关理论和技术

## 3.1 词的表示

### 3.1.1 One-hot 模型

  将自然语言处理任务用数学模型去描述，就必须将文本用数学方法表示。One-hot 模型是最简单的表示方式，该模型将词表示为一个与词表长度一样的向量，向量中只有一个特定维度（词在词表中的位置）置1，其余维度置0。One-hot 表示法广泛应用在支持向量机和条件随机场等传统机器学习算法中，但这种方法不能够体现词语之间的语义关系。

### 3.1.2 分布式表示

  为了让词的表示能够体现词的语义，1986年 Hinton 等人[30]提出分布式表示(Distributed Representation)方法。分布式表示法认为词的语义与词周围词的分布有关系，例如有如下四句话：

* A bottle of tesguino is on the table.
* Everybody likes tesguino.
* Tesguino makes you drunk.
* We make tesguino out of corn.

  这4句话中都有一个单词tesguino，从这四句话的上下文中大致可以猜到这个单词是一种酒，因为这个单词的上下文也经常出现在其他酒类单词周围。基于以上思想，发展出一种基于共现向量(Co-occurrence matrix)的词嵌入模型，这种词嵌入模型输出的是稀疏向量；另外一类词嵌入模型输出稠密向量，例如：基于奇异值分解(SVD)的词嵌入模型，基于聚类的词嵌入模型以及目前流行的基于神经语言模型的词嵌入模型。本节将主要介绍一下基于神经语言模型(Neural Language Modle)的两种词嵌入模型：Word2vec模型和ELMo模型。

### 3.1.2.1 Word2vec模型

  2013年，Mikolov 等人[31][32]提出Word2vec框架,该框架集成了两个模型：Skip-gram模型和CBOW模型，两个模型可以更快的通过预训练大规模语料得到词的向量表示。本节将主要介绍这两个模型。

（1）Skip-gram模型

  Skip-gram模型的基本思想是通过中心词预测上下文，模型结构如下图所示：

图3-1：Skip-gram 模型(Mikolov 等人于2013年发表)

  Skip-gram模型可看做一个多任务分类模型，其中 均为词的one-hot表示。训练完成后，模型取作为词的词向量。

（2）CBOW(Continous Bag of Words Model)模型

  CBOW模型的基本思想是通过上下文预测中心词，模型结构如下图所示：

图3-2：CBOW 模型(Mikolov 等人于2013年发表)

  CBOW模型可以简单的看作一个分类任务。与Skip-gram模型类似， 均为词的one-hot表示。训练完成后，模型取作为词的词向量。

### 3.1.2.2 ELMo模型

  2018年，Matthew 等人[33]提出ELMo(Embeddings from Language Models)模型，在多个自然语言理解的数据集上都有较好的表现，在性能指标上都有显著的提升。预训练文本表示在自然语言理解领域具有重要地位，其中Word2vec[31][32]和Glove[34]对预训练词表示起到了关键作用。但是Word2vec和Glove都只是将一个词用一个向量表示，而词在不同的语境中会有不同的语义，而用一个固定的向量来表示一个词则无法体现词的多义性。ELMo的则是引入了一种基于语境的词嵌入方式。

  ELMo模型是基于大量文本语料，从深层的双向语言模型（deep bidirectional language model）中的内部状态(internal state)训练而来的。深层双向语言模型由多层双向长短记忆网络(BiLSTM)构建而成。ELMo是将BiLSTM中的各层状态向量进行线性组合而得。对于某个词，可以由一个层的BiLSTM中的个状态向量表示如下：

  其中,表示BiLSTM第i层的状态向量。然后ELMo将组合成一个词向量，实验发现[33]，将中的所有向量进行加权求和组合并进行缩放具有最好的效果，具体的组合公式如下：

