# Memory Networks的阶段总结

目录

一. 原始模型介绍 2

二. 基于MemN2N的垂直领域人机交互系统 4

2.1 代码学习 4

2.2 试验测试及模型改动 5

2.3 交互系统的设计和调试 11

三、基于MemN2N的三元组QA问答和三元组抽取测试 19

四、MemN2N特点总结 23

## 一. 原始模型介绍

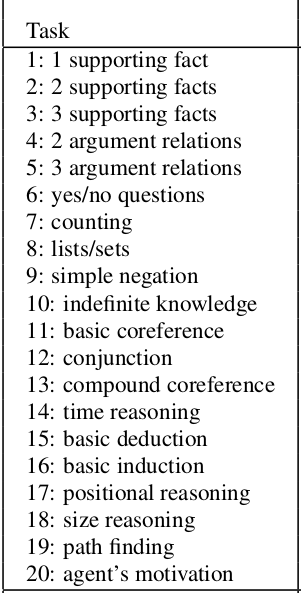
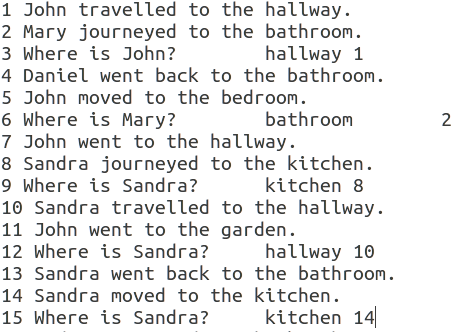
J. Weston, S. Chopra, and A. Bordes. Memory networks. In International Conference on

Learning Representations (ICLR), 2015.

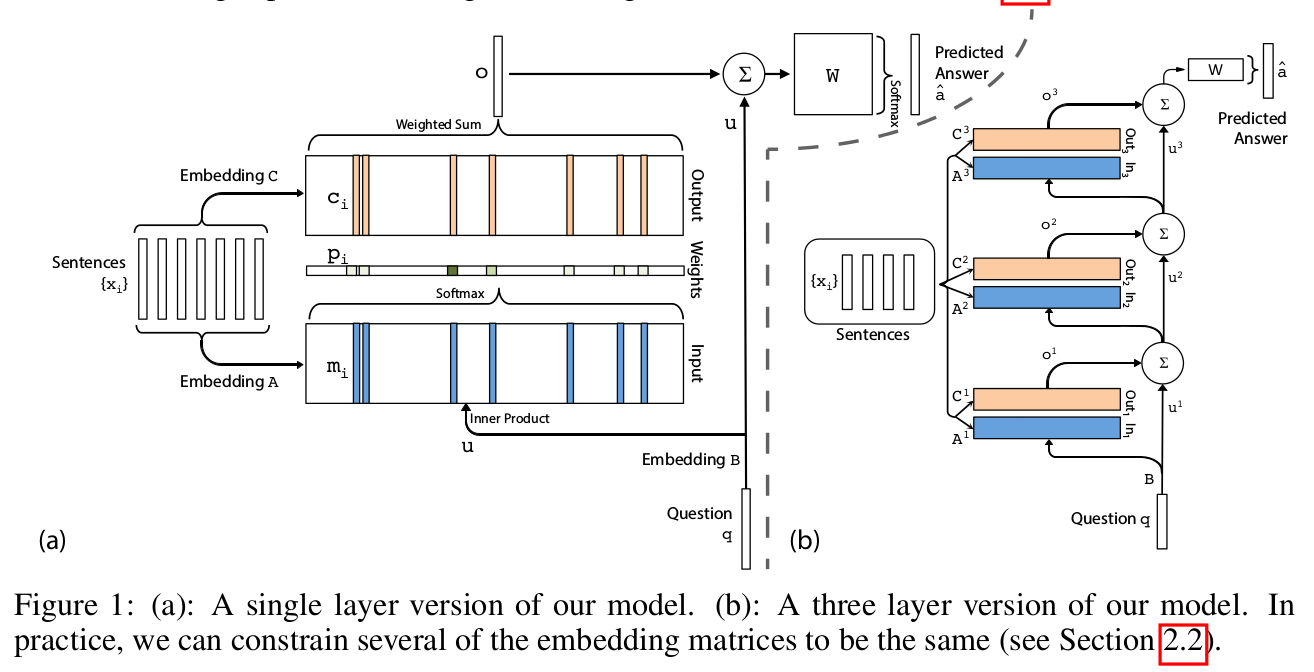
Memory networks是FB的Weston团队在2014年提出的一个具有记忆单元和效应的模型。

Memory networks reason with inference components combined with a long-term memory component; they learn how to use these jointly.The long-term memory can be read and written to, with the goal of using it for prediction.

