角色识别系统

# Emory NLP的角色标注系统

## 共指消解、实体关系和角色识别

### 共指消解（coreference resolution）

mentions的定义：文本中所有人名、代词等名词性实体。

co-reference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。

co-resolution的定义：共指消解就是将不同的mentions连接到同一实体上。

### 实体关系（entity linking）

关系抽取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，判别两个实体之间的关系。

### 角色识别（character identification）

在一段对话中找到被提及的人mentions（人名、人称代词等）所对应角色的过程就是角色识别。

### 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）角色识别需要将分组标注为一个具体的参与对话的角色（2）并非所有共指消解系统都可以直接拿来完成角色识别，因为角色识别处理的文本是多人参与的对话，它是在多个文本之间执行共指消解的。

## 角色识别的难点

### 交叉文档共指消解

普通共指消解指的是在一篇文本或一个句子中识别名词对应的实体，名词和实体都位于一篇文本或一个句子，而角色识别中，文本是对话语料，因此将mentions映射到真实角色的过程中涉及交叉文档的实体消解。

### 没有实体知识库

传统的实体关系抽取需要实体知识库的支持，而对于角色识别，我们并没有关于mentions的先验知识。

## 角色识别的应用

### 问答系统、文本摘要系统的基础步骤

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。

### 对智能助理系统、教育系统提出更高的复杂性要求

同问答系统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，只能根据字面义给出相应响应。

