多人会话文本中的角色识别

# 研究现状

1. 角色识别的**目的**：角色识别旨在通过多人对话将**mention（例如她，妈妈）链接到**其指代的全局实体，这些全局实体又指向了**特定角色（发言者）**。
2. 角色识别的**必要性**：由于多人对话的性质，几个发言者轮流完成一个上下文，因此角色识别是完成高端NLP任务（例如摘要，**问答**，机器翻译等）的**关键步骤**。
3. 角色识别的**难点**：对话为口语形式，文本中存在俚语、语法错误等，给角色识别带来更多挑战。（1）普通共指消解是在一篇文本或一个句子中找出名词对应的实体，而角色识别是在多人对话文本中找出mentions对应的角色，mentions与其对应的角色通常不在同一个发言文本中，这就是**交叉文档上的共指消解**。（2）传统的实体连接可以利用实体知识库中的知识，但角色识别**没有关于mentions的先验知识**。因此在用向量表达实体时会有一定困难。
4. 相关研究：

SemEval2018-Task4的任务为角色识别，网址为：

<https://competitions.codalab.org/competitions/17310#learn_the_details>。

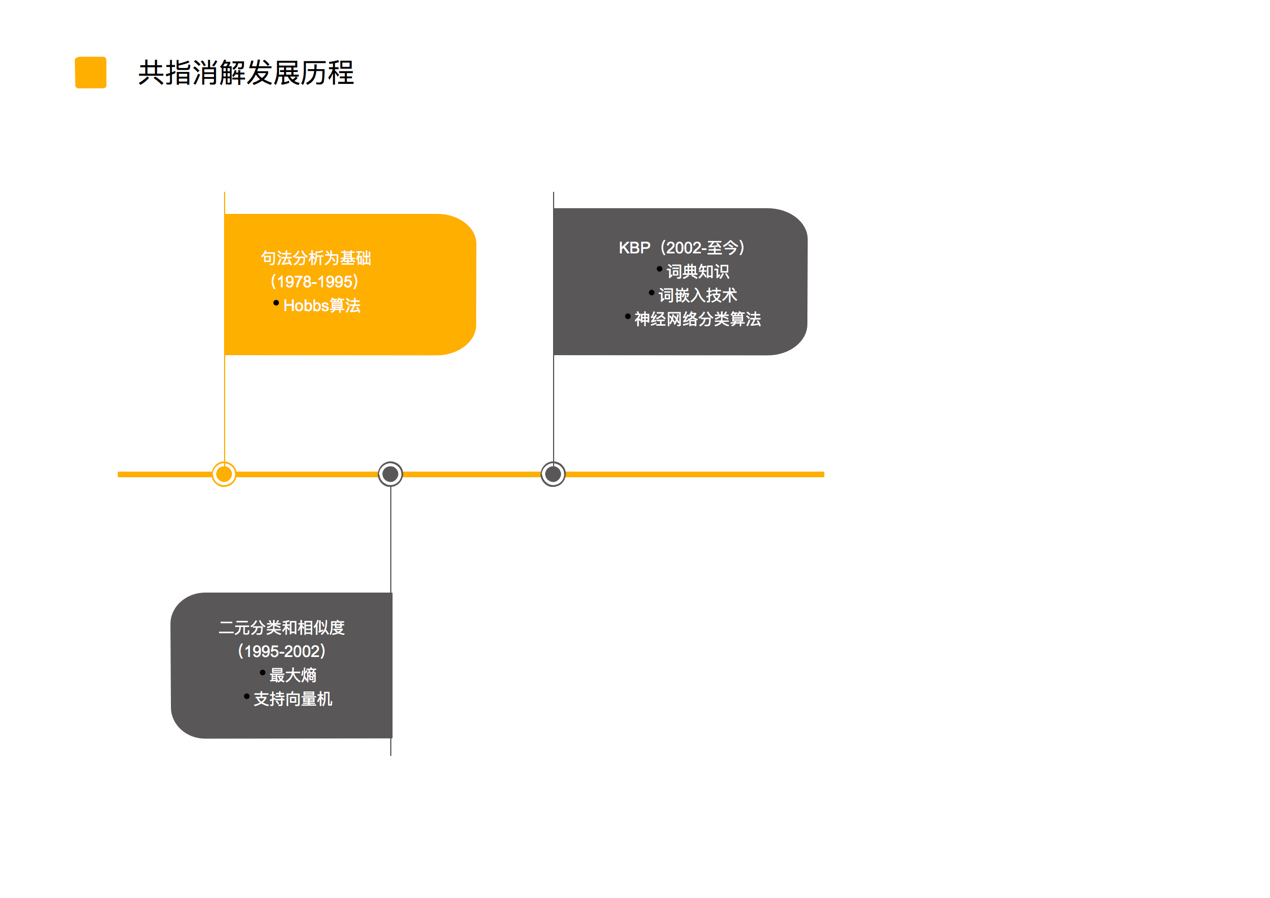
Emory NLP Group利用其文本挖掘的研究成果完成角色识别任务，网址为：

