FudanNLP 说明文档 v0.01

邱锡鹏

微博：http://weibo.com/xpqiu

复旦大学计算机科学技术学院

xpqiu@fudan.edu.cn

2012 年 12 月 15 日

序

本书的部分内容和参考材料来源于互联网，如：维基百科、百度百科以及一些论坛、博客等。后面正文中不再一一引用，在此一并致谢。若有需要特别标注引用的，请联系作者更正。

FudanNLP 项目的主要研发人员还包括：计峰、高文君、赵嘉亿、曹零、赵建双、田乐、缪有栋、刘昭等。此外，复旦大学计算机科学技术学院的部分其他研究生和本科生也贡献了部分代码。在此一并感谢大家的工作。

作者

目录

第一章　前言

> “人工智能之父”图灵 1950 年在《机器能思维吗?》一文中提出著名的“图灵测试”：一个人在不接触对方的情况下，通过一种特殊的方式，和对方进行一系列的问答，如果在相当长时间内，他无法根据这些问题判断对方是人还是计算机。那么就可以认为这个计算机是智能的。

1.1　人工智能与自然语言处理

自然语言处理（Natural Language Processing，NLP）是人工智能和语言学领域的分支学科，主要是研究如何让计算机处理及运用自然语言。自然语言处理广义分为两大部分。**自然语言理解**（Natural Language Understanding，NLU）是指让电脑“懂”人类的语言。**自然语言生成**（Natural Language Generation，NLG）是指把计算机数据转化为自然语言。

2011 年 2 月 14 日到 2 月 16 日期间，IBM 超级计算机 Watson[1] 在美国最受欢迎的智力竞猜节目《Jeopardy》中击败了两名人类选手，最终获得胜利。Watson 的目标是建造一个能与人类回答问题能力匹敌的计算系统。在比赛中，参赛者必须要回答一系列的问题，主要涉及历史，文学，政治，电影，流行文化和科学。这要求计算机具有足够的速度、精确度和置信度，并且能使用人类的自然语言回答问题。比赛题目需要分析人类语言中微妙的含义、讽刺口吻、谜语等，这些通常是人类擅长的方面，一直以来计算机在这方面毫无优势可言。

Watson 的成功也是人工智能的又一次标志性进展，也使得工业界对自然语言处理有了新的认识。

1.2　自然语言处理研究的难点[2]

单词的边界界定在口语中，词与词之间通常是连贯的，而界定字词边界通常使用的办法是取用能让给定的上下文最为通顺且在文法上无误的一种最佳组合。在书写上，汉语也没有词与词之间

[1] http://www-03.ibm.com/innovation/us/watson/index.html

[2] 参考http://zh.wikipedia.org/wiki/

的边界。词义的消歧许多字词不单只有一个意思，因而我们必须选出使句意最为通顺的解释。句法的模糊性自然语言的文法通常是模棱两可的，针对一个句子通常可能会剖析 (Parse) 出多棵剖析树 (Parse Tree)，而我们必须要仰赖语意及前后文的资讯才能在其中选择一棵最为适合的剖析树。有瑕疵的或不规范的输入例如语音处理时遇到外国口音或地方口音, 或者在文本的处理中处理拼写, 语法或者光学字符识别 (OCR) 的错误。语言行为与计划句子常常并不只是字面上的意思；例如，“你能把盐递过来吗”，一个好的回答应当是把盐递过去；在大多数上下文环境中，“能”将是糟糕的回答，虽说回答“不”或者“太远了我拿不到”也是可以接受的。再者，如果一门课程去年没开设，对于提问“这门课程去年有多少学生没通过?”回答“去年没开这门课”要比回答“没人没通过”好。

1.3 当前自然语言处理研究的发展趋势

第一，传统的基于句法 -语义规则的理性主义方法受到质疑，随着语料库建设和语料库语言学的崛起，大规模真实文本的处理成为自然语言处理的主要战略目标。

第二，统计数学方法越来越受到重视，自然语言处理中越来越多地使用机器自动学习的方法来获取语言知识。

第三，浅层处理与深层处理并重，统计与规则方法并重，形成混合式的系统。

第四，自然语言处理中越来越重视词汇的作用，出现了强烈的“词汇主义”的倾向。词汇知识库的建造成为了普遍关注的问题。

统计自然语言处理

统计自然语言处理运用了推测学、机率、统计的方法来解决上述，尤其是针对容易高度模糊的长串句子，当套用实际文法进行分析产生出成千上万笔可能性时所引发之难题。处理这些高度模糊句子所采用消歧的方法通常运用到语料库以及马可夫模型（Markov models）。统计自然语言处理的技术主要由同样自人工智能下与学习行为相关的子领域：机器学习及资料采掘所演进而成。

1.4 统计机器学习

机器学习(Machine Learning) 是从观测数据（样本）中寻找规律，或模拟人类的学习行为，并利用掌握的规模对未知或无法观测的数据进行预测。Mitchell [1997] 在他的《机器学习》一书中定义机器学习时提到，“机器学习是对能通过经验自动改进的计算机算法的研究”。（Machine Learning is the study of computer algorithms that improve automatically through experience.）

狭义地讲，机器学习是给定一些观测样本 $(x_i, y_i), 1 \leq i \leq N$ （其中 x_i 是样本，y 是类别标签），让计算机自动寻找一个模型 $\hat{y} = f(\phi(x))$，对于所有已知或未知的 (x, y)，使得 $\hat{y}$ 和 y 尽可能地一致。

机器学习可以分为下面几类：

监督学习 也称分类，如果 y 是离散的，机器学习问题就成了分类问题。这时，$f(\cdot)$ 也称为**分类器**。

半监督学习 如果 y 是连续的，就是回归问题。

回归问题 如果 y 是连续的，就是回归问题。

聚类问题 如果给定的数据不包含 y，就是聚类问题，也称无监督学习问题。

机器学习有很多优秀的参考文献，这里我们推荐如下：

1. R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2nd edition, 2001. ISBN 0471056693

2. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001

3. M.I. Jordan. *Learning in Graphical Models*. Kluwer Academic Publishers, 1998

4. I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011

1.5 机器学习基本概念

要了解机器学习，首先需要了解一些基本概念，比如“数据”，“样本”，“特征”，“数据集”，“分类器”、“学习算法”，“评价方法”等。

1.5.1 数据

在计算机科学中，**数据**是指所有能计算机程序处理的对象的总称，可以是数字、字母和符号等。在不同的任务中，表现形式不一样。比如：在文本分类中，数据可以是一篇文档，也可以是一句话。

样本是按照一定的抽样规则从全部数据中取出的一部分数据，是实际观测得到的数据。我们用 x 表示抽象概念的样本，可以有多种表示形式，比如字符串、数组、集合等。

1.5.2 样本特征

在机器学习中，为了更好地表示样本的属性，一般将样本表示成代数形式，称为**样本特征**，我们用 $\phi(x)$。样本特征可以是一维或多维向量，$\phi(x) \in \mathbb{R}^k$，k 是向量维数。在自然语言处理中，样本特征是非常稀疏的，可以用稀疏向量来表示。

1.5.3 数据集

数据集，在自然语言处理中也称为**语料库**，是指多个样本组成的集合。我们用 X 表示，$X = \{x_1, \cdots, x_N\}$。

1.6 FudanNLP 简介

FudanNLP 主要是为中文自然语言处理而开发的工具包，也包含为实现这些任务的机器学习算法和数据集。本工具包及其包含数据集使用 LGPL3.0 许可证。FudanNLP 是基于 Java 的开源项目，利用统计机器学习和规则方法来处理中文自然语言处理的经典问题，比如：分词、词性标注、句法分析、实体名识别等。

1.6.1 组织结构

FudanNLP 的组织结构可分为 5 层，如图1.1所示。

1. 最底层的操作。比如数据结构、数据表示、数据类型、数据预处理、特征转换等。
2. 结构化机器学习和人工规则框架。涉及到特征抽取，学习算法、推理算法和模型建立等。
3. 可插拔的具体算法。比如分类、聚类、半监督和优化等。
4. 中文自然语言处理应用，比如分词、句法分析等。
5. 系统应用，比如文本分类、主题词抽取等。

