1．语料库标注格式

1.1 实例级(i-instance)

例如：NER(CoNLL 2003, JNLPBA 2004), RE(SemEval 2010)

特点：

1. 仅适合于一个单一任务，如NER, RE；
2. 对于RE，标注单位是一个实例；对于NER，标注单位是一个句子；
3. 对于训练集或测试集，只有一个语料文件 NER(\*.iob), RE(\*.wrd)。

1.2 句子级(s-sentence)

例如：NER(BC2GM), BEL(BEL)

特点：

1. 对于训练集或测试集，有多个文件：文本(\*.txt)，实体(\*.ent)，或关系(\*.sbel)；
2. 文本文件中有句子的编号和内容两个字段：snt-id, snt；
3. 实体、关系文件中标注了每个句子的所有实体或关系。

1.3 摘要级(a-abstract)

例如：NER(NCBI, CHEMD), RE(CPR), VE(GE09)

特点：

1. 语料库众多，文件格式与句子级标注相似；
2. 区别在于文本文件中含有三个字段: pmid, title和abstract；

1.4 文本级(f-fulltext)

例如：NER(LINNAEUS)

特点：

1. 训练集有实体文件(\*.ent)，文本文件位于train/test目录下，每篇文章一个文件；
2. 实体标注格式与句子级和摘要级相同。
3. 语料库的转换

功能：将原始语料库转换成标准的实例、句子、摘要或全文级语料库。

源码：bio\_io\_conversion.py

列表：

1. convert\_asq\_corpus: BIOASQ ()
2. convert\_bc2gm\_corpus: (Gene NER)
3. convert\_cdr\_corpus: CDR (CDI RE)
4. convert\_chemd\_cemp\_corpus: CHEMD(Chemical NER), CEMP
5. convert\_cpr\_corpus: CPR (Biomedial RE)
6. convert\_euadr\_corpus: EUADR
7. convert\_gad\_corpus: GAD
8. convert\_ge09\_corpus: Genia Event 2009 (Event extraction)
9. convert\_linn\_corpus: LINNAEUS (Species NER)
10. convert\_ncbi\_corpus: NCBI: (Disease NER)
11. convert\_s800\_corpus:
12. convert\_jnlpba\_corpus：(GENE NER)
13. convert\_conll2003\_corpus：(NEWS NER)
14. 主要功能介绍

r-review, t-train, v-validation, p-prediction

3.1 审视(r-review)

功能：检查每个语料库的训练、测试集能否产生docset，并对标注实体、关系、事件等进行统计。对于复杂任务，还进行子任务流水线处理。

列表：每种任务一个

1. review\_ner\_corpus：统计语料库实体情况
2. review\_re\_corpus：统计语料库关系情况
3. review\_ve\_corpus：分解为ve\_trg和ve\_arg两个NER/RE子任务，并进行统计
4. review\_sbel\_corpus：分解为sbel\_bre和sbel\_ure两个RE子任务，并进行统计

输出：

1. NER或RE配置文件：
   * 1. '{}/{}\_{}\_cfg.json'.format(wdir, task, cps\_file): 目录，任务，数据集(train/test)
     2. 复杂任务名称为子任务+简单任务，如ve\_trg\_ner, ve\_arg\_ner
     3. 例如：CPR/re\_train\_cfg.json, GE09/ve\_trg\_ner\_train\_cfg.json
     4. 正式训练时，需要一个总的配置文件，如re\_cfg.json, ve\_trg\_ner.json，因此需要将训练集的配置文件手工改名为总配置文件；
     5. PS: 每次审视语料库时，如果配置文件不存在，则自动创建。如果需要更新配置文件，则需要手工删除配置文件。对于某些语料，如SBEL，由于测试集只包含部分一元关系类型，因此需要将训练集的配置文件复制到测试集配置文件中，否则统计一元关系时会出错。
2. NER或RE统计文件：entities.cnt, relations.cnt, events.cnt, ...

3.2 训练(t-train)

功能：

1. 单独使用时，对所有的标注语料库进行训练，以产生用于最终生产的模型；
2. 与功能v和p合用时，对除测试集外的所有标注语料进行训练，以产生用于验证或预测的模型。
3. 与功能v合用且语料只有一个数据集(如训练集)时，进行多倍交叉验证。交叉验证参数由FOLD\_NUM和FOLD\_NUM\_RUN定义；
4. 模型文件名：'{}/{}\_{}\_e{}\_f{}.hdf5'.format(wdir, task, model\_name, epo, fold)
   * 1. task为任务全称，epo为轮编号(从1开始), fold为交叉验证中的数据集编号(从1开始)或main中指定的初始种子(从0开始)；
     2. 如：SMV/re\_Bert\_e3\_f0.hdf5, GE09/ve\_trg\_ner\_e2\_f0.hdf5
5. PS: 在训练时，epo参数无效，训练轮数由EPOCHS决定，而fold参数则决定初始种子。在验证或预测时, epo和fold决定了采用哪一个训练好的模型。