  其中是归一化后的权重向量,可以对整个词向量进行放缩调整。这两个参数都是根据具体任务参与训练的，而从预训练双向语言模型中抽取获得的。

  当在问答系统、文本蕴含、机器翻译等监督学习任务中，ELMo词向量可以与普通的预训练词向量进行合并得来表示一个词；同时，实验表明将具体任务（例如：SNLI,SQuAD）中的RNN模型的输出替换成，会获得更好的实验效果。

## 3.2 循环神经网络(Recurrent Neural Network, RNN)

  循环神经网络 (Recurrent Neural Networks，RNNs) 是一种常见的人工神经网络，网络中结点间的连接线形成有向环，RNN 在许多自然语言处理任务中都有重要的应用。区别于前向反馈神经网络(Feed-forward Neural Networks，FNNs) 中输入输出是相互独立的关系，RNN 能够有效利用到上一时刻的输出结果。因此，RNN 用来处理序列数据比较合适。RNN 在语言模型、文本生成、机器翻译、语言识别和图像描述生成等任务中都取得了很不错的效果。其展开图如图所示:

图3-3：RNN网络结构

  在传统RNN网络的梯度后向传播阶段，梯度信号最后会与RNN隐含层中相关的权重相乘多次（次数与步长一样），这就意味着，相关权重的大小会对网络的学习训练过程产生巨大的影响。如果权重太小（或者说是权重矩阵的特征向量小于1.0），就会导致“梯度消失”，梯度变得越来越小使得网络学习过程缓慢，甚至完全停止。对于“长距离依赖”问题，由于梯度消失问题，使得RNN的学习变得非常困难。如果权重太大（或者说是权重矩阵的特征向量大于1.0），会导致梯度爆炸，无法收敛。为了解决 RNN 存在的这些问题，Hochreiter 等人[35]提出了长短期记忆神经网络单元(Long Short-term Memory, LSTM)。

### 3.2.1 长短记忆神经网络(Long Short-term Memory, LSTM)

  LSTM 神经网络就是将 RNN 网络中的隐含层节点替换为 LSTM 单元后形成的网络。一个 LSTM 记忆单元的结构如图所示：

图3-4：LSTM网络结构

  LSTM 的核心是Cell的状态，它通过“门”来控制流入到 Cell 信息，Sigmoid 层的输出为 1 则代表信息全部通过，输出为 0 表示内容被完全阻隔。LSTM 单元包含三个门来控制 Cell 状态，分别为输入门、输出门和遗忘门。在时刻t，Cell 的状态通过以下公式进行更新：

  其中，分别表示，在t时刻，输入门、遗忘门、输出门、输入模块和Cell的输出;分别为偏置向量;为权重矩阵。

  LSTM学习能力强、表达力强并且容易训练，在NLP任务中应用广泛。实际上，取得领先水平的RNN网络，绝大多数都是LSTM网络。

### 3.2.2 双向长短记忆神经网络(Bidirectional LSTM, BiLSTM)

  单向 LSTM 的一个缺点是它只能利用出现在当前节点之前的信息。在句子文本建模任务中，充分利用过去和未来的信息能够更全面的掌握语义信息，生成更有效的句子向量。双向 LSTM 利用两个独立的隐含层双向处理文本来达到同时利用上下文的目的，两个隐含层的结果最后都送入到同一个输出层[36]。如图所示:

图3-4：BiLSTM网络结构

  设分别是BiLSTM的两个单向LSTM的输入隐状态序列，则BiLSTM的输出为：

## 3.3 Attention机制

  Attention机制首先出现在视觉图像领域，2014年，Bahdanau 等人[37]将Attention机制应用于机器翻译任务中，这是Attention机制首次应用于自然语言处理领域，之后各种Attention机制结合深度学习用来处理自然语言处理任务。机器翻译主要是依据Seq2seq框架[38]进行建模，而传统的Seq2seq模型中的编码器部分把源语言编码为一个上下文向量，这会造成源语言信息的损失。本节将首先介绍在首次在机器翻译中运用的Attention机制。