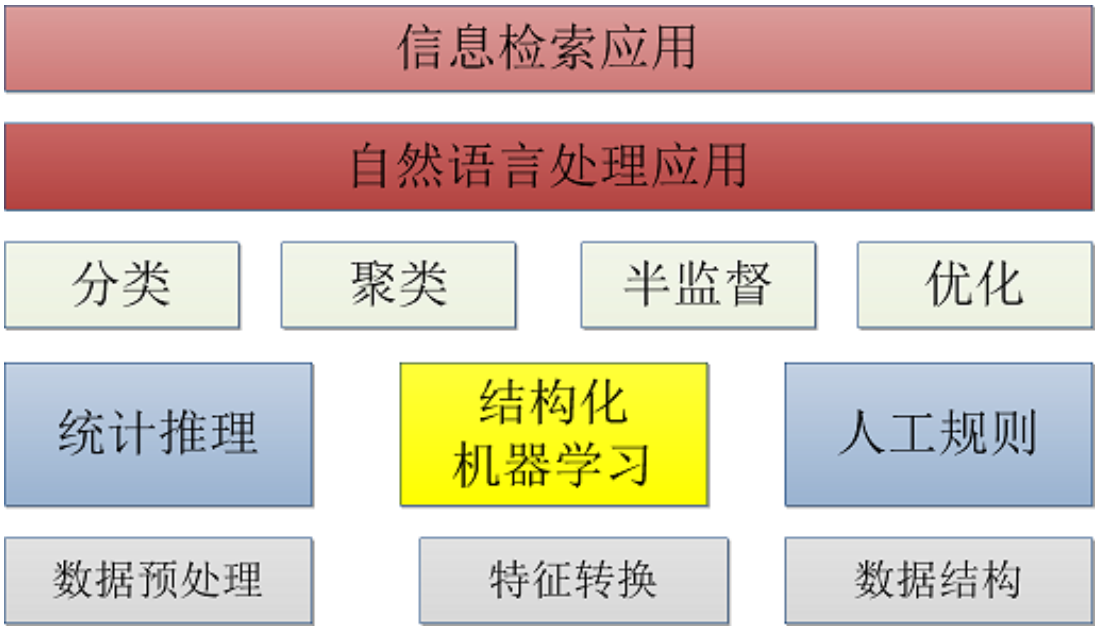


图 1.1: FudanNLP 组织结构图

1.6.2 FudanNLP 命令行调用使用示例

下载 FudanNLP 工具包后，可以通过命令行调用测试主要功能。[3]

```
@echo 分词实例
java -classpath fudannlp.jar;lib/commons-cli-1.2.jar;lib/trove.jar; edu.fudan.nlp.cn.tag.CWSTagger -s models/seg.m "自然语言是人类交流和思维的主要工具，是人类智慧的结晶。"

@echo 词性标注实例
java -classpath fudannlp.jar;lib/commons-cli-1.2.jar;lib/trove.jar; edu.fudan.nlp.cn.tag.POSTagger -s models/pos.m "自然语言是人类交流和思维的主要工具，是人类智慧的结晶。"
```

[3]最新示例请参考发布包内“主要功能命令行测试.cmd”的调用示例。

```
6
7 %@echo 句法分析实例
8 java -classpath fudannlp.jar;lib/commons-cli-1.2.jar;lib/trove.
      jar; edu.fudan.nlp.cn.tag.NERTagger -s models/seg.m models/
      pos.m "詹姆斯·默多克和丽贝卡·布鲁克斯鲁珀特·默多克旗下的美国小报
      《纽约邮报》的职员被公司律师告知，保存任何也许与电话窃听及贿赂有关的文
      件。"
9 %@pause>nul
10
11 ====System Output====
12 自然 语言 是 人类 交流 和 思维 的 主要 工具 ， 是 人类 智慧
      的 结晶 。
13 自然/AD 语言/NN 是/VC 人类/NN 交流/NN 和/CC 思维/NN 的/DEG
      主要/JJ 工具/NN ，/PU 是/VC 人类/NN 智慧/NN 的/DEG 结
      晶/NN 。/PU
14 {詹姆斯·默多克 = 人名, 丽贝卡·布鲁克斯鲁珀特·默多克 = 人名, 纽约 = 地名, 美
      国 = 地名}
```

1.6.3 FudanNLP 目录组织结构

表 1.1: FudanNLP 目录组织结构

目录	描述
/src	主要功能代码，主目录。
/example	对外 API 使用示例代码。
/example-data	使用示例需要的数据。
/apps	基于 FudanNLP 的应用
/models	必须的模型文件或知识文件
/docs	项目文档

1.6.4 FudanNLP Java 包组织结构

机器学习相关的 Java 包：

表 1.2: FudanNLP 目录组织结构

目录	描述
edu.fudan.ml.types	数据类型。
edu.fudan.ml.data	数据读取器。通过 Reader 接口将原始数据读入，并生产 Instance 对象。Reader 为一个迭代器，依次返回一个 Instance 对象。
edu.fudan.ml.classifier	机器学习。包括分类器和训练器两部分。按照结构化学习的思想分为：特征生成、损失函数和统计推理三部分。
edu.fudan.ml.feature .generator	特征生成。
edu.fudan.ml.loss	损失函数
edu.fudan.ml.inf	统计推理。这里对于离散的类别，使用简单遍历计算，然后求最大值得方法。

自然语言处理基础相关的 Java 包：

表 1.3: 自然语言处理基础相关的 Java 包

目录	描述
edu.fudan.nlp.pipe	数据特征变换器。这里进行数据不同形式表示之间的转换。比如从文本到向量的转换。
edu.fudan.nlp.parser	句法分析包。
edu.fudan.nlp.tag	序列标注任务训练等。
edu.fudan.nlp.cn	分词、词性标注、实体名识别、以及中文处理一些规则方法。

自然语言处理应用相关的 Java 包：

表 1.4: 自然语言处理应用相关的 Java 包

目录	描述
edu.fudan.nlp.app.keyword	关键词抽取。
edu.fudan.nlp.app.tc	文本分类器。

1.6.5 FudanNLP 总体流程

FudanNLP 项目大概结构组织如下：

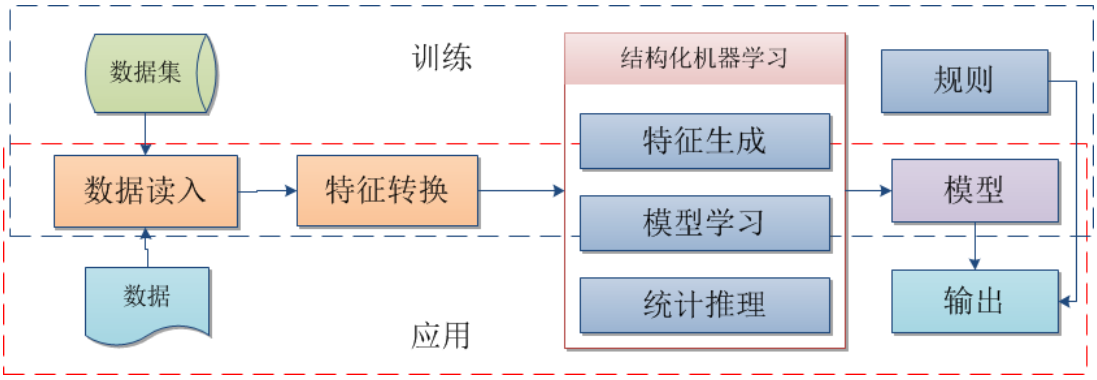


图 1.2: FudanNLP 总体流程图

第二章　自然语言处理基础

自然语言是人类交流和思维的主要工具，是人类智慧的结晶。人类使用的语言都会被视为“自然”语言，以相对于如编程语言等为计算机而设的“人造”语言。

自然语言和程序语言的一个重要区别是自然语言普遍存在不确定性或歧义性 [Chomsky and Miller, 1963]。

2.1　自然语言处理

自然语言处理 (Natural Language Processing, NLP) 是计算机科学领域与人工智能领域中的一个重要部分，甚至核心部分，也是人工智能中最为困难的问题之一。它研究能实现人与计算机之间用自然语言进行有效通信的各种理论和方法。自然语言处理是一门融语言学、计算机科学、数学于一体的科学。它与语言学的研究有着密切的联系，但又有重要的区别。自然语言处理并不是一般地研究自然语言，而在于研究能有效地实现自然语言通信的软件系统，特别是大规模的智能处理。

从广义上讲，自然语言处理可分为两部分：自然语言理解和自然语言生成。**自然语言理解**是使计算机能理解自然语言文本的意义，而**自然语言生成**是让计算机能以自然语言文本来表达给定的意图、思想等。