3.3 验证(v-validation)

1. 仅对有标注的测试集进行验证，并计算相应性能；
2. 验证评估的级别：表明是在摘要/句子/实例的哪一个层面上进行评估：
   1. inst: 在实例级进行评估，包括正例和负例；
   2. sent: 在句子级进行评估，只有正例；
   3. docu: 在摘要、全文级进行评估，只有正例；
   4. PS: 在摘要向句子转换，或句子向实例转换过程中，由于会发生正例损失现象，因此通常实例级性能不小于句子级，句子级不小于摘要级。
3. 输出的性能文件为：'{}/{}\_{}.log'.format(wdir, task, cps\_file)
4. 输出的结果文件为：'{}/{}\_{}.rst'.format(wdir, task, cps\_file)
5. 输出文件的详细程度由参数verbose的值来决定：
   1. verbose = 0：仅在屏幕上输出总体性能
   2. verbose >= 1：在日志文件(log)中输出分类结果的混淆矩阵和分类性能；
   3. verbose >= 2：在结果文件(rst)中输出分类错误的实例；
   4. verbose >= 3：在结果文件(rst)中输出所有实例的分类结果；

3.4 预测(p-prediction)

(1) 对测试集进行预测，没有性能结果；

(2) 预测结果文件的格式同实体或关系标注文件格式相同；

(3) 预测结果输出文件为：'{}/{}\_{}.prd'.format(wdir, task, cps\_file)

1. Classes





类说明:

ieDocSets: 语料库，包含多个文件集(filesets)和三个数据集(datasets[3])，后者用于训练验证和测试。其主要功能有创建语料库、获取语料库文件集、模型训练和评估、语料库统计等。

ieDocSet: 文档集，包含文档字典(docdict)，主要功能有产生文档集、获取文档集的字符型和整数型数据、评估模型性能等。

ieDoc: 文档，包含文本(text)和句子集(sntlist), 以及实体字典(emdict)和列表(emlist)，及其它标注信息。主要功能有产生文档句子、产生句子实例等。

EntityMention: 实体提及，包含实体类型、位置、标准名称等信息。

Instances：针对某一任务的实例，如标签序列(SequenceLabel)、分段标签序列(segSequenceLabel)、关系实例(RelationMention)等。主要方法有产生实例特征、输出实例等。



其中：实线箭头表示继承关系，虚线表示任务之间的调用关系

1. 平台总体结构 main()

convert\_bio\_corpus: 转换语料库

review\_[task]\_corpus: 审视语料库

train\_eval\_corpus: 对语料库进行训练、评估和预测

* 1. convert\_bio\_corpus
  2. review\_[task]\_corpus
  3. train\_eval\_corpus

ieDocSets=create\_corpus\_filesets()：创建语料库

load\_json\_file: 加载任务配置文件

load\_word\_voc\_file: 加载词表文件（非BERT）

load\_bert\_tokenizer: 加载BERT词表和符号化接口

load\_pretrained\_embedding\_from\_file: 加载预训练向量（非BERT）

ieDocSets.prepare\_corpus\_filesets: 为语料库生成文件集

prepare\_corpus\_fileset：针对单一语料文件

copy2docset：从语料库产生文件集对象

ieDocSet.*prepare\_docset\_instance*: 为实例级(i)标注语料准备文件集

ieDocSet.*prepare\_docset\_abstract*: 句子级/摘要级/全文级(s/a/f) 【后续】

ieDocSet.prepare\_docset\_dicts\_features: 为文件集准备标签字典，词表和实例特征

*create\_type\_label\_dict*：建立类别和标签字典

*set\_type\_label\_dict*：设置类别和标签字典

generate\_docset\_instance\_features： 为文档集产生特征

inst.*generate\_instance\_features*：为一个实例产生特征

create\_docset\_word\_vocab：建立词表

set\_docset\_word\_vocab：设置词表

ieDocSets.train\_eval\_docsets: 对生成的文件集进行训练、验证和预测【后续】

ieDocSets.calculate\_docsets\_instance\_statistics: 统计语料库实例情况

ieDoc.generate\_docset\_instance\_statistics: 统计文档集实例情况

ieDocSets.output\_docsets\_instance\_statistics：输出语料库实例情况

ieDocSets.output\_aspect\_statistics：输出一个维度(如级别、文件名)的实例统计数据

* + 1. ieDocSets.train\_eval\_docsets

set\_random\_seed：设置随机种子值

TrainConfig：产生训练配置

prepare\_tdt\_docsets：常规验证，从文件集中划分出训练、验证和测试文档集

ieDocSet.*get\_docset\_data*: 获取文档集字符型训练数据

如果是验证或预测，则将最后一个数据集设为测试集，其它为训练集。

对于模型训练，根据验证比率进一步划分训练集和验证集。

prepare\_cv\_tdt\_docsets：交叉验证，从文件集中划分出训练、验证和测试文档集

ieDocSet.*get\_docset\_data*:

create\_training\_model：创建训练模型

train\_eval\_model：进行多轮训练验证模型

在训练集上进行训练

get\_model\_filename：获得模型文件名

data\_generator：产生训练用数据

model.fit\_generator：训练模型

model.save\_weights：保存模型权重

对验证集或测试集：

data\_generator：产生预测用数据

predict\_on\_batch：批量预测

*assign\_docset\_predicted\_results*：将预测结果赋给文档集中的实例

*calculate\_docset\_performance*：计算文档集的P/R/F1性能

cv\_train\_eval\_model：交叉训练验证

对每一份数据集的每一轮

在训练集上进行训练

get\_model\_filename

data\_generator/model.fit\_generator/model.save\_weights

评估验证集或测试集：

data\_generator/predict\_on\_batch\_keras/

*assign\_docset\_predicted\_results/calculate\_docset\_performance*

预测

test\_with\_model: 用单模型预测

ensemble\_classify: 用集成模型预测

ieDocSet.evaluate\_docset\_model: 评估模型性能

*assign\_docset\_predicted\_results*: 将预测结果赋给文档集中的实例

*calculate\_docset\_performance*: 计算文档集的P/R/F1性能

*dispatch\_docset\_predicted\_results*: 将预测结果分配到文档中

*output\_docset\_predicted\_results*: 输出文档集预测结果

1. NER中需要重写的方法
   1. ieDocSets.create\_training\_model (ie\_docsets.py)

RE/URE是句子级分类模型

NER/segNER是符号级分类模型，BERT或LSTM-CRF模型

* 1. ieDocSet

prepare\_docset: 为标注语料库准备文档集

*prepare\_docset\_instance*：实例级(i)标注

*prepare\_docset\_abstract*：句子级/摘要级/全文级(s/a/f)标注

ieDocSet.load\_docset\_bio\_text: 加载文本

ieDocSet.load\_docset\_entity\_mentions: 加载实体列表

ieDocSet.check\_docset\_entity\_mentions: 检查实体标注和文本的一致性

ieDocSet.generate\_docset\_sentences: 产生文档集句子

ieDoc.*generate\_document\_sentences*: 产生文档句子

ieDocSet.collect\_docset\_instances: 收集文档集实例

ieDoc.*generate\_sentence\_instances*：从句子产生实例

label\_type\_dict: 处理任务所使用的标签类别字典

*create\_type\_label\_dict*：创建

create\_entity\_type\_dict: 收集实体类型字典 (.etypedict)

create\_entity\_label\_dict: 创建实体标签字典 (.elabeldict)

*set\_type\_label\_dict*：设置

*get\_label\_dict*：获取标签字典

create\_entity\_type\_dicts: 为每一个文件集创建实体类型字典 (.etypedict)

*get\_docset\_data*：获取文档集字符型训练数据，NER/RE在标签数量上在区别

get\_label\_dict: (.etypedict)

*output\_docset\_instance\_candidates*：向模型输出文档集的整数型训练数据

inst.*output\_instance\_candidate*: 输出实例

get\_word\_idx：获得词索引

*assign\_docset\_predicted\_results*：将整数型预测结果赋给文档集实例

*calculate\_docset\_performance*：计算实例级/句子级/文档级任务性能

*collect\_docset\_performance*: 收集性能数据，包括混淆矩阵，PRFs

match\_gold\_pred\_instances: 匹配预测和标准实体

collect\_ner\_confusion\_matrix: 收集混淆矩阵

get\_ner\_labeled\_sentences: 获得人工或自动标注的句子

sum\_confusion\_matrix\_to\_prfs: 汇总混淆矩阵

calculate\_classification\_prfs: 计算分类性能

*output\_docset\_performance*: 收集性能数据，包括混淆矩阵，PRFs

output\_confusion\_matrix: 输出混淆矩阵

output\_classification\_prfs: 输出分类性能

*dispatch\_docset\_predicted\_results*：将预测结果分配到文档中

*output\_docset\_predicted\_results*：输出文档集预测结果

* 1. ieDoc

*generate\_document\_sentences*: 将文档转换为句子集，同时转换各种标注信息。

replace\_bio\_special\_chars：替换文档特殊字符

split\_bio\_sentences：对文档分句

为每个句子：

bio\_sentence\_tokenize：句子符号化

recover\_entity\_mentions：从句子中恢复实体

reDoc.transfer\_relation\_mentions：从句子中恢复关系实例

ieDoc.align\_document\_with\_sentences：将句子中的单词对齐到文本中的字符

ieDoc.transfer\_entity\_mentions\_from\_documents\_to\_sentences：

将文本中的实体转移到句子中，其位置定位到单词序列中。

*generate\_sentence\_instances*：从句子中产生标注实例

* 1. Instances: SequenceLabel/RelationMention/...