原始的语料集是基于FB的Q20数据集，对应二十种推理任务。



在MemNN的基础上，2015年三月，其团队又发表了改进过后的基于端对端的模型，发表在论文《End-To-End Memory Networks》中，该方法使得在模型训练的过程中需要更少的监督，有利于在更小的数据集上得到相似的结果。



我们采用的模型就是基于端对端的MemNN，源码是由FB官方团队公开，包括Matlab和Lua两个版本，我们选择matlab版本作为我们学习的模型。

## 

## 二. 基于MemN2N的垂直领域人机交互系统

在得到源码之后，我们想要利用其推理和记忆的功能，在我们目标任务上得到好的结果。我们开始按照代码学习、试验测试、应用搭建的流程，逐一展开工作。

### 2.1 代码学习

MemN2N是Facebook首先提出的Memory Network问答系统框架的端对端版本。Memory Network是一种依赖于存储的QA框架，通过训练集的监督信息，对词表中的每一个词进行语义投影。而MemN2N则不需要关于句子相似度的监督信息，而只依赖于问句和答句，从中通过多轮Attention机制来定位回答问题所最需要的句子，并从这些句子的向量表示中生成最终的答案。



### 2.2 试验测试及模型改动

由于原语料全部基于英文，尤其是Q20数据集，我们首先按照其格式，在得到的原始语料的基础上进行修改，得到与之对应的可使用的语料，具体如下。

（1）中文语料的修改及实验

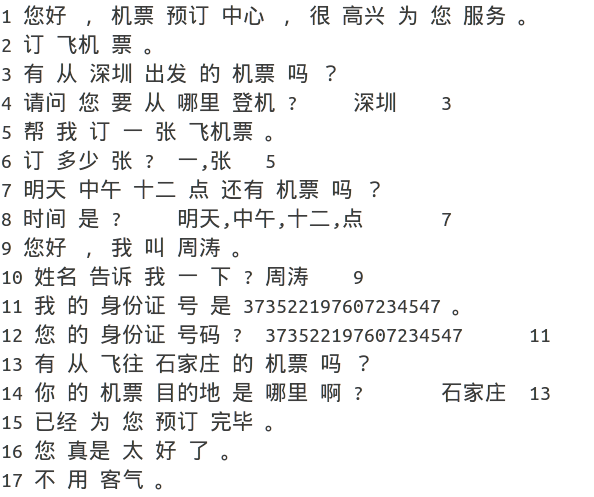
原始语料是一批经过标注的多个垂直领域的问答语料，其形式如下：



每个垂直领域具有相同的语义槽。如上订机票领域内的6个语义槽是：

name、departure、destination 、idnumber 、count、 time .

按照Q20的逻辑，我们修改了语料，使其成为 Story -- Question -- Answer的形式，如下所示：

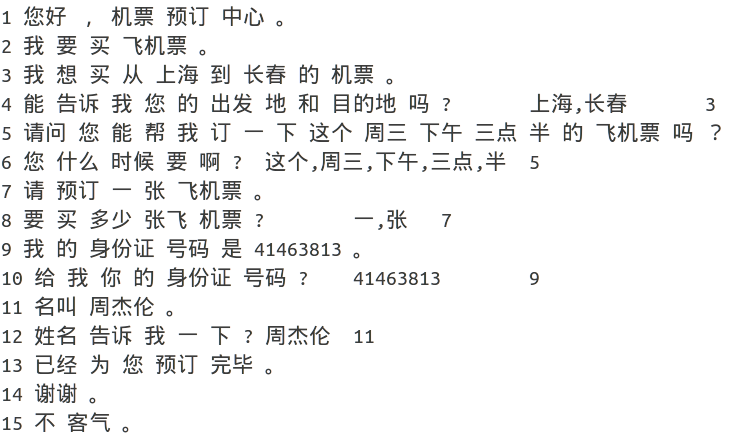


实验结果显示，该中文语料上的single fact问题也能很好的解决。

（2）随机词汇的生成和模型

　　原始语料中，包括Q20和我们的中文垂直领域语料中出现的问题是，词典非常小、词汇比较单一，某些语义槽的选择范围太狭窄，如时间问题，“周三下午三点半”这个回答就占了40%左右的时间槽，例如姓名中几乎全部是“周涛”。我们决定对语料词汇进行扩充，以验证模型的能力。

