用Bi-GRU和字向量做端到端的中文关系抽取

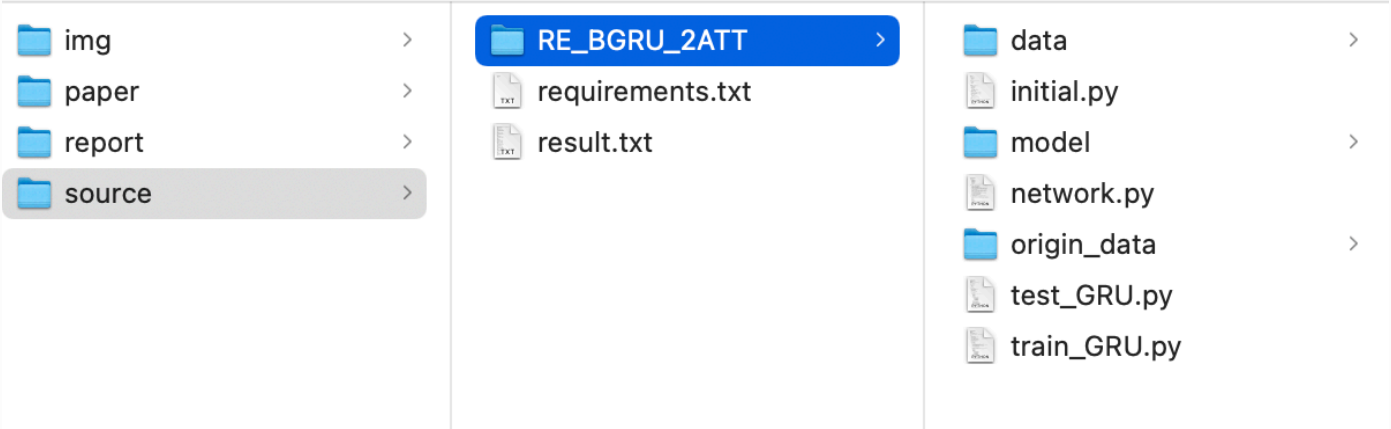
组长姓名：赵炫皓 学号：1120200603 班级：07112006 负责项目代码编写和报告撰写

组员：韦杰文 学号：1120200605 班级：07112005 负责项目代码的修改及训练和报告的审核修改

作业二代码文件如下：

```
---homework2
            --- img // 保存report中使用的图片
            --- paper // 模型所参考论文
            --- source // 作业源码
                --- RE_BGRU_2ATT
                    --- data // 保存数据
                    --- initial.py // 初始化数据
                    --- model // 保存训练模型
                    --- network.py // 实现网络
                    --- origin_data // raw data
                    --- test_GRU.py // 测试文件
                    --- train_GRU.py // 训练主文件
```

文件如图所示：



实体识别和关系抽取是构造知识图谱的基础。简单来说，给定两个实体同时出现的文本，关系抽取可以判断两个实体之间的关系。

本次作业以实验为目的，使用一个用双向GRU，字与句子的双重Attention模型，以天然适配中文特性的character embedding作为输入，网络开源数据作为训练预料构建中文关系抽取模型。

部分模型代码参考清华的开源项目thunlp/TensorFlow-NRE。此代码中tensorflow版本较低，为适应tensorflow(2以上版本)做了较多改动。

双向GRU加Dual Attention模型

双向GRU加字级别attention的模型想法来自文章“Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification” (Zhou et al.,2016)。此处的模型改为GRU，且对句子中的每一个中文字符输入为character embedding。原论文模型和修改后的模型分别由图1，图2所示。