### 3.3.1 机器翻译中的Attention机制

  Attention机制应用于机器翻译的示意图如下：

图3-5：Attention机制(Ashish 等人于2017年发表)

  设编码器输出的各个时刻的隐状态向量 ，解码器t时刻需要输出的隐状态向量为 。机器翻译任务在预测t+1时刻的词时，首先根据当前隐状态向量和输入序列的所有隐层输出进行一个打分运算，得分数；然后经过softmax操作后得到Attention对于t时刻解码器隐状态向量的权重，再通过加权求和得到目标语言t时刻状态对应的上下文向量，最后应用上下文向量与t时刻隐状态向量共同预测t+1时刻的词。Attention机制在机器翻译中的运算过程如下所示：

  其中Attention中的打分函数 可以有多种操作。最简单的打分运算函数是将两个维度相同的向量做內积操作，运算公式如下：

  Bahdanau 等人[37]首次在机器翻译中运用Attention机制的打分运算如下：

  其中均为Attention机制中可训练的参数。2015年，Luong 等人[39]简化了上述打分操作，提出如下打分运算函数：

  其中是Attention机制中可训练的参数。

### 3.3.2 Transformer模型[40]

  2017年，Vaswani 等人[40]提出一种新的Seq2Seq网络结构Transformer，其完全依赖Attention机制，舍弃了循环神经网和卷积操作，并在机器翻译领域取得重大成功。Transformer整体网络结构如图所示：

图3-6：Transformer整体结构(Ashish 等人于2017年发表)

  Transformer主要由编码器(Encoder)与解码器(Decoder)组成。Transformer的编码器主要由多个编码层堆叠而成，每个编码层结构相同，上层编码层的输出作为下层的输入。每个编码层的分别由两部分组成：第一部分是多头自注意力层(Multi-head Self-attention)，第二部分是基于位置的全连接神经网络(Position-wise feed-forward network)。两个部分都有一个残差链接(residual connection)，然后接着一个 Layer Normalization 层。Transformer的解码器部分与编码器部分结构相似，只是在 Multi-head Self-attention 与 Position-wise feed-forward network 之间加上 Multi-head Cross-attention 层，用来对齐解码器编码层产生的内容。由于本文利用Multi-head Attention 对序列进行编码，下文将主要阐述Multi-head Attention 机制。

#### 3.3.2.1 Multi-head Self-attention 机制

  本节将先介绍Scaled Dot-product Attention 机制。Vaswani 等人[40]在论文中曾将Attention机制描述为一个问题(query)与一组键值对(key-value pairs)的函数，函数的输出则是由键值(value)的加权求和得到，其中权重则是由问题(query)与对应的关键字(key)计算得到。设问题(query)，键值对的关键字(key)，键值对的键值(value)，Scaled Dot-product Attention函数的输出，则函数输出的计算公式如下：

  Scaled Dot-product Attention与普通的基于內积的Attention机制的唯一区别就在于scaled dot-product attention 有一个缩放因子，表示词向量的维度，当词向量维度很大时，就会偏大，使得softmax函数处于梯度较小的区域，而乘上缩放因子，可以在一定程度上修正这种情况。

  为在多个层次上获取语义关联，Vaswani 等人[40]提出Multi-head Attention结构，该机制将问题,键值对中的关键字和键值分别进行多次线性变换之后，进行Scaled Dot-product Attention操作，再将结果进行合并操作，再次经过线性变换，得到最后最终的输出。Multi-head Attention的网络结构如下图所示：

图3-7：Multi-head Attention网络结构

  Multi-head Attention 使得模型能够获取来自不同位置的多个表示子空间的信息。Multi-head Attention操作的公式如下：

  其中分别表示线性变换的权重，表示将Q、K、V 进行线性映射的次数。原始模型使用个并行的attention 层。

  当Multi-head Attention中的问题Q，键值对中的关键字K与键值V均对应着同一个序列时，K=V=Q，则称这种结构为Multi-head Self-attention。这样做的主要是为了学习序列内部各个元素的依赖关系，捕获序列中的内部结构。