*generate\_instance\_features*: 产生一个实例的特征

SequenceLabel.generate\_wordpiece\_bert\_sequence

产生BERT所用的词片序列，并计算每一个词所对应的词片序列范围boffsets

SequenceLabel.locate\_entity\_mention\_in\_bert\_sequence

根据boffsets将所有实体重新定位到词片序列中

SequenceLabel.generate\_sequence\_labels

根据一个序列中的实体列表和标签方案产生标签序列

*output\_instance\_candidate*: 返回模型所要用到的实例特征，即词序列和类别标签(序列)

1. 扩展类
   1. 二元关系抽取
      1. reDocSets (ieDocSets)

calculate\_docsets\_relation\_statistics：统计语料库关系实例情况

reDoc.generate\_docset\_relation\_statistics: 统计文档集关系实例情况

output\_docsets\_relation\_statistics：输出关系类型统计结果

* + 1. reDocSet (ieDocSet)

**rmdict**, **rrmdict**: dicts of gold and recognized relation mentions

*prepare\_docset\_instance*：实例级(i)标注

*prepare\_docset\_abstract*：句子级/摘要级/全文级(s/a/f)标注

reDocSet.load\_docset\_relation\_mentions: 加载文档集二元关系实例

reDocSet.preprocess\_docset\_entity\_mentions: 对文档集的实体进行预处理

ieDoc.preprocess\_document\_entity\_mentions: 对文档的实体进行预处理

sort\_entity\_mentions: 实体排序

mask\_nested\_entity\_mentions: 屏蔽嵌套实体

replace\_entity\_mention\_with\_placeholder: 用占位符替换实体

replace\_bio\_special\_tokens: 替换特殊符号

ieDocSet.prepare\_docset\_dicts\_features:

*create\_type\_label\_dict*：创建标签类型字典

create\_relation\_type\_dict: 创建关系类型

create\_relation\_label\_dict: 创建标签字典

*calculate\_docset\_performance*：计算实例级/句子级/文档级任务性能

inst.collect\_re\_confusion\_matrices: 收集关系抽取混淆矩阵

Inst.get\_re\_labeled\_instance: 获取人工或自动识别的实例

* + 1. reDoc (ieDoc)

reDoc.*generate\_document\_sentences*: 产生文档句子

reDoc.transfer\_relation\_mentions：将二元关系从文档转移到句子中

reDoc.*generate\_sentence\_instances*：从句子产生实例

reDoc.generate\_relation\_mention\_candidate：为一对实体或一个实体产生实例

rm.insert\_entity\_mentions\_delimiters：插入实体两边的分界符

* + 1. Instance: RelationMention

rm.insert\_entity\_mentions\_delimiters

* 1. 一元关系抽取
     1. reDocSets
     2. ureDocSet (reDocSet)

ureDocSet.load\_docset\_relation\_mentions: 加载文档集一元关系实例

* + 1. ureDoc (reDoc)

ureDoc.transfer\_relation\_mentions: 转移一元关系

ureDoc.generate\_sentence\_instances: 从句子产生一元关系实例

* + 1. Instance: RelationMention (emid2=None)
  1. 分段序列标注 (SSL, segmented sequence labeling)
     1. ieDocSets
     2. sslDocSet (ieDocSet)

**slabel\_schema**, **stypedict**, **slabeldict**: 段的标签方案，类型字典和标签字典

create\_type\_label\_dict：

super.create\_entity\_type\_dict/super.create\_entity\_label\_dict：

create\_segment\_type\_dict：获得段的类型字典

create\_segment\_label\_dict：获得段的标签字典

generate\_docset\_instance\_features：

inst.generate\_instance\_features: 多一个参数slabel\_schema

output\_docset\_instance\_candidates:

get\_label\_dict：

get\_segment\_label\_dict：

* + 1. sslDoc (ieDoc)

smlist: 分段实体列表

generate\_sentence\_instances：

segSequenceLabel()：多一个参数(.smlist)

* + 1. Instance: segSequenceLabel (SequenceLabel)

smlist, slabels: 分段实体列表和标签序列

generate\_instance\_features：

super.generate\_instance\_features(): 为正常实体产生特征和标签

locate\_entity\_mention\_in\_bert\_sequence(smlist): 定位分段实体

generate\_sequence\_labels(smlist): 为分段实体准备标签序列

output\_instance\_candidate：(words, labels, slabels)