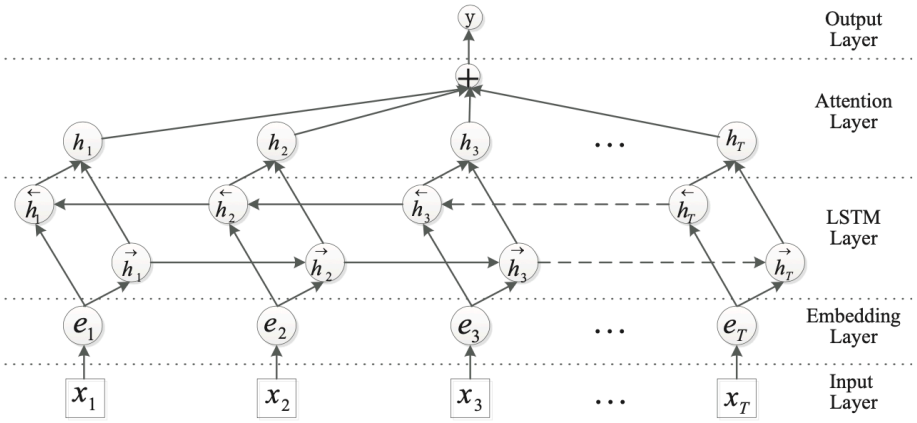


图1.论文原文模型

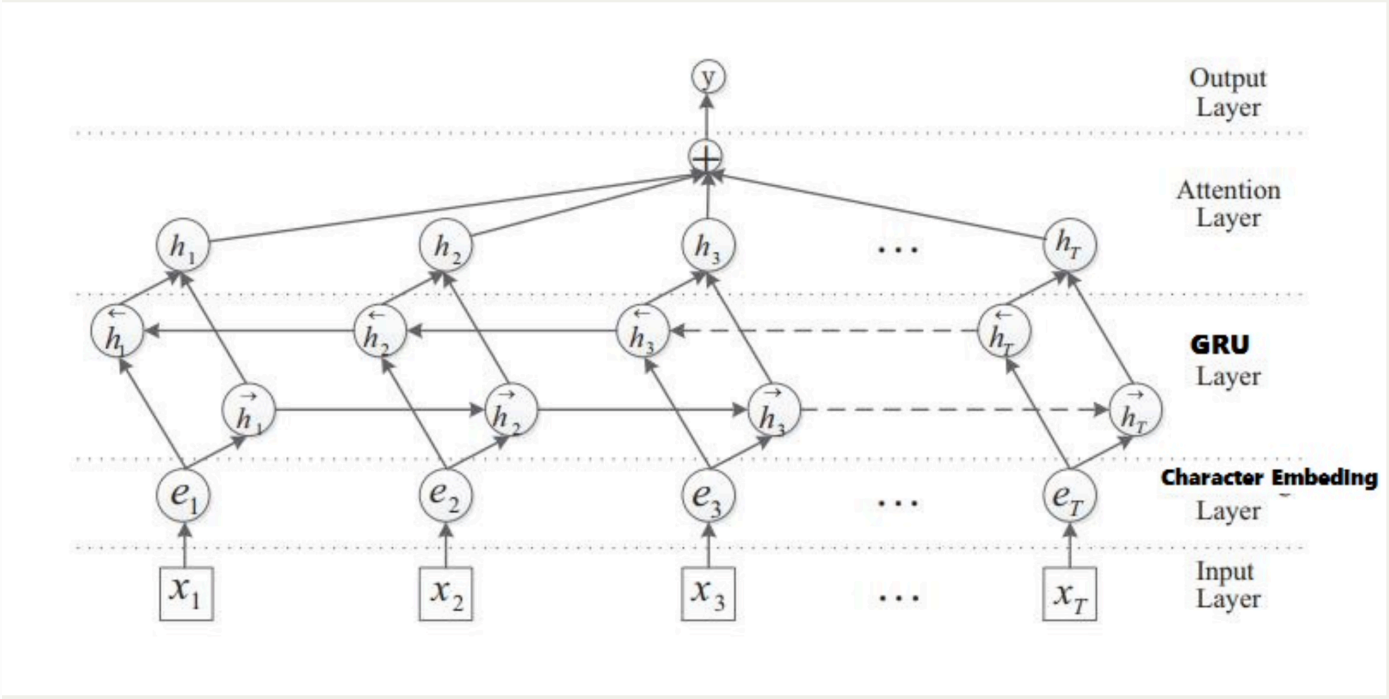


图2.修改后的模型

句子级别attention的想法来自文章“Neural Relation Extraction with Selective Attention over Instances” (Lin et al.,2016)。同上所述，将模型改为GRU,这样的模型对每一种类别的句子输入做共同训练，加入句子级别的attention。图三和图四分别为原文模型和修改后的模型。

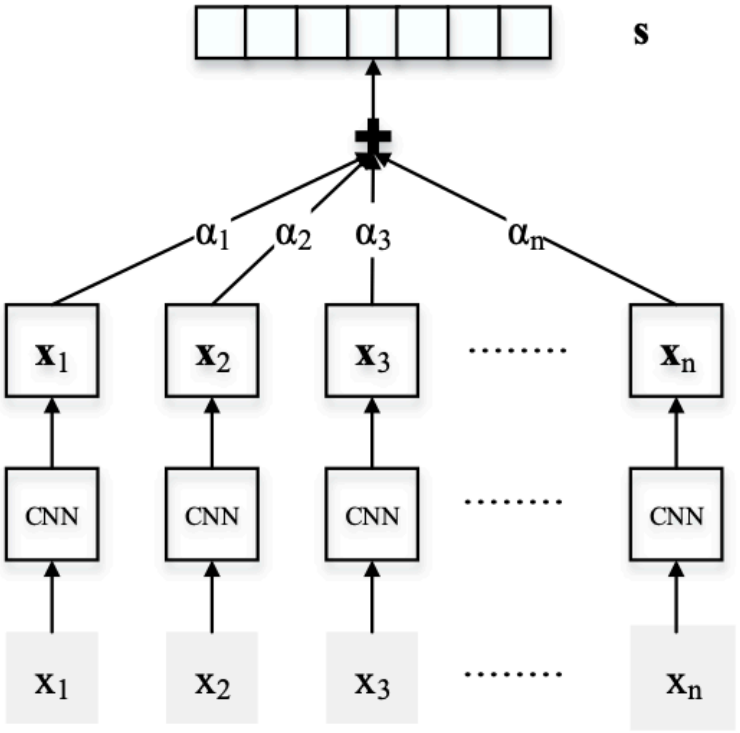


Figure 1: The architecture of sentence-level attention-based CNN, where x_i and $\mathbf{x}_i$ indicate the original sentence for an entity pair and its corresponding sentence representation, α_i is the weight given by sentence-level attention, and $\mathbf{s}$ indicates the representation of the sentence set.