<https://github.com/emorynlp/character-identification>。

1. 角色识别应用：（1）问答系统是一种特殊的对话系统，参与对话的人数一般为2人，其中不免也会有一些被提及的人或物，因此识别这些人或物所对应的真实实体对于机器理解文本进而给出合理的回答有重要作用。（2）通过角色识别可以让机器了解关于独立个体的更精细的信息，因此角色识别系统为对话的推断理解提供了帮助。（3）智能助理也需要根据用户的输入信息给出相应的响应，因此系统也需要明确用户输入信息中的mentions所指代的真实实体，而目前的大多数智能助理并没有角色识别功能，只能根据字面义做出相应动作。
2. **群聊语料上的角色识别**：指在微信或QQ群聊语料上识别出“@昵称”所指代的特定发言者。与SemEval2018-Task4在剧本语料上的角色识别不同，群聊语料上很少出现能与发言者具有共指关系的实体（比如，发言者的人名，称谓等），因此在群聊语料上的角色识别更加困难。

# 共指消解与角色识别

## 共指消解的历程



## 共指消解与角色识别的关系

照应语（mentions）的定义：文本中所有人名、代词等名词性实体。

* 共指（coreference）的定义：共指指的是两个（或多个）mentions指向真实世界中的同一实体。
* **共指消解（coreference-resolution）**的定义：共指消解就是将指向同一实体的不同mentions连接到同一实体上。
* 实体连接（entity linking）的定义：实体连接可以简单理解为一个分类问题，即给定两个实体和两个实体共同出现的句子文本，预测两个实体之间的关系。
* **角色识别（character identification）**的定义：在一段多人对话中找出被提及的人（人名、人称代词等）所对应角色（说话人）的过程就是角色识别。
* **角色识别与实体连接的关系**：实体连接的目的是预测两个实体之间的某种关系，角色识别的目的是判别代词与其指代的角色之间是否具有指代关系，也可以表述为给定代词预测其指代的某一种或某几种（复数代词）角色。因此，可以说角色识别任务是实体连接任务的子任务。
* **角色识别与共指消解的关系：**两者的共同点是：它们都会对提及的人（mentions）进行分组并标注出每组对应的真实实体。区别在于：（1）角色识别中的共指消解系统需要预测每个mentions组对应的角色（2）角色识别中的共指消解系统处理的是多人参与的对话文本，它是在多个文本之间进行共指消解的。可以认为共指消解是角色识别的一个预处理步骤，利用共指消解找出联系紧密的角色-mentions组。