若希望全面了解自然语言处理，可以参考下列文献或相对应的中文版：

1. J. Allen. *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA, 1995

2. C.D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999

3. D. Jurafsky, J.H. Martin, A. Kehler, K. Vander Linden, and N. Ward. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, volume 2. Prentice Hall New Jersey, 2000

2.2 中文

中文（Chinese），也称为汉语，按照字型分为简体中文和繁体中文。按照时间可分为古代汉语和现代汉语。本书所述的中文在不做特别说明是一般指现代汉语。

中文的语素绝大部分是单音节的。语素和语素可以组合成词（比如：吃 + 饭 $\Longrightarrow$ 吃饭）。有的语素本身就是词（手、洗），有的语素本身不是词, 只能跟别的语素一起组成复合词（民 $\rightarrow$ 人民 失 $\rightarrow$ 丧失）。现代汉语里双音节词占的比重最大。大部分双音词都是按照上面提到的复合方式造成的。有些语素虽然在现代汉语里不能作为一个词单独用，但是有时候在借用古汉语的词句时，也偶尔作为词来使用。

下面几节简单介绍下中文自然语言处理中的主要研究问题。

2.3 中文分词

词是最小的能够独立活动的有意义的语言成分，英文单词之间是以空格作为自然分界符的，而汉语是以字为基本的书写单位，词语之间没有明显的区分标记，因此，中文分词是中文信息处理的基础与关键。

中文由于继承自古代汉语的传统，词语之间没有分隔。古代汉语中除了连绵词和人名地名等，词通常就是单个汉字，所以当时没有分词书写的必要。现代汉语的基本表达单元为“词”，以双字或多字词居多，一个字不再等同于一个词。

2.3.1 FudanNLP 中文分词 API

FudanNLP 中中文分词对应的类为 edu.fudan.nlp.cn.tag.CWSTagger。我们可以通过如下 API 来调用[1]：

```
1 import edu.fudan.nlp.cn.tag.CWSTagger;
2
3 CWSTagger tag = new CWSTagger("./models/seg.m");
4
5 String str = "他说的确实在理";
6 String s = tag.tag(str);
7 System.out.println(s);
8
9 ======System Output======
10 他 说 的 确实 在理
```

目前 FudanNLP 还支持自定义词典来辅助机器学习分词算法的不足。

```
1 import edu.fudan.nlp.cn.tag.CWSTagger;
2
```

[1]更多使用示例请参考最新版发布包内 example 文件夹里的示例代码 ChineseWordSegmentation.java

```
CWSTagger tag = new CWSTagger("./models/seg.m");
String str = "高级数据挖掘很难。";
String s = tag.tag(str);
System.out.println(s);

//注：词典里只能有中文字符，英文与数字不支持
//设置临时词典
ArrayList<String> al = new ArrayList<String>();
al.add("数据挖掘");
//false 表示词典是没有歧义的，没有包含关系的（比如“数据”与“数据库”）
//系统会自动设为 false
tag.setDictionary(new Dictionary(al, false));
s = tag.tag(str);
System.out.println(s);
=====System Output=====
高级 数据 挖掘 很 难 。
高级 数据挖掘 很 难 。
```

2.4 中文词性标注

词性（Part-of-Speech, POS）指作为划分词类的根据的词的特点。现代汉语的词可以粗分为 12 类。

实词 名词、动词、形容词、数词、量词和代词

虚词 副词、介词、连词、助词、拟声词和叹词

但这样的划分对于后续的语义分析远远不够。因此不同的语料构建者都会对词性进行更细的划分。比如“名词”可以再分为“一般名词”、“专用名词”、“抽象名词”和“方位名词”等。“专用名词”可用再分为“人名”、“地名”、“机构名”等。

词性标注 (POS Tagging) 是给句子中每个词标记出最合适的词性。中文和英语的一个重要区别是中文没有词性变化。中文的每个词会有多种词性（比如“希望”即是名词又是动词）。但特定的使用场合下，比如一个句子中，每个词都有唯一确定的词性。即在若干词性候选项中选择一个合适的词性。

词性标注规范有很多种，比如 Chinese TreeBank （CTB）[2]标准。

比如下面例子中，“DT”，“PN”，“VV”，“AS”，“JJ”，“DEG”，“NN”和“PU”都是 CTB 定义的词性标签。

去年/DT 他/PN 取得/VV 了/AS 可喜/JJ 的/DEG 进步/NN 。/PU

[2]http://www.cis.upenn.edu/ chinese/ctb.html

具体规范见下面文档。F. Xia. *The part-of-speech tagging guidelines for the penn chinese treebank (3.0)*, 2000

在 FudanNLP 中，词性分为：动词、能愿动词、趋向动词、把动词、被动词、形谓词、形容词、副词、名词、方位词、人名、地名、机构名、时间短语、邮件、网址、型号名、实体名、疑问代词、指示代词、人称代词、量词、介词、数词、惯用词、限定词、连词、叹词、序数词、省略词、语气词、结构助词、时态词、标点、拟声词、表情词等。

FudanNLP 中词性标注对应的类为 edu.fudan.nlp.cn.tag.POSTagger。我们可以通过如下 API 来调用[3]：

2.5 命名实体识别

命名实体识别（NE）是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名和其它实体名等。

FudanNLP 中中文分词对应的类为 edu.fudan.nlp.cn.tag.NERTagger。我们可以通过如下 API 来调用[4]：

2.6 句法分析

句法分析 (Parsing) 是指对句子中的词语语法功能进行分析。

FudanNLP 提供了依存句法分析功能。对应的类为 edu.fudan.nlp.parser.dep.JointParser。FudanNLP 的依存关系类型包括：核心词、主语、补语、宾语、定语、状语、并列、同位语、数量、介宾、连动、疑问连动、兼语、关联、重复、标点、的字结构、地字结构、得字结构、语气、时态等。

我们可以通过如下 API 来调用[5]：

2.7 指代消解

共指（Coreference）和指代（Anaphora）是自然语言表达中的常见现象。在语言学中，为了避免已经出现的字词重复出现在文章的句子上，导致语句结构过于赘述和语意不够清晰，所以使用代名词、普通名词或缩略语来代替已经出现过的字、词、短语或句子。共指是指两个词 (或短语) 都指向同一对象 (或实体)。指代是指一个指代词对先行词的依赖关系，比如，等价关系、上下位关系、整体和部分关系等。一般意义上讲，共指可以脱离上下文存在。而指代是在一个小的范围内存在。

例如：下面一段文字中，“北极熊”、“白熊”、“肉食动物”和“熊”都指向同一个实体，属于共指现象。而“那里”和“北极”是指代关系。

北极熊又称白熊，是陆上最庞大的肉食动物。这种熊在北极生活，那里非常寒冷。

近年来，共指消解 (Coreference Resolution) 和指代消解（anaphora resolution）的研究受到了格外的关注，2000 年开始的 ACE(Automatic Content Extraction) 评测会议中共指消解也是重要内

[3]更多使用示例请参考最新版发布包内 example 文件夹里的示例代码 PartsOfSpeechTag.java
[4]更多使用示例请参考最新版发布包内 example 文件夹里的示例代码 NamedEntityRecognition.java
[5]更多使用示例请参考最新版发布包内 example 文件夹里的示例代码 DepParser.java

容之一。中文的共指消解研究开始于二十世纪末。中文共指消解评测开始于 2003 年 ACE 会议。

这里我们主要考虑指代消解问题。王厚峰 [2002]

按照指向，可以分为回指和预指。回指就是代词的先行语在代词前面，预指就是代词的先行语在代词后面。按照指代的类型可以分为三类：人称代词、指示代词、有定描述、省略、部分－整体指代、普通名词短语。这些类别中前四个都是和语言学息息相关的。