图3.原文的模型

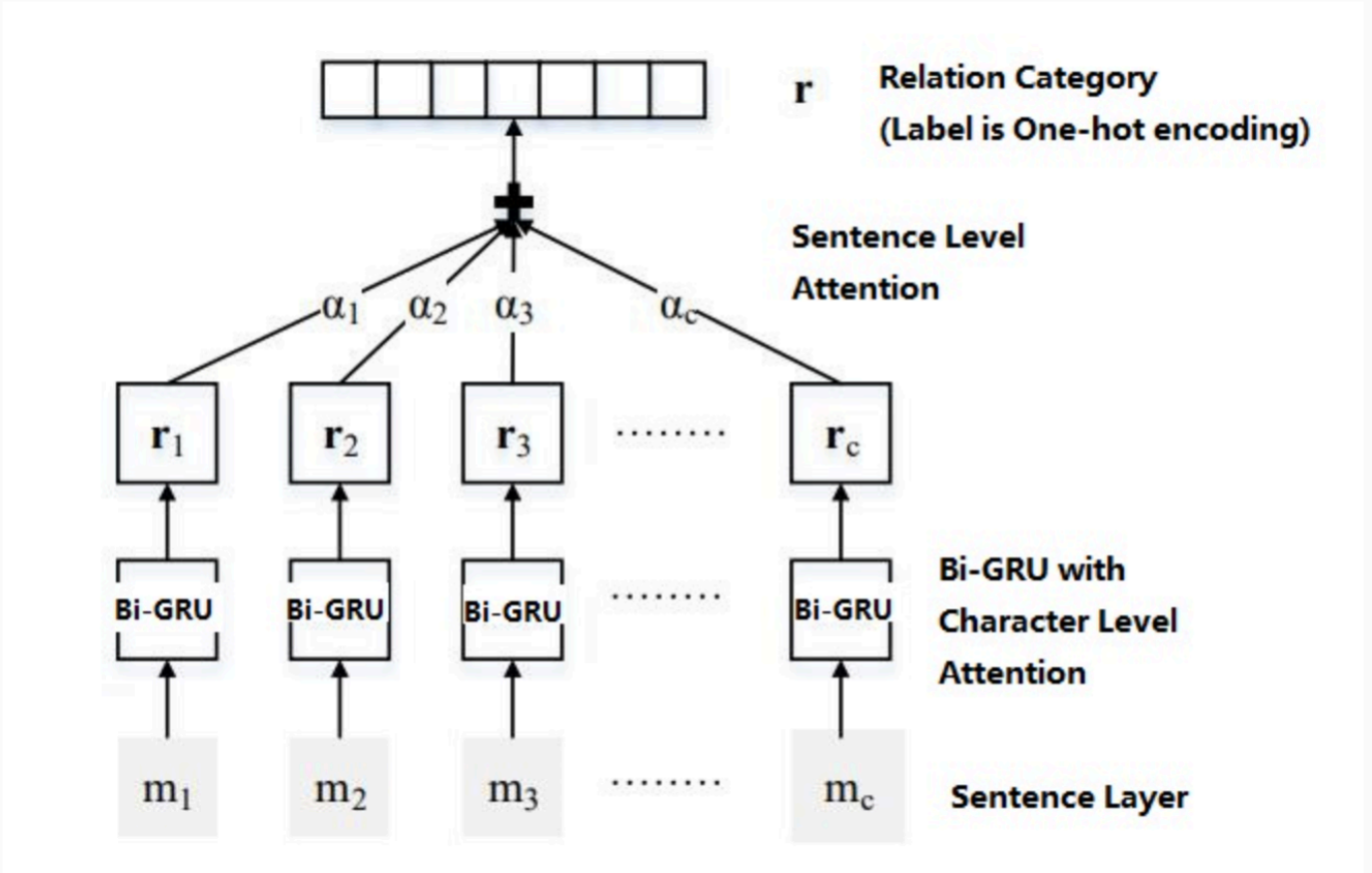


图4.修改后的模型

考虑到用GRU替换原论文中的LSTM原因是：考虑到GRU中的parameter更少，模型更加简单，从实用的角度来讲容错性更高，可以方便实践。

语料获取

语料取自开源项目Roshanson/TextInfoExp和复旦知识工厂，得到了具有确定关系的实体对。

Roshanson中dataset5和复旦知识工程中的语料信息如图5，6所示

郭台铭 郭晓玲 父母 孝顺的郭晓玲，昨大清早7时就和父亲郭台铭及曹斯杰，带着喜饼到爱物园给母亲上香，这是她今年3个月内第4度前往扫
郭台铭 曹斯杰 父母 孝顺的郭晓玲，昨大清早7时就和父亲郭台铭及曹斯杰，带着喜饼到爱物园给母亲上香，这是她今年3个月内第4度前往扫
诸葛亮 刘备 上下级 刘备-相关语录刘备托孤时候对诸葛亮说：“君才十倍曹丕，必能安邦定国，终定大事。
刘备 诸葛亮 上下级 刘备-相关语录刘备托孤时候对诸葛亮说：“君才十倍曹丕，必能安邦定国，终定大事。
沈坚强 庄泳 夫妻 可尽管沈坚强和庄泳同为国家队队友，但是两人的结合却是后来的事情。
庄泳 沈坚强 夫妻 可尽管沈坚强和庄泳同为国家队队友，但是两人的结合却是后来的事情。
邵逸夫 方逸华 夫妻 1952年，邵逸夫在登台期间，认识现时的丈夫方逸华，当时邵逸夫对她的歌艺尤为赏识。
方逸华 邵逸夫 夫妻 1952年，邵逸夫在登台期间，认识现时的丈夫方逸华，当时邵逸夫对她的歌艺尤为赏识。
李铭顺 范文芳 夫妻 两位弟弟家中排行：老二信仰：佛教2009年4月15日，李铭顺与新加坡演员范文芳注册结婚，结束爱情长跑。
范文芳 李铭顺 夫妻 两位弟弟家中排行：老二信仰：佛教2009年4月15日，李铭顺与新加坡演员范文芳注册结婚，结束爱情长跑。
孙天牧 孙墨佛 父母 向中华慈善总会捐赠孙天牧四尺整纸书谱、孙墨佛四尺整纸山水画《雪霁图》参加义拍。
孙墨佛 孙天牧 父母 向中华慈善总会捐赠孙天牧四尺整纸书谱、孙墨佛四尺整纸山水画《雪霁图》参加义拍。
