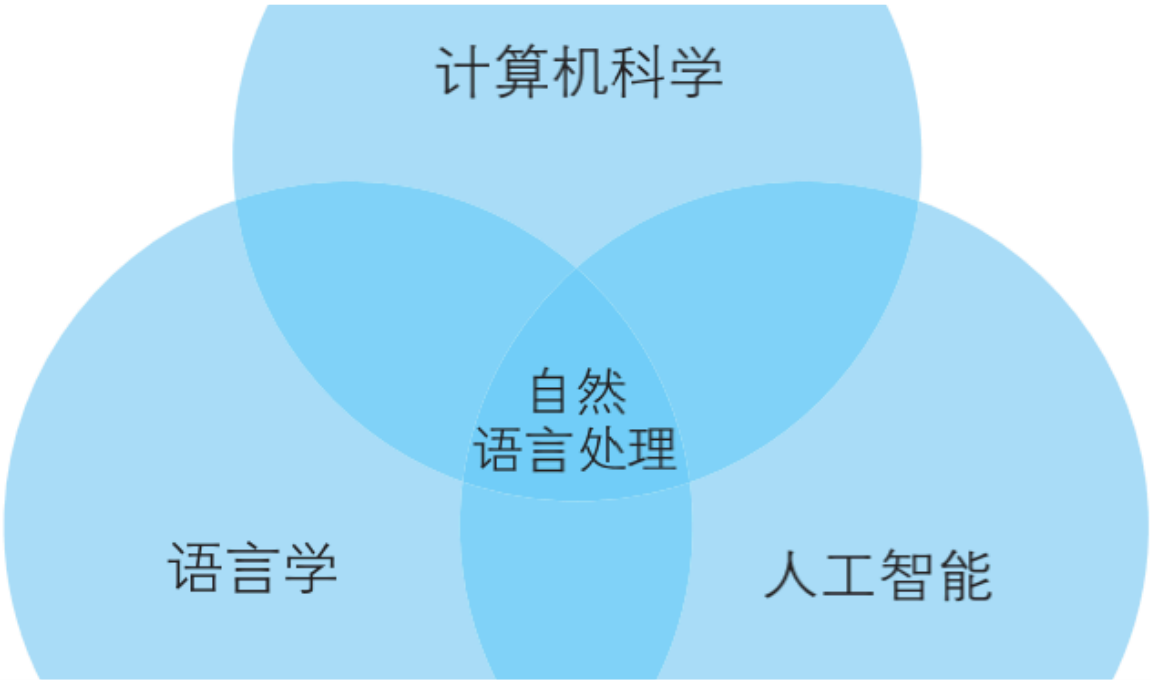
1-导论

自然语言处理是一门涉及语言和计算的跨学科领域，其目的是让计算机能够理解、生成和处理人类语言。

自然语言处理(Natural Language Processing, NLP)是一门融合了计算机科学、人工智能及语言学的交叉学科，它们的关系如下图所示。这门学科研究的是如何通过机器学习等技术，让计算机学会处理人类语言，乃至实现终极目标——理解人类语言或人工智能。



> 现代的自然语言处理更多和深度学习结合。

美国计算机科学家Bill Manaris在《计算机进展》(Advances in Computers)第47卷的《从人机交互的角度看自然语言处理》一文中曾经给**自然语言处理提出了如下的定义:**

> “自然语言处理可以定义为研究在人与人交际中以及在人与计算机交际中的语言问题的一门学科。自然语言处理要研制表示语言能力和语言应用的模型，建立计算框架来实现这样的语言模型，提出相应的方法来不断地完善这样的语言模型，根据这样的语言模型设计各种实用系统，并探讨这些实用系统的评测技术。”

NLP和其他语言的区别

自然语言之所以高度灵活，主要是因为人类语言本身就是一种灵活的符号系统，可以通过不同的词汇、语法结构和语境来表达各种不同的意义和信息。人类语言具有丰富的表达能力，可以表达复杂的概念和情感，同时也可以根据不同的情境进行灵活的调整和变化。