# 4 研究内容与实施方案

## 4.1 研究内容

### 4.1.1 数据集介绍

#### 4.1.1.1 SNLI数据集

  2015年，Bowman 等人[5]为了促进深度学习在文本蕴含识别中的应用，通过众包的方式，收集了大规模的文本蕴含识别的语料 Stanford Natural Language Inference (SNLI)。SNLI 语料中包含大量的句子对，其中句子对的关系共分为三种，分别是：蕴含关系、冲突关系和中立关系。如果前提文本可以推断出假设文本，那么两句话之间就是蕴含关系（Entailment），如何前提文本可以推断出假设文本是错误的，那么两句话之间就是冲突关系（Contradiction），否则则是中立关系。从数据集中随机选取的部分样本如下表所示，

表4-1：SNLI语料库中部分数据样例

  蕴含关系：前提中描述多名男性在玩足球，由此可以推断出多个男人在做运动，从前提可以推断出假设，因此两句话是蕴含关系。冲突关系：前提中描述一个男人在检查一个 人的制服，与这个男人在睡觉显然冲突。中立关系：前提中一个大一点的和一个年轻一点的男人在微笑，而假设中描述两个男人在微笑，并且因为猫在地板上玩耍而大笑。从前提中无法推断出假设是否正确，因此两句话是中立关系。

  SNLI中的原始的训练数据为 550152 对，开发集数据和测试集数据均为10000对。但是在众包过程中的人工标注往往会有一些误差，每对前提和假设经过五人标注，最终以三人及三人以上标注结果相同的标注称为黄金标注。经过规则过滤非黄金标注后的训练数据为549367对，开发集为9842对，测试集为9824对。并且所有数据中，前提的平均长度为14.1，假设的平均长度为8.3，说明通常前提文本中包含的信息比假设文本中包含的信息更加丰富，可以根据这一点来辅助进行文本蕴含识别。SNLI的统计信息如下表所示：

表4-2：SNLI统计信息表

#### 4.1.1.2 MultiNLI数据集

  2017年，Williams 等人[5]建立了更加复杂的文本蕴含任务的数据集MultiNLI (Multi-Genre Natural Language Inference)。该数据集的建立方式与SNLI类似，但数据的来源更加丰富，语言形式更加多样。SNLI数据集中的文本形式主要以书面文本为主，MultiNLI数据集中的文本包含口语形式的句子；SNLI数据集的数据主要来源于Flickr30k语料库的图像标题，而MultiNLI数据集的内容数据来源可分为如下10类：

* FACE-TO-FACE：出自“夏洛特叙事与对话集”(the Charlotte Narrative and Conversation Collection)，21世纪初进行的双方面对面的对话；
* GOVERNMENT：公共领域政府网站的报告，演讲，信件和新闻稿；
* LETTERS：20世纪90年代末至21世纪初期印第安纳州慈善筹款论坛跨文化交流中心(he Indiana Center for Intercultural Communication of Philanthropic Fundraising Discourse)的来信；
* 9/11：美国国家恐怖袭击委员会(the National Commission on Terrorist Attacks Upon the United States)于2004年7月22日发布的关于911的公众报告；
* OUP:牛津大学出版社(the Oxford University Press)出版的五篇关于纺织工业和儿童发展的非小说类作品；
* SLATE:1996年至2000年间撰写的Slate杂志档案中的流行文化文章；
* TELEPHONE:来自宾夕法尼亚大学的语言数据联盟，1990年或1991年间进行的电话交谈的录音文字;
* VERBATIM:贝立兹出版社(Berlitz Publishing)于21世纪初出版的旅行指南；
* FICTION:一些在1912年至2012年完成的当代小说，包含各种内容主题。

