苏州大学实验报告

院、系	计算机学院	年级专业	计算机科学	姓名	张昊	学号	1927405160
课程名称	人工智能与知识工程					成绩	
指导教师	陈文亮	同组实验者	无		实验日期	2021 年 10 月 14 日	

实 验 名 称　　语义网络：isA/AKO 关系抽取

一、 实验内容

语义网络：isA/AKO 关系抽取

1、中文分词：最大匹配算法

要求：编程平台不限（windows、linux），编程语言不限（C、C++）

子任务 1：使用最大匹配算法、词典文件（corpus.dict.txt），对语料（corpus.sentence.txt）进行分词

-- 将分词的结果输出到文件 corpus.out.txt 中；

-- 对比 corpus.answer.txt 和 corpus.out.txt，给出算法的 P/R/F 指标

输出：一个 corpus.out.txt 文件（格式参照 corpus.answer.txt）

P/R/F 指标(格式类似于：Precision = 36 / 100 = 36.00%)

子任务 2：使用最大匹配算法、人名文件（per.dict.txt），对语料（corpus.sentence.txt）进行人名识别

-- 人名文件无人数数目和最长字数

-- 将识别的结果输出到文件 per.out.txt 中

输出：一个 per.out.txt 文件（格式参照 corpus.answer.txt）

2、isA/AKO 关系抽取

实验数据：10 万篇文档（例如 百科页面），10 万个相关实体（例如 姚明、榴莲）

输出：

不少于 1 万关系对

每个关系对出现的实例句子

随机采样正确率不低于 50%（随机看 10 对）

输出格式（三元组）：

[实体，关系，类别] 句子

[李娜, isA 冠军] 李娜…(把整个句子附上)

二、 实验步骤

使用最大匹配算法的中文分词：

使用了双向最大匹配的分词算法，首先分别对句子进行前向最大匹配和反向最大匹配，分别应用前向最大匹配算法和逆向最大匹配算法对语句进行分词，若前向和逆向分词结果词数不同，取分词数量较少的，否则返回单字词语的数量较少的。

具体实现如下：

```python
def segmentation_sentence(sentence: str, words: set, join_str=' ', max_length=10, only=None) -> str:
    """
    最大匹配算法（双向最大匹配算法）
    :param sentence: 待分词句子
    :param words: 词典
    :param join_str: 拼接字符串
    :param max_length: 最大步长
    :param only: 仅使用某一算法进行分词，接受的参数为：'fmm'（前向最大匹配算法），'bmm'（逆向最大匹配算法）
    :return: 使用 join_str 拼接后的分词过的句子
    """

    def fmm(_sentence: str) -> list:
        """
        前向最大匹配算法
        :param _sentence: 待分词句子
        :return: 组成句子的词语
        """
        length = max_length
        seg = []
        while len(_sentence) > 0:
            word = _sentence[0:length]
            while word not in words:
                if len(word) <= 1:
                    break
                else:
                    length -= 1
                word = word[0:length]
            length = max_length
            seg.append(word)
            _sentence = _sentence[len(word):]
        return seg

    def bmm(_sentence: str) -> list:
        """
        逆向最大匹配算法
        :param _sentence: 待分词句子
        :return: 组成句子的词语
        """
        length = max_length
        seg = []
        while len(_sentence) > 0:
            word = _sentence[-length:]
            while word not in words:
                if len(word) <= 1:
                    break
                else:
                    length -= 1
                word = word[-length:]
            length = max_length
            seg.append(word)
            _sentence = _sentence[:-len(word)]
        seg.reverse()
        return seg
```

```
    # 分别应用前向最大匹配算法和逆向最大匹配算法对语句进行分词
    fmm_res = fmm(sentence)
    bmm_res = bmm(sentence)
    # 若前向和逆向分词结果词数不同，取分词数量较少的
    if len(fmm_res) < len(bmm_res):
        return join_str.join(fmm_res)
    elif len(fmm_res) > len(bmm_res):
        return join_str.join(bmm_res)
    else:
        fmm_single, bmm_single = 0, 0  # 前向和逆向分词结果中单字词语的数量
        same = True
        for x, y in zip(fmm_res, bmm_res):
            if x != y:
                same = False
            if len(x) == 1:
                fmm_single += 1
            if len(y) == 1:
                bmm_single += 1
        if same:
            # 分词结果相同，随便返回哪个都行
            return join_str.join(fmm_res)
        else:
            # 分词结果不同，返回单字词语的数量较少的
            if fmm_single <= bmm_single:
                return join_str.join(fmm_res)
            else:
                return join_str.join(bmm_res)
```