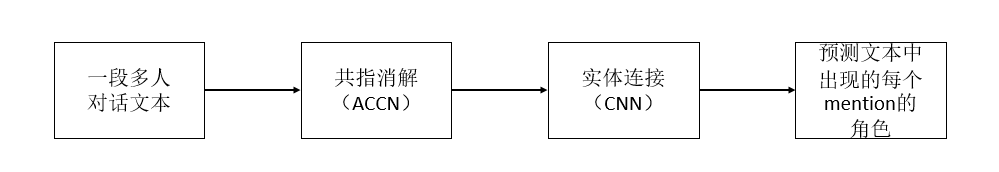
# Emory角色识别系统

## 语料构建



## 系统结构

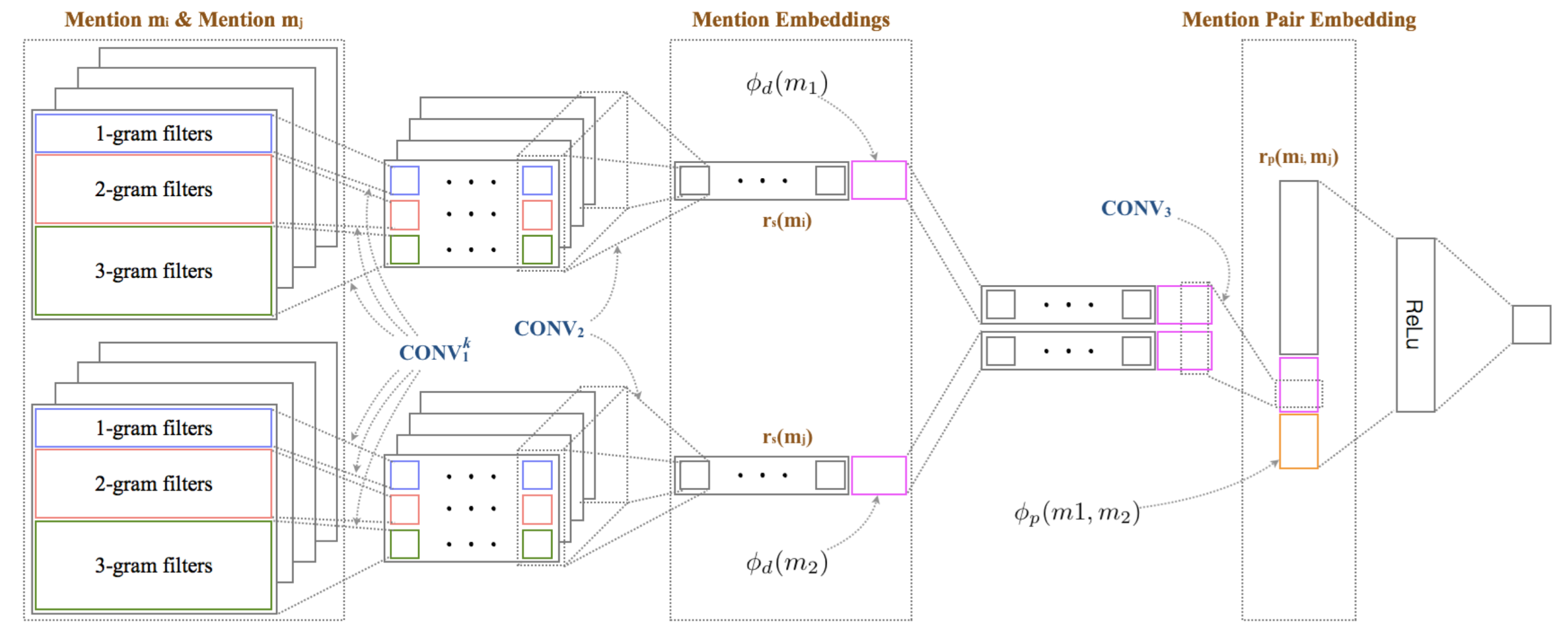
### 整体结构



角色识别系统由两个子系统构成。第1个子系统为共指消解系统。共指消解系统的核心为凝聚卷积神经网络（Agglomerative Convlution Neural Network, ACCN），ACCN结合预训练词向量和语言学特征对mentions、mentions-pair进行嵌入表达，完成共指消解任务。第2个子系统为实体连接子系统。该系统借助共指消解系统从mentions簇中抽取丰富的特征实现角色识别，因此角色识别的效果强烈依赖共指消解的质量。