  MultiNLI数据集共包含432702对句子，其中训练集包含392702对句子，验证集与测试集各包含20000对句子。其中训练集中数据主要来源共有5类，也就是说，验证集和测试集中有5类其他来源的数据是没有在训练集中没有出现过的。据此，我们把验证集和测试集分为匹配(matched)和非匹配(mis-matched)两类。并且所有数据中，前提句的最大长度为401，平均长度为22.3；假设句的最大长度70，平均长度为8.3；其中有91%的句子是完整的。MultiNLI数据集与SNLI数据集的统计情况如下表所示：

表4-3：MultiNLI统计信息表

## 4.2 实施方案

  鉴于Chen 等人所建立的增强序列推理模型(Enhanced Sequential Inference Model, ESIM)ESIM[41]在文本蕴含数据集SNLI与MutiNLI上的成功，本文依据ESIM模型总结出基于Bi-LSTM与Attention的文本蕴含识别模型的基本架构，并希望能够对模型的并行计算能力与数据性能指标进行改善。下文将对模型的主要部分进行描述并提出改进计划。

### 4.2.1 基本模型结构

  文本蕴含任务可以抽象为一个三分类任务，模型大致可分为5层，分别为：序列输入层、序列编码层、序列间相关信息抽取层、特征提取及融合层、分类层。序列输入层对文本序列进行向量化的表示；序列编码层将融合词的上下文信息；序列间相关信息抽取层将从假设序列中提取与前提序列相关的信息，同理也从前提序列中提取与假设序列相关的信息；特征提取层提取语义特征信息并将两序列的特征信息进行融合；分类层将提取后的特征信息映射到类别标签。模型的结构如图所示：

图4-1：模型基本网络结构

1. 序列输入层

  对于序列输入层，本文仅使用预训练的词向量表，并对词向量表中未出现的词进行随机初始化。通过词向量映射表将字符映射为向量，我们得到前提序列 与假设序列 ，其中。

1. 序列编码层

  为了让序列中每个元素都融合上下文信息，模型利用双向长短记忆网络(Bi-LSTM)分别对前提序列和假设序列进行编码，编码过程可由如下公式所示：

  其中 分别是一个h维的向量，分别是前提序列的第i个词向量和假设序列的第j个词向量在经过双向长短记忆网络后输出的隐状态向量。双向长短记忆神经网络每个时刻输出的隐状态向量不仅具有序列在该时刻词的信息，并且还包含该时刻词的上下文信息。

1. 序列间相关信息抽取层

  对输入序列进行编码之后，前提序列 和假设序列 中每个元素包含着对应时刻词的信息和其上下文的信息，但两个序列之间的联系并没有描述，模型需要对两个序列的关系进行建模描述。局部推理模块利用软注意力(soft attention)机制对两个被编码的序列的各个元素之间的关系进行建模。为了判断假设序列中的每个词是否可以从前提序列中推理出来，最直观的想法是比较两个序列中每个元素之间是否具有蕴含、矛盾或中立关系。模型利用Attention机制对两元素之间的差异进行衡量，首先计算两个序列各个元素之间的相似度，然后通过softmax操作得到一个权重向量，再对前提序列中的每个元素进行加权求和，得到前提序列针对假设序列相关信息向量。同理可得，假设序列针对前提序列的相关信息向量。上述过程可由如下公式表示：

  其中表示前提序列的第i个元素与假设序列的第j个元素相关程度，计算元素之间关联程度的方式有很多，本文通过最简单的方式，计算两序列中各个词向量之间的内积来衡量两词之间关联程度，公式如下：

  为抽取序列元素与序列间相关信息之间的语义关系，序列元素和序列间相关信息需要进行交互。本文采用三种交互方式：向量连接、按位相乘和按位相减。交互过程如下公式所示：

  其中表示按位相乘，是深度学习中向量交互的常用方式；需要指出的是，按位相减操作后进行线性变换与直接进行连接操作后进行线性变换是等价操作，但这种方式在许多实验中发现对实验效果有明显提升。