评价指标计算：首先把词序列转换为词语在句子中的位置集合。由于直接将词转换为集合会失去位置信息，故使用词语在句中的位置（起始位置和结束位置）来唯一地标识一个词。在计算全体的真正例 TP, 假正例 FP, 假反例 FN 时，对于输出和答案不同的：若结果文件结束但答案文件未结束（即有句子没输出），则作为 FN；若答案文件结束但结构文件未结束（即多输出了其他的内容），则作为 FP。具体实现如下：

```
def evaluation(output_file_path: str, answer_file_path: str):
    """
    评价函数
    :param output_file_path: 结果文件
    :param answer_file_path: 答案文件
    :return: 真正例TP, 假正例FP, 假反例FN
    """
    def index_set(seg: list) -> set:
        """
        将词序列转换为词语在句子中的位置集合
        :param seg: 词序列
        :return: 位置集合
        """
        seg_index_set = set()
        index = 0
        for word in seg:
```

```
            length = len(word)
            seg_index_set.add((index, index + length))
            index += length
        return seg_index_set
    def evaluation_sentence(output: list, answer: list):
        """
        评价一句话的分词结果
        :param output: 分词结果
        :param answer: 答案
        :return: 真正例TP, 假正例FP, 假反例FN
        """
        output_index_set = index_set(output)
        answer_index_set = index_set(answer)
        TP = len(output_index_set & answer_index_set)
        FP = len(output_index_set - answer_index_set)
        FN = len(answer_index_set - output_index_set)
        return TP, FP, FN
    TP_ALL, FP_ALL, FN_ALL = 0, 0, 0  # 全体的真正例TP, 假正例FP, 假反例FN
    with open(output_file_path, 'r', encoding='utf-8') as output_file, \
            open(answer_file_path, 'r', encoding='utf-8') as answer_file:
        while True:
            output_line = output_file.readline()
            answer_line = answer_file.readline()
            if not output_line and not answer_line:  # 已到达文件尾
                break
            if not output_line:  # 结果文件结束但答案文件未结束（有句子没输出）
                FN_ALL += len(answer_line.strip().split())  # 作为假反例
                continue
            elif not answer_line:  # 答案文件结束但结构文件未结束（多输出了其他的内容）
                FP_ALL += len(output_line.strip().split())  # 作为假正例
                continue
            evaluation_res = evaluation_sentence(output_line.strip().split(),
answer_line.strip().split())
            TP_ALL += evaluation_res[0]
            FP_ALL += evaluation_res[1]
            FN_ALL += evaluation_res[2]
    return TP_ALL, FP_ALL, FN_ALL
```

基于规则的 isA/AKO 关系抽取：

整体思路：首先将句子分词（采用 jieba 加自己实现的最大匹配混合的方法），将句子转化为词列表，且由于数据中的噪声比较多，所以只关注前 m 个句子（经实验测试，m=1 时效果最佳），对每个句子去除停用词和大部分的标点符号，按照句号来划分句子。

```
@staticmethod
def __handle_sentence(sentence: str) → list:
    # 分句，并将句子转换为词列表，只考虑前 $(CONFIG.sentence_max_count) 个句子，并去除停用词以及大部分标点符号
    sentence = re.sub('。?(%+)', ', ', sentence)
    sentence = re.sub('#+|-{3,}', '', sentence)
    words = []
    _cnt = 0
    for sentence in filter(lambda x: len(x.strip()) ≠ 0 and x.find('。') == -1,
                           re.split(r'(\s*。\s*)', sentence)):
        _words = jieba.lcut(sentence)
        words.append(EDict({
            'sentence': ''.join(_words),
            'words': [word for word in _words if word not in CONFIG.stop_words],
        }))
        _cnt += 1
        if _cnt ⩾ CONFIG.sentence_max_count:
            break
    return words
```