从名字，身份证号下手，我们将名字的数量进行扩充，身份证号完全随机。得到的丰富的语料如下：



修改后的语料，词典大小扩充到3000以上，直接用模型开始训练得到的结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train error | Valid error | Test Error |
| 10轮后 | 0.0050223 | 0.11979 |  |
| 20轮后 | 0 | 0.09375 |  |
| 40轮后（20轮线性，20轮softmax) | 0.97824 | 0.99479 |  |
| 训练语料规模为1000个完整对话，约5000个问题和答案。  测试语料与训练语料按照1：1规模配置。 | | | |

结论：

在线性规约的先行训练中，经过20轮，大致也能成功学出了目标，但是加入softmax规约之后，完全不能有效预测。

　按照以上的实验结果修改源代码，去掉了softmax归一化的过程，得到的结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train error | Valid error | Test Error |
| 10轮后 | 0.0089286 | 0.19792 |  |
| 20轮后 | 0 | 0.11979 |  |
| 40轮后（线性) | 0 | 0.11458 |  |
| 100轮后（线性） | 0 | 0.11458 | 0.075605 |
| 训练语料规模为1000个完整对话，约5000个问题和答案。  测试语料与训练语料按照1：1规模配置。 | | | |

把“词向量”的维度进一步加大（从20到50）之后，经过５轮线性训练，test valid test 误差几乎都到了0，验证了模型的能力，成为了继续做下一步工作的基础。

（3）未登录词的处理和实验

在源码中，一开始处理语料时，就把train和test的所有词汇都加入了词典。为了按照初步设想的比较鲁棒的交互系统的初衷设计，我们发现目前存在的一个大的问题是未登录词的问题。

例如 “我要从北京到堪培拉”，由于训练中没有堪培拉这个词汇，在最后回答的过程中，永远不可能回答出destination=堪培拉 的正确结果。

经过讨论，我们设计了在字典中引入未登录槽的方式，通过只构建train的字典，用以完成对test 中可能出现的丰富词汇的较好的回答。

具体的实现如图所示：

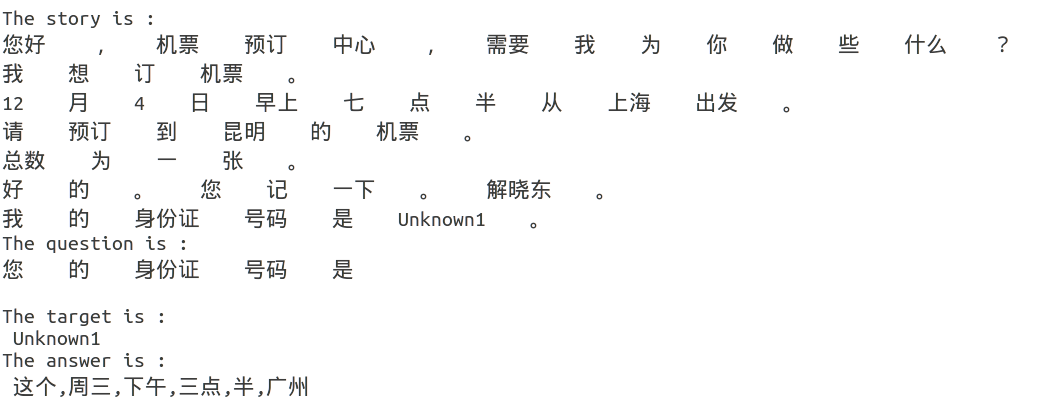


实验结果：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train error | Valid error | Test Error |
| 5轮后 | 0.049287 | 0.054688 |  |
| 10轮后 | 0.010696 | 0.019531 | 0.046212 |
| 训练语料规模为1000个完整对话，约5000个问题和答案。  测试语料与训练语料按照1：1规模配置。 | | | |

实验结果表明，未登录词的解决方式可行。

我们也同时修改了源码，生成了一系列测试集中出现错误的对话内容，其中一则如下：



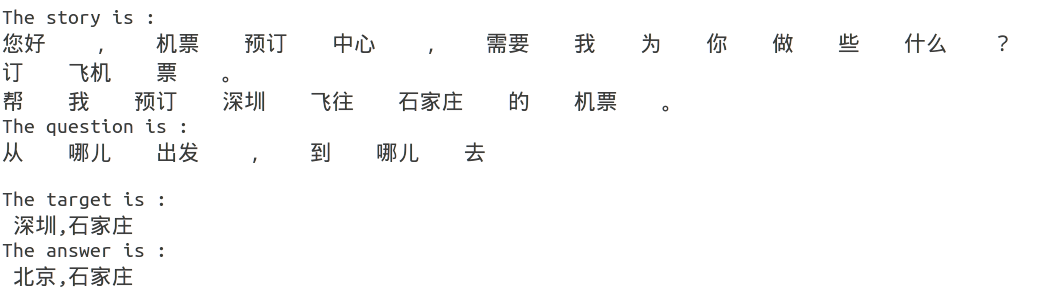
其中Unknown1本来是一个随机的十八位数字，我们将它处理成了一个槽。

根据错误文本来看，出现错误的地方没有规律，几乎是随机的。这说明我们的模型到现在并没有出现比较明显的系统误差无法解决的情况。

（4）回答范围的限定

　　按照FB的设计思路，我们的回答均是在上文的story中出现过的一个词，系统不应该凭空生成某些词汇，另外根据我们垂直领域的抽取语义槽的思路，例如这种情况是我们需要极力避免的：