  本文采用BiLSTM对交互后的信息进行进一步编码，计算过程如下：

1. 特征提取与融合层

  本文通过池化操作对语义特征进行提取，并将特征向量进行连接操作对特征进行融合。池化操作主要是将平均池化和最大池化相结合，然后将池化后的向量进行连接操作，形成一个长度固定的向量，向量的计算过程如下：

1. 分类层

  最后，模型利用多层感知机(Multi-Layer Perceptron)对句子蕴含关系进行分类。本模型的多层感知机只有一个隐藏层，激活函数是tanh，最后输出层利用softmax进行数据归一化。分类的计算过程如下：

  其中和分别是全连接层的权重和偏置，输出向量中的每一个元素表示相对应蕴含关系的概率。

### 4.2.2 实验细节与部分实验结果

  本文选用 300-D Glove 840B 中预训练的词向量来初始词向量。对于Glove中未出现的词，我们对其词向量进行随机初始化。训练过程中，我们选用Adam来优化梯度下降算法，其中Adam优化器中的超参数，初始学习率，每次训练数据的数目。模型参数方面，所有的长短记忆神经网络(LSTM)的隐状态向量和初始词向量的维度都是300维。我们在每层全连接层的输入处应用dropout机制，以加快训练速度并防止过拟合，dropout的比例均为0.5。

  目前，我们分别在数据集SNLI和MultiNLI数据集上进行了训练并测试。训练及测试结果如表所示：

表4-4：实验结果表

Train Corpus

Model

Embedding-Dimention

Params

Train Acc

SNLI Dev Acc

MNLI Dev-matched Acc

MNLI Dev-mismatched Acc

SNLI

EMSI

300

-

91.06%

86.24%

55.12%

55.88%

Multi-NLI

EMSI

300

-

91.06%

80.19%

73.33%

73.61%

### 4.2.3 改进计划

1. 预处理

  文本蕴含数据集中包含一定量的指示代词、数字、近义词和反义词。为了直观观察这些语言现象的影响，我们需要统计这些语言现象。为削弱这些语言现象的影响，我们需要进行指代消解、共指消解、数字规格化、近义词与反义词的替换等操作。

1. 序列输入

  本文仅仅利用词向量对文本进行表示，而一个向量表示一个词，不足以体现词的多义性。2017年，Vasmani 等人[40]提出ELMo词嵌入模型，该模型在具体任务中使用多个词向量的线性组合表示一个词，不同层次的词向量表示语义的不同层次。当词在不同语境下时，多个层次的词向量通过不同的线性组合对词进行表示，这样使得词具备语境信息。

  同时为提高文本表示的准确性，我们可以加入更多的序列特征，例如：词性特征、序列重合特征和句法特征。

1. 序列编码

  本文利用BiLSTM对序列进行编码，但LSTM存在串行计算并且不能够获取长期的依赖信息。2017年，Vaswani 等人[40]利用Attention机制来替换循环神经网络，并在机器翻译任务上获得明显的性能提升。受此次启发，我们可以利用Multi-Head Attention 加 CNN 对序列进行编码。利用Multi-Head Self-Attention机制有如下优点：1）并行计算，提升训练和测试速度；2）获取长距离依赖信息；3）允许模型在不同的表示子空间里学习相关信息。Vaswani 等人[40]在论文中利用卷积核尺寸为1的两层的一维卷积操作对向量空间进行线性变换，我们可以尝试其他尺寸的卷积核。

1. 特征提取及融合操作

  本文采用池化和连接操作对特征信息进行提取并融合。受到 Gong 等人[42]的启发，我们可以利用一种新的信息融合方式。设与分别是序列间相关信息抽取层输出的前提序列和假设序列，我们将两序列中的每个元素两两做连接操作，公式如下：