### 为阅读理解系统提供帮助

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。

## Emory NLP的角色识别系统

### 角色识别系统语料构建

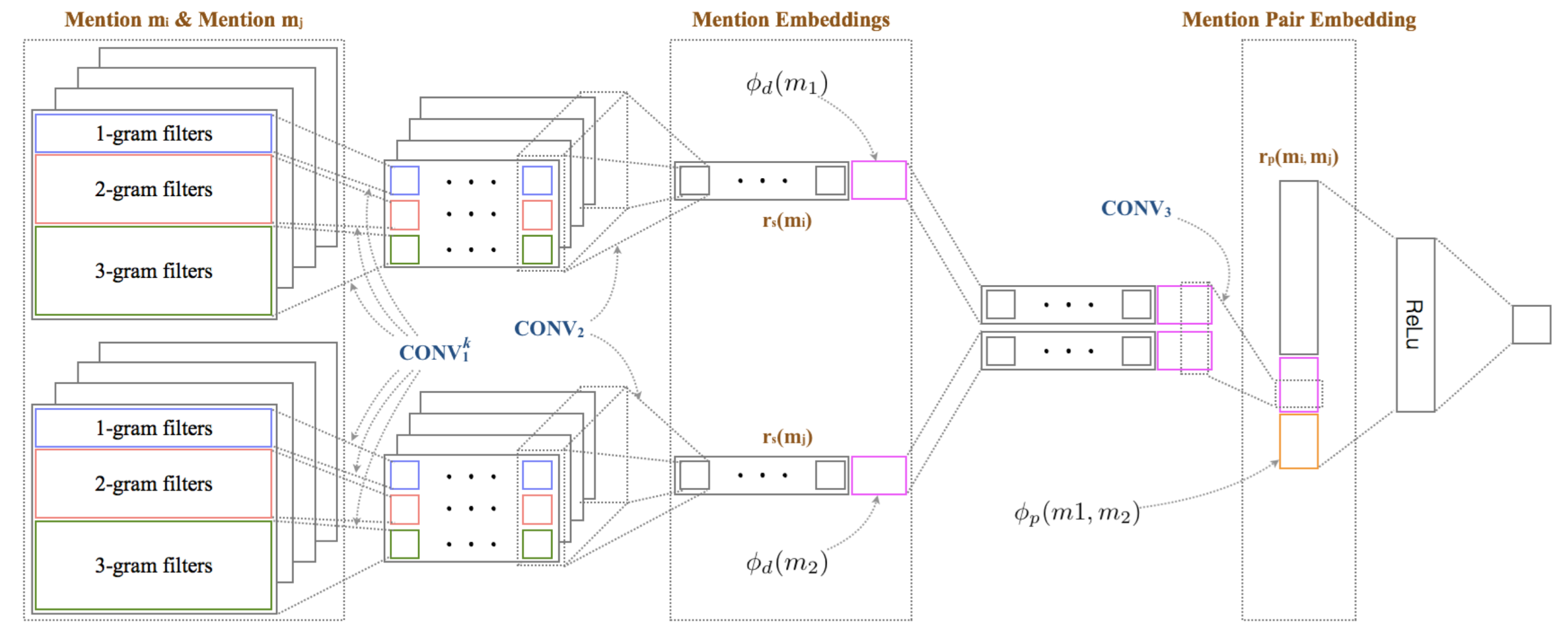
### 角色识别系统整体模型结构



凝聚卷积神经网络（Agglomerative Convlution Neural Network, ACCN），可以将众多的特征组加入到不同的卷积层中，从而创建出更加强壮的嵌入表达。系统利用ACCN的特性，学习输入预训练词向量的特征组合，同时融合其他的语言学特征，对mentions、mentions对、mentions簇进行嵌入表达，完成共指消解任务（对mentions簇给予对应的实体标签）。角色识别任务，借助共指消解系统从mentions簇中抽取丰富的特征，因此角色识别系统的效果强烈依赖于共指消解模型的质量。

在完成了共指消解任务后，ACCN模型会给出每个 mentions簇到具体实体的标注结果，将该结果送入后边的实体关系识别模型，该模型是一个卷积神经网络（Convolution Neural Network, CNN）模型，它完成实体关系（entity linking）标注，也就是角色识别任务。实体关系识别之所以可以借助CNN神经网络的原因是：利用了mentiosn分布式嵌入表达可以表达语义层面的特性。

### 角色识别系统的共指消解模型



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；

为抽取出的特征经1号卷积层的输出向量；

为2号卷积层；

||表示concatenate。

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；

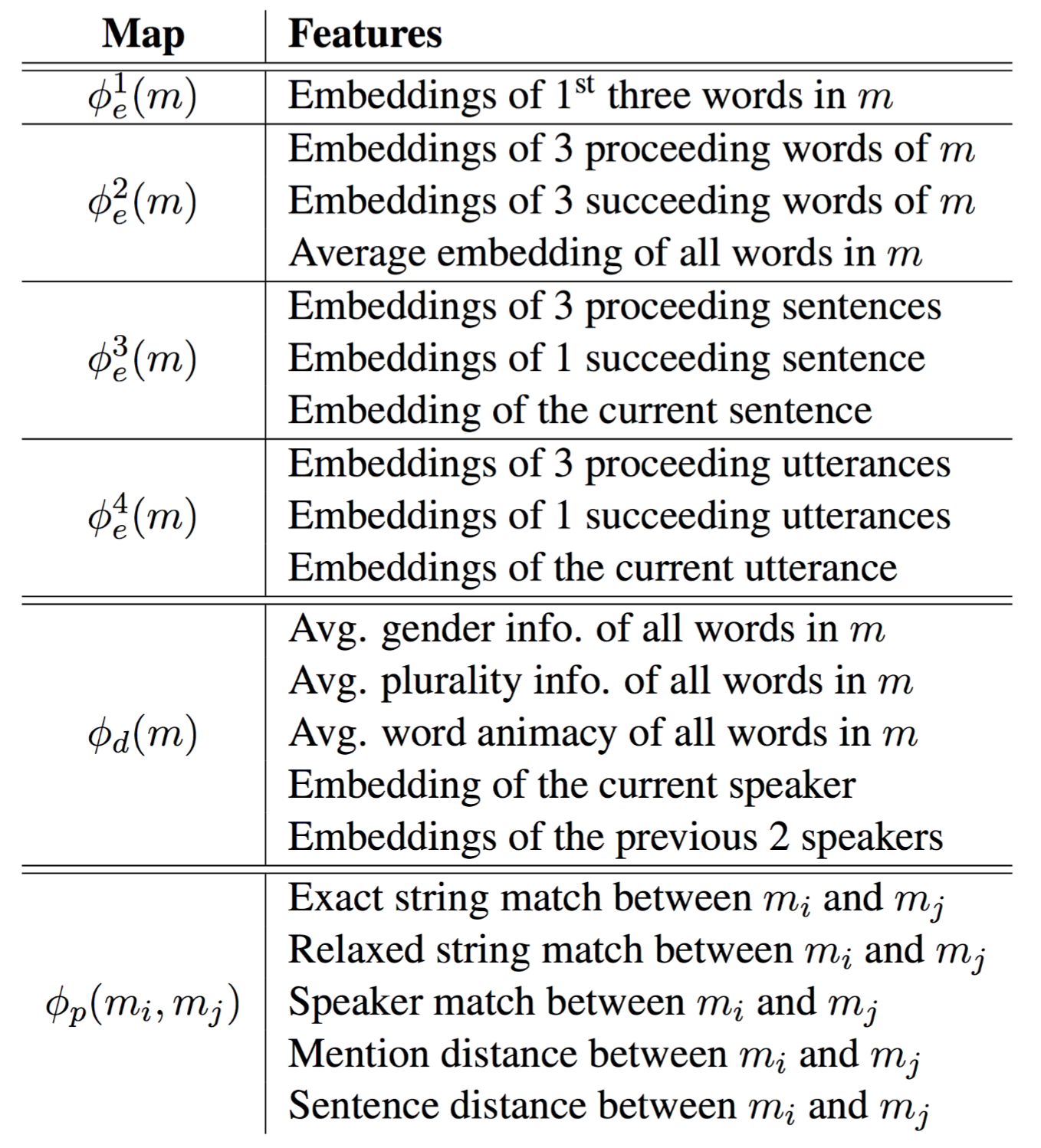
为mention-pair的特征模板，抽取两个mention字面值匹配度，说话者匹配度，距离，所在句子距离；