　　　　我们在这里加入了一个条件，所有回答的答案，必须在该问题对应的Story中出现过，即在最后选取答案词的时候，将范围缩小在story中的词汇，可以保证得到story范围内的最大概率的答案词汇。

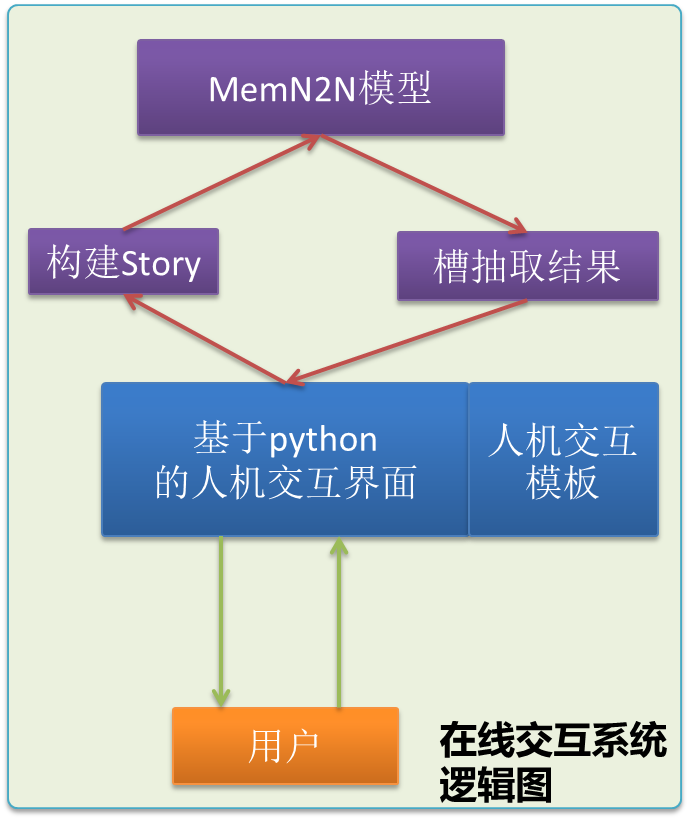


### 2.3 交互系统的设计和调试

为仿真应用场景，我们设计实现了简单的在线交互系统。系统主要模块包括人机交互界面、人机交互模板、MemN2N模型三个部分。

1. 人机交互界面：完成向用户主动提问、获取用户输入、提供MemN2N模型需要的Story格式、获取语义槽抽取结果、判断信息是否收集完全；
2. 人机交互模板：提供greeting、不同语义槽向用户发问的问句模板
3. MemN2N模型：利用离线学习的模型，对在线story进行语义槽抽取。

　　　　 在线系统



具体地，垂直领域问答系统的任务是：系统根据缺失的语义槽提出问句Q，用户给出答案A（自然语言），系统从答句A中抽取答案词语a，使得a能够正确填充Q所对应的语义槽。这个抽取过程可以用MemN2N来实现，方法是用Q和A作为Story，放入MemN2N的存储部分，再生成一个问句Q’，Q’可以设计成是一个单词，就对应语义槽的语义，例如time? Departure?等等，作为MemN2N的问句。MemN2N输出抽取结果作为答句。

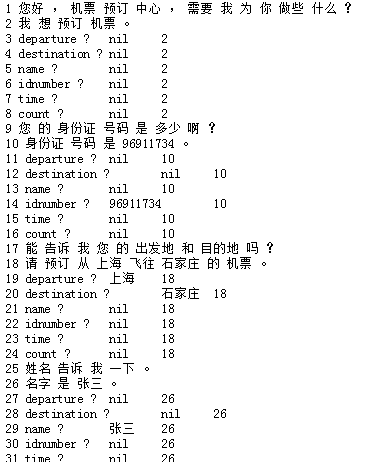
系统的整体流程图如下：



我们在这部分的工作如下：

1. 语料的改动

为了适应语义槽的要求，我们将原来的语料进行改动，在问题-答案对的基础上，扩充出来问题-nil，即没有回答的情况。扩展时候，每个Story大概有50个问题。其具体如下：