  操作之后的融合信息的张量(tensor)。然后，我们可以利用卷积神经网对融合信息进行特征提取。

1. 数量蕴含问题

  基于深度学习的文本蕴含模型很难完成数量关系的蕴含关系识别，例如：前提序列为“本人今年24岁”，而假设句为“本人今年不到30岁”，基于深度学习的文本蕴含系统很可能误判关系。而对于数量蕴含关系问题，Roy 等人[23]通过数学演算的方法解决了数量蕴含问题。我们可以参考 Roy 等人的解决方案，对数量蕴含问题进行解决。

1. 集成学习

  针对不同蕴含问题会有不同解决方案，并且针对不同特征的输入会有不同处理方法，我们可以运用集成学习策略对多个方法的结果进行整合，以提高性能指标。

### 4.2.4 预计困难

1）模型的调试。在模型的每个功能层之间会出现值域不在激活区的情况，这需要分析每个功能层的输入输出，决定是调整功能层结构还是增加标准化(normalization)模块。

2）序列元素影响可视化。为观察模型判断蕴含关系是否是根据正确的关键词，我们需要显式的通过可视化程序进行观察。而如何进行可视化操作，选举那些特征进行可视化描述，这些还需要后期学习调试。

3）模型后期会加入一定的传统文本特征以观察是否能够提升，传统文本特征（例如：词性特征、文本相似度特征和句法特征），需要一些自然语言处理的技术（序列标注模型和文本相似度计算）的支持，这些本人之后需要继续学习相关理论并实践获取。

# 参考文献

[1] QUIÑONERO-CANDELA J, DAGAN I, MAGNINI B, 等. Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment, First Pascal Machine Learning Challenges Workshop, MLCW 2005, Southampton, UK, April 11-13, 2005, Revised Selected Papers[M]. Springer, 2006, 3944.

[2] MACCARTNEY B. Natural Language Inference[D]. 2009: 179.

[3] 郭茂盛, 张宇, 刘挺. 文本蕴含关系识别与知识获取研究进展及展望[J]. 计算机学报, 2017.

[4] BOWMAN S R, ANGELI G, POTTS C, 等. A large annotated corpus for learning natural language inference[J]. 2015.

[5] WILLIAMS A, NANGIA N, BOWMAN S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, 2018: 1112–1122.

[6] JIJKOUN V, RIJKE M de, OTHERS. Recognizing textual entailment using lexical similarity[C]//Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment. Citeseer, 2005: 73–76.

[7] ADAMS R, NICOLAE G, NICOLAE C, 等. Textual entailment through extended lexical overlap and lexico-semantic matching[C]//Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Association for Computational Linguistics, 2007: 119–124.

[8] MEHDAD M, MATTEO N, ELENA C, 等. Edits: An open source framework for recognizing textual entailment[J]. 2009.

[9] 任函, 盛雅琦, 冯文贺, 等. 基于知识话题模型的文本蕴涵识别[J]. 中文信息学报, 2015, 29(6): 119–126.

[10] ZHANG Z, YAO D, CHEN S, 等. Chinese textual entailment recognition based on syntactic tree clipping[G]//Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer, 2014: 83–94.

[11] 基于混合主题模型的文本蕴涵识别[J]. 计算机工程, 2015, 41(5): 180–184.

[12] 赵红燕, 刘鹏, 李茹, 等. 多特征文本蕴涵识别研究[J]. 中文信息学报, 2014, 28(2): 109–115.

[13] HUANG W-J, LIU C-L. Exploring lexical, syntactic, and semantic features for Chinese textual entailment in NTCIR RITE evaluation tasks[J]. Soft Computing, Springer, 2017, 21(2): 311–330.

[14] MARNEFFE M-C, RAFFERTY A N, MANNING C D. Finding contradictions in text[J]. Proceedings of ACL-08: HLT, 2008: 1039–1047.

[15] MACCARTNEY B, GALLEY M, MANNING C D. A phrase-based alignment model for natural language inference[C]//Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, 2008: 802–811.

[16] BASAK R, NASKAR S K, PAKRAY P, 等. Recognizing textual entailment by soft dependency tree matching[J]. Computación y Sistemas, Centro de Investigación en Computación, IPN, 2015, 19(4): 685–700.

