自然语言处理初步 大作业二

姓名	学号	电话	邮箱
刘志丹	1120193154	15664420680	1318257253@qq.com
任凯文	1120193291	15802527880	mine268@foxmail.com
赖昱行	1120192236	13677610313	loadinglyh@qq.com
卜梦煜	1120192419	18110246062	1061146405@qq.com

目录

1 实验目的

构造一个汉语词义自动消歧系统。词义消歧的目的在于对于拥有多个词义的词语，需要结合上下文确定该词语在文中的具体含义。例如句子:“诸葛亮卒于二三四年。”一句中词语“卒”的意思有“士兵”和“死亡”两种含义，在本句中语义消歧的目的就是确定“卒”应该解释为“士兵”还是解释为“死亡”。

在工程实践或学术研究视角下，词义消歧不应该是用词典中的词语解释词典中的词语——即“自解释”的。每个词语拥有一个或多个词义，这个词义以一种唯一的记号标记，例如“来到 Ae01 Ak01 Jk93”表示词语“来到”拥有词义“来到 Ae01 Ak01 Jk93”，词义消歧的目的则是确定在当前语句中词语“来到”对应的意思应该是“Ae01”“Ak01”还是“Jk93”。同样地，给出同义词的定义，对于两个词语，如果他们拥有至少一个相同的词义，那么它们就可以被认为是同义词。

我们采用了两种实验方法完成了词义消歧的任务，即 Lesk 算法消歧和基于 Attention-based Bi-LSTM 模型的消歧。

2 实验原理

2.1 基于 Attention-based Bi-LSTM 的消歧模型

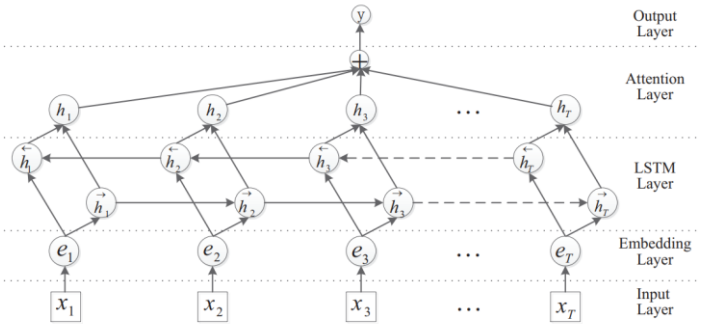
消歧模型参考了《Chinese Word Sense Disambiguation using a LSTM》[1]中提出的模型，这里简述模型完成分词的基本思想。对于一个多义词，为了消除歧义，我们采用同义词替换的方法，将这个问题转换为另一个分类问题。具体而言，即对于一个分好词的句子中的需要消歧的一个词语，我们通过同义词库的方式查询这个词语的其他同义词，将这些词语分别替换到句子中，一般认为在这些替换得到的句子中只有一句是通顺的，这句通顺的句子对应的词语就可以作为这个词语的正确解释。

例如句子“联想/是/中国/企业/。”中我们希望消除词语“希望”的词义，已知“希望”的同义词有“联想企业”和“关联想象”。分别替换得到句子：

1．联想企业/是/中国/企业/。

2．关联想象/是/中国/企业/。

显然句 1 正确而句 2 语义不通。故“联想”在这句话中的正确含义为“联想企业”。我们定义在替换得到的句子中，语义正确的句子为正类，语义错误的句子为负类，对每个句子进行使用 Attention-based Bi-LSTM 作为二分类区分正类和负类。



图表 2-1 Attention-based Bi-LSTM

形式化的，对于句子

$$S=(w_1,w_2,\cdots,w_i,\cdots,w_n),$$

希望确定词语 w_i 的歧义。我们有该词语的所有同义词集合

$$W_i=\{\hat{w}_1^i,\cdots,\hat{w}_j^i,\cdots,\hat{w}_m^i\},$$

我们用每一个 W_i 中的词语替换 w_i 得到新的句子

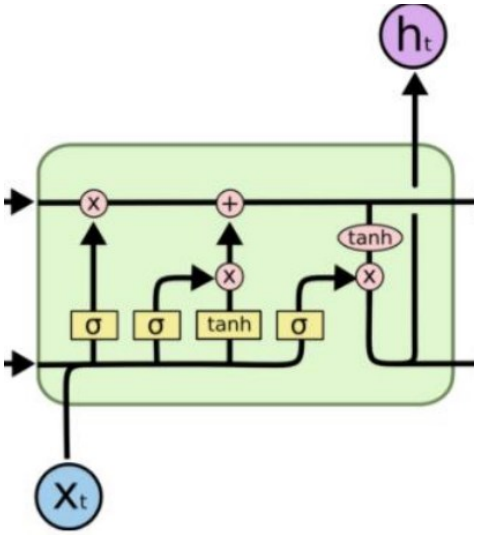
$$\mathbb{S}_i=\{(w_1,\cdots,\hat{w}_j^i,\cdots,w_n):1\leqslant j\leqslant m\},$$