1. 交互系统的功能测试及调整

调试、修改交互系统，找出测试过程跑不通/交互效果不好的原因并加以修正。经验上，系统对回答为nill的问题基本不会错，而交互中遇到的问题主要是答案非nill的。如果系统对一个完整对话的30~40个问题中的6个答案不是nill的问题回答错1~2个，测试集看到的错误率仍然只有0.03，但实际交互效果就很不理想。如果测试集错误率降到0.01以下，交互时表现比较稳定。为了降低系统错误率，使用了三个方法：完全使用Linear Start Training，去掉最后的softmax指数层。增大词投影维度至50维。增大训练轮数至30轮。

1. 多样性的改进

针对答句句式多样性问题和问句敏感问题，讨论提出多样化答句、固定问句的解决方案，并参与制作新语料模板。垂直领域问答的场景就是问句可以完全固定，而用户的答句句式却难以确定，所以在训练语料中直接将问句固定，同时加入更丰富的答句句式，能够提高系统的性能。

具体地，我们针对不同的语义槽，依次设计了问句、答句的多句式模版，其中一部分如下：

|  |  |  |
| --- | --- | --- |
|  | **问句** | **答句** |
| **姓名** | 1. **您好，请问您的姓名是？** 2. **请问您的姓名是？** 3. **请告诉我您的姓名。** 4. **请您告诉我您的名字。** 5. **请问您要购买机票的用户姓名是？** | 1. **[槽\_name]。** 2. **叫[槽\_name]。** 3. **姓名是[槽\_name]。** 4. **我是[槽\_name]。** 5. **您好，我叫[槽\_name]。** |
| **身份证号** | 1. **请问您的身份证号码是多少？** 2. **请问您的身份证号是？** 3. **请告诉我您的身份证号码？** 4. **您好，我需要输入您的身份证号码。** 5. **请您告诉我您的身份证号码。** | 1. **[槽\_idnumber]。** 2. **我的身份证号是[槽\_idnumber]。** 3. **是[槽\_idnumber]。** 4. **身份证号码是[槽\_idnumber]。** 5. **我的身份证是[槽\_idnumber]。** |
| **电话号码** | 1. **请问您的电话号码是多少？** 2. **请告诉我您的联系方式？** 3. **请问您的联系电话是？** 4. **请告诉我您的联系电话。** 5. **您好，我需要输入您的联系电话。** | 1. **[槽\_phone]。** 2. **我的电话号码是[槽\_phone]。** 3. **联系电话是[槽\_phone]。** 4. **您好，我的联系电话是[槽\_phone]。** 5. **我的联系电话是[槽\_phone]。** |
| **出发地** | 1. **请问您从哪里起飞？** 2. **请问您从哪里出行？** 3. **请问您从哪个城市出行？** 4. **请告诉我您的起飞城市。** 5. **请您告诉我，您从哪个城市出行？** | 1. **[槽\_departure]。** 2. **从[槽\_departure]起飞。** 3. **在[槽\_departure]登机。** 4. **请帮我预订从[槽\_departure]起飞的机票。** 5. **预订从[槽\_departure]出发的机票。** |
| **目的地** | 1. **请问您的目的地是哪里？** 2. **请问您要预定的目的地是？** 3. **请问您要预定飞往哪个城市的机票？** 4. **请问您此次出行的目的地是？** 5. **请问您此次前往哪个城市？** | 1. **[槽\_destination]。** 2. **去[槽\_destination]。** 3. **目的地是[槽\_destination]。** 4. **飞往[槽\_destination]。** 5. **是[槽\_destination]。** |
| **张数** | 1. **请问您需要几张机票？** 2. **请问您需要预定几张飞机票？** 3. **您需要预定几张机票？** 4. **需要预定几张飞机票？** 5. **请问您总共需要预定几张机票？** | 1. **[槽\_count]。** 2. **[槽\_count]张就可以了。** 3. **帮我预定[槽\_count]张。** 4. **[槽\_count]张。** 5. **[槽\_count]张机票。** |
| **出发时间** | 1. **请问您需要购买什么时间的机票？** 2. **您需要预定哪个时间的机票？** 3. **请告诉我您预计的出行时间是？** 4. **请问您的出行时间是？** 5. **请告知您的出行时间。** | 1. **[槽\_time]。** 2. **时间是[槽\_time]。** 3. **帮我预订[槽\_time]的机票。** 4. **出行时间是[槽\_time]。** 5. **订[槽\_time]的机票。** |