下一步是查找在句子（词列表）中首次出现在实体集中的实体，并将其作为本句的实体。

```
@staticmethod
def __find_name_in_sentence(sentence: list, names: set):
    # 查找句子中首次出现在实体集中的实体，未找到返回None
    for word in sentence:
        if word in names:
            return word
    return None
```

之后基于规则来查找标签有哪些（代码过长，请看附件）。对于多个标签的冲突，越靠前的标签认为是最符合的标签。规则的定义如下：首先在实体词之后查找有 isa 关系的关键词，将其后的标签抽取出来；其次是注意到存在很多“XX 简介”“XX 介绍”之类的词语，而这些“XX”往往指示了这一实体的标签；再有就是实体本身可能存在“小说”“公司”“企业”之类的词语，这也表明了它的标签。这三类规则词语定义如下：

```
'relation_start_words': {'简介', '介绍', '类型', '说明', '类型'},
'isa_words': {'是', '从属', '附属', '归属', '包括', '包含', '蕴含', '囊括'},
'accelerated_words': {'小说', '电影', '游戏', '公司', '企业', '学院'},
```

抽取的代码如下：（具体函数实现参见附件）

```
def __extract(self, texts: list) -> list:
    result = []
    for text in tqdm(texts):
        for sentence in self.__handle_sentence(text):  # 分句，并将句子处理成为词列表
      first_name = self.__find_name_in_sentence(sentence.words, self.names)  # 首次出现在实体集中的实体
```

```
            if first_name is not None:
                    all_tags = self.__find_tags_in_sentence(sentence.words,
first_name, self.tags)  # 找到所有标签
                if len(all_tags) != 0:
                        result.append(EDict({
                            'name': first_name,
                            'tags': all_tags,
                            'sentence': sentence.sentence,
                        }))
        return result
```

基于这种规则，每个实体对应的标签有多个，会拉低查准率，所以最后在处理输出的时候，缩减的输出的数量：

```
@staticmethod
def neatly_string(result, file=None):
    output = ''
    for item in tqdm(result):
        cnt = 0
        for tag in item.tags:
            if tag.isdigit():
                continue
            output += f'[{item.name}, isA, {tag}]  {item.sentence}\n'
            cnt += 1
            if cnt ⩾ math.floor(len(item.tags) / CONFIG.relation_number_limit):  # 缩减冗余的标签，提高accuracy
                break
    if file:
        with open(file, 'w', encoding=CONFIG.encoding) as f:
            f.write(output)
    return output
```