[17] SULTAN M A, BETHARD S, SUMNER T. Feature-rich two-stage logistic regression for monolingual alignment[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. 2015: 949–959.

[18] NOH T-G, PADÓ S, SHWARTZ V, 等. Multi-level alignments as an extensible representation basis for textual entailment algorithms[C]//Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics. 2015: 193–198.

[19] HOBBS J R, STICKEL M E, APPELT D E, 等. Interpretation as abduction[J]. Artificial intelligence, Elsevier, 1993, 63(1-2): 69–142.

[20] RAINA R, NG A Y, MANNING C D. Robust textual inference via learning and abductive reasoning[C]//AAAI. 2005: 1099–1105.

[21] MOLDOVAN D, CLARK C, HARABAGIU S, 等. Cogex: A logic prover for question answering[C]//Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Association for Computational Linguistics, 2003: 87–93.

[22] AKHMATOVA E. Textual entailment resolution via atomic propositions[C]//Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment. Citeseer, 2005, 150.

[23] ROY S. Reasoning about quantities in natural language[D]. University of Illinois at Urbana-Champaign, 2017.

[24] VENDROV I, KIROS R, FIDLER S, 等. Order-Embeddings of Images and Language[J]. CoRR, 2015, abs/1511.06361.

[25] MOU L, MEN R, LI G, 等. Recognizing entailment and contradiction by tree-based convolution[J]. CoRR, abs/1512.08422, 2015, 2.

[26] BOWMAN S R, GAUTHIER J, RASTOGI A, 等. A fast unified model for parsing and sentence understanding[J]. arXiv preprint arXiv:1603.06021, 2016.

[27] ROCKTÄSCHEL T, GREFENSTETTE E, HERMANN K M, 等. Reasoning about entailment with neural attention[J]. arXiv preprint arXiv:1509.06664, 2015.

[28] WANG S, JIANG J. Learning natural language inference with LSTM[J]. arXiv preprint arXiv:1512.08849, 2015.

[29] CHENG J, DONG L, LAPATA M. Long short-term memory-networks for machine reading[J]. arXiv preprint arXiv:1601.06733, 2016.

[30] HINTON G E, MCCLELLAND J L, RUMELHART D E, 等. Distributed representations[M]. Carnegie-Mellon University Pittsburgh, PA, 1984.

[31] MIKOLOV T, CHEN K, CORRADO G, 等. Efficient Estimation of Word Representations in Vector Space[J]. 2013: 1–12.

[32] MIKOLOV T, SUTSKEVER I, CHEN K, 等. Distributed Representations of Words and Phrases and their Compositionality[J]. 2013: 1–9.

[33] PETERS M E, NEUMANN M, IYYER M, 等. Deep contextualized word representations[J]. 2018.

[34] PENNINGTON J, SOCHER R, MANNING C D. GloVe: Global Vectors for Word Representation[J]. 2014.

[35] HOCHREITER S, SCHMIDHUBER J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8): 1735–1780.

[36] GRAVES A. Supervised sequence labelling[G]//Supervised sequence labelling with recurrent neural networks. Springer, 2012: 5–13.

[37] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. CoRR, 2014, abs/1409.0473.

[38] SUTSKEVER I, VINYALS O, LE Q V. Sequence to Sequence Learning with Neural Networks[J]. 2014: 1–9.

[39] LUONG M, PHAM H, MANNING C D. Effective Approaches to Attention-based Neural Machine Translation[J]. CoRR, 2015, abs/1508.04025.

[40] VASWANI A, SHAZEER N, PARMAR N, 等. Attention Is All You Need[J]. 2017.

[41] CHEN Q, ZHU X, LING Z, 等. Enhanced LSTM for Natural Language Inference[J]. Research in Immunology, 2016, 141(4): 431–440.

[42] GONG Y, LUO H, ZHANG J. Natural Language Inference over Interaction Space[J]. 2017(2017): 1–15.