||表示concatenate。

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



### 共指消解系统评测

1. Bcube

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。

1. Ceafe

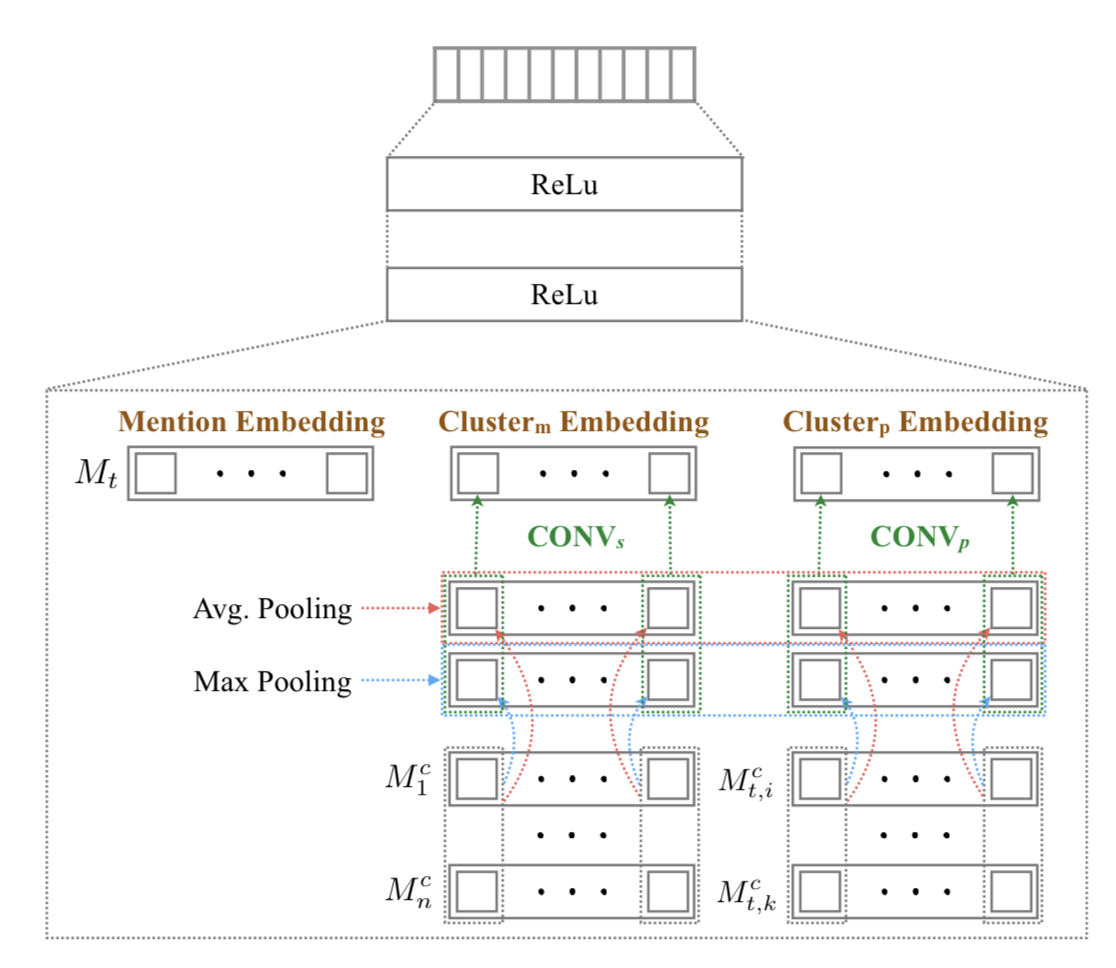
设实体Si表示一个系统给出的实体，Gj表示标准的一个实体，定义

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

### 角色识别系统评测

### 角色识别系统的实体关系预测

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。

### 将英文角色识别系统改为中文角色识别系统

* **Emory角色识别系统效果**

（1）共指消解系统效果

|  |
| --- |
| Evaluating trained model on Tst  2019-09-11 00:14:09,097 - LatestSystem.SubsystemTypes.COREF - INFO - Bcube - **0.9064**/0.4304/0.5836  2019-09-11 00:17:20,738 - LatestSystem.SubsystemTypes.COREF - INFO - Ceafe - 0.3691/0.6922/0.4815  2019-09-11 00:17:20,773 - LatestSystem.SubsystemTypes.COREF - INFO - Blanc - 0.8223/0.6297/0.6492 |

（2）角色识别系统效果

|  |
| --- |
| Evaluating trained model  2019-09-11 00:22:44,704 - LatestSystem.SubsystemTypes.ENTITY\_LINKING - INFO - Test accuracy: 0.3322/**0.9386** |

* **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。

识别效果

（1）共指消解系统效果

|  |
| --- |