在丰富了的句式语料模版上，我们对语料进行了再一次的扩充和调整，具体如下：

|  |  |  |  |
| --- | --- | --- | --- |
|  | Question | Answer | Remarks |
| Name | 40 | 32 | 姓：51个  名：63个 |
|  | | | |
| Count | 37 | 21 | 50%是从（‘一张’到‘十张’）  50%是随机（ xxxxx张） |
| Departure | 15 | 29 | 50%从113个真实城市名  50%是“地方代号-xxxxx” |
| Destination | 22 | 33 | 同上 |
|  | | | |
| Idnumber | 32 | 36 | 18位随机数 |
| time | 18 | 37 | '2016年02月15日23点14分' 这种形式，从现在时刻起的100天内完全随机形式 |
| phone | 37 | 36 | 11位随机数 |

总体上，每个语义槽的问题顺序也是随机的。

在新语料上进行训练，测试集错误率见下表。

|  |  |  |  |
| --- | --- | --- | --- |
| 改造前语料，dim = 20 | 改造后语料，dim = 20 | 改造后语料，dim = 50 | 改造后语料，dim = 50，固定问句 |
| 0.0076 | 0.034 | 0.0039 | 0.0026 |

1. 语义的预处理及后处理

编写针对身份证、电话以及机票数量的后处理模块。MemN2N能够从用户答句中抽取与答案相关的部分，但这部分可能和最终答案之间有格式上的差距。为了使系统实用性更好，可以针对每个特定任务设计后处理过滤模块。

预处理阶段：

1. 前处理主要是利用正则表达式来合并因使用分词工具分开的时间词。

|  |  |  |
| --- | --- | --- |
| 用户输入 | 分词后结果 | 前处理后结果 |
| 今天下午五点半出发 | 今天 下午 五点半 出发 | 今天下午五点半 出发 |

1. 解决方法：

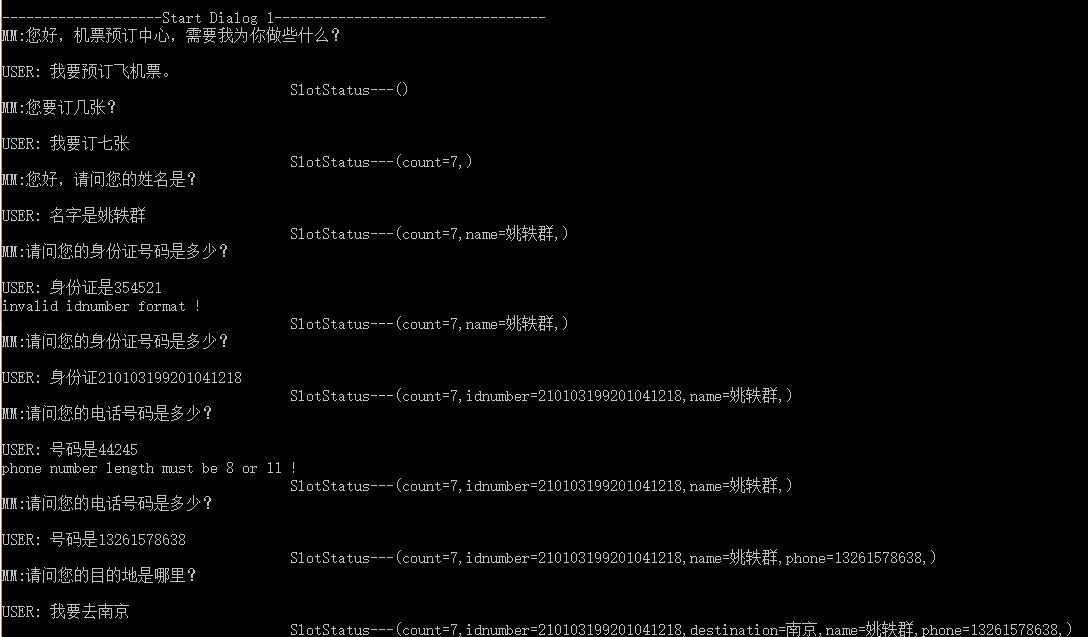
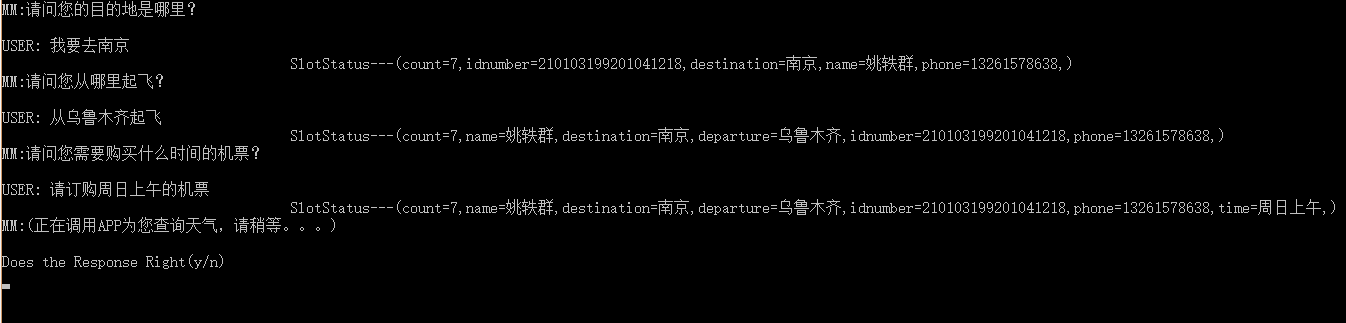
利用正则表达式检测分词后结果中出现的所有时间词(Date/Time/Duration)，比如今天/下午/五点半，然后再将这些词顺序拼接。

