Text Processing 第四章 IE

Notebook:	Text Processing		
Created:	1/29/2020 8:44 AM	**Updated:**	1/30/2020 4:44 PM
Author:	Alvin		

1.Information Extraction的定义：

对于每一个非结构性的自然语言(Unstructured natural language)文本，提取出预先定义的entities(对象), relationships(关系)，events(事件)，然后通过以下几种结构化的类型来记录：

- 标注(Annotating)过的源文本，比如使用XML tags
- 从文本中将信息分离填充到一个结构数据中。比如，一个结构化数据模版或者数据库

Information Extraction 也可以被理解为使用无结构的自然语言，构建一个结构化信息数据库。
我们得到的结构化数据可以用来：

- 使用常规的数据查询来搜索和分析
- 数据挖掘
- 生成报告

与Information Retrival 对比：

	Information Extraction	Information Retrival
Task:	• 输入input: 一个预先定义的entities(对象),	• 输入input：一个文本库，和一个用户查询

	relationships(关系)，events(事件) 集合和数据集 • 输出return：包括entities(对象), relationships(关系)，events(事件)的结构化数据	• 返回return：一个list的与用户查询相关的doc
Strengths:	• 从文本中提取重要的事实信息，而不仅是从文本集中筛选出文本 • 可以给数据库，语义索引引擎，数据挖掘工具提供数据。	• (rapid) 能够快速查询庞大的文本库 • (insensitive)对文本的类别和领域不敏感 • (straightforward) 实际实施起来比简单。
Weakness:	• 系统倾向于产生对应类别，或者领域的数据。如果对应新兴的类别或者领域额外需要专业人员，同样费时间。 • 精确度有限（limited） • 算力消耗大，对应非常大的数据集会有算力问题。	• 用户必须看输出的内容来额外获得信息 • 输出内容是非结构化的，所以不能直接用来做数据挖掘(Data mining)

	(computationally demanding)	

IE的例子：

- 从网页上提取信息，生成结构化数据。通俗点说，爬虫(web crawler)就是其中一种。
- 从人工维护的数据库中，通过从科学文献中提取entities和relations来构建医学数据库
- 辅助公司获取竞争对手的信息，包括：研发成果，新产品，新项目。

2.Entity Extraction: Named Entity Recognition：

Task: 对于每一个entity中的文本，形成一个能够归纳其extent和type的fixed set。

Cable and Wireless today announced ...	Extent: 0-3; Type = ORG
IBM and Microsoft today announced ...	Extent: 0-1; Type = ORG
	Extent: 2-3 Type = ORG
John Lewis hired ...	Extent: 0-2; Type = ORG
Theresa May hired ...	Extent: 0-2; Type = PER

已经被IE 系统包括的entity类型有：

- 被命名的个体
- 被命名的种类
- 时间
- 量化数据(measure)

Coreference:

- 同一个entity也可以有多个references，而且使用相同的文本(string)

- 对于同一个现实中的实体的不同文字表达方式叫做corefer
- 如果可以认知到coreferential的文本，IE system会更加有用。
- Coreference Task: 忽视其表面的是否为一个名字的表达形式，把所有的文本refer到一个相同的现实entity，这个过程叫做Coreference Task.

Approaches:
一共分为四种方法来解决NER(Named Entity Recognition)

- Knowledge Engineering Approaches
- Supervised Learning Approaches
- Bootstrapping Approaches
- Distant Supervision Approaches

我们一般只考虑前两种

1.Knowledge Engineering Approaches to NER:
这样的系统通常只用：

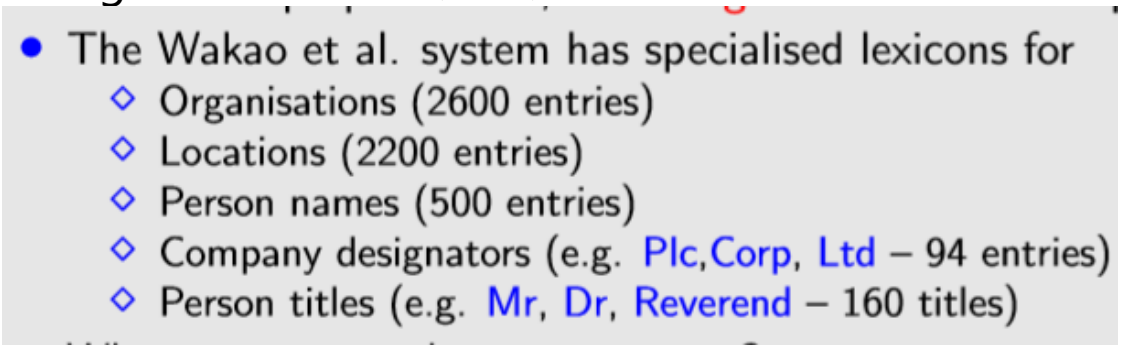
- **named entity lexicons**
- 人工编写的**pattern/action rules or regular expression**