翁狄森 蒋怡 夫妻 -婚姻生活2013年，35岁的翁狄森与42岁的香港珠宝商、设计师男友蒋怡（Dickson）爱情长跑12年，两人已在1月18日悄悄在南
蒋怡 翁狄森 夫妻 -婚姻生活2013年，35岁的翁狄森与42岁的香港珠宝商、设计师男友蒋怡（Dickson）爱情长跑12年，两人已在1月18日悄悄在南
白崇禧 马佩璋 夫妻 白崇禧还特地对马佩璋夸赞谢和赓为人忠诚，胆识过人。
孔令侃 宋子文 亲戚 孔令侃一直就看不惯宋子文那股横不讲理的劲。
吴尊 赵晨浩 好友 聽被美国著名《人物》杂志评为亚洲新四小天王后，凭着超高人气，允浩、韩庚、郑赵晨浩、吴尊四人的身价倍增，代言以及参与影视节目录制等，身价均超过
允浩 韩庚 好友 聽被美国著名《人物》杂志评为亚洲新四小天王后，凭着超高人气，允浩、韩庚、郑赵晨浩、吴尊四人的身价倍增，代言以及参与影视节目录制等，身价均超过
允浩 赵晨浩 好友 聽被美国著名《人物》杂志评为亚洲新四小天王后，凭着超高人气，允浩、韩庚、郑赵晨浩、吴尊四人的身价倍增，代言以及参与影视节目录制等，身价均超过
允浩 吴尊 好友 聽被美国著名《人物》杂志评为亚洲新四小天王后，凭着超高人气，允浩、韩庚、郑赵晨浩、吴尊四人的身价倍增，代言以及参与影视节目录制等，身价均超过
林立果 林立衡 兄弟姐妹 、林立衡二人不由得笑起来，林立果和蔼地向张宁说：“你今后一定要多掌握些党的历史知识。
林立衡 林立果 兄弟姐妹 、林立衡二人不由得笑起来，林立果和蔼地向张宁说：“你今后一定要多掌握些党的历史知识。
林月云 邱嘉雄 情侣 [2]郭纯美情敌郭纯美2004年，林月云承认与另一个有妇之夫邱嘉雄保持了长达22年的地下情。
陈致中 黄百禄 父母 ”当有人提到黄百禄跟陈致中在美国有四栋豪宅，黄睿靓激动到不雅的言语都说出口。
黄百禄 陈致中 父母 ”当有人提到黄百禄跟陈致中在美国有四栋豪宅，黄睿靓激动到不雅的言语都说出口。
黄百禄 黄睿靓 父母 ”当有人提到黄百禄跟陈致中在美国有四栋豪宅，黄睿靓激动到不雅的言语都说出口。