另外，自然语言还受到文化、社会和个体差异的影响，不同的人群在使用语言时会有不同的习惯、风格和表达方式，这也为语言的灵活性增添了更多的维度。

在自然语言处理领域，研究人员通过结合语言学、计算机科学和统计学等多个学科的知识，开发出各种算法和技术来实现对自然语言的自动处理和理解。这些技术可以帮助计算机识别语言中的词汇、句法和语义结构，从而实现文本分析、信息检索、情感分析、语音识别等功能。下面看下自然语言的特点。

词汇量

和编程语言相比，NLP的词汇量巨大。

> 编程语言需要在有限的词汇和语法规则下来描述和执行程序。

例如，C语言中的关键词包括if、else、while、int等，总共有约32个关键词；Python中的关键词包括if、else、while、def等，总共有33个关键词。因此，编程语言中的关键词数量是有限制的，每种编程语言有自己固定的关键词集合。

结构化

编程语言有明确的结构关系，人类认为简单的一句话。计算机却很难理解。

歧义性

编程语言是明确的，如果有歧义，会产生报错。

容错性

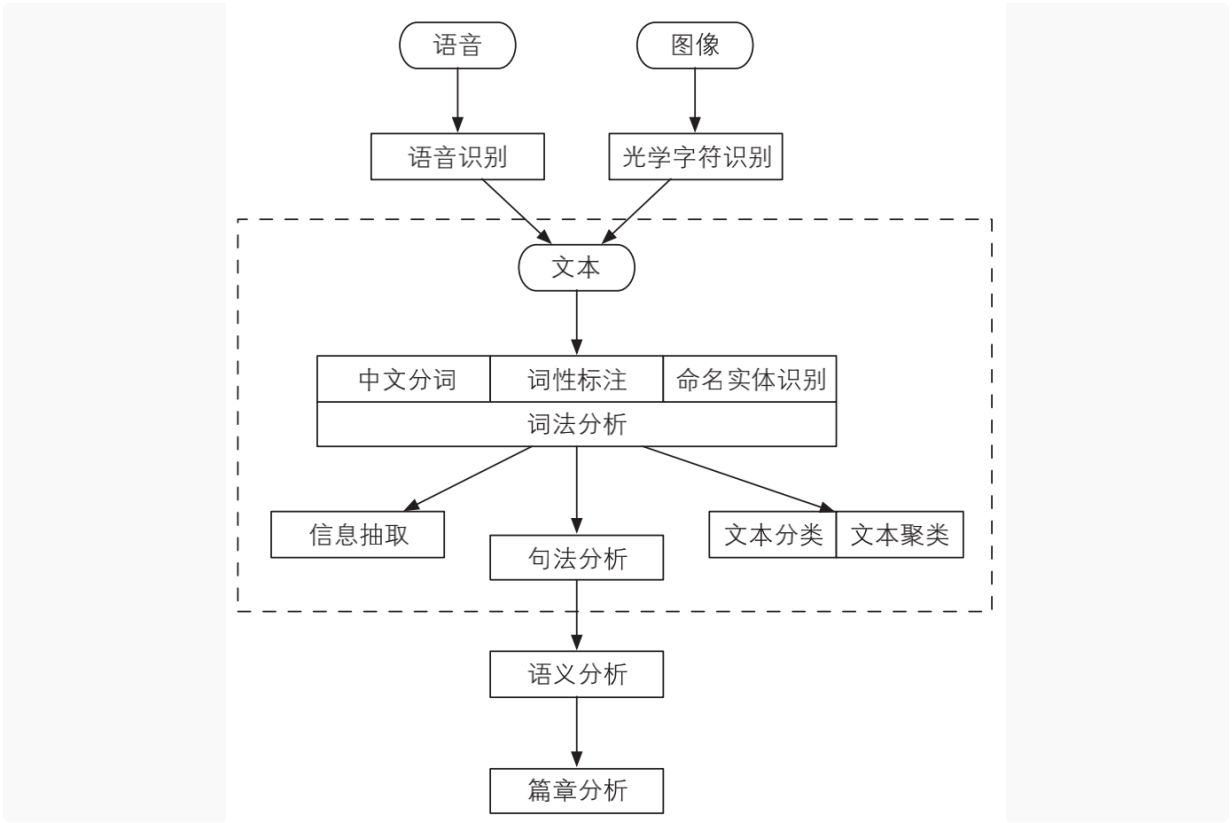
易变性

简略性

总结:

比较	不同	例子
词汇量	自然语言中的词汇比编程语言中的关键词丰富，我们还可以随时创造各种类型的新词	蓝瘦、香菇
结构化	自然语言是非结构化的，而编程语言是结构化的	
歧义性	自然语言含有大量歧义，而编程语言是确定性的	这人真有意思:没意思
容错性	自然语言错误随处可见，而编程语言错误会导致编译不通过	的、地的用法错误
易变性	自然语言变化相对迅速嘈杂一些，而编程语言的变化要缓慢得多	新时代词汇
简略性	自然语言往往简洁、干练，而编程语言就要明确定义	“老地方”不必指出

自然语言处理的层次



语音、图像、文本

自然语言处理系统的输入源一共有3个，即语音、图像与文本。语音和图像这两种形式一般经过识别后转化为文字，转化后就可以进行后续的NLP任务了。

中文分词、词性标注和命名实体识别

这3个任务都是围绕词语进行的分析，所以统称**词法分析**。词法分析的主要任务是将文本分隔为有意义的词语(**中文分词**)，确定每个词语的类别和浅层的歧义消除(**词性标注**)，并且识别出一些较长的专有名词(**命名实体识别**)。对中文而言，词法分析常常是后续高级任务的基础。

信息抽取

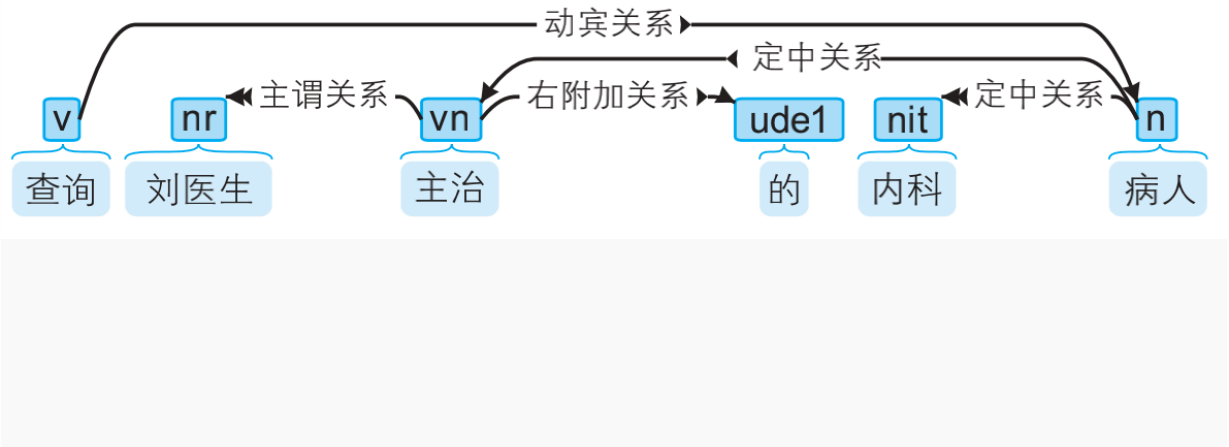
词法分析之后，文本已经呈现出部分结构化的趋势，根据分析出来的每个单词和附有自己词性及其他标签的数据，抽取出一部分有用的信息，关键词、专业术语等，也可以根据统计学信息抽取出更大颗粒度的文本。

文本分类、文本聚类

将文本拆分为一系列词语之后，就可以对文本进行分类和聚类操作，找出相类似的文本。

句法分析

词法分析只能得到零散的词汇信息，计算机不知道词语之间的关系。在一些问答系统中，需要得到句子的主谓宾结构，这就是句法分析得到的结果，如下图所示：



不仅是问答系统或搜索引擎，句法分析还经常应用有基于短语的机器翻译，给译文的词语重新排序。

语义分析和篇章分析

相较于句法分析，语义分析侧重语义而非语法。它包括**词义消歧**(确定一个词在语境中的含义，而不是简单的词性)、**语义角色标注**(标注句子中的谓语与其他成分的关系)乃至**语义依存分析**(分析句子中词语之间的语义关系)。