三、 实验结果

所有输出都保存在 output 目录下

任务 1：子任务 1：（task1_1.py）

```
戴相龙 说 中国 经济 发展 为 亚洲 作出 积极 贡献
新华社 福冈 5月 11日 电 （ 记者 乐绍延 ）
中国 人民 银行 行长 戴相龙 今天 在 亚洲 开发 银行 第30 届 年会 的 “ 亚洲 未来 30 年 ” 研讨会 上 说 ， 中国 的 经济 发展 为 亚洲 的 繁荣 与 发展 作出 了 积
戴相龙 在 发言 时 说 ， 中国 的 发展 得益于 亚洲 国家 和 地区 的 经济 发展 与 合作 ， 与 亚洲 的 繁荣 息息相关 。
他 指出 ， 随着 经济 的 持续 增长 和 改革 开放 政策 的 深入 ， 中国 将 在 亚洲 经济 区域 合作 中 发挥 更 积极 的 作用 。
中国 经济 的 快速 增长 将 为 亚洲 地区 创造 更 多 的 贸易 机会 ， 在 今后 四 年 中 ， 中国 将 为 世界 提供 将近 7000亿 美元 的 市场 。
关于 香港 回归 中国 后 的 国际 金融 地位 问题 ， 戴相龙 强调 ， 香港 的 国际 金融 地位 不但 能够 维持 ， 而且 还 会 得到 加强 。
在 谈到 亚洲 经济 的 发展 前景 时 ， 戴相龙 认为 ， 亚洲 经济 将 继续 保持 稳定 的 发展 势头 ， 仍 将 成为 推动 世界 经济 发展 的 主导 力量 。
戴相龙 同时 指出 ， 亚洲 经济 发展中 还 存在 工资 上涨 过快 削弱 竞争力 ； 高级 研究 、 管理 人才 严重 匮乏 ； 能源 、 交通 等 基础 设施 相对 落后 等 制约 经济
戴相龙 认为 ， 要 保持 亚洲 地区 经济 增长 ， 既 需要 亚洲 各 国 继续 开发 利用 自身 的 经济 潜力 ， 也 需要 进一步 加强 区域 经济 合作 。
亚洲 国家 和 地区 今后 除了 在 商品 、 投资 领域 加强 合作 外 ， 还 应 在 科技 和 环保 以及 货币 政策 和 金融 监管 方面 加强 合作 。
亚洲 开发 银行 总裁 佐藤光夫 主持 了 这 次 研讨会 。
日本 前 首相 宫泽喜一 、 印度 财政 部长 奇丹巴拉姆 和 芬兰 环境 部长 佩卡·哈维斯托 也 在 研讨会 上 发 了 言 。
（ 完 ）
美国 罗素 二O - - 二O 协会 考察 中国 中西部 地区
新华社 重庆 五月 十二日 电 （ 记者 李佩 ）
拥有 万亿 美元 可投 资金 、 每 年 都 要 选择 一 个 新兴 市场 作为 投资 机会 的 美国 罗素 二O - - 二O 协会 ， 如今 把 目光 转向 中国 的 中西部 。
从 十日 起 ， 这 个 协会 的 成员 对 刚刚 升为 直辖市 的 重庆市 进行 了 三 天 的 参观 考察 。
重庆 位 于 中国 的 西南 地区 ， 是 中国 西部 最大 的 工业 城市 ， 也是 中国 西部 利用 外国 投资 最 多 的 城市 。
今年 三月 十四日 重庆 成为 中国 第四 个 、 也是 西部 唯一 的 一 个 直辖市 。
在 重庆 期间 ， 协会 成员 听取 了 市政府 关于 重庆市 发展 战略 、 对外 开放 优惠 政策 及 投资 环境 和 合作 机会 的 介绍 ， 并 参观 了 中外 合资 企业 庆铃 汽车
此前 ， 这 个 协会 的 成员 还 考察 了 中部 工业 名城 武汉 。
罗素 二O - - 二O 协会 是 由 世界 最大 的 养老金 管理 公司 弗兰克·罗素 公司 董事长 乔治·罗素 于 一九九O年 六月 发起 成立 的 组织 ， 由 二十 家 养老金 基金 组
据 介绍 ， 该 协会 在 中国 现有 投资 达 二十亿 美元 ， 此 次 访 华 的 主要 目的 是 考察 中国 中西部 地区 的 投资 环境 ， 了解 中国 对外 开放 政策 ， 特别是 在
```

评价指标：

```
C:\lang\Anaconda3\python.exe D:/Code/AI/E1/task1_1.py
Precision = 20301 / 20396 = 99.53%
Recall = 20301 / 20454 = 99.25%
F = 0.9953 * 0.9925 * 2 / (0.9953 + 0.9925) = 99.39%
```