图5.Roshanson项目中Data5的数据

儿子	小海绵
公益基金	黄晓明明天爱心基金
出生地	山东省青岛市市南区
出生日期	1977年11月13日
别名	教主、猫、钢钉侠、熊猫明、囧明
国籍	中国
外文名称	Huang Xiaoming
妻子	angelababy(杨颖)
影友会	明教
星座	天蝎座
毕业院校	北京电影学院表演系、北京大学国家发展学院EMBA2012级
民族	汉族
生肖	蛇
经纪人	郭亭婷

图6，7.复旦知识工程所抽取的数据效果

其中部分数据需要手动预处理，例如将"配偶"替换为"夫妻"以提高模型精度。

模型训练

配置要求：

```
Python >= 3.5
TensorFlow (>= 2.0)
scikit-learn(>= 0.18)
```

训练

1 所有数据准备在origin_data/中，包括了关系种类(relation2id.txt),训练数据(train.txt),测试数据(test.txt) 和中文字向量(vet.txt)。现有数据包括以下12种关系种类：

```
unknown，父母，夫妻，师生，兄弟姐妹，合作，情侣，祖孙，好友，亲戚，同门，上下级
```

2 将所有的数据通过字向量整理成numpy，存储至data/

```
python initial.py
```

初始化结果如图8，9所示：

```
reading word embedding data...
reading relation to id
reading train data...
reading test data ...
organizing train data
organizing test data
reading training data
seprating train data
seperating test all data
(base) myfile@zhaoxuanhaodeMacBook-Pro RE_BGRU_2ATT %
```