其他高级任务

自动问答、自动摘要、机器翻译

> 注意，一般认为信息检索(Information Retrieve, IR)是区别于自然语言处理的独立学科，IR的目标是查询信息，而NLP的目标是理解语言。

NLP的流派

基于规则的专家系统

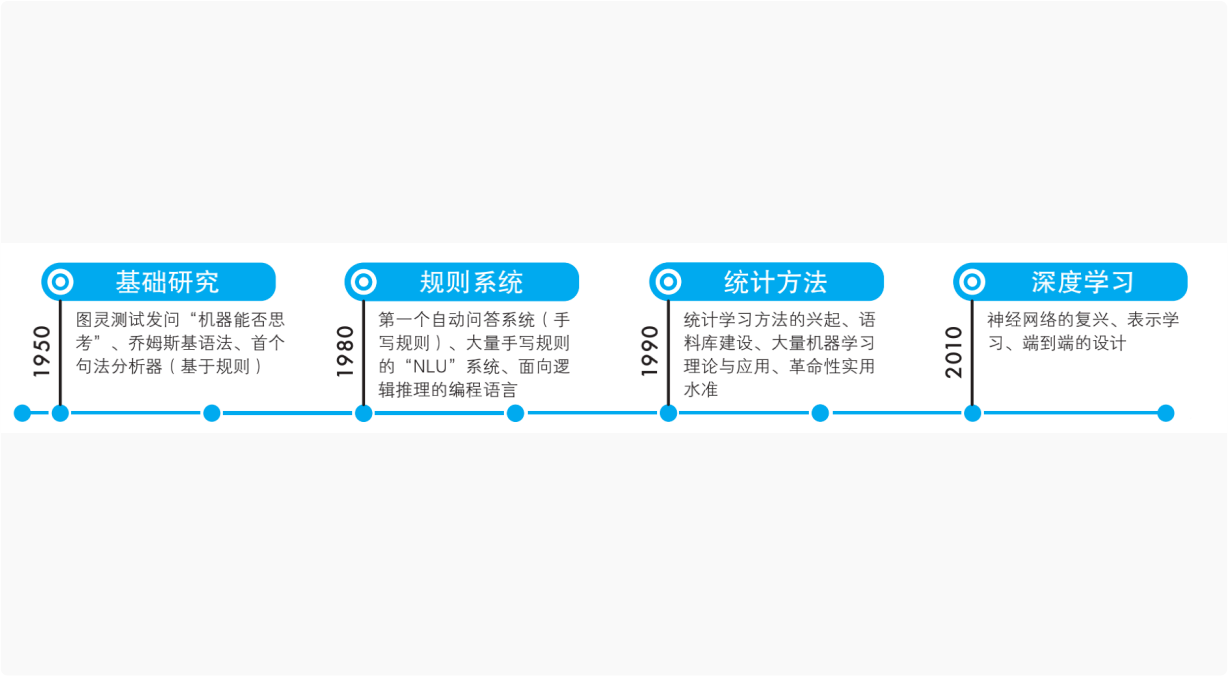
规则，指的是由专家手工制定的确定性流程。专家系统要求设计者对所处理的问题具备深入的理解，并且尽量以人力全面考虑所有可能的情况。它最大的弱点是难以拓展。当规则数量增加或者多个专家维护同一个系统时，就容易出现冲突。

基于统计的学习方法

人们使用统计方法让计算机自动学习语言。所谓“统计”，指的是在语料库上进行的统计。所谓“语料库”，指的是人工标注的结构化文本。

统计学习方法其实是机器学习的别称，而机器学习则是当代实现人工智能的主要途径。

历史



规则与统计

专家系统是一种基于规则的人工智能系统，其核心思想是通过人工制定的规则来模拟专家在特定领域中的决策过程。这种系统曾经备受关注，被认为是人工智能的未来发展方向之一。然而，随着统计学习方法和深度学习的兴起，专家系统逐渐走向没落。

专家系统的主要问题在于其基于规则的设计方式，需要专家们花费大量时间和精力来构建和维护规则库。随着问题复杂度的增加和领域知识的更新，规则库变得庞大且难以维护，导致系统变得僵化和难以适应变化。另外，专家系统对于复杂的、模糊的问题无法提供有效的解决方案，因为规则往往无法覆盖所有可能的情况。