FudanNLP 提供了指代消解功能。对应的类为 edu.fudan.nlp.cn.anaphora.Anaphora。我们可以通过如下 API 来调用[6]：

2.8 语义分析

2.9 其他

语言生成

语义消岐

语音识别语音合成

文本分类和聚类信息检索和过滤信息抽取问答系统对话系统机器翻译情感分析文本挖掘

知识工程方法　　后来兴起过一段时间的知识工程的方法则借助于专业人员的帮助，为每个类别定义大量的推理规则，如果一篇文档能满足这些推理规则，则可以判定属于该类别。这里与特定规则的匹配程度成为了文本的特征。由于在系统中加入了人为判断的因素，准确度比词匹配法大为提高。但这种方法的缺点仍然明显，例如分类的质量严重依赖于这些规则的好坏，也就是依赖于制定规则的“人”的好坏；再比如制定规则的人都是专家级别，人力成本大幅上升常常令人难以承受；而知识工程最致命的弱点是完全不具备可推广性，一个针对金融领域构建的分类系统，如果要扩充到医疗或社会保险等相关领域，则除了完全推倒重来以外没有其他办法，常常造成巨大的知识和资金浪费。统计学习法　　后来人们意识到，究竟依据什么特征来判断文本应当隶属的类别这个问题，就连人类自己都不太回答得清楚，有太多所谓“只可意会，不能言传”的东西在里面。人类的判断大多依据经验以及直觉，因此自然而然的会有人想到何让机器像人类一样自己来通过对大量同类文档的观察来自己总结经验，作为今后分类的依据。这便是统计学习方法的基本思想。　　统计学习方法需要一批由人工进行了准确分类的文档作为学习的材料（称为训练集，注意由人分类一批文档比从这些文档中总结出准确的规则成本要低得多），计算机从这些文档中挖掘出一些能够有效分类的规则，这个过程被形象的称为训练，而总结出的规则集合常常被称为分类器。训练完成之后，需要对计算机从来没有见过的文档进行分类时，便使用这些分类器来进行。这些训练集包括 sogou 文本分类分类测试数据、中文文本分类分类语料库，包含 Arts、Literature 等类别的语料文本、可用于聚类的英文文本数据集、网易分类文本分类文本数据、tc-corpus-train(语料库训练集，适用于文本分类分类中的训练)、2002 年中文网页分类训练集 CCT2002-v1.1 等。　　现如今，统计学习方法已经成为了文本分类领域绝对的主流。主要的原因在于其中的很多技术拥有坚实的理论基础（相比之下，知识工程方法中专家的主观因素居多），存在明确的评价标准，以及实际表现良好。

[6]更多使用示例请参考最新版发布包内 example 文件夹里的示例代码 AnaphoraResolution.java

第三章　监督学习算法

利用一组已知类别的样本调整分类器的参数，使其达到所要求性能的过程，也称为监督训练或有教师学习。正如人们通过已知病例学习诊断技术那样，计算机要通过学习才能具有识别各种事物和现象的能力。

给定一些观测样本 $(x_i, y_i), 1 \leq i \leq N$ （其中 x_i 是样本，y 是类别标签），让计算机自动寻找一个模型 $\hat{y} = f(\phi(x))$，对于所有已知或未知的 (x, y)，使得 $\hat{y}$ 和 y 尽可能地一致。

在监督学习中，我们假设 $y = \{1, \cdots, C\}$ 共 C 个类别。

$f(\cdot)$ 的形式是多种多样的。

当 $f(\cdot)$ 为线性时，

$$f(\phi(x)) = \underset{y=1}{\overset{C}{\operatorname{argmax}}} \mathbf{w}_y \cdot \phi(x), \tag{3-1}$$

这里 $\mathbf{w}$ 为参数。

$f(\cdot)$ 也可以表示为概率形式，

$$f(\phi(x)) = \underset{y=1}{\overset{C}{\operatorname{argmax}}} p(y|\phi(x)). \tag{3-2}$$

公式3-2可以用贝叶斯公式进行扩展：

$$\begin{aligned} f(\phi(x)) &= \underset{y=1}{\overset{C}{\operatorname{argmax}}} p(y|\phi(x)) && (3\text{-}3) \\ &= \underset{y=1}{\overset{C}{\operatorname{argmax}}} \frac{p(\phi(x)|y)p(y)}{\sum_y p(\phi(x)|y)p(y)} && (3\text{-}4) \\ &\propto \underset{y=1}{\overset{C}{\operatorname{argmax}}} p(\phi(x)|y)p(y). && (3\text{-}5) \end{aligned}$$

如果我们假设 $p(\phi(x)|y)$ 是高斯分布 $p(\phi(x)|y) \sim N(\mu_y, \sigma_y^2)$，并且 y 是均匀分布，公式3-2可以写为：

$$f(\phi(x)) \propto \underset{y=1}{\overset{C}{\operatorname{argmax}}} g(\phi(x); \mu_y, \sigma_y^2), \tag{3-6}$$

这里 $g(\phi(x); \mu_y, \sigma_y^2)$ 是在类 y 中 $\phi(x)$ 的正态分布函数。

3.1 训练算法

在公式和中，我们分别需要设定参数 w_y 和 μ_y, σ_y 的值，这里 $y = 1, \cdots, C$。

训练算法就是给定一组样本，我们计算这些参数的方法。本节我们简要介绍下常用的机器学习算法为，比如决策树，朴素贝叶斯，神经网络，支持向量机，线性最小平方拟合，kNN，最大熵等。

3.1.1 两类感知器

这里我们先考虑两类分类问题。对于一个样本 (x, y)，$\phi(x)$ 是特征向量。y 是类别标签。对于两个问题来说，我们可以令 y 等于 1 或 -1。

我们需要找到一个映射函数 $f(\phi(x))$，把特征向量 $(\phi(x)$ 映射为一个二元值 $\{1, -1\}$。

在自然语言处理中，最常用也非常有效的映射函数是如下的线性函数：

$$f(\phi(x)) = \begin{cases} 1 & \text{if } \phi(x) \cdot \mathbf{w} + b > 0 \\ -1 & \text{else} \end{cases}, \tag{3-7}$$

这里，$\mathbf{w}$ 是权重向量，b 表示常数偏置。$\mathbf{w}, b$ 就是我们需要学习的参数。

我们也可以采用等价的简化表示方式：

$$f(\phi(x)) = \begin{cases} 1 & \text{if } \widehat{\phi(x)} \cdot \hat{\mathbf{w}} > 0 \\ -1 & \text{else} \end{cases}, \tag{3-8}$$

这里，$\hat{\mathbf{w}} = [\mathbf{w}; b]$，$\widehat{\phi(x)} = [\phi(x); 1]$。在后面的分类器描述中，我们都采用简化的表示方法，并直接写为 $\mathbf{w}$ 和 $\phi(x)$。

这里我们介绍一种简单有效的在线学习算法，感知器算法 [Rosenblatt, 1958]。感知器算法通过多次迭代来更新参数 $\mathbf{w}$。每轮迭代 t 中，从样本集中选取样本，如果分类正确，参数不作修改，如果分类错误，这通过下面公式更新 $\mathbf{w}$，

$$\mathbf{w}_t = \mathbf{w}_{t-1} + \rho y_t \phi(x_t) \tag{3-9}$$

这里 ρ 是一个常数，控制每次迭代中参数修正的幅度。

两类感知器算法的训练过程如算法3.1所示。

算法 3.1: 两类感知器算法

输入: 训练集: $(\phi(x)_i, y_i), i = 1, \cdots, N$，参数：$\rho, T$

输出: $\mathbf{w}$

```
w_0 ← 0 ;
for t = 1 ··· T do
    选取一个样本 (x_t, y_t);
    预测：ŷ_t = f(φ(x_t));
    if ŷ_t != y_t then
        w_t = w_{t-1} + ρ y_t φ(x_t) ;
    end
end
return w_T ;
```

如果存在一个正的常数 γ 和权重向量 $\mathbf{w}$，对所有 i 的满足 $y_i \mathbf{w} f(\phi(x)_i) > \gamma$，训练集 $\mathcal{D}$ 就被叫被做**线性可分**的。Novikoff 证明如果训练集是线性可分的，那么感知器算法可以在有限次迭代后收敛 [Cristianini and Shawe-Taylor, 2000]，迭代次数最多 $\left(\frac{2R}{\gamma}\right)^2$，其中 R 为输入向量的最大平均值。然而，如果训练集不是线性分隔的，那么这个算法则不能确保会收敛。

为了避免过拟合，可以使用Collins [2002] 提出的平均策略。

3.2 多类感知器

对于处理多类问题（$C > 2$）时，可以将多类问题转换成两类问题。一般有两个转换方法。一种转换方法是构建 C 个一对多的分类器，另一种转换方法是 $C(C-1)/2$ 个两两分类器。这两个方法都有一定的缺点：在得到两类分类结果后，还需要通过投票方法进一步确定多类的分类结果。

更有效的方法是直接建立多类分类器。比如建立如下线性多类分类器：

$$f(\phi(x)) = \underset{y=1}{\overset{C}{\operatorname{argmax}}}\, w_y \cdot \phi(x), \tag{3-10}$$