图8.数据预处理初始化

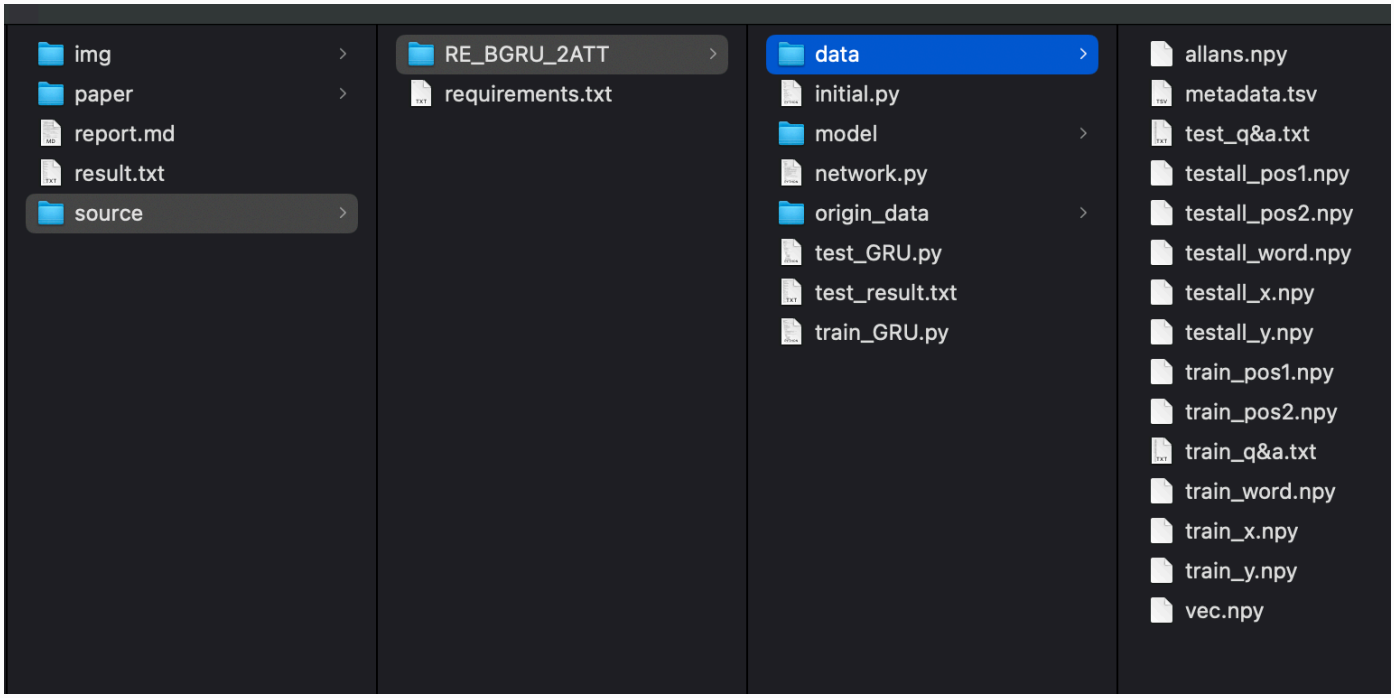


图9.处理后的文件

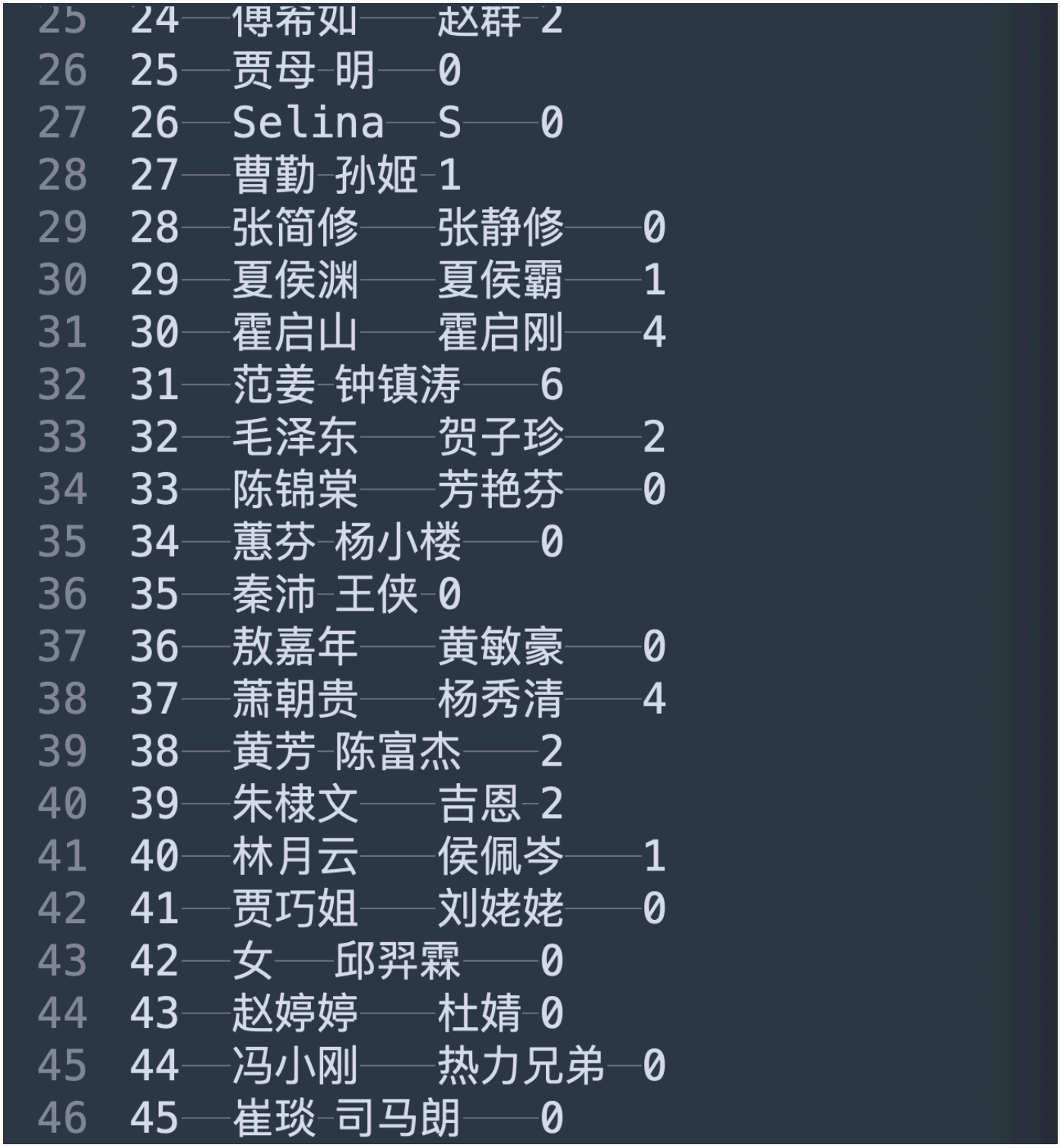


图10.字向量文件

3 运行train_GRU通过训练已经经过处理后的npy文件进行训练，模型会存储在model/。对于英文版模型参数几乎没有做改变。

运行

```
python train_GRU.py
```

训练效果如图11，12所示：

```
2022-05-16110:24:59.524880:step50,softmaX_loss 74.5103,acc 0.5

2022-05-1610:25:13.585860:step108,softmax_loss 69.0191, acc 0.56

2022-05-1610:25:27.834800:step158,softmax_loss 70.7606, acc 0.6
```