子任务 2：(task1_2.py)

```
戴相龙说中国经济发展为亚洲作出积极贡献
新华社福冈5月11日电（记者乐绍延）
中国人民银行行长戴相龙今天在亚洲开发银行第30届年会的“亚洲未来30年”研讨会上说，中国的经济发展为亚洲的繁荣与
戴相龙在发言时说，中国的发展得益于亚洲国家和地区的经济发展与合作，与亚洲的繁荣息息相关。
他指出，随着经济的持续增长和改革开放政策的深入，中国将在亚洲经济区域合作中发挥更积极的作用。
中国经济的快速增长将为亚洲地区创造更多的贸易机会，在今后四年中，中国将为世界提供将近70000亿美元的市场。
关于香港回归中国后的国际金融地位问题，戴相龙强调，香港的国际金融地位不但能够维持，而且还会得到加强。
在谈到亚洲经济的发展前景时，戴相龙认为，亚洲经济将继续保持稳定的发展势头，仍将成为推动世界经济发展的主导力量。
戴相龙同时指出，亚洲经济发展中还存在工资上涨过快削弱竞争力；高级研究、管理人才严重匮乏；能源、交通等基础设施相对
戴相龙认为，要保持亚洲地区经济增长，既需要亚洲各国继续开发利用自身的经济潜力，也需要进一步加强区域经济合作。
亚洲国家和地区今后除了在商品、投资领域加强合作外，还应在科技和环保以及货币政策和金融监管方面加强合作。
亚洲开发银行总裁佐藤光夫主持了这次研讨会。
日本前首相宫泽喜一、印度财政部长奇丹巴拉姆和芬兰环境部长佩卡·哈维斯托也在研讨会上发了言。
（完）
美国罗素二O--二O协会考察中国中西部地区
新华社重庆五月十二日电（记者李佩）
拥有万亿美元可投资金、每年都要选择一个新兴市场作为投资机会的美国罗素二O--二O协会，如今把目光转向中国的中西
从十日起，这个协会的成员对刚刚升为直辖市的重庆市进行了三天的参观考察。
重庆位于中国的西南地区，是中国西部最大的工业城市，也是中国西部利用外国投资最多的城市。
今年三月十四日重庆成为中国第四个、也是西部唯一的一个直辖市。
在重庆期间，协会成员听取了市政府关于重庆市发展战略、对外开放优惠政策及投资环境和合作机会的介绍，并参观了中外合
此前，这个协会的成员还考察了中部工业名城武汉。
罗素二O--二O协会是由世界最大的养老金管理公司弗兰克·罗素公司董事长乔治·罗素于一九九O年六月发起成立的组织，由
据介绍，该协会在中国现有投资达二十亿美元，此次访华的主要目的是考察中国中西部地区的投资环境，了解中国对外开放政
```

任务 2：(task2.py)