| Evaluating trained model on Tst  2019-09-11 13:14:22,763 - LatestSystem.SubsystemTypes.COREF - INFO - Bcube - **0.8309**/0.5249/0.6434  2019-09-11 13:18:27,650 - LatestSystem.SubsystemTypes.COREF - INFO - Ceafe - 0.3670/0.6728/0.4749  2019-09-11 13:18:27,695 - LatestSystem.SubsystemTypes.COREF - INFO - Blanc - 0.7289/0.6706/0.6881 |

（2）角色识别系统效果

|  |
| --- |
| Evaluating trained model  2019-09-11 13:23:45,483 - LatestSystem.SubsystemTypes.ENTITY\_LINKING - INFO - Test accuracy: 0.3060/**0.9385** |

* **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。

识别效果

（1）共指消解系统效果

|  |
| --- |
| Evaluating trained model on Tst  2019-09-11 17:45:05,319 - LatestSystem.SubsystemTypes.COREF - INFO - Bcube - **0.8860**/0.4766/0.6198  2019-09-11 17:48:16,179 - LatestSystem.SubsystemTypes.COREF - INFO - Ceafe - 0.3766/0.6999/0.4897  2019-09-11 17:48:16,224 - LatestSystem.SubsystemTypes.COREF - INFO - Blanc - 0.7989/0.6656/0.6917 |

（2）角色识别系统效果

|  |
| --- |
| Evaluating trained model  2019-09-11 17:53:43,555 - LatestSystem.SubsystemTypes.ENTITY\_LINKING - INFO - Test accuracy: 0.4230/**0.9445** |