### 共指消解模型（ACCN模型）

**1. ACCN模型**



**第1部分 计算mention嵌入**

其中，为mention特征模板，抽取mention上下文特征；

为离散特征模板，抽取mention内部特征和说话者特征；

为1号卷积层，对各离散特征向量进行卷积操作；

为2号卷积层，对1号卷积层输出进行卷积操作，||表示concatenate；

2号卷积层的输出作为mention的嵌入向量。

**第2部分 计算mention-pair嵌入**

表示两个mention的嵌入表示的堆叠；

表示3号卷积层，对堆叠后的mention嵌入进行卷积操作；

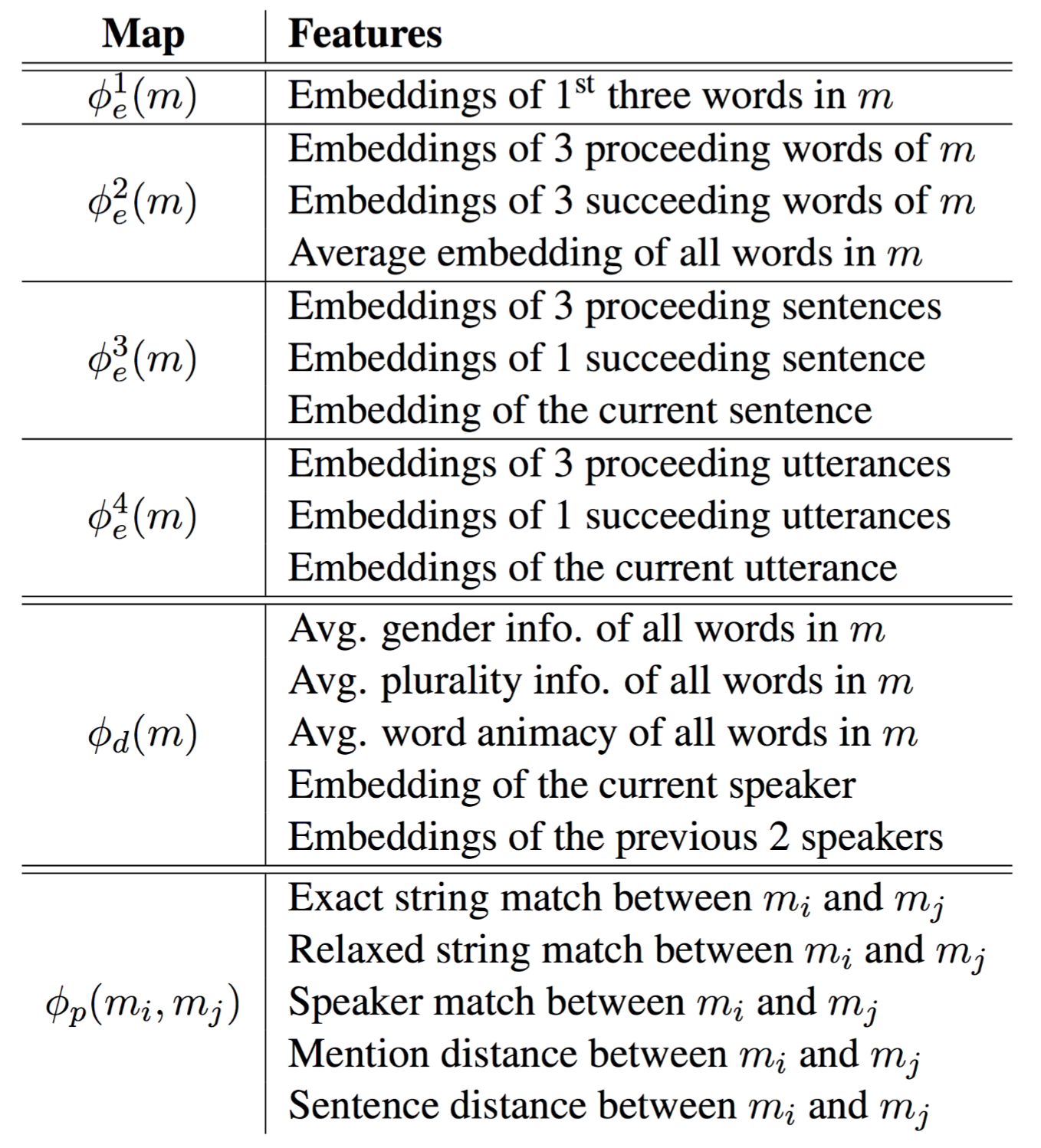
为mention-pair的特征模板，抽取两个mention字面值匹配度，说话者匹配度，距离，所在句子距离；||表示concatenate；