这里 w 为参数。

在公式3-12中，y 为离散变量，但是在很多场合，类别 y 可以是更复杂的表示，比如多标签、层次化以及结构化等形式。为了更好的描述这些情况，公式3-12可以改写为如下形式：

$$f(x) = \underset{y=1}{\overset{C}{\operatorname{argmax}}}\, f(\phi(x,y)) = \underset{y=1}{\overset{C}{\operatorname{argmax}}}\, \mathbf{w} \cdot \phi(x,y), \tag{3-11}$$

这里 $\phi(x,y)$ 是包含了 x 和 y 信息的特征向量，$\mathbf{w} = (\mathbf{w}_1, \cdots, \mathbf{w}_C)$。

当 y 为离散变量时（$y = \{1, \cdots, C\}$），$y = c$ 可以表示为向量：$(\overbrace{0, \cdots, 0}^{c-1}, 1, \overbrace{0, \cdots, 0}^{C-c})^T$。

在多标签分类时，$y = \{c, k\}$ 可以表示为向量：$(\cdots, \overbrace{1}^{c}, \cdots, 0, \cdots, \overbrace{1}^{k}, \cdots)^T$。

$\phi(x,y)$ 可以看成是 $\phi(x)$ 和 y 向量表示的笛卡尔积。

$$\phi(x, y=c) = \begin{pmatrix} \vdots \\ 0 \\ \phi(x) \\ 0 \\ \vdots \end{pmatrix} \leftarrow \text{第 } c \text{ 个位置} \tag{3-12}$$

多类感知器算法的训练过程如算法3.2所示。

算法 3.2: 多类感知器算法

输入: 训练集: $(\phi(x)_i, y_i), i = 1, \cdots, N$，参数：$\rho, T$

输出: $\mathbf{w}$

```
w_0 ← 0 ;
for t = 1 ··· T do
    选取一个样本 (x_t, y_t);
    预测：ŷ_t = f(x_t);
    if ŷ_t != y_t then
        w_t = w_{t-1} + ρ(φ(x_t, y_t) − φ(x_t, ŷ_t));
    end
end
return w_T ;
```

3.3 决策树算法

决策树算法是一种典型的分类方法，

如何构造精度高、规模小的决策树是决策树算法的核心内容。决策树构造可以分两步进行。

决策树的生成 由训练样本集生成决策树的过程。一般情况下，训练样本数据集是根据实际需要有历史的、有一定综合程度的，用于数据分析处理的数据集。

1. 树以代表训练样本的单个结点开始。
2. 如果样本都在同一个类．则该结点成为树叶，并用该类标记。
3. 否则，算法选择最有分类能力的属性作为决策树的当前结点。
4. 根据当前决策结点属性取值的不同，将训练样本数据集分为若干子集，每个取值形成一个分枝。
5. 针对上一步得到的一个子集，重复进行先前步骤，形成每个划分样本上的决策树。
6. 递归划分步骤仅当下列条件之一成立时停止：
   (a) 给定结点的所有样本属于同一类。
   (b) 没有剩余属性可以用来进一步划分样本。以样本组中个数最多的类别作为类别标记。

决策树的剪技 决策树的剪枝是对上一阶段生成的决策树进行检验、校正和修下的过程，主要是用新的样本数扼集（称为测试数据集）中的数据校验决策树生成过程中产生的初步规则，将那些影响预衡准确性的分枝剪除。由于数据表示不当、有噪声或者由于决策树生成时产生重复的子树等原因，都会造成产生的决策树过大。因此，简化决策树是一个不可缺少的环节。寻找一棵最优决策树，主要应解决以下 3 个最优化问题：

1. 生成最少数目的叶子节点；
2. 生成的每个叶子节点的深度最小；

3. 生成的决策树叶子节点最少且每个叶子节点的深度最小。

决策树的典型算法有 ID3，C4.5，CART 等。相对于其它算法，决策树易于理解和实现，人们在通过解释后都有能力去理解决策树所表达的意义。决策树可以同时处理不同类型的属性, 并且在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

3.4 贝叶斯分类算法

贝叶斯分类（Bayes）算法是计算在给定样本情况下不同类别的后验概率，其表示形式见公式3.1：

$$f(\phi(x)) \propto \underset{y=1}{\overset{C}{\operatorname{argmax}}}\, p(\phi(x)|y)p(y). \tag{3-13}$$

这里需要估计概率函数 $p(\phi(x)|y)$，当 $\phi(x)$ 为高维向量时，这是比较困难的工作。

在实际应用中，经常使用**朴素贝叶斯（Naïve Bayes，NB）算法** 。在朴素贝叶斯算法中，$p(\phi(x)|y)$ 近似为样本向量中每一维变量概率的乘积。

$$p(\phi(x)|y) = \prod_{i=1}^{M} p(\phi_i(x)|y), \tag{3-14}$$

这里 M 是特征向量 $\phi(x)$ 的维数，$\phi_i(x)$ 是 $\phi(x)$ 第 i 维的值。

$p(\phi(x)|y)$ 之所以能展开成公式（3-14）的连乘积形式，就是假设样本每一维之间是彼此独立的。但这种情况在实际情况中经常是不成立的，因此其分类准确率可能会下降。比如在文本分类中，一个样本是一篇文档，每一维特征可以看出是一个词。但是词语之间有明显的所谓“共现”关系，在不同主题的文章中，可能共现的次数或频率有变化，但彼此间绝对谈不上独立。

但在许多场合，朴素贝叶斯分类算法的假设依然取得很好的性能，并且十分简单，可以与很多复杂的分类算法相媲美，是自然语言处理中最为常用的算法之一。

3.5 k 最近邻算法（kNN 算法）

k 最近邻 (k-Nearest Neighbor，kNN) 算法，是最简单有效的机器学习算法之一。该算法的步骤是：对于一个样本，在特征空间中找到和它最相似 (即特征空间中最邻近) 的 k 个样本，如果 k 个样本中的大多数属于某一个类别，则该样本也属于这个类别。kNN 算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。这也意味着 kNN 算法根本没有真正意义上的“训练”阶段。这种判断方法很好的克服了线性不可分问题的缺陷，也很适用于分类标准随时会产生变化的需求（只要删除旧训练样本，添加新训练样本，就改变了分类的准则）。当 $k = 1$ 时，也称为**最近邻（NN）算法**。

kNN 方法相当于非参数密度估计方法，在决策时只与极少量的相邻样本有关。由于 KNN 方法主要靠周围有限的邻近的样本，因此对于类域的交叉或重叠较多的非线性可分数据来说，kNN 方法较其他方法更为适合。

kNN 的一个不足是判断一个样本的类别时，需要把它与所有已知类别的样本都比较一遍，这样计算开销是相当大的。比如一个文本分类系统有上万个类，每个类即便只有 20 个训练样本，为

了判断一个新样本的类别，也要做 20 万次的向量比较。这个问题可以通过对样本空间建立索引来弥补。

kNN 也有另一个不足是：当样本不平衡时，如一个类的样本容量很大，而其他类样本容量很小时，有可能导致当输入一个新样本时，该样本的 k 个邻居中大容量类的样本占多数，导致分类错误。

3.6 支持向量机 (SVM)

支持向量机（Support Vector Machine，SVM）是一个经典的监督学习算法，它在解决小样本、非线性及高维模式识别中表现出许多特有的优势，并能够推广应用到函数拟合等其他机器学习问题中。

支持向量机方法是建立在统计学习理论的 VC 维理论和结构风险最小化原理基础上的，根据有限的样本信息在模型的复杂性和学习能力（即无错误地识别任意样本的能力）之间寻求最佳折衷，以期获得最好的泛化能力。

SVM 引入边际距离的概念，

$$\gamma = f(\phi(x,y)) - \operatorname*{argmax}_{\hat{y}\neq y} f(\phi(x,\hat{y})) \tag{3-15}$$

这里 $f(\phi(x),y)$ 是线性函数，见公式3-11。

给定训练样本 $\{(x_i,y_i\}_{i=1}^n$，公式3-15可以写为：

$$\gamma_i(\mathbf{w}) = w\cdot\phi(x,y) - \operatorname*{argmax}_{\hat{y}\neq y} \mathbf{w}\cdot\phi(x,\hat{y}) \tag{3-16}$$

SVM 目标是寻找一个 w^* 使得，

$$w^* = \operatorname*{argmax}_{\mathbf{w}:||\mathbf{w}||=1} \min_{i=1}^{n} \gamma_i(\mathbf{w}) \tag{3-17}$$