### 在《我爱我家》剧本语料上的评测

# Emory NLP4J自然语言处理系统

## 简介

Natural Language Processiong for JVM languages（NLP4J）项目提供的主要功能有：

* 可在各种研究过程中使用的工具
* 开发高效健壮的NLP组件
* 操作NLP特有数据结构的API（例如：dependecy graph）

该项目目前由 [Emory NLP](http://nlp.mathcs.emory.edu/) 研究组创建并管理。

* 最新释放版本：1.1.3
* 论坛（国内无法访问）
* demo（国内无法访问）

## 快速开始

### 怎样安装

方法1为MVN安装包（在Windows下总提示文件系统IO有错误）。这里请使用Linux系统。

注意项目为嵌套项目，执行编译安装时在项目根目录下，即在项目根目录运行mvn clean install。命令行验证安装是否成功时，在cli目录下，即在cli目录下运行mvn exec:java -Dexec.mainClass="edu.emory.mathcs.nlp.bin.Version"，返回下边结果表示执行成功。

|  |
| --- |
| ========================================  NLP4J Version 1.1.3  Contact: choi@mathcs.emory.edu  Webpage: http://emorynlp.github.io/nlp4j  ======================================== |

方法2为直接下载汇编包，该包中包含了所有需要的class文件以及针对英文的模型。下载汇编包[nlp4j-appassembler-1.1.3.tgz](http://nlp.mathcs.emory.edu/nlp4j/nlp4j-appassembler-1.1.3.tgz)后解压并运行下边的命令，如果返回下表所示内容，表示安装成功。

|  |
| --- |
| > bin/version  =========================================  NLP4J Version 1.1.3  Contact: choi@mathcs.emory.edu  Webpage: https://emorynlp.github.io/nlp4j  ========================================= |

### 怎样训练

1. nlptrain的命令行参数

命令参数

|  |  |
| --- | --- |
| 参数 | 意义 |
| -c | 配置文件 |
| -m | 模型文件 |
| -p | 已有模型文件 |
| -t|d | 训练集|验证集数据所在目录或单独一个文件 |
| -te|de | 训练数据文件的扩展名，默认为任意 |
| -cv | 交叉验证集比例 |
| -mode | 训练算法名称，可用算法见下表 |

NLP算法表

|  |  |  |
| --- | --- | --- |
| 算法名称 | 说明 | 备注 |
| custom | 定制NLP | 不知用途 |
| pos | 词性标注 |  |
| ner | 命名实体识别 |  |
| dep | 依存句法分析 |  |
| srl | 语义角色标注 |  |
| doc | 文档分类 |  |
| it | it分类 | 不知用途 |
| sentiment\_ensemble | 情感分析：ensemble方法 |  |

配置文件

对模型的的配置说明，对训练数据格式的说明。具体参考

* <https://emorynlp.github.io/nlp4j/quickstart/train.html#configuration>
* <https://github.com/emorynlp/nlp4j/tree/master/api/src/main/resources/edu/emory/mathcs/nlp/configuration/>

1. 示例

这里给出一个训练依存句法分析模型的示例。

执行命令如下

|  |
| --- |
| > java -Xmx1g -XX:+UseConcMarkSweepGC java edu.emory.mathcs.nlp.bin.NLPTrain -mode dep -c config-train-sample.xml -t sample-trn.tsv -d sample-dev.tsv -m sample-dep.xz |

选项说明：

[-XX:+UseConcMarkSweepGC](http://www.oracle.com/technetwork/java/tuning-139912.html) 用于减少内存消耗。

[log4j.properties](https://github.com/emorynlp/nlp4j/blob/master/api/src/main/resources/edu/emory/mathcs/nlp/configuration/log4j.properties) 日志工具

模型文件的生成：

一旦训练结束，sample-dep.xz会被创建，该功能可以在configuration文件中的指定，要求dependency parsing功能。

训练过程的终端信息说明：

L：类别数量

SF：离散特征数量

NZW：非零特征数量

N/S：每秒中处理的节点数

### 怎样decode

## NLP组件

## 补充说明