将3号卷积层输出与mention-pair特征向量拼接后作为mention-pair的特征向量。

**第3部分 预测构成共指关系可能性并构建coreference resolution cluster**

Single-Pass思想构建coreference resolution cluster。

**特征模板**



**2. 评测**

**（1）B3**

给定一个mention记为mi，系统给出的结果中包含mi的共指链集合为Smi，标准中包含mi的共指链为Gmi，则mi的准确率和召回率分别为

整体准确率为对所有mention准确率的算术平均值，整体召回率为对所有mention召回率的算术平均值。

**（2）Ceaf**

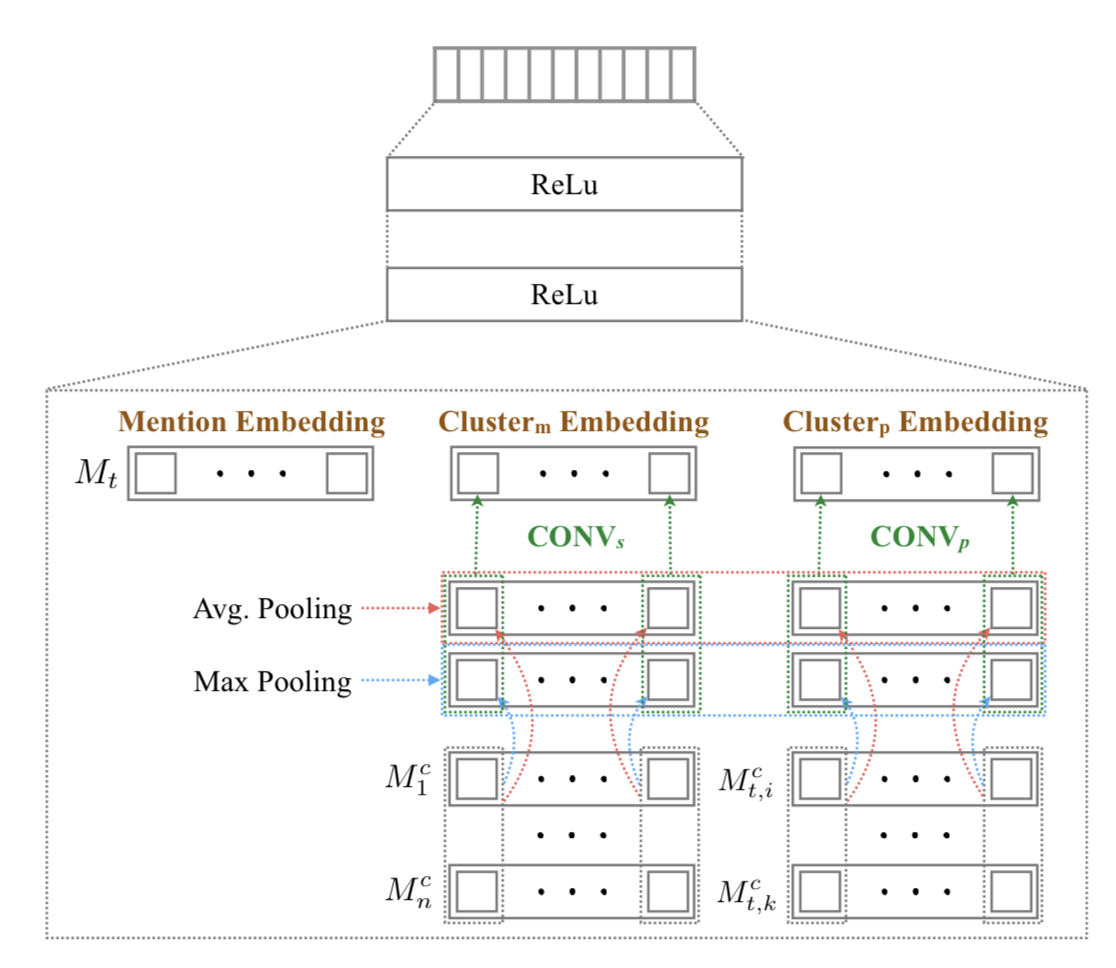
设实体Si表示一个系统给出的实体，Gj表示标准的一个实体，定义

其中表示引用实体Si的mention集合与引用实体Gj的mention集合的交集大小。则系统准确率和召回率为

### 实体连接模型（CNN模型）

**1. CNN模型**

语料经过共指消解系统输出后，得到共指链（簇），得到每个mention的嵌入向量，在实体关系预测过程中，会再次利用CNN神经网络得到每个簇的嵌入向量，结合mention向量送入输出层得到一个mention对应各种角色标记的概率。如下图所示：



图中最上边的两个隐藏层构成一个多类别分类器，下边的部分是构建分类器输入的过程。显然对于一个mention，该模型的下边部分会生成3个向量分别是mention向量，包含mention的簇向量Rs，mention与簇中每个mention构成的mention-pair向量Rs，其中Rp和Rs向量需要各自经过1个卷积和2个池化层得到。

**2. 评测**

（1）用神经网络模型自带的metric度量方法计算准确率。

（2）作者编写的针对每个角色预测的准确率、召回率、F值的评测方法。

（3）作者编写的针对每个角色预测的宏平均准确率、宏平均召回率、宏平均F值的评测方法。

## 中文角色识别系统

### Emory原版系统转换为中文系统

* **语料的修改**

（1）借助翻译API将语料中tokenize后的英文单词序列翻译为中文单词序列。