相比之下，基于统计学习方法和深度学习的自然语言处理系统可以从大量的数据中学习规律和模式，无需人工干预。这种数据驱动的方法在处理复杂、多变和模糊的自然语言问题时表现更加优异，为自然语言处理领域带来了革命性的进步。因此，专家系统逐渐被淘汰，取而代之的是基于统计学习和深度学习的新技术和方法。

总的来说，专家系统的没落主要是因为其规则化、僵化、难以维护和适应变化的特点，无法适应发展。

传统方法和深度学习

深度学习在现在大放异彩，从计算机视觉到大语言模型，可以说阳光普照的地方就有神经网络，但是在NLP的基础任务上却意外的发力不大。深度学习用到了大量矩阵运算，需要GPU、TPU等硬件的加速，成本高昂，传统的机器学习反而更适合中小企业。

机器学习

什么是机器学习

美国工程院院士 Tom Mitchell 给过一个更明确的定义，**机器学习** 指的是计算机通过某项任务的经验数据提高了在该项任务上的能力

模型

模型是对现实问题的数学抽象，由一个假设函数以及一系列参数构成。以下就是最简单的模型公式：

$$f(x) = w * x + b$$

其中，w 和b 是函数的参数，而 x 是函数的自变量。不过模型并不包括具体的自变量x ，因为自变量是由用户输入的。自变量 x 是一个特征向量，用来表示一个对象的特征。

特征

- 特征指的是事物的特点转化的数值。
- 如何挑选特征，如何设计特征模板，这称作特征工程。特征越多，参数就越多；参数越多，模型就越复杂。

数据集

样本的集合在机器学习领域称作**数据集**，在自然语言处理领域称作**语料库**。

监督学习

如果数据集附带标准答案 y，则此时的学习算法称作**监督学习**。学习一遍误差还不够小，需要反复学习、反复调整。此时的算法是一种迭代式的算法，每一遍学习称作**一次迭代**。这种在有标签的数据集上迭代学习的过程称作**训练**。

无监督学习

如果我们只给机器做题，却不告诉它参考答案，机器仍然可以学到知识吗？可以，此时的学习称作**无监督学习**，而不含标准答案的数据集被称作**无标注的数据集**。无监督学习一般用于聚类和降维，**降维** 指的是将样本点从高维空间变换成低维空间的过程。

其他类型的机器学习算法

- **半监督学习**：如果我们训练多个模型，然后对同一个实例执行预测，会得到多个结果。如果这些结果多数一致，则可以将该实例和结果放到一起作为新的训练样本，用力啊扩充训练集。这样的算法被称为半监督学习。
- **强化学习**：现实世界中的事物之间往往有很长的因果链：我们要正确地执行一系列彼此关联的决策，才能得到最终的成果。这类问题往往需要一边预测，一边根据环境的反馈规划下次决策。这类算法被称为强化学习。

语料库

中文分词语料库

中文分词语料库指的是，由人工正确切分的句子集合。以著名的1998年《人民日报》语料库为例：

> 先 有 通货膨胀 干扰，后 有 通货 紧缩 叫板。

词性标注语料库

它指的是切分并为每个词语制定一个词性的语料。依然以《人民日报》语料库为例：

> 迈向/v 充满/v 希望/n 的/u 新/a 世纪/n --/w 一九九八年/t 新年/t 讲话/n

这里每个单词后面用斜杠隔开的就是词性标签。

命名实体识别语料库

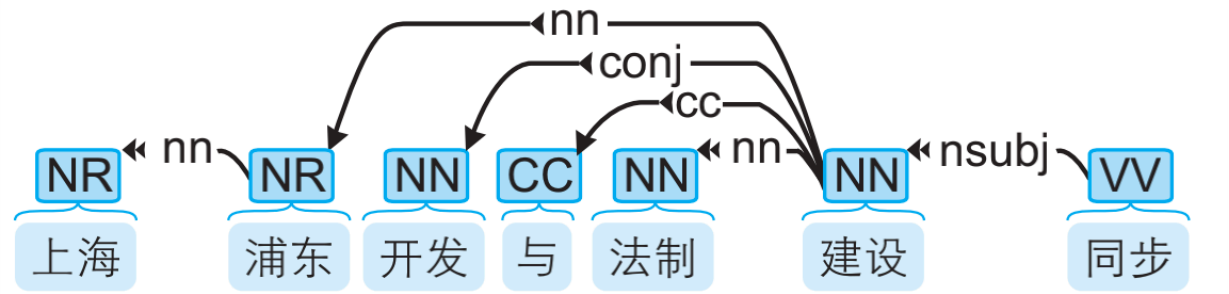
这种语料库人工标注了文本内部制作者关心的实体名词以及实体类别。比如《人民日报》语料库中-共含有人名、地名和机构名3种命名实体:

> 萨哈夫**/nr** 说/v ,/w 伊拉克**/ns** 将/d 同/p **[**联合国**/nt** 销毁**/v** 伊拉克**/ns** 大规模**/b** 杀伤性**/n** 武器**/n** 特别**/a** 委员会**/n] /nt** 继续/v 保持/v 合作/v 。/w

这个句子中的加粗词语分别是人名、地名和机构名。中括号括起来的是复合词，我们可以观察到:有时候机构名和地名复合起来会构成更长的机构名，这种构词法上的嵌套现象增加了命名实体识别的难度。

句法分析语料库

汉语中常用的句法分析语料库有CTB(Chinese Treebank，中文树库)，其中一个句子可视化后如下图所示：



中文单词上面的英文标签标示词性，而箭头表示有语法联系的两个单词，具体是何种联系由箭头上的标签标示。

文本分类语料库

它指的是人工标注了所属分类的文章构成的语料库。

语料库建设

语料库建设指的是构建一份语料库的过程，分为规范制定、人员培训与人工标注这三个阶段。针对不同类型的任务，人们开发出许多标注软件，其中比较成熟的一款是brat，它支持词性标注、命名实体识别和句法分析等任务。

常用工具介绍

1. NumPy

numpy系统是Python的一种开源的数值计算包。 包括：1、一个强大的N维数组对象Array；2、比较成熟的（广播）函数库；3、用于整合C/C++和Fortran代码的工具包；4、实用的线性代数、傅里叶变换和随机数生成函数。numpy和稀疏矩阵运算包scipy配合使用更加方便。

```
conda install numpy
```

2. NLTK

Natural Language Toolkit，自然语言处理工具包，在NLP领域中， 最常使用的一个Python库。

```
conda install nltk
```

3. Gensim

Gensim是一个占内存低，接口简单，免费的Python库，它可以用来从文档中自动提取语义主题。它包含了很多非监督学习算法如：TF/IDF，潜在语义分析（Latent Semantic Analysis，LSA）、隐含狄利克雷分配（Latent Dirichlet Allocation，LDA），层次狄利克雷过程 （Hierarchical Dirichlet Processes，HDP）等。

Gensim支持Word2Vec,Doc2Vec等模型。

```
conda install gensim
```

4. Tensorflow

TensorFlow是谷歌基于DistBelief进行研发的第二代人工智能学习系统。TensorFlow可被用于语音识别或图像识别等多项机器学习和深度学习领域。TensorFlow是一个采用数据流图（data flow graphs），用于数值计算的开源软件库。节点（Nodes）在图中表示数学操作，图中的线（edges）则表示在节点间相互联系的多维数据数组，即张量（tensor）。它灵活的架构让你可以在多种平台上展开计算，例如台式计算机中的一个或多个CPU（戒GPU），服务器，移动设备等等。TensorFlow 最初由Google大脑小组（隶属于Google机器智能研究机构）的研究员和工程师们开发出来，用于机器学习和深度神经网络方面的研究，但这个系统的通用 性使其也可广泛用于其他计算领域。

```
conda install tensorflow
```