将句子送入网络进行二分类，凡是分类为正类的句子对应的词语都能对 w_i 进行正确的解释，凡是分类为负类的句子都不能对 w_i 进行正确的解释。这样即可训练出一个对多义词词义敏感的模型。模型预训练完成后，对于需要消歧的词语，将其替换后产生的所有新句子依次放入模型中进行预测，选取正确的可能性最大的那一个作为词义正确的句子，即可完成词义消歧。

2.1.1 Embedding 层

在向模型输入句子时，首先需要经过一个预训练的 Embedding 层将词转化成相应词向量。这里采用的是网上下载的 50 维词向量的预训练模型 ctb.50d.vec[3]。

2.1.2 LSTM



图表 2-2 LSTM

LSTM 是一种特殊的 RNN，主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题。简单来说，就是相比普通的 RNN，LSTM 能够在更长的序列中有更好的表现。这样的特性在 NLP 问题的处理中体现出了独特的优势。LSTM 的神经元结构从前往后分为遗忘阶段，选择记忆阶段，输出阶段。这三个门的数学表示如下：

遗忘门公式：

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f)$$

$$i_t=\sigma(W_i\cdot[h_{t-1},x_t]+b_i)$$

选择记忆公式：

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_C)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_O)$$

输出公式：

$$h_t = o_t \times \tanh(C_t)$$
$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t$$

2.1.3 Attention 层

本次实验所用的模型在双向 LSTM，即 Bi-LSTM 的基础上，还增加了一个 Attention 层来加强词义与句子中上下文的关联性。对于 T 个时间序列的 Bi-LSTM 的输出H: $[h_1, \ldots, h_T]$，进行以下处理$M = \tanh(H)$，$\alpha = softmax(w^T M)$，$r = H\alpha^T$，$h^* = \tanh(r)$其中 w 为 Attention 层中经过训练学习的参数矩阵h^*是最终的输出向量

2.1.4 Linear 层

模型最后使用一个 Linear 层，将经过 Attention 层的输出向量映射成到正类和负类的概率。

2.1.5 损失函数

由于是分类问题，训练过程中使用交叉熵作为损失函数。

2.2 基于 Lesk 算法的消歧

2.2.1 算法原理

Lesk 算法认为，一个词在词典中的词义解释与该词所在句子具有相似性，相似性可以由相同单词的个数来表示。对某个歧义词的多种语义计算与句子的相似性，相似性最高的认为是该歧义词在句子中的词义。

2.2.2 词频-逆文件频率 TF-IDF

TF-IDF 是一种用于资讯检索与咨询探勘的常用加权技术，用于评估一个词对于一个语料库的重要程度。TF-IDF 技术的基本思想是，词的重要性与在一个语料文件中出现的频率成正比，与词在整个语料库中出现的频率成反比。TF-IDF 可分为词频 TF 和逆向文件频率 IDF 两部分。词频 TF 用于统计一个词在一个语料文件中出现的频率。公式如下：

$$\mathrm{TF}_i = \frac{\text{词在语料文件}i\text{中出现的次数}}{\text{语料文件}i\text{的总词数}}$$

逆向文件频率 IDF 用于衡量一个词对整个语料库的类别区分能力。基本思想为：如果语料库中包含词 w 的文档越少，词 w 越能代表该语料库。公式如下：

$$\mathrm{IDF} = \log\left(\frac{\text{语料库中语料文件的个数}}{\text{包含词}w\text{的语料文件个数}+1}\right)$$

根据 TF-IDF 技术的基本思想，可将词对语料库的重要程度定义如下：

$$\mathrm{TF}-\mathrm{IDF}=\mathrm{TF}\times\mathrm{IDF}$$

在实际应用中，由于常用词在此算法下容易获得较高的权值，为减少常用词的影响，TF-IDF 方法还需过滤掉常用词，只对重要词语进行计算。

3 实验步骤

3.1 基于 Attention-based Bi-LSTM 的消歧模型

数据处理部分是指用于训练 Attention-based Bi-LSTM 的消歧模型的数据处理。数据来源哈尔滨工程大学。[2] 词向量来源网络。[3]