|  |
| --- |
| 相关正则表达式：   1. (\d{1,4}|[一二三四五六七八九零]{1,4}|今|去|前|明|来)年 Date 2. (\d{1,2}|[一二三四五六七八九十]{1,2}|上个?|下个?|这个?|本)月份?([月底|[上中下]旬])? Date 3. ((\d{1,2}|[一二三四五六七八九十]{1,2})[日|号]) Date 4. (上个?|下个?|这个?|本)?(周|星期){1,2}[一二三四五六日几] Date 5. (上个?|下个?|这个?|本)?[季度|世纪]([初|中|末]期?) Date 6. [今明昨后]天 Date 7. 上午|下午|中午|晚上|傍晚|凌晨|早[上晨]|午夜|半夜|夜里 Time 8. (\d|[零一二两三四五六七八九十])+[点时] Time 9. ((\d|[一二两三四五六七八九十])+分)钟? Time 10. (([13]|[一三])+刻)钟? Time 11. [早晚]((\d|[零一二两三四五六七八九十])+[点时]) Time 12. ((\d|[一二两三四五六七八九十])+个?)半?(小时|分钟|秒) Duration 13. 半(小时|分钟|秒) Duration 14. ((\d|[一二两三四五六七八九十])+|半)(天|个月|年) Duration 15. ((\d|[一二两三四五六七八九十])+)(天|个月|年)半 Duration |

目前对于订机票领域做了如下后处理：

|  |  |
| --- | --- |
| 身份证号码 | 使用18位正则表达式进行匹配过滤。 |
| 电话号码 | 对数字的位数进行检验，只允许8或11位的作为合理答案。 |
| 张数 | 去除量词，限定数值范围，并将中文大写数字转换为阿拉伯数字。 |

目前效果如下：



## 

## 三、基于MemN2N的三元组QA问答和三元组抽取测试

1. **利用Memery Networks测试“定餐馆”数据的语义槽抽取**

1）语义槽有三个，时间、人数、姓名

2）语料：训练-20000条 测试7764条

|  |  |
| --- | --- |
|  |  |
| Diner\_A | 名是王雪。 这个周三下午三点 6 0 |
| Diner\_Q | 好的，王雪，为您预订了这个周三下午三点6人桌。 这个周三下午三点 6 王雪 |
| Diner\_QA | 1 名是 王雪 。  2 名是 王雪 。  3 name ? 王雪 2 |

3）测试结果： 训练集/测试集 均为100%

1. **利用Memory Networks 测试三元组QA**
2. 问题描述：已知实体的若干属性和属性值 (A,r1,B1) (A, r2, B2)… 预测(A, rk, ?)

目前数据情况

|  |  |  |
| --- | --- | --- |
| **DataSetA** | **训练集：401964条**  **测试集：39726条** | **包含关系15种：**  去世年月/民族/毕业院校/所在国家/出生地/所属市/建造时间/所属国/籍贯/所属宗教/出生年月/主要民族/国籍/职业/政党 |
| **DataSetB** | **训练集：671719条**  **测试集：39726条** | 去世年月/民族/毕业院校/所在国家/出生地/所属市/建造时间/所属国/籍贯/所属宗教/出生年月/主要民族/国籍/职业/政党 |