（2）借助翻译API将语料中transcript后的英文句子翻译为中文。

（3）去掉依存句法标注：将dependency\_tags赋值为null，将dependency\_heads赋值为null。

（4）去掉tokens\_with\_note标注。

* **词典资源的修改**

（1）将英文词向量模型文件替换为中文词向量模型文件。

（2）去掉加载的animate词典，inanimate词典，gender词典，自定义词典。

### Emory原版系统效果

英文的实体连接子系统的效果并没有像论文中所叙述的那样那么理想，说明利用mention向量+mentions簇向量经过卷积层得到的向量来预测mention角色的方法并不理想。

### 中文系统效果

转换为中文的共指消解系统Ceafe法计算的准确率和Blanc法计算的准确率下降了50%，具体原因需要分析。

## 结论

1. Emory算法的准确率过低，无法应用于工业实践。
2. 剧本语料上的共指消解任务对多人对话文本上的角色识别任务没有帮助。

# 4. 基于问答匹配的多人对话文本上的角色识别

## 疑问句分类及识别

### 疑问句分类与识别

疑问句的作用是提出问题。在书面上，疑问句末尾都用问号。疑问句内部又可细分为四小类。

1. 是非问句。

他们在上课？

他并不知道？

今年是建校三十周年？

你叫他把那桌子搬到外面去？

不准抽烟？

显然，这里如果没有句子最后的“？”，我们无法判断该句是疑问句还是陈述句还是祈使句，回答时，只需给出是、不是两种结果。**实践中不匹配其他三种疑问句的正则，又以问号结束，则判定为是非问句。**

1. 特指问句。

刚才谁来了？

你想喝点什么？

他准备去哪儿旅行？

这大衣多少钱？

她最近身体怎么样？

显然，这类句子必须含有疑问代词比如 “谁”、“什么”、“哪儿”、“多少”、“怎么样”，回答时需要给出具体答案。**实践中通过疑问代词可以很容易识别特指问句，但是我们很难找到判定问答匹配的规则。识别特指问句的正则如下：**

|  |
| --- |
| 谁[^,，?？]+[?？]  什么[^,，?？]+[?？]  哪儿[^,，?？]+[?？]  多少[^,，?？]+[?？]  怎么样[^,，?？]+[?？] |