公式3-17可以写为：

$$\begin{gathered}\min_{\mathbf{w}} \tfrac{1}{2}||\mathbf{w}||^2 \\ \text{s.t.}\gamma_i(w) \geq 1, (\forall i)\end{gathered} \tag{3-18}$$

公式3-18的约束条件比较强，为了能够容忍部分不满足约束的样本，可以引入松弛变量 ξ：

$$\begin{gathered}\min_{\mathbf{w},\xi} \tfrac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^n \xi_i \\ \text{s.t.}\gamma_i(w) \geq 1-\xi_i, \xi_i \geq 0, (\forall i)\end{gathered} \tag{3-19}$$

求解 SVM 的目标函数可以通过二次优化问题（指目标函数为二次函数，约束条件为线性约束的最优化问题），得到的是全局最优解，这使它有着其他统计学习技术难以比拟的优越性。同时使用核函数将原始的样本空间向高维空间进行变换，能够解决原始样本线性不可分的问题。

SVM 的缺点是核函数的选择缺乏指导，难以针对具体问题选择最佳的核函数；另外 SVM 训练速度极大地受到训练集规模的影响，计算开销比较大。

SVM 分类器的优点在于通用性较好，且分类精度高、分类速度快、分类速度与训练样本个数无关，是最常用的分类器之一。

3.7 评价方法

为了衡量一个分类算法好坏，需要给定一个测试集，用分类器对测试集中的每一个样本进行分类，并根据分类结果计算评价分数。常见的评价标准有正确率、准确率、召回率和 F 值等。

给定测试集 $T = (x_1, y_1), \cdots, (x_M, y_M)$，其中 $\forall i, y_i \in [\omega_1, \cdots, \omega_C]$。假设分类结果为 $Y = \hat{(}y_1), \cdots, \hat{(}y_M)$。

则**正确率**（Accuracy，Correct Rate）为：

$$Acc = \frac{\sum_{i=1}^{M} |y_i = \hat{y_i}|}{M} \tag{3-20}$$

其中，$|\cdot|$ 为指示函数，若条件为真，$|\cdot| = 1$；否则 $|\cdot| = 0$。

和正确率相对应的就是**错误率**（Error Rate）。

$$Err = \frac{\sum_{i=1}^{M} |y_i \neq \hat{y_i}|}{M} \tag{3-21}$$

正确率是平均的整体性能。

在很多情况下，我们需要对每个类都进行性能估计，这就需要计算准确率和召回率。正确率和召回率是广泛用于信息检索和统计学分类领域的两个度量值，在机器学习的评价中也被大量使用。

准确率（Precision，P），也叫查准率，精确率或精度，是识别出的个体总数中正确识别的个体总数的比例。对于类 c 来说，

$$P_c = \frac{\sum_{\substack{i=1 \\ \hat{y_i}=c}}^{M} |y_i = \hat{y_i}|}{\sum_{\substack{i=1 \\ \hat{y_i}=c}}^{M} 1} \tag{3-22}$$

召回率（Recall，R），也叫查全率，是测试集中存在的个体总数中正确识别的个体总数的比例。

$$R_c = \frac{\sum_{\substack{i=1 \\ y_i=c}}^{M} |y_i = \hat{y_i}|}{\sum_{\substack{i=1 \\ y_i=c}}^{M} 1} \tag{3-23}$$

F1 值是根据正确率和召回率二者给出的一个综合的评价指标，具体定义如下：

$$F1_c = \frac{P_c * R_c * 2}{(P_c + R_c)} \tag{3-24}$$

为了计算分类算法在整个数据集上的总体准确率、召回率和 F1 值，经常使用两种平均方法，分别称为**宏平均**（macro average）和**微平均**（micro average）。

宏平均是每一个类的性能指标的算术平均值，

$$R_{macro} = \sum_{i=1}^{C} R_c / C, \tag{3-25}$$

$$P_{macro} = \sum_{i=1}^{C} P_c/C, \tag{3-26}$$

$$F1_{macro} = \frac{P_{macro} * R_{macro} * 2}{(P_{macro} + R_{macro})}. \tag{3-27}$$

而微平均是每一个样本的性能指标的算术平均。对于单个样本而言，它的准确率和召回率是相同的（要么都是 1，要么都是 0）因此准确率和召回率的微平均是相同的，根据 F1 值公式，对于同一个数据集它的准确率、召回率和 F1 的微平均指标是相同的。

第四章　监督学习实践：文本分类

文本分类是指按照预先定义的文本主题类别，将一篇文章归于预先给定的某一类或某几类的过程。20 世纪 90 年代以前，文本分类由专业人员手工进行分类，非常费时，效率非常低。之后，众多的机器学习方法应用于自动文本分类，文本分类研究引起了研究人员的极大兴趣，并在信息检索、网页自动分类、数字图书馆、自动文摘、新闻组分类、文本过滤以及文档的组织和管理等多个领域得到了广泛的应用。

文本分类（Text Categorization），或称**自动文本分类**，是指在给定分类体系下，根据文本内容自动确定文本类别的过程。文本分类是文本挖掘的一个重要内容。

在文本分类方面，经典的文献如下：

- T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. *Proc. of Euro. Conf. on Mach. Learn. (ECML)*, pages 137–142, 1998
- F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys*, 34(1):1–47, 2002
- Y. Yang. A study of thresholding strategies for text categorization. In *Proc. of Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval(SIGIR)*, pages 137–145. ACM Press New York, NY, USA, 2001
- Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proc. of SIGIR*. ACM Press New York, NY, USA, 1999

本节下面内容以 FudanNLP 工具集为参考，介绍基本的文本分类方法并实现一个简单的文本分类器。最新示例参考/FudanNLP/example/edu/fudan/example/TextClassification.java 文件。

4.1　文本分类数据集

中文文本分类方面还没有广泛使用的数据集。

表 4.1: 文本分类数据集

数据集	网址
20 Newsgroup	http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html
TechTC	http://techtc.cs.technion.ac.il/
Reuters 21578	http://www.daviddlewis.com/resources/testcollections/reuters21578/
搜狗文本分类语料库	http://www.sogou.com/labs/dl/c.html

4.2 样本表示

机器学习中样本 (x, y) 至少包含两部分：一是样本属性 x，二是类别 y。样本属性在机器学习算法中一般是以连续变量或离散变量的形式存在的（也成为特征），而在自然语言处理中一般都是文字形式。因此，样本属性可以是多种多样的方式存在的，

4.2.1 样本特征

为了更好的区分样本属性的概念。我们用 x 表示抽象概念的样本，可以有多种表示形式，比如字符串、数组、集合等。而用 $\phi(x)$ 表示**样本特征**，一般表示为一维或多维变量，$\phi(x) \in \mathbb{R}^k$，k 是向量维数。多维变量在自然语言处理中基本上都是非常稀疏的，可以用稀疏向量来表示。

4.2.2 FudanNLP 中的样本表示

FudanNLP 中样本用 edu.fudan.ml.types.Instance 类（简称 Instance 类）描述。每一样本都是 Instance 类的实例化对象。Instance 类有两个主要的成员变量 data 和 target，它们分别表示样本的属性 x 和类别 y。

Instance 类描述：

```
1 /**
2  * 表示单个样本 (x,y)。x,y 分别对应 data,target.
3  */
4 public class Instance {
5     /**
6      * 样本属性，相当于 x
7      */
8     protected Object data;
9     /**
```

```
10      * 标签或类别，相当于 y
11      */
12     protected Object target;
13 }
```

4.2.3 FudanNLP 中的样本集合表示

FudanNLP 中**样本集合**用 edu.fudan.ml.types.InstanceSet 类（简称 InstanceSet 类）描述。样本集合定义为 InstanceSet 类的实例化对象。

InstanceSet 类描述：

```
1  /**
2   * 样本集合
3   */
4  public class InstanceSet extends ArrayList<Instance> {
5      /**
6       * 本样本集合默认的数据类型转换管道
7       */
8      private Pipe pipes = null;
9      /**
10      * 本样本集合对应的特征和标签索引字典管理器
11      */
12     private AlphabetFactory factory = null;
13
14     /**
15      * 用本样本集合默认的“数据类型转换管道”通过“数据读取器”批量建立样
            本集合
16      * @param reader 数据读取器
17      * @throws Exception
18      */
19     public void loadThruPipes(Reader reader) throws Exception;
20 }
```