图11.训练过程

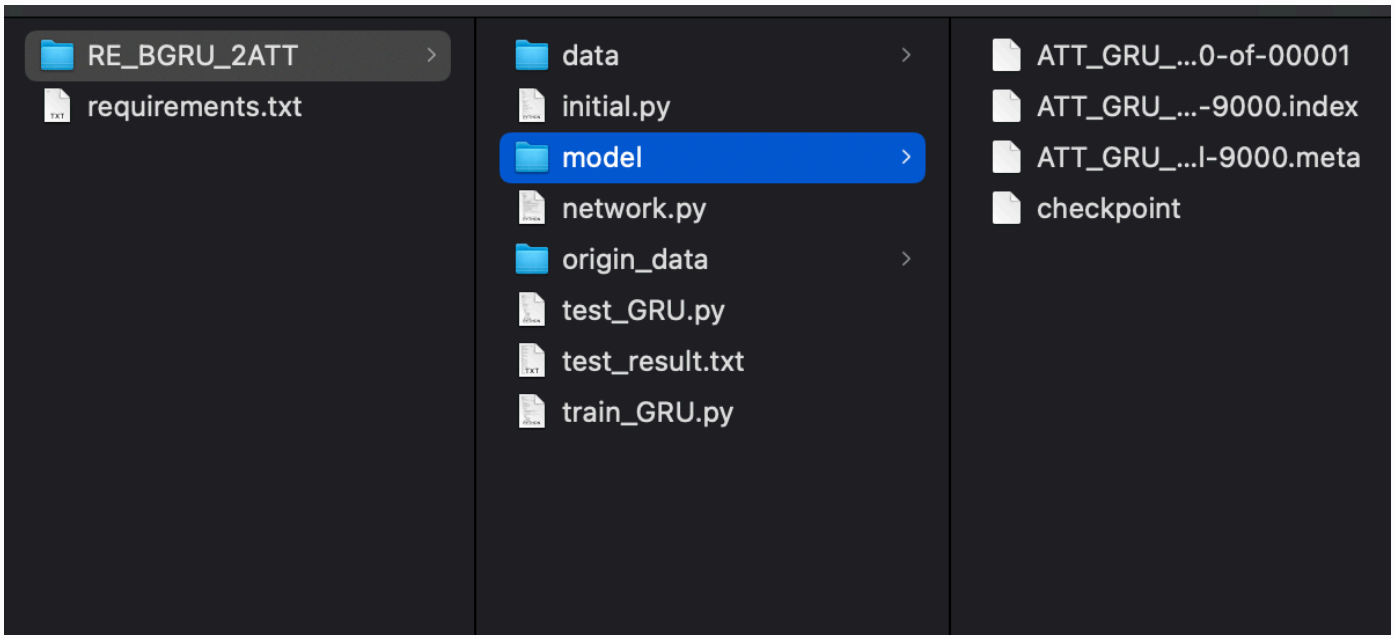


图12.训练后的模型

预测

代码一个main函数是用测试数据来测试准确率的，另一个则是供用户输入进行inference.

自动预测可以运行

```
python test_GRU.py
```

手动预测运行代码如下：

```
while True:
    #try:
        #BUG: Encoding error if user input directly from
command line.
```

```
        line = input('请输入中文句子，格式为 "name1 name2 sentence":')
        #Read file from test file
        '''
        infile = open('test.txt', encoding='utf-8')
        line = ''
        for orgline in infile:
            line = orgline.strip()
            break
        infile.close()
        '''
        en1, en2, sentence = line.strip().split()
        print("实体1: " + en1)
        print("实体2: " + en2)
        print(sentence)
        relation = 0
        en1pos = sentence.find(en1)
        if en1pos == -1:
            en1pos = 0
        en2pos = sentence.find(en2)
        if en2pos == -1:
            en2post = 0
        output = []
        # length of sentence is 70
        fixlen = 70
        # max length of position embedding is 60 (-60~+60)
        maxlen = 60
        #Encoding test x
        for i in range(fixlen):
            word = word2id['BLANK']
            rel_e1 = pos_embed(i - en1pos)
            rel_e2 = pos_embed(i - en2pos)
            output.append([word, rel_e1, rel_e2])
        for i in range(min(fixlen, len(sentence))):
            word = 0
            if sentence[i] not in word2id:
                #print(sentence[i])
                #print('==')
                word = word2id['UNK']
                #print(word)
            else:
                #print(sentence[i])
```

```
                #print('||')
                word = word2id[sentence[i]]
                #print(word)
            output[i][0] = word
        test_x = []
        test_x.append([output])
        #Encoding test y
        label = [0 for i in range(len(relation2id))]
        label[0] = 1
        test_y = []
        test_y.append(label)
        test_x = np.array(test_x)
        test_y = np.array(test_y)
        test_word = []
        test_pos1 = []
        test_pos2 = []
        for i in range(len(test_x)):
            word = []
            pos1 = []
            pos2 = []
            for j in test_x[i]:
                temp_word = []
                temp_pos1 = []
                temp_pos2 = []
                for k in j:
                    temp_word.append(k[0])
                    temp_pos1.append(k[1])
                    temp_pos2.append(k[2])
                word.append(temp_word)
                pos1.append(temp_pos1)
                pos2.append(temp_pos2)
            test_word.append(word)
            test_pos1.append(pos1)
            test_pos2.append(pos2)
        test_word = np.array(test_word)
        test_pos1 = np.array(test_pos1)
        test_pos2 = np.array(test_pos2)
         prob, accuracy = test_step(test_word, test_pos1,
test_pos2, test_y)
        prob = np.reshape(np.array(prob), (1,
test_settings.num_classes))[0]
        print("关系是:")
```

```
        #print(prob)
        top3_id = prob.argsort()[-3:][::-1]
        for n, rel_id in enumerate(top3_id):
            print("No." + str(n+1) + ": " + id2relation[rel_id]
+ ", Probability is " + str(prob[rel_id]))
```