首先进行同义词词典的读取。同义词词库选用 HIT-IRLab-Cilin，其中的词语按照词义整理，每一行按照空格分隔首先是该行词语的意思与该行词语之间关系（有相同、相似、不同、单义等），然后是若干个该意向下的词语。首先需要做的是统计出同义词词典中拥有两个及以上意思的多义词以及每个多义词的意思，其次还要记录下每个意思所拥有的词语。

代码中分别采用 syn_dict 和 syn_sense 存储这两个必要的数据结构，两个结构都是字典，分别可以用于查询每个词语拥有的意思和拥有某个意思的词语，例如 syn_dict['为']能够以列表形式读取词语“为”的所有意思，syn_sense['Ae01']能够读取所有包含意思“Ae01”的词语。实现按照词语和意思两个索引方向的查询。

然后进行词义语料库的处理。词义数据使用了哈工大数据库中的数据样本，该数据库不仅标注了每条句子中词语的词性，还标注了每个词语的词义，词义的格式同 HIT-IRLab-Cilin。最初的思路是希望筛选出该数据库中只包含一个多义词的句子，但是符合这个要求的句子数量只有 38 条，不足以支撑后续的训练过程，故将多义词数量上限有一个提升到 15 个，从而获取了 923 条句子，每个句子识别出其中多义词的位置并于句子共同存储到 syn_lines 中。

syn_lines 是一个列表，每一个元素结构为([num], [(str, str, str)])，其中前一个列表中的每一个数字表示了句子中多义词的下标位置，后一个列表中的每个元组表示该句子的每一个词语的内容、词性和意思。

需要特别注意的是，部分特殊词语和标点的词义被标注为“-1”，注意不要使这些标注影响到数据读取。

然后生成训练数据。训练数据生成思路如下，对于表明了多义词位置的 syn_lines 中的每一条句子，首先将其原封不动作为正类，然后任选其中任意一个多义词替换为该词语另一意思下的另一个词语。例如句子“为/人民/服务。”中“为”是多义词，拥有两个不同的意思“A”和“B”，在这句话中的意思是“A”，那么我们选择同样拥有意思“B”的另一个词语“作为”替换到句子中形成新的句子“作为/人民/服务”，这个句子作为负类。

然后生成验证数据。验证数据的生成为了能够计算准确率和召回率，其格式与训练数据不同。对于一条句子，其中必然包含一个或以上多义词，对每一个多义词我们选择其每个其他词义下的每一个词语替换原来的多义词，这样的句子标成负类且与原来的标为正类的句子形成一组。按照这样的分组策略有助于计算准确率。

3.1.1 训练

在应用模型之前，首先要对模型进行训练。在 constant.py 中存有训练和应用时需要使用的常量和文件路径；prework.py 用于读取预训练的词向量并进行词到编号，编号到词向量的映射；model.py 中存放有 Attention-based Bi-LSTM 模型和调用模型以及对应的优化器的方法；

dataset.py 用于将预处理过的语料读入自定义的数据集中；evaluate.py 用于使用验证集评估模型的准确率；train.py 中含有在训练集上前后向传播然后进行参数迭代的代码，并在训练中进行验证；trainEntrance.py 是训练部分的入口，用于模型的实例化和 DataLoader 的载入，以及训练函数的调用。

运行 trainEntrance.py 后，会调用 get_Attention_LSTM()获取模型，然后使用自定义的 MyDataSet 读取语料，用 DataLoader 载入它，再使用 read_word2ve()函数读入预训练词向量，然后调用 fit()进行训练。训练过程中，每进行一轮都对模型参数进行了保存，下次训练或者部署应用时可以直接读入使用。

3.1.2 应用

与应用有关的模块主要有图形界面和接口部分。main.py，myUi.py 为整个应用程序提供了 GUI 和输入接口；interface.py 为 GUI 和模型之间提供了接口。运行 main.py 后，等待一段时间载入词向量与训练模型。在弹出的窗口中的文本框输入一个句子之后，点击分词进行切分，切分后点击歧义词汇。然后点击 ABL 消歧，显示消歧结果与窗口左下角。

