Atitit nlp 常见功能 总结

目录

[1. NLP的最基本任务和知识，比如tagging, 各种parsing，coreference, semantic role labeling等等等等 2](#_Toc21873)

[1.1. 中文分词（Character-Based Generative Model） 2](#_Toc19873)

[1.2. 词性标注（TnT 3-gram 隐马） 2](#_Toc6485)

[1.3. NER识别其中的命名实体，最后使用关系识别搜索相近实体间的可能的关系。 2](#_Toc3932)

[1.4. 关系抽取 2](#_Toc16)

[1.5. 情感分析 2](#_Toc23301)

[1.6. 文本分类（Naive Bayes） 3](#_Toc4774)

[1.7. 提取文本关键词（TextRank算法） 3](#_Toc25396)

[1.8. 提取文本摘要（TextRank算法） 3](#_Toc22630)

[1.9. 观点抽取,需要将文本当中有价值的信息提取出来 3](#_Toc7343)

[1.10. 词云 词频计算 3](#_Toc6890)

[2. 其他人物 3](#_Toc23269)

[2.1. 转换成拼音（Trie树实现的最大匹配） 3](#_Toc22300)

[2.2. 繁体转简体（Trie树实现的最大匹配） 3](#_Toc10609)

[2.3. tf，idf 3](#_Toc11066)

[2.4. 同义词 反义词 3](#_Toc4724)

[2.5. NLTK词干提取单词词干提取就是从单词中去除词缀并返回词根。 3](#_Toc21048)

[2.6. LDA主题建模。 4](#_Toc16649)

[2.7. 文本分类（Naive Bayes） 4](#_Toc32494)

[2.8. 转换成拼音（Trie树实现的最大匹配） 4](#_Toc10279)

[2.9. 繁体转简体（Trie树实现的最大匹配） 4](#_Toc15857)

# NLP的底层任务由易到难大致可以分为词法分析、句法分析和语义分析

NLP的底层任务由易到难大致可以分为词法分析、句法分析和语义分析。

## 分词是词法分析（还包括词性标注和命名实体识别）中最基本的任务

可以说既简单又复杂。说简单是因为分词的算法研究已经很成熟了，大部分的准确率都可以达到95%以上

# NLP的最基本任务和知识，比如tagging, 各种parsing，coreference, semantic role labeling等等等等

## 中文分词（[Character-Based Generative Model](http://aclweb.org/anthology//Y/Y09/Y09-2047.pdf" \t "http://www.ppkanshu.com/index.php/post/_blank)）

## 词性标注（[TnT](http://aclweb.org/anthology//A/A00/A00-1031.pdf" \t "http://www.ppkanshu.com/index.php/post/_blank) 3-gram 隐马）

## NER识别其中的命名实体，最后使用关系识别搜索相近实体间的可能的关系。

## **关系抽取**

## 情感分析

* （现在训练数据主要是买卖东西时的评价，所以对其他的一些可能效果不是很好，待解决）

## 文本分类（Naive Bayes）

## 提取文本关键词（[TextRank](http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf" \t "http://www.ppkanshu.com/index.php/post/_blank)算法）

## 提取文本摘要（[TextRank](http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Mihalcea.pdf" \t "http://www.ppkanshu.com/index.php/post/_blank)算法）

## 观点抽取,需要将文本当中有价值的信息提取出来

## 词云 词频计算

# 其他人物

## 转换成拼音（Trie树实现的最大匹配）

## 繁体转简体（Trie树实现的最大匹配）

## tf，idf

## 同义词 反义词

## **NLTK词干提取**单词词干提取就是从单词中去除词缀并返回词根。

（比方说 working 的词干是 work。）搜索引擎在索引页面的时候使用这种技术，所以很多人通过同一个单词的不同形式进行搜索，返回的都是相同的，有关这个词干的页面。

词干提取的算法有很多，但最常用的算法是 Porter 提取算法。NLTK 有一个 PorterStemmer 类，使用的就是 Porter 提取算法。

## LDA主题建模。

## 文本分类（Naive Bayes）

## 转换成拼音（Trie树实现的最大匹配）

## 繁体转简体（Trie树实现的最大匹配）