手动检验结果

手动输入的句子应该是如下格式

```
name1 name2 sentence(which includes name1 and name2)
```

进行如下输入：

```
A B
A和她的丈夫B前日一起去英国旅行了。

A B
A和她的高中同学B两个人前日一起去英国旅行。

A B
B命令A在周末前完成这份代码。

A B
B非常疼爱他的孙女A小朋友。

A B
谈起曾经一起求学的日子，B非常怀念他的师妹A。



A B
B对于他的学生A做出的成果非常骄傲!

A B
B和A是从小一起长大的好哥们

A B
B的表舅叫A的二妈为大姐

A B
这篇论文是B负责编程，A负责写作的。
```

```
A B
B和A为谁是论文的第一作者争得头破血流。
```

以A和B作为name1和name2为例，可以得到如图12结果。

```
INFO:tensorflow:Restoring parameters from ./model/ATT_GRU_model-900
reading word embedding data...
reading relation to id

实体1: A
实体2: B
A和她的丈夫B前日一起去英国旅行了。
关系是:
No.1: 夫妻, Probability is 0.996217
No.2: 父母, Probability is 0.00193673
No.3: 兄弟姐妹, Probability is 0.00128172

实体1: A
实体2: B
A和她的高中同学B两个人前日一起去英国旅行。
关系是:
No.1: 好友, Probability is 0.526823
No.2: 兄弟姐妹, Probability is 0.177491
No.3: 夫妻, Probability is 0.132977

实体1: A
实体2: B
B命令A在周末前完成这份代码。
关系是:
No.1: 上下级, Probability is 0.965674
No.2: 亲戚, Probability is 0.0185355
No.3: 父母, Probability is 0.00953698

实体1: A
实体2: B
B非常疼爱他的孙女A小朋友。
关系是:
No.1: 祖孙, Probability is 0.785542
No.2: 好友, Probability is 0.0829895
No.3: 同门, Probability is 0.0728216

实体1: A
实体2: B
```

图13.手动预测结果

具体数据如下：

```
INFO:tensorflow:Restoring parameters from
./model/ATT_GRU_model-9000
reading word embedding data...
reading relation to id

实体1: A
实体2: B
A和她的丈夫B前日一起去英国旅行了。
关系是:
No.1: 夫妻, Probability is 0.996217
No.2: 父母, Probability is 0.00193673
No.3: 兄弟姐妹, Probability is 0.00128172

实体1: A
实体2: B
A和她的高中同学B两个人前日一起去英国旅行。
关系是:
No.1: 好友, Probability is 0.526823
No.2: 兄弟姐妹, Probability is 0.177491
No.3: 夫妻, Probability is 0.132977

实体1: A
实体2: B
B命令A在周末前完成这份代码。
关系是:
No.1: 上下级, Probability is 0.965674
No.2: 亲戚, Probability is 0.0185355
No.3: 父母, Probability is 0.00953698

实体1: A
实体2: B
B非常疼爱他的孙女A小朋友。
关系是:
No.1: 祖孙, Probability is 0.785542
No.2: 好友, Probability is 0.0829895
No.3: 同门, Probability is 0.0728216

实体1: A
实体2: B
谈起曾经一起求学的日子，B非常怀念他的师妹A。
关系是:
No.1: 师生, Probability is 0.735982
```

```
No.2: 同门, Probability is 0.159495
No.3: 兄弟姐妹, Probability is 0.0440367

实体1: A
实体2: B
B对于他的学生A做出的成果非常骄傲!
关系是:
No.1: 师生, Probability is 0.994964
No.2: 父母, Probability is 0.00460191
No.3: 夫妻, Probability is 0.000108601

实体1: A
实体2: B
B和A是从小一起长大的好哥们
关系是:
No.1: 兄弟姐妹, Probability is 0.852632
No.2: 亲戚, Probability is 0.0477967
No.3: 好友, Probability is 0.0433101

实体1: A
实体2: B
B的表舅叫A的二妈为大姐
关系是:
No.1: 亲戚, Probability is 0.766272
No.2: 父母, Probability is 0.162108
No.3: 兄弟姐妹, Probability is 0.0623203

实体1: A
实体2: B
这篇论文是B负责编程，A负责写作的。
关系是:
No.1: 合作, Probability is 0.907599
No.2: unknown, Probability is 0.082604
No.3: 上下级, Probability is 0.00730342

实体1: A
实体2: B
B和A为谁是论文的第一作者争得头破血流。
关系是:
No.1: 合作, Probability is 0.819008
No.2: 上下级, Probability is 0.116768
No.3: 师生, Probability is 0.0448312
```