首先会进行关系抽取，之后会随机抽 10 个句子，输出运行结果。

结果文件：output/extract_result.txt

```
Run - isA_AKO
运行: task2
读取标签 4000 个
100000 行文本加载完成
100%|██████████| 100000/100000 [03:58<00:00, 418.88it/s]
100%|██████████| 67533/67533 [00:17<00:00, 3841.70it/s]
100%|██████████| 10/10 [00:00<00:00, 639.97it/s]
========== 测试用例 ==========
香菇韭菜生肉包是一道由韭菜等为主要食材做成的菜品，属于面食。###香菇韭菜生肉包所需食材:猪肉350克%%%韭菜250克%%%香菇4只%%%酵母8克%%%蒜头适量%%%料酒适量%%%酱油适量%%%盐适量%%%糖适
桐城市第十一中学坐落在皖西南商贸名镇—青草镇。背依源远流长的大沙河，东邻206国道，西接桐潜公路，交通十分便利。%%%学校创办于1945年，最初定名为清河小学。1986年更名为桐城县青草塌高中，
“掌中加农炮”是CAPCOM公司在2005年开发的游戏：生化危机4中的1把左轮枪。###掌中加农炮入手方法:在佣兵模式中，全人物，全地图打出5星评价后可在商人处以0元购得。###掌中加农炮武器性能:火力
《网游之超级江湖》是连载于起点中文网的网游小说，作者是龙魂九段。###网游之超级江湖小说类型:游戏网游###网游之超级江湖内容简介:使命、骏马、美人、烈酒、秘籍、奇遇、恩怨、情仇、杀戮、侠义
SKF607-Z轴承，是SKF品牌轴承，型号为607-Z。###SKF607-Z轴承SKF607-Z轴承尺寸参数:系列：深沟球轴承,单列,单面防尘罩%%%外径：19mm%%%厚度：6mm
刘占兴，男，1966年3月生，汉族，内蒙古克什克腾旗人，1987年4月入党，1987年7月参加工作，北京大学哲学系伦理学专业研究生毕业，哲学硕士，副研究员。%%%现任北京市人民政府研究室党组书记、主
###（物体要达到绕地球飞行作圆周运动的速度）:第一宇宙速度分为两个别称：航天器最小发射速度、航天器最大运行速度。在一些问题中说，当某航天器以第一宇宙速度运行，则说明该航天器是沿着地球表
###诡鬼合集小说类型:灵异怪谈###诡鬼合集内容简介:罗逸作为一个普通人,却一次次陷入诡异的灵异当中,没有超人的能力,没有逆天的法宝,更没有奇遇,有的只是坚持和信念,看他如何逃出怪异的迷离,敬
字两石，清朝一代名臣，正一品衔。湖北大冶人，顺治九年进士，康熙年间官至武英殿大学士、吏部尚书。###余国柱生平简介:在朝为官36年：前32年清正廉明，励精图治，为大清王朝立下不朽功勋；后4年
###谁知花开花落小说类型:青春校园###谁知花开花落内容简介:当一个孤僻的女生,碰到一个开朗的男生,会碰擦出什么爱的火花呢。当五个男生围着一个女生团团转的时候又会有什么事情发生呢。让我们来
========== 运行结果 ==========
[香菇韭菜生肉包，isA，菜品]  香菇韭菜生肉包是一道由韭菜等为主要食材做成的菜品，属于面食
[掌中加农炮，isA，公司]  “掌中加农炮”是CAPCOM公司在2005年开发的游戏：生化危机4中的1把左轮枪
[网游之超级江湖，isA，小说]  《网游之超级江湖》是连载于起点中文网的网游小说，作者是龙魂九段
[SKF607-Z轴承，isA，轴承]  SKF607-Z轴承，是SKF品牌轴承，型号为607-Z
[诡鬼合集，isA，小说]  诡鬼合集小说类型:灵异怪谈诡鬼合集内容简介:罗逸作为一个普通人,却一次次陷入诡异的灵异当中,没有超人的能力,没有逆天的法宝,更没有奇遇,有的只是坚持和信念,看他如何
[谁知花开花落，isA，小说]  谁知花开花落小说类型:青春校园谁知花开花落内容简介:当一个孤僻的女生,碰到一个开朗的男生,会碰擦出什么爱的火花呢

100%|██████████| 6/6 [00:00<?, ?it/s]
```

四、 实验总结

通过本次实验，我掌握了正向最大匹配算法以及逆向最大匹配算法，并实现了中文分词以及分词

结果的评估。理解了语义网络的概念，理解实体，标签，关系抽取的概念，实现了基于规则的 isa/ako 关系的抽取方法。

五、 附录：源代码与运行方法

源代码过长，详见附件。

运行方法：

任务一的数据保存在 seg_data 目录下，任务二的数据保存在 isA_AKO_shuffled_data 目录下（由于文件过大，提交的附件中不包含这些文件），所有输出都会保存到 output 目录下。

使用 python 3.9 或更高版本运行。

首先安装依赖：

pip install -r requirements.txt

运行任务 1 的子任务 1：python3 task1_1.py

运行任务 1 的子任务 2：python3 task1_2.py

运行任务 2：python3 task2.py