其中，Pipe 和 AlphabetFactory 分别是数据类型转换管道和字典管理器，我们在下节进行介绍。

4.3 数据处理与特征生成

自然语言处理的数据都是以字符形式存在的，在构造了样本和样本集合之后，为了和后面的机器学习算法相结合，我们将样本 x 转变成向量 $\phi(x)$。在将字符表示转换成向量表示的过程中需要很多中间步骤，我们把这些中间步骤都成为**数据处理**，并且尽可能的模块化。

下面我们集中将文本数据向量化的模型。向量空间模型 (Vector Space Model,VSM) 是近几年来应用较多且效果较好的方法之一 [4]。1969 年，Gerard Salton 提出了向量空间模型 VSM，它是文档表示的一个统计模型。该模型的主要思想是：将每一文档都映射为由一组规范化正交词条矢量张成的向量空间中的一个点。对于所有的文档类和未知文档，都可以用此空间中的词条向量（T1 ,W 1,T 2,W2,…, Tn, Wn）来表示 (其中，Ti 为特征向量词条；Wi 为 Ti 的权重)[5]。一般需要构造一个评价函数来表示词条权重，其计算的唯一准则就是要最大限度地区别不同文档。这种向量空间模型的表示方法最大的优点在于将非结构化和半结构化的文本表示为向量形式，使得各种数学处理成为可能。

4.3.1 词袋模型

一种简单的方法是简单假设文本（如一个句子或一个文档）是由字、词组成的无序集合，不考虑语法甚至词序。这就是在自然语言处理和信息检索中常用的**词袋模型**（Bag of words），或简称**词袋**。

词袋模型将文本仅仅看作词的集合，这种假设虽然对文本表示进行了简化，但在不要需要深层分析的应用（比如信息检索和文本分类）来说，也有一定的合理性，也便于模型化。比如在文本分类中，如果我们观察“体育”和“科技”两个类别的新闻网页，会发现这两类网页中词汇的分布是有明显区别的。在“体育”类的网页中经常会出现“比赛”、“胜利”、“冠军”等词，而“科技”类网页中经常会出现“电脑”、“手机”、“产品”等词。

词袋模型在需要深层分析的场合就会显得太过简化了。例如在语义分析里，“你打了我”和“我打了你”，意思是相反的，但用词袋模型表示后，这两句话是向量表示的等价的，这显然是不合理的。

4.3.2 N 元特征

N 元特征（N-gram 特征）可以看出是对词袋模型的一种改进方法。与 N 元特征相关的概念是**N 元语法模型**。N 元语法模型 (N-gram Model) 一种统计方法，假定一个字只与前面的 $N-1$ 个字有关，相当于 $N-1$ 阶的马尔可夫模型，在信息检索、语音识别和机器翻译中被广泛使用。

“N 元特征”，顾名思义，就是由 N 个单元组成的字符串，单元可以是字或词。这里 N 是大于等于 1 的任意整数。

例如中文句子“机器学习算法研究”，以字为基本单元的二元特征为“机器/器学/学习/习算/算法/法研/研究”，可以看出，每一个特征项都是由二个字组成的，可以是有含义的词（例如：“学习”、“算法”），也可以是无任务含义的字串（例如：“法研”、“器学”）。若以词为基本单位，首先需要进行分词，句子转换成“机器 学习 算法 研究”，二元特征为“机器学习/学习算法/算法研究”。

可以看到，随着 N 的增加，特征空间呈指数增加。并且特征集中会存在大量毫无意义的特征项，这些特征项没有对分类也没有太多帮助，还会直接影响着后续处理的效率与复杂度。

4.3.3 TF-IDF

TF-IDF（Term Frequency - Inverse Document Frequency，词频 -逆向文档频率) 是一种统计方法，用以评估**词**（Term）对于文本集中的某一个文档的重要程度，被公认为信息检索中重要的发明。

词的权重跟它在文本中出现的次数成正比，与包含词条的文档数量成反比。TF-IDF 加权的各种形式常被搜索引擎、文本分类等应用，可以用来衡量两个文档之间相关程度。

TF-IDF 其主要思想是，如果某个词在一个文档中出现的频率 TF 高，并且在其他文档中很少出现，则认为该项具有很好的类别区分能力，适合用来分类。

词频（Term Frequency，TF）$tf_{i,j}$ 表示文档 i 中词汇 j 出现的频率，计算公式如下：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{4-1}$$

其中，$n_{i,j}$ 表示词 j 在文档 i 中出现的次数，分母则是文档 j 中所有字词出现的次数之和。

逆向文档频率（Inverse Document Frequency，IDF）idf_i 是一个词普遍重要性的度量，由下面的式子计算：

$$idf_i = \log \frac{|D|}{|d : d \ni t_i|} \tag{4-2}$$

其中，$|D|$ 是文档总数，分母是包含词 t_i 的文档数目。

权值 $tfidf_{i,j} = tf_{i,j} * idf_i$。权值就是最终要得到的结果，权值的高低直接表明了该题词是否反应了文档的主题。

4.3.4 FudanNLP 中的数据转换

FudanNLP 中数据转换通过**管道**来进行，定义为 edu.fudan.nlp.pipe.Pipe 类（简称 Pipe 类），该类是一个抽象类。

```
/**
 * 数据类型转换管道，通过一系列的组合将数据从原始方式转为需要的数据类型
 * Pipe 只能每次连续流水处理一个样本，不能按阶段多遍执行
 * 要分阶段多遍执行参见 InstanceSet 类 loadThruStagePipes 方法
 */
public abstract class Pipe{

    /**
     * 基本的数据类型转换处理操作，继承类需重新定义实现
     * @param inst 样本
     */
    public abstract void addThruPipe(Instance inst) throws
        Exception;
```

所有的数据转换器都继承 Pipe 类，通过重新定义 addThruPipe 方法来实现不同的数据类型转换。

一个特殊的继承类是 edu.fudan.nlp.pipe.SeriesPipes 类（简称 SeriesPipes 类），该类是一个或多个 Pipe 对象的串行组合。

4.3.5 FudanNLP 中的特征生成

我们定义一个抽象类 edu.fudan.ml.feature.Generator（简称 Generator 类）来进行特征生成。不同的特征生成方法都继承 Generator 类。

```
/**
 * 生成特征向量，包含类别信息
 */
public abstract class Generator implements Serializable {
    public SparseVector getVector(Instance inst);
}
```

4.4 分类算法

第五章　非监督学习

导言

非监督学习（Unsupervised Learning）是指直接对输入的无类别标记的数据集进行建模。

监督学习必须要有标记好的训练集，在训练集中找规律，以求对训练集数据达到某种最优，并能推广到新数据。而非监督学习只有无标记的数据，在该组数据集内寻找规律。非监督学习方法只有要分析的数据集本身，没有标号。如果发现数据集呈现某种聚集性，则可按自然的聚集性分类，但并没有预先的分类标号。

样本数据类别未知，需要根据样本间的相似性对样本集进行分类，这个过程可以称为聚类。

聚类（Clustering）就是将数据分组成为多个类（Cluster）。在同一个类内对象之间具有较高的相似度，不同类之间的对象差别较大。

非监督式学习是一种机器学习的方式，并不需要人力来输入标签。它是监督式学习和强化学习等策略之外的一种选择。在监督式学习中，典型的任务是分类和回归分析，且需要使用到人工预先准备好的范例。一个常见的非监督式学习是数据聚类。在人工神经网络中，自我组织映射（SOM）和适应性共振理论（ART）则是最常用的非监督式学习。

5.1　聚类算法

目标是我们不告诉计算机怎么做，而是让它（计算机）自己去学习怎样做一些事情。非监督学习一般有两种思路。第一种思路是在指导 Agent 时不为其指定明确的分类，而是在成功时采用某种形式的激励制度。需要注意的是，这类训练通常会置于决策问题的框架里，因为它的目标不是产生一个分类系统，而是做出最大回报的决定。这种思路很好的概括了现实世界，Agent 可以对那些正确的行为做出激励，并对其他的行为进行处罚。　　强化学习的一些形式常常可以被用于非监督学习，由于没有必然的途径学习影响世界的那些行为的全部信息，因此 Agent 把它的行为建立在前一次奖惩的基础上。在某种意义上，所有的这些信息都是不必要的，因为通过学习激励函数，Agent 不需要任何处理就可以清楚地知道要做什么，因为它（Agent）知道自己采取的每个动作确切的预期收益。对于防止为了计算每一种可能性而进行的大量计算，以及为此消耗的大量时间（即使所有世界状态的变迁概率都已知），这样的做法是非常有益的。另一方面，在尝试出错上，这也是一种非常耗费时间的学习。　　不过这一类学习可能会非常强大，因为它假定没有事先分类的样本。在某些