1. 选择问句。

今天是星期三还是星期四？

他想去北京还是上海，还是广州？

你想喝可乐，喝咖啡，还是喝红茶？

她个儿高，还是矮，还是不高不矮？

显然，这类疑问句具有句式“X还是Y”，回答时需要选择一种问题中的选项，回答时应该出现问题中的X或Y。**在识别选择问句时，如果仅通过“还是”这个词很容易造成误识别，因此应当注意加强识别规则限制，句子不仅需要出现“还是”同时句子末尾应当有问号。我们用下边的正则来识别。**

|  |
| --- |
| 还是[^,，?？]+[?？] |

1. 反复问句。

你去不去？

他明天来不来？

那衣服贵不贵？

那地方远不远？

你看没看今天的电视新闻？

你去没去他家？

**显然，这类疑问句具有句式“V不V”或“V没V”，回答时应该出现问题中的动词。**

口语中：

你去不？

那地方远不？

你看今天的电视新闻没有？

你去他家没有？

显然，是在句末给出一个否定副词“不”或“没有”，回答时应该出现问题中的动词。

他们在上课吗/吧？

这是谁告诉他的呢？

他究竟去了福州，还是去了广州呢？

他去没去学校呢？

显然，口语中也经常在句末加疑问语气词**“吗”、“吧”、“呢”来增强疑问语气。**

你的笔呢？

你病了呢？

口语中，一种“非是非问句”，句式为“一个名词性词语+呢”，“动词性词语+呢”，显然在回答时，应该出现这些名词性词语和动词性词语。

**显然，可以通过模板“V不V”或“V没V”来识别反复问句，但需要注意V的位置的词语词形必须相同。用下边的正则来识别。**

|  |
| --- |
| (\w)不\1  (\w)没\1 |

### 实验与结论

1. 语料从QQ群聊记录中导出
2. 实验代码为　疑问句识别.py
3. 结果

|  |  |
| --- | --- |
| 发言限制 | 数量 |
| 无限制 | 10706 |
| 含有”@用户名” | 609 |
| 含有”@用户名”且含有”你” | 82 |
| 含有”@用户名”且含有”你”且为疑问句 | 10 |

1. 结论

从上边数据可以看到，含有”@用户名”且含有”你”且为疑问句占含有”@用户名”发言数1.6%，即使后续的问答匹配准确率达到100%，这样的**召回率是否可以忍受有待商榷**。

## 基于bert的问答匹配

Bert是从transformer中衍生出来的预训练语言模型，transformer是谷歌于2017年底提出的seq2seq模型，transformer与LSTM最大的区别是它可对输入的字符序列并行运算，通过位置嵌入表示保留字符之间的顺序信息。这里我们基于<https://github.com/yfyvan/keras-bert-core/blob/master/demo_predict_is_next_sentence.py> 这个公共项目搭建了一个基于bert的问答匹配基线系统。需要注意的是，这个系统是对两个句子是否构成上下文进行预测的，我们的问句与答句当然也构成上下文，因此，我们可以借用这个系统来搭建我们的实验系统。从初步的实验结果来看，效果并不理想。基本上没有预测正确的问答对。

### Badcase举例

**例1：特指疑问句的问答匹配**

user\_name= 地质地球所-王锴 utterance= 你要做什么？[表情] @高能所 em

('非上下文', -0.9999632) pro\_user\_name= 半导体所-窦智强 pro\_utterance= @大连化物所-王悦诚 螳螂啊

('构成上下文', 0.8556638) pro\_user\_name= 大连化物所-王悦诚 pro\_utterance= 螳螂幼崽的话也太小只了吧而且颜色有点像异型

('非上下文', -0.7600244) pro\_user\_name= 东北地理所-胡家晴 pro\_utterance= 有点可爱

('非上下文', -0.9341632) pro\_user\_name= 大连化物所-王悦诚 pro\_utterance= 哈哈哈来来来分你几只

('非上下文', -0.9927073) pro\_user\_name= 东北地理所-胡家晴 pro\_utterance= [图片]