5. jieba

“结巴”中文分词：是广泛使用的中文分词工具，具有以下特点：

1）三种分词模式：精确模式，全模式和搜索引擎模式

2）词性标注和返回词语在原文的起止位置（ Tokenize）

3）可加入自定义字典

4）代码对 Python 2/3 均兼容

5）支持多种语言，支持简体繁体

项目地址：https://github.com/fxsjy/jieba

```
pip install jieba
```

6. Stanford NLP

Stanford NLP提供了一系列自然语言分析工具。它能够给出基本的 词形，词性，不管是公司名还是人名等，格式化的日期，时间，量词， 并且能够标记句子的结构，语法形式和字词依赖，指明那些名字指向同样的实体，指明情绪，提取发言中的开放关系等。 1.一个集成的语言分析工具集； 2.进行快速，可靠的任意文本分析； 3.整体的高质量的文本分析; 4.支持多种主流语言; 5.多种编程语言的易用接口; 6.方便的简单的部署web服务。

安装

```
Python 版本stanford nlp 安装
• 1)安装stanford nlp自然语言处理包: pip install stanfordcorenlp
• 2)下载Stanford CoreNLP文件
https://stanfordnlp.github.io/CoreNLP/download.html
• 3)下载中文模型jar包, http://nlp.stanford.edu/software/stanford-chinese-
corenlp-2018-02-27-models.jar,
• 4)把下载的stanford-chinese-corenlp-2018-02-27-models.jar
放在解压后的Stanford CoreNLP文件夹中, 改Stanford CoreNLP文件夹名为stanfordnlp (可选)
• 5)在Python中引用模型:
• from stanfordcorenlp import StanfordCoreNLP
• nlp = StanfordCoreNLP(r'path', lang='zh')
例如:
nlp = StanfordCoreNLP(r'/home/kuo/NLP/module/stanfordnlp/', lang='zh')
```

测试

```
#-*-encoding=utf8-*-
from stanfordcorenlp import StanfordCoreNLP
nlp = StanfordCoreNLP(r'/home/kuo/NLP/module/stanfordnlp/', lang='zh')

fin=open('news.txt','r',encoding='utf8')
fner=open('ner.txt','w',encoding='utf8')
ftag=open('pos_tag.txt','w',encoding='utf8')
for line in fin:
    line=line.strip()
    if len(line)<1:
        continue

    fner.write(" ".join([each[0]+"/"+each[1] for  each in nlp.ner(line) if
len(each)==2 ])+"\n")
    ftag.write(" ".join([each[0]+"/"+each[1] for each in nlp.pos_tag(line) if
len(each)==2 ])+"\n")
fner.close()
ftag.close()
print ("okkkkk")
```

7. Hanlp

HanLP是由一系列模型与算法组成的Java工具包，目标是普及自然 语言处理在生产环境中的应用。HanLP具备功能完善、性能高效、架构 清晰、语料时新、可自定义的特点。　　功能：中文分词 词性标注 命名实体识别 依存句法分析 关键词提取 新词发现 短语提取 自动摘要 文本分类 拼音简繁

Hanlp环境安装

- 1、安装Java:我装的是Java 1.8
- 2、安装Jpype,

```
> conda install -c conda-forge jpype1
>[或者]pip install jpype1
```

- 3、测试是否按照成功:

```
from jpype import *
startJVM(getDefaultJVMPath(), "-ea")
java.lang.System.out.println("Hello World")
shutdownJVM()
```

Hanlp安装

- 1、下载hanlp.jar包: https://github.com/hankcs/HanLP
- 2、下载data.zip: https://github.com/hankcs/HanLP/releases 中 http://hanlp.linrunsoft.com/release/data-for-1.7.0.zip 后解压数据包。
- 3、配置文件
- 示例配置文件:hanlp.properties
- 配置文件的作用是告诉HanLP数据包的位置,只需修改第一行:root=usr/home/HanLP/
- 比如data目录是/Users/hankcs/Documents/data,那么root=/Users/hankcs/Documents/

测试

```
#-*- coding:utf-8 -*-
from jpype import *

startJVM(getDefaultJVMPath(), "-Djava.class.path=/home/kuo/NLP/module/hanlp/
hanlp-1.6.2.jar:/home/kuo/NLP/module/hanlp",
         "-Xms1g",
         "-Xmx1g") # 启动JVM，Linux需替换分号;为冒号:

print("=" * 30 + "HanLP分词" + "=" * 30)
HanLP = JClass('com.hankcs.hanlp.HanLP')
# 中文分词
print(HanLP.segment('你好，欢迎在Python中调用HanLP的API'))
print("-" * 70)
```