1. 数据转换 (三元组转换成story+Q)

|  |  |  |
| --- | --- | --- |
|  | 三元组 | Story+Q |
| Train  /Test | 一凡<->毕业院校<->中共中央党校  一凡<->民族<->汉族  一凡<->出生年月<->1949年4月8日  一凡<->国籍<->中国  一凡<->政党<->中国共产党 | 1 一凡 出生年月 1949年4月8日  2 一凡 国籍 中国  3 一凡 毕业院校 中共中央党校  4 一凡 民族 汉族  5 一凡 政党 ? 中国共产党 1 |

|  |  |  |
| --- | --- | --- |
| relation | Train/Test size | Precision |
| 籍贯 | 2851/1664 | 87/19/15 |
| 政党 | 10090/1152 | 99.8/93/93.2 |
| 民族 | 5728/4480 | 99.5/91.2/90.7 |
| 所在国家 | 196/32 | 95/NaN/96.8 |
| 出生地 | 8588/1088 | 81/19/23 |

|  |  |
| --- | --- |
| 出生地 | 籍贯 |
| -----------------------------------------------  3th Dialog, story is:  1 jj·雷迪克 国籍 美国  2 jj·雷迪克 职业 职业球员  query is: jj·雷迪克 出生地 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:田纳西州  \*\*\*\*\*\*\*\*\*\*WRONG:洛杉矶  -----------------------------------------------  4th Dialog, story is:  1 macaulay\_culkin 出生年月 1980年8月26日  2 macaulay\_culkin 国籍 美国  query is: macaulay\_culkin 出生地 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:纽约  \*\*\*\*\*\*\*\*\*\*WRONG:印第安纳州  -----------------------------------------------  5th Dialog, story is:  1 m·克拉吉塞克 出生年月 1989年1月9日  2 m·克拉吉塞克 国籍 荷兰  query is: m·克拉吉塞克 出生地 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:荷兰  \*\*\*\*\*\*\*\*\*\*WRONG:不列颠哥伦比亚省  -----------------------------------------------  6th Dialog, story is:  1 丁学飞 国籍 中国  2 丁学飞 民族 汉族  query is: 丁学飞 出生地 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:泾县  \*\*\*\*\*\*\*\*\*\*WRONG:临川  -----------------------------------------------  7th Dialog, story is:  1 丁家立 出生年月 1857年  2 丁家立 去世年月 1930年  3 丁家立 国籍 美国  query is: 丁家立 出生地 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:波士顿  \*\*\*\*\*\*\*\*\*\*WRONG:加利福尼亚州 | -----------------------------------------------  1th Dialog, story is:  1 hans\_wegner 出生年月 1914年  2 hans\_wegner 去世年月 2007年  3 hans\_wegner 国籍 丹麦  query is: hans\_wegner 籍贯 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:欧登塞  \*\*\*\*\*\*\*\*\*\*WRONG:丹麦  -----------------------------------------------  5th Dialog, story is:  1 thom 出生地 阿伯丁  2 thom 出生年月 1968年10月7日  3 thom 国籍 英国  query is: thom 籍贯 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:阿伯丁  \*\*\*\*\*\*\*\*\*\*WRONG:萨里郡  -----------------------------------------------  7th Dialog, story is:  1 丁则民 国籍 中国  2 丁则民 民族 汉族  query is: 丁则民 籍贯 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:北京  \*\*\*\*\*\*\*\*\*\*WRONG:新化县  -----------------------------------------------  8th Dialog, story is:  1 丁志超 出生年月 1941年  2 丁志超 国籍 中国  3 丁志超 民族 汉族  query is: 丁志超 籍贯 ?  ANSWER is:  \*\*\*\*\*\*\*\*\*\*RIGHT:晋江市  \*\*\*\*\*\*\*\*\*\*WRONG:肃宁县 |

1. **利用Memory Networks做句子抽三元组**
2. 语料转换

|  |  |
| --- | --- |
| 原始语料 | Story+Q |
| 陈廷敬 生 在 泽州 。  ->陈廷敬 人物\_出生地 泽州 | 1 陈廷敬 生 在 泽州 。  2 陈廷敬 生 在 泽州 。  3 subj ? 陈廷敬 1  4 relation ? 人物\_出生地 1  5 obj ? 泽州 1 |

1. 测试结果

|  |  |  |
| --- | --- | --- |
| relation | Train/Test size | Precision |
| 人物\_出生地 | 2799/336 | 94.2/99.6/98.4 |
| 人物\_国籍 | 1893/621 | 99.6/99.1/99.7 |
| 人物\_职业 | 3069/336 | 75/79/78 |
| 人物\_民族 | 6093/672 | 99.9/99.4/99.2 |
| 动物\_目 | 7516/608 | 94/99.6/99.4 |

## 四、MemN2N特点总结

|  |  |
| --- | --- |
| MemN2N优点 | 训练参数少，训练速度快，逻辑性强 |
| MemN2N缺点 | 输出通常为结构化的词汇， |
| MemN2N使用场合 | 垂直领域、QA、Language Model |