**分析：**

显然上边的疑问句是**特指疑问句**，如果用人来辨识的话，我们会认为“螳螂啊”可能是问题的回答。但是绝不会认为“螳螂幼崽的话也太小只了吧而且颜色有点像异型”这句话是回答。我们的bert系统居然会认为这句是回答。特指问句对于机器来说是比较难以应付的。

**例2：**

user\_name= 数学与系统科学研究院林海鹏 utterance= 你是账号登录还是怎么弄呀？@物构所-叶黄黄

('非上下文', -0.9954126) pro\_user\_name= 物构所-叶黄黄 pro\_utterance= [表情]我也不懂，一登就能上

('非上下文', -0.93734664) pro\_user\_name= 数学与系统科学研究院林海鹏 pro\_utterance= ucas那个WiFi吗？

('非上下文', -0.99766433) pro\_user\_name= 生物物理所谷子堆 pro\_utterance= 我也登录不上，显示当然用户数满了好像

('非上下文', -0.6729023) pro\_user\_name= 物构所-叶黄黄 pro\_utterance= 我连lan

('构成上下文', 0.7105062) pro\_user\_name= 物构所-叶黄黄 pro\_utterance= 有线

**分析：**

显然这是一个**选择问句**。核心动词是“登陆”，bert系统所认为的回答是“有线”简直莫名其妙，显然“ [表情]我也不懂，一登就能上”这句才是回答。

例3：

user\_name= 微生物研究所 刘兰功 utterance= @心理所 刘亚男 你是从心里所拉东西？

('非上下文', -0.9999554) pro\_user\_name= 心理所 刘亚男 pro\_utterance= 不是，我从昌平区[表情]@微生物研究所 刘兰功

('非上下文', -0.71423125) pro\_user\_name= 微生物研究所 刘兰功 pro\_utterance= 哦[表情][表情]

('构成上下文', 0.99481374) pro\_user\_name= 海洋所-王鑫 pro\_utterance= 从怀柔城区到雁栖湖学校有没有知道怎么过去方便的同学

('构成上下文', 0.88865334) pro\_user\_name= 海洋所-王鑫 pro\_utterance= H58遥遥无期

('非上下文', -0.9999881) pro\_user\_name= 心理所 刘亚男 pro\_utterance= @微生物研究所 刘兰功 你想找从心理所的吗

**分析：**

显然这是一个**是非问句**，回答应该有不是，是这样的词语，我们bert系统选出的居然是“H58遥遥无期”，“从怀柔城区到雁栖湖学校有没有知道怎么过去方便的同学”。显然第1句“不是，我从昌平区[表情]@微生物研究所 刘兰功”才是回答。

## 基于公共词数量的问答匹配

陈俊杰在《网络购物环境下的问句答案匹配方法研究》提出了基于公共词序列的问答匹配方法。规则是提问与候选回答公共词序列的数量越多认为候选答案与提问越匹配。

## 基于词共现计算相关度的问答匹配

陈俊杰在《网络购物环境下的问句答案匹配方法研究》提出了利用已有的问答语料统计问与答词语共现矩阵，利用该矩阵计算提问与候选回答的相关度，相关度越大认为候选答案与提问越匹配。

### 训练阶段

统计词项相关度矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | … |  |
|  |  |  | … |  |
|  |  |  | … |  |
| … | … | … | … | … |
|  |  |  | … |  |

其中，,,…表示提问文本集的词汇表，,,…,表示回答文本集的词汇表。表示与共同出现的次数。

### 预测阶段

计算问句答案关联矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | … |  |
|  |  |  | … |  |
|  |  |  | … |  |
| … | … | … | … | … |
|  |  |  | … |  |

其中，为问句中的词语，为答案中的词语，元素值为在训练时统计得到的共现次数。

定义问句中某个词语与答案的相关度为

上式表示行向量的长度，t为答案文本中词语数。

问句与答案的相关度为

上式s表示问句中的词语数。

## 基于最大熵的问答匹配

## 基于神经网络的问答匹配