系统有三个阶段：

- **Lexical processing**
- **NE parsing**
- **Discourse Interpretation**

第一步：**Lexical Processing:**

许多 rule-based NER 大量使用了特定的词汇名称，比如gazetteers（翻译过来是"地名"）



为什么不适用更大的地名呢？

- 原因是许多Named entity出现在多个类别中，词汇越长，越容易成生ambiguity(歧义)
- 命名的列表是不可能完整的，所以需要一些机制来输入一些从未见过的named entity

Wakao et al. system principal lexical processing sub-steps:

- 词语标签化，句子分割，词汇形态分析
- Part-of-speech tagging，对于已知的名称和未知的开头首字母大写的词汇打上合适的标签名。
- Name List/Gazetter Lookup and Tagging (organisations, locations, persons, company designators, person titles)
- Trigger Word Tagging， 特定单词在多单词的名字里面起到了trigger words的作用，允许对名字的分类。

Example:
Norwich Investment Bank plc. today announced ... $\longrightarrow$
$\text{Norwich}_{NNP/LOC}$ Investment_{NNP} $\text{Bank}_{NNP/ORG-TRIGGER}$ $\text{plc.}_{NN/CDG}$
today_{RB} announced_{VBD} ...

第二步：NE(Named Entity) Parsing

具体过程看这个图就行了

The system has 177 hand-produced rules for proper names: 94 for organisation; 54 for person; 11 for location; 18 for time expressions.

A fragment of the proper name grammar:

```
NP--> ORGAN_NP
ORGAN_NP --> LIST_LOC_NP NAMES_NP CDG_NP
ORGAN_NP --> LIST_ORGAN_NP NAMES_NP CDG_NP
ORGAN_NP --> NAMES_NP '&' NAMES_NP
NAMES_NP --> NNP NAMES_NP
NAMES_NP --> NNP
```

The rule ORGAN NP --> NAMES_NP '&'' NAMES_NP means:

If an unclassified proper name (NAMES_NP) is followed by '&' and another unclassified proper name, then it is an organisation name.

第三步：Discourse Interpretation – Coreference Resolution

- 当一个词的名字类别的对应词是已知的，通过建立coreference可以建立这样的一种名字类别和对应词的关系

例子：

E.g., Alice is my sister. She is CEO of MS. (Antecedent – Alice; anaphor – she)

- 一个没有分类的 PN(Proper name)和另外一个 分类过的 PN的不同格式可以是co-referential 关系。

例子：

Ford – Ford Motor Co.
CAA – Creative Artists Agency

从这个例子中看出没分类的PN可能是从已分类的PN中推断出来是相同的种类。

- 一个未分类的PN也可以和一个definite NP(有定名词词组)是co-referential关系，这样PN的种类也可以被推断出来。

例子：Kellogg ... the breakfast cereal manufacturer

第三步：Discourse Interpretation – Semantic Type Inference

在涉及到语法(syntactic relations)的语义形式(Semantic type)上的信息也可以被用来推断 PN的种类。

- noun-noun qualification：当一个未分类的PN可以qualify一个organisation-related object，那么这个PN也可以分类成一个organisation.