3.2 基于 Lesk 算法的消歧

一个歧义词的义项可通过爬取百度百科网页，利用类名“polysemantList-wrapper cmn-clearfix”或“custom_dot para-list list-paddingleft-1”获取。对每个义项，获取相应百度百科网页下类名”main-content”中的内容，截取含有该词的句子作为该歧义词在该词义下的语料。

本算法参考了网络资料[4]。

3.2.1 爬虫模块算法设计

1）输入：歧义词 word；
2）调用 requests.get()方法建立网页链接，创建 BeautifulSoup 对象；
3）定位到类名"polysemantList-wrapper cmn-clearfix"或"custom_dot para-list list-paddingleft-1"处，提取歧义词义项和对应的网页 url；
4）对每个词义，打开对应网页，定位到类名”main-content”处，对正文按句子切分，选取含有歧义词的句子作为该歧义词在该义项下的语料；
5）输出：歧义词义项，各义项语料。

3.2.2 Lesk 模块算法设计

1）输入：句子 sent，歧义词 word；
2）爬取语料。调用爬虫模块，获取歧义词的所有义项和各个义项的语料；
3）分词。对 sent，调用 jieba 库分词，并去掉停用词，作为 sent 预处理结果 sent_cut；
4）对 sent_cut 中每个词，计算在各义项语料上的 TF、IDF、TF-IDF 值。再对 sent_cut 中所有词的 TF-IDF 值求和，作为歧义词该义项与句子 sent 的相似度；
5）对该歧义词各义项与 sent 的相似度排序，最大相似度对应的义项认为是歧义词在该句子下的词义；
6）输出：歧义词词义。

4 实验结果

4.1 基于 Attention-based Bi-LSTM 模型的词义消歧

生成的训练集摘要：

```
1 迈向 充满 希望 的 新 世纪 —— 一九九八年 新年 讲话 （ 附 图片 1 张 ）

0 迈向 充满 希望 的 新 世纪 —— 一九九八年 新年 讲话 （ 附 图片 1 张 大 ）

1 只要 我们 进一步 解放思想 ， 实事求是 ， 抓住 机遇 ， 开拓进取 ， 建设 有 中国 特色 社会主义 的 道路 就 会 越 走 越 宽广 。

0 只要 我们 进一步 解放思想 ， 实事求是 ， 抓住 机遇 ， 开拓进取 ， 建设 里里外外 中国 特色 社会主义 的 道路 就 会 越 走 越 宽广 。

1 我们 伟大 祖国 在 新 的 一 年 ， 将 是 充满 生机 、 充满 希望 的 一 年 。

0 我们 伟大 祖国 在 新 的 一 年 ， 将 是 充满 生机 、 充满 射 的 一 年 。
```

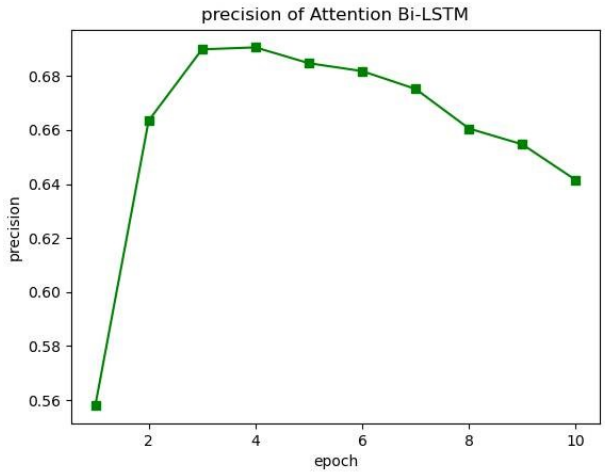
训练集格式如下。对于词义标注语料库[2]中的每一条语料，取其原有形式并去除词性标注和词义标注的分词结果作为正类，标记为 1 写入到训练集中（如上例中第一行所示）；同时，对于每个正类语料通过同义词替换的方式生成的负类语料，我们标记为 0 写入到训练集中（如上例中第二行所示）。按照训练集与测试机二八分的原则，训练集中包含语料 1476 行，共有 738 行词义标注语料库[2]中的语料被用来生成训练集。

生成的验证集摘要：

```
1 改革 的 重大 问题 讨论 。

0 改革 弩 重大 问题 讨论 。

0 改革 活脱 重大 问题 讨论 。

$

1 改革 的 重大 问题 讨论 。

0 改革 的 重大 疑问 讨论 。

0 改革 的 重大 心脏 讨论 。

0 改革 的 重大 中医学 讨论 。

0 改革 的 重大 体裁 讨论 。

$

1 改革 的 重大 问题 讨论 。

0 改革 的 重大 问题 推翻 。

$
```

