周报

3月13日——3月17日

本周周一组会明确了任务目标：对4000万条机构数据进行机构（实体）消歧。并且初步规划了方法和步骤。数据格式：

```
    ],
    [
        "Chinese Acad Sci, Inst Phys, Beijing Natl Lab Condensed Matter Phys, Beijing 100190, Peoples R China",
        13503
    ],
    [
        "Beijing University of Aeronautics and Astronautics(Beijing University of Aeronautics and Astronautics,Beijing Un
iv. of Aero. and Astron.),Beijing,China",
        13310
    ],
    [
        "Hefei University of Technology(Hefei University of Technology),Hefei,China",
        13195
    ],
    [
        "Univ London Imperial Coll Sci Technol & Med, London, England",
        13089
    ],
    [
        "Beth Israel Deaconess Med Ctr, Boston, MA 02215 USA",
        12940
    ],
    [
        "China Agricultural University(China Agricultural University),Beijing,China",
        12929
    ],
    [
        "university of tubingen",
        12914
    ],
    [
        "Harvard Smithsonian Ctr Astrophys, Cambridge, MA 02138 USA",
        12763
    ],
    [
```

周二确定方法和流程

1. 第一步 做字符串拆解

利用第三方包，将一级机构，二级机构，邮编，地址抽取出来。

2. 第二步 **纠错**和扩展

一级机构做做纠错和扩展（如果有必要，可以加入NER）

明确一下**纠错和扩展**怎么做！需要用到哪些方法

3. 第三步 匹配

与库中的实体进行匹配（匹配算法找宏博哥）

4. 第四步 聚类

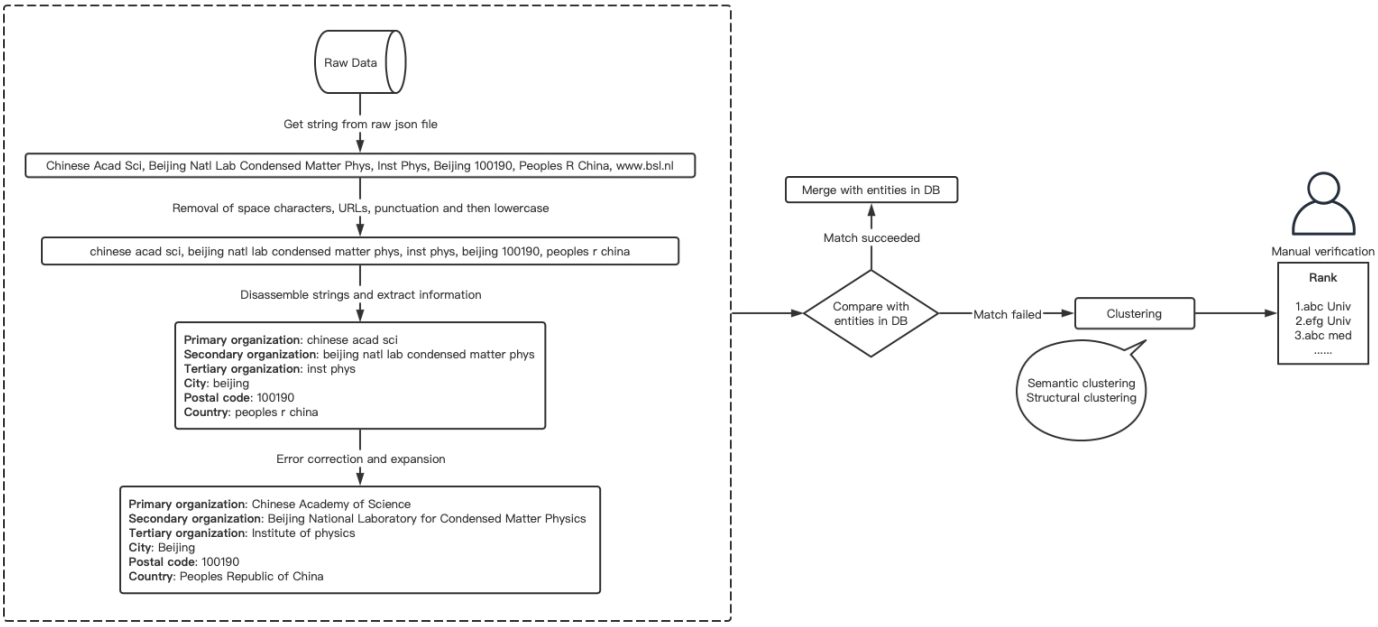
匹配不上的做聚类

（需要用到什么聚类方法？语义聚类，结构聚类）

不停迭代几次之后还是匹配不到，那么就把常见的聚类出来，再把开头排名靠前的人工查看

这一阶段如果发现匹配算法不准确，需要修改匹配算法（重新计算相似度）。

最后计算精度和召回率



周三利用第三方包，对字符串中的一级组织、二级组织、三级组织、城市、邮编、国家进行抽取。

抽取结果：


```
{"primary organization": [{"text": "institute of metal research", "start": 0, "end": 27, "probability": 0.9958675281032257}, {"text": "chinese academy of sciences", "start": 28, "end": 55, "probability": 0.5427676645675632}], "secondary organization": [{"text": "institute of metal research", "start": 0, "end": 27, "probability": 0.983910715558963}, {"text": "chinese academy of sciences", "start": 28, "end": 55, "probability": 0.47576641860911906}], "tertiary organization": [{"text": "institute of metal research", "start": 0, "end": 27, "probability": 0.9953853514769833}, {"text": "chinese academy of sciences", "start": 28, "end": 55, "probability": 0.797398826513323}], "city": [{"text": "shenyang", "start": 85, "end": 93, "probability": 0.9998965885268021}], "country": [{"text": "china", "start": 94, "end": 99, "probability": 0.9999813438184191}]}
```


周五：针对primary organization、secondary organization、tertiary organization、postal code、city、country中的probability中最大的值，进行存储

```
def get_items():
    """
    It opens the file, reads each line, and then extracts the information we need
    :return: the maximum probability of the organization, postal code, country and city.
    """
    with open("./raw_org_extracted.txt", "r") as f:
        for line in f:
            line = ast.literal_eval(line)
            # print(line)
```

```
            if 'primary organization' not in line:
                continue
            orgs = line['primary organization']
            max_prob_o = max(orgs, key=lambda x: x['probability'])
            max_prob_o_text = max_prob_o['text']
            # print(max_prob_o_text)

            if 'postal code' not in line:
                continue
            zip_code = line['postal code']
            max_prob_p = max(zip_code, key=lambda x: x['probability'])
            max_prob_p_text = max_prob_p['text']
            # print(max_prob_p_text)

            if 'country' not in line:
                continue
            country = line['country']
            max_prob_c = max(country, key=lambda x: x['probability'])
            max_prob_c_text = max_prob_c['text']
            # print(max_prob_c_text)

            if 'city' not in line:
                continue
            city = line['city']
            max_prob_ci = max(city, key=lambda x: x['probability'])
            max_prob_ci_text = max_prob_ci['text']
            # print(max_prob_ci_text)

    return max_prob_o_text,max_prob_p_text,max_prob_c_text,max_prob_ci_text
```