  例子： Erickson stocks

- possessives: 当一个未分类的PN和一个organisation post处于一种所属关系中的时候，那么这个PN也可以分类成一个organisation

  例子：vice president of ABC, ABC's vice president

- apposition: 当一个未分类的PN和一个organisation post处于并列关系中，前面的名字被分类成人名

  例子：Miodrag Jones, president of XYZ

Knowledge Engineering Approaches to NER: Strengths and Weaknesses

优点：

- 高性能，只比人类差一点点
- 透明，可解释性强

缺点：

- 当应用到另一个领域的时候，需要重新构造基本的规则
- 只得到对应领域的lexicon
- 写规则需要很高的专业技术

2.Supervised learning approaches to NER:

- 监督学习方法解决了knowledge engineering NER中的存在的可移植性的问题。

1.不需要人工来编写规则，直接从标好的样本中学习

2.移植到另外一个领域只需要对应那个领域的标注的数据。

- 很多监督学习的模型都被尝试过了：

- Hidden Markov models
- Decision Trees
- Maximum Entropy
- Support Vector Machines
- Conditional Random Fields
- AdaBoost
- Deep Learning

Sequence Labelling:

- 系统可以学到

1.可以和extraction targets匹配的pattern
2.分类器(classifiers)可以对tokens进行打标签(label)
as beginning/inside/outside a tag type

- 最近的研究都使用了sequence labelling的方法
- 在sequence labelling for NER中，每一个token被给予了三种形式的标签

- B_{Type} if the token is at the beginning of a named entity of type $= Type$ (here, e.g., $Type \in \{ORG, PER, LOC\}$).
- I_{Type} if the token is inside a named entity of type $= Type$
- O if the token is outside any named entity

这种方法被叫做BIO，或者IOB sequence labelling

Features for Sequence Labelling:

这一段直接看**note**的例子，这一部分主要是在选取训练使用的**features**，也就是特征

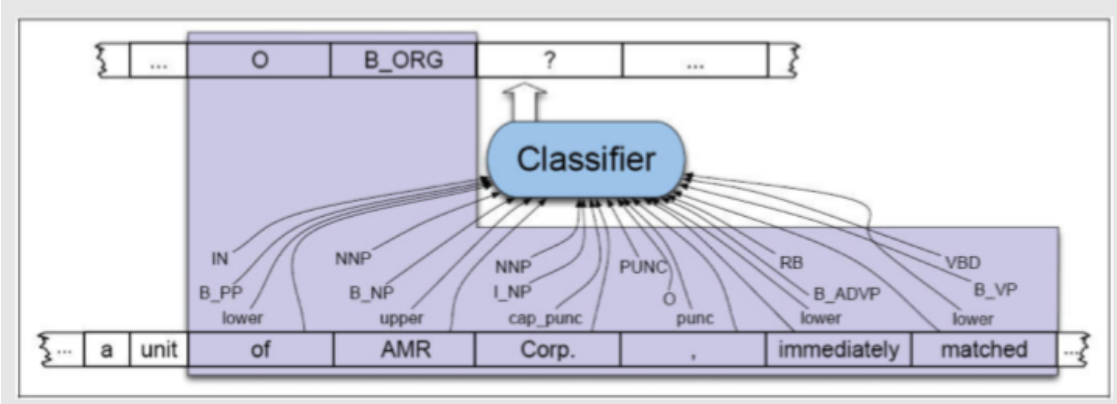
- Given a BIO-type encoding, each training instance (token) is typically represented as a set of features.
- Features can be not only characteristics of the token itself but of neighbouring tokens as well
  - ◇ usually consider tokens in a window of e.g. ± 2 or 3 tokens either side of the training instance
- Features commonly used for NER sequence labelling include:

Feature	Explanation
Lexical items	The token to be labeled
Stemmed lexical items	Stemmed version of the target token
Shape	The orthographic pattern of the target word
Character affixes	Character-level affixes of the target and surrounding words
Part of speech	Part of speech of the word
Syntactic chunk labels	Base-phrase chunk label
Gazetteer or name list	Presence of the word in one or more named entity lists
Predictive token(s)	Presence of predictive words in surrounding text
Bag of words/Bag of N-grams	Words and/or *N*-grams occurring in the surrounding context

- For case sensitive languages like English the orthographic pattern of a token carries significant information.
- Commonly used "shape" features include:

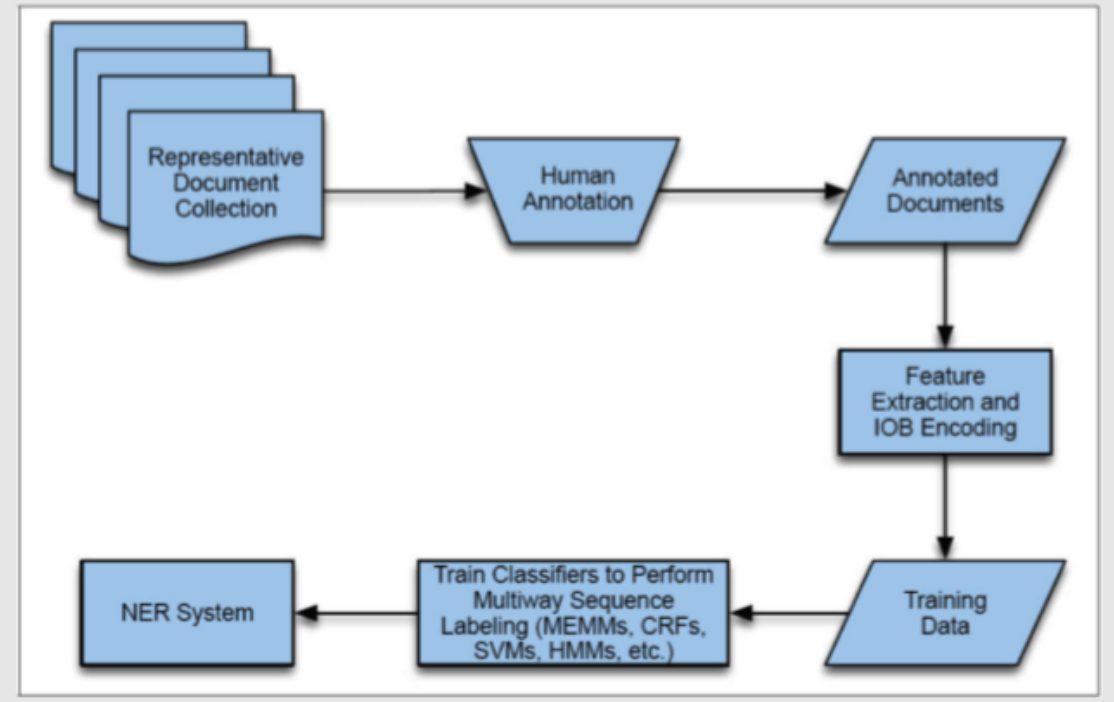
Shape	Example
Lower	cummings
Capitalized	Washington
All caps	IRA
Mixed case	eBay
Capitalized character with period	H.
Ends in digit	A9
Contains hyphen	H-P

在模型训练完之后，分类器从输入string和其预测结果中提取出了特征。



上图灰色部分是可以用来分类**(classification)**的特征**(features)**

整个sequence labelling approach to NER的流程:



然后介绍了Carreras et al这个人做的算法:
他们把问题分成了两部分:

- NE detection: in a first pass over the text BIO tags are assigned without regard to type – i.e. boundaries are found for all NE's regardless of whether they are organisations, persons, locations, etc.

第一部分就是说文本的BIO tags是不依赖type来分配的

- NE classification: in a second pass the NE's detected in the first pass are assigned a class (organisation, person, location, etc.)

第二部分是第一个过程中被检测过的NE被分类到一个class中。
分两个步骤的好处是，所有NE class的训练数据可以被用来做NE detection的task。

- They used the Adaboost classifier
- They used all features mentioned above plus some additional ones, e.g.
  - Type pattern of consecutive words in context – functional (f), capitalized (C), lowercased (l), punctuation mark (.), quote (), other (x) – e.g. word type pattern for the phrase John Smith payed 3 euros is CClxl.

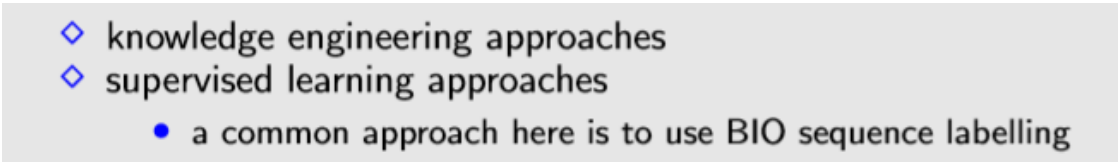
这一部分记住就行了。

Entity Linking:

- 一个非常重要的IE的应用实例就是 knowlege base population (KBP).通过从开放的网站资源中来构建结构化的数据库。
- 为了让KBP很好的工作，entities不仅需要被detected，而且他们必须被连接到合适的entry上。前提是这些事实信息是正确集合在一起的。
- 这引入了 Entity Linking Task，给定一个文本和在这个文本中熟悉的 NE mention，再加上一个knowledge base 比如wikipedia。连接所有NE 到KB里匹配的entry。如果没有这个entry，那就创建一个entry。
- 这个工作是很难的。因为部分词条的entry太多了

结论：

- NER已经是一个相对成熟而且达到“可用”水平的技术了。
- NER 用来检测并且分类一个文本中给定的entity type set中的所有提及到的named entity。
- 被用到的技术包括：



- 公开的challenge包括：

1.减少训练样本

2.利用现有的结构化数据来生成“weakly labelled”（弱标签化）的训练集

3.拓展entity的种类。
4.开发非英语语言的NER

3.Relation Extraction:

- Task:给定一个文本T，和一个关系的集合，R. 确定所有从T里面关于R的关系定义。

- 注意：

1.所有在R中的关系通常为binary，理解为要么有，要么没有就行了。
2.R 中的关系的entity type 是那些在entity extraction process中的同时存在的子集。

- 也可以分为两个sub task:

1. Relation detection: 找到一对有关系的entity。
2. Relation classification: 对于有关系的entity，决定他们之间的关系是什么。

例子：

- Examples
  - LOCATION_OF holding between
    - ORGANISATION and GEOPOLITICAL_LOCATION
    - medical INVESTIGATION and BODY_PART
    - GENE and CHROMOSOME_LOCATION
  - EMPLOYEE_OF holding between PERSON and ORGANISATION
  - PRODUCT_OF holding between ARTIFACT and ORGANISATION
  - IS_EXPOSED_TO holding between ORGANIZATION and RISK
  - IS_ASSOCIATED_WITH holding between DRUG and SIDE_EFFECT
  - INTERACTION holding between PROTEIN and PROTEIN

Relation Extraction 的challenges:

- 同样的关系可以有不同的表达方式
- 所需要的信息可能会被分散到多个句子中，发现关系可能取决于following coreference links.
- 需要被提取的信息可能会被文本隐喻，不是直接阐述，所以提取还需要推断。

Relation Extraction的方法：

这里只讲一下几种方法的优缺点，具体过程直接看一下**IE**的第三个**PDF**，从第**9**页到**32**页。

Knowledge-engineering approaches:

优点：

- 高精度（这一点和**NER**对应的同样方法是一样的）
- 系统的行为对于人类而言可解释性高

缺点：

- 写规则是写不完的
- 每一个新的领域都需要写**rule**

Supervised learning approaches:

优点：

- 不需要对每一个领域写**rule**
- 只要有训练集，同样的系统可以直接部署到新的领域

缺点：

- **Relation Extraction**的质量取决于训练集的数量和质量，产生可能很难也很费时间

- 开发 **feature extractors(特征提取器)**是很难的，而且噪点可能很大，影响整体性能。

Bootstrapping Approaches:

优点：

- 不需要人工标注的数据集

缺点：

- 会导致 **semantic drift**，一个错误的模版会引进错的数据组合，从而导致更多的错误模版
- 在某些表达关系的特定模版中，确定特定数据组合有多余**(redundant)**的信息时候才会有好用。
- 当同一对**entity**有多个关系的时候，就有问题。

Distant Supervision Approaches:

优点：

- 不需要人工标注的数据集
- 可以快速的得到多种样式的**extractors**。

缺点：

- 精准度**(precision)**比 **knowledge-engineered/directly supervised learning approaches**要差。
- 只有在提供良好的并且能够给确立关系有用的结构化数据时才有用。

结论：

- Relation Extraction是用来检测和分类：给定文本中，特定entity与所有涉及关系的集合。
- 由于自然语言中，关系可以表达的方式太多，Relation extraction是一项很难的技术。
- 包含技术：

  1.Knowledge-engineering approaches:
  2.Supervised learning approaches:
  3.Bootstrapping Approaches:'
  4.Distant Supervision Approaches:

公开的**challenges:**

- 提高精准度(precision) 和召回率(recall)
- 分析超过一句话的关系阐述
- 改进bootstrapping techniques，来减小"semantic drift"
- 开发非英语的relation extractors