验证集的格式不同于训练集，验证集中每一个“$”符号到前一个“$”符号或文件开头之间的句子算作一组，由同一条句子的同一个多义词通过同义词替换的方式生成，这样可以用于计算模型的准确率。最终生成的语料包含 5916 行语料，分为 1364 组。

在网络模型中训练，训练轮数与准确率的关系如下图所示。最终准确率可以达到 69%。在训练到第四轮的时候根据验证集上的准确率可以发现明显的过拟合现象。



图表 4-1 训练准确率与轮数的关系图

运行截图：

图表 4-2 运行界面 1

图表 4-3 运行界面 2-选词消歧

图表 4-4 运行截图 3-消歧结果

4.2 基于 Lesk 算法的消歧

爬取语料截图：

图表 4-5 爬取语料截图

运行实例截图：

图表 4-6 运行截图

5 实验总结

5.1 基于 Attention-based Bi-LSTM 的消歧模型

这次实验中，虽然在模型训练时达到了不错的准确率，但在实际使用的过程中效果并不如人意。而后检查数据集，发现其中存在很多脏数据，再加上数据集中语料稀少，推测模型学到的知识不够，并且过拟合严重，主要原因是这次没有向 HIT 申请到全部的数据集。实验后对模型分析可以发现一些可能的改进之处，比如在 Bi-LSTM 中添加 dropout 减轻过拟合现象；在歧义词的左右加上“<”和“>”符号，可以使模型对歧义词词义的注意力更集中；同理，可以在每个词的词向量后加上对于歧义词的 posing feature embedding。另外，限于时间原因，没有对模型的超参数进行研究，以后可以通过实验探究最合适的超参数取值。

5.2 基于 Lesk 算法的消歧

使用该方法获取语料的优点有：

1）语料方面。本方法不需要专门的语料库，且百度百科作为百科知识库，爬取的语料准确率较高；

2）Lesk 算法简单有效，在 SenVal-2 的评测任务中准确率可达到 50%-70%。

同时缺点有：

1）百度百科爬取的语料库较小，且计算算法准确率的验证集需要手动构造；

2）用 TF-IDF 值表示词对语料的重要程度并不总是准确的。比如，对于一些专有名词，虽然 IDF 值较高，但因为专有名词在整个语料库中出现的频率很低，会导致 TF-IDF 值较低，但这些专有名词往往是识别歧义词的关键词。

针对缺点可以提出一些改进思路：

1）使用语料更多、语料种类更丰富的语料库；

2）扩充停用词，进一步降低常用词对算法准确率的影响；

3）改进算法，对关键词的筛选应考虑词义因素，而不仅仅是词频因素。

6 小组分工

姓名	工作内容	完成文件
赖昱行	Attention-based Bi-LSTM 模型及其训练与预测	constant.py，dataset.py，evaluate.py，graph.py，interface.py 中的 MyWSD 类，train.py，trainEntrance.py，prework.py，model.py
刘志丹	GUI	myUI.py，main.py
任凯文	基于 Attention-based Bi-LSTM 模型消歧的训练数据预处理	data_preprocess.ipynb，gen_corpus.plain.utf8.txt，gen_valid.plain.utf8.txt，interface.py 中的 SentenceSubstitution 类
卜梦煜	Lesk 算法消歧	Lesk_bug.py，Lesk_main.py

表格 1 小组分工明细

7 引用

[1] SUN X-R, LV S-H, WANG X-D, 等. Chinese Word Sense Disambiguation using a LSTM[J]. LONG L, LI Y, LI X, 等. ITM Web of Conferences, 2017, 12: 01027.

[2] 哈工大信息检索研究中心(HIT CIR)语言技术平台共享资源和程序步骤[EB/OL]. [2021-12-26]. http://ir.hit.edu.cn/demo/ltp/Sharing_Plan.htm.

[3] Hins/Flat-Lattice-Transformer: code for ACL 2020 paper: FLAT: Chinese NER Using Flat-Lattice Transformer[EB/OL]. [2021-12-26]. https://github.com/Hins/Flat-Lattice-Transformer.

[4] 自然语言处理(NLP)之词义消歧(WSD)的简介与实现_IT 之一小佬的博客-CSDN 博客_词义消歧算法[EB/OL]. [2021-12-26]. https://blog.csdn.net/weixin_44799217/article/details/116498179.