情况下，例如，我们的分类方法可能并非最佳选择。在这方面一个突出的例子是 Backgammon（西洋双陆棋）游戏，有一系列计算机程序（例如 neuro-gammon 和 TD-gammon）通过非监督学习自己一遍又一遍的玩这个游戏，变得比最强的人类棋手还要出色。这些程序发现的一些原则甚至令双陆棋专家都感到惊讶，并且它们比那些使用预分类样本训练的双陆棋程序工作得更出色。 一种次要的非监督学习类型称之为聚合（clustering）。这类学习类型的目标不是让效用函数最大化，而是找到训练数据中的近似点。聚合常常能发现那些与假设匹配的相当好的直观分类。例如，基于人口统计的聚合个体可能会在一个群体中形成一个富有的聚合，以及其他的贫穷的聚合。

在机器学习 (Machine learning) 领域，监督学习 (Supervised learning)、非监督学习 (Unsupervised learning) 以及半监督学习 (Semi-supervised learning) 是三类研究比较多，应用比较广的学习技术，wiki 上对这三种学习的简单描述如下：监督学习：通过已有的一部分输入数据与输出数据之间的对应关系，生成一个函数，将输入映射到合适的输出，例如分类。非监督学习：直接对输入数据集进行建模，例如聚类。半监督学习：综合利用有类标的数据和没有类标的数据，来生成合适的分类函数。以上表述是我直接翻译过来的，因为都是一句话，所以说得不是很清楚，下面我用一个例子来具体解释一下。其实很多机器学习都是在解决类别归属的问题，即给定一些数据，判断每条数据属于哪些类，或者和其他哪些数据属于同一类等等。这样，如果我们上来就对这一堆数据进行某种划分 (聚类)，通过数据内在的一些属性和联系，将数据自动整理为某几类，这就属于非监督学习。如果我们一开始就知道了这些数据包含的类别，并且有一部分数据 (训练数据) 已经标上了类标，我们通过对这些已经标好类标的数据进行归纳总结，得出一个“数据 –> 类别”的映射函数，来对剩余的数据进行分类，这就属于监督学习。而半监督学习指的是在训练数据十分稀少的情况下，通过利用一些没有类标的数据，提高学习准确率的方法。铺垫了那么多，其实我想说的是，在 wiki 上对于半监督学习的解释是有一点点歧义的，这跟下面要介绍的主动学习有关。主动学习 (active learning)，指的是这样一种学习方法：有的时候，有类标的数据比较稀少而没有类标的数据是相当丰富的，但是对数据进行人工标注又非常昂贵，这时候，学习算法可以主动地提出一些标注请求，将一些经过筛选的数据提交给专家进行标注。这个筛选过程也就是主动学习主要研究的地方了，怎么样筛选数据才能使得请求标注的次数尽量少而最终的结果又尽量好。主动学习的过程大致是这样的，有一个已经标好类标的数据集 K(初始时可能为空)，和还没有标记的数据集 U，通过 K 集合的信息，找出一个 U 的子集 C，提出标注请求，待专家将数据集 C 标注完成后加入到 K 集合中，进行下一次迭代。按 wiki 上所描述的看，主动学习也属于半监督学习的范畴了，但实际上是不一样的，半监督学习和直推学习 (transductive learning) 以及主动学习，都属于利用未标记数据的学习技术，但基本思想还是有区别的。如上所述，主动学习的“主动”，指的是主动提出标注请求，也就是说，还是需要一个外在的能够对其请求进行标注的实体 (通常就是相关领域人员)，即主动学习是交互进行的。而半监督学习，特指的是学习算法不需要人工的干预，基于自身对未标记数据加以利用。至于直推学习，它与半监督学习一样不需要人工干预，不同的是，直推学习假设未标记的数据就是最终要用来测试的数据，学习的目的就是在这些数据上取得最佳泛化能力。相对应的，半监督学习在学习时并不知道最终的测试用例是什么。也就是说，直推学习其实类似于半监督学习的一个子问题，或者说是一个特殊化的半监督学习，所以也有人将其归为半监督学习。而主动学习和半监督学习，其基本思想上就不一样了，所以还是要加以区分的，如果 wiki 上对半监督学习的解释能特别强调一下“是在不需要人工干预的条件下由算法自行完成对无标记数据的利用”，问题就会更清楚一些了。

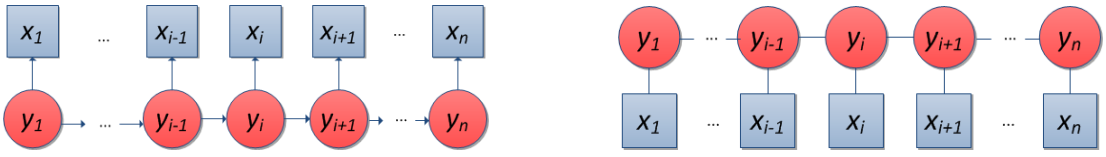
第六章　序列标注模型

序列标注是自然语言处理领域的一个非常常见的问题，从分词、词性标注，到较深层的组块分析以至更为深层的完全句法分析、语义角色标注等任务，都可以看作是典型的序列标注问题。

在自然语言处理应用中，比如分词、词性标注等，不再是简单的离散分类问题，样本的不同元素标记之间有很强的相关性，因此不能孤立地对每个待标注对象进行分类和标记，必须进行联合标记。

序列标注问题是指对于序列形式的输入样本 $x = x_1, \ldots, x_n$，对序列中每个元素进行标记，输出标记序列 $y = y_1, \ldots, y_n$，n 是序列的长度。若 y_i 的取值范围定义为 $S = \{s_i\}_{i=1}^{C}$，输出序列的可能组合数为 C^L。变量 y_i 的不同取值也叫不同的状态。y_i 所有可能取值的集合 S，也被称为“状态空间”。因为一个序列状态的组合数非常多，也不能直接用传统的学习方法通过枚举 y 来得到最佳的标记，需要用动态优化的方法来求解 y。

序列中每个元素间关联紧密，元素标记之间也具有很强的相关性，传统的单点分类器方法难以获得整个序列的最优标记。图6.1是两种线性链序列标注结构，每个元素标记只与相邻的元素相关，构成了线性链式结构。其中，图6.1a是有向图结构，每个元素标记只与前一个元素标记相关，图6.1b是无向图结构，每个元素标记与左右两个相邻元素标记相关。



(a) 线性链序列标注有向图　　(b) 线性链序列标注无向图

图 6.1: 线性链序列标注

序列标注问题是结构化机器学习的一种特例。**结构化机器学习**就是指处理的样本 (x, y)，y 不再是离散的类别，而是有结构的，比如序列、树结构等。序列标注模型都可以和图模型 [Jordan, 1998] 进行对应。

序列标注问题的经典参考文献如下:

1. C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, page 93, 2007

2. John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL http://portal.acm.org/citation.cfm?id=645530.655813

3. A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598. Citeseer, 2000

4. Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 2002

5. Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *In proceedings of the 17th Annual Conference on Neural Information Processing Systems*, Whistler, B.C., Canada, 2003

6. I. Tsochantaridis, T. Hofmann, T. Joachims, and Y Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning(ICML)*, 2004

6.1 序列标注问题

序列标注问题可以用下面公式表示：

$$\hat{y} = \overbrace{\underset{y}{\text{argmax}}}^{\text{解码}} \underbrace{f(\overbrace{\Phi(x,y)}^{\text{特征}}, \overbrace{\mathbf{w}}^{\text{模型参数}})}_{\text{模型}} \tag{6-1}$$

其中，x 和 y 分别是输入序列和标记序列，$\Phi(x,y)$ 是在序列上抽取的特征向量，$f(\Phi(x,y),\mathbf{w})$ 是序列标注模型，$\mathbf{w}$ 是模型参数。

从公式6-1可知，序列标注问题需要解决四个问题：

1. 如何选择合适的序列标注模型？确定标记之间的关联关系。
2. 怎样从序列上抽取特征？
3. 如何进行求解？也就是解码问题。
4. 如何进行参数学习？

下面我们分别来讨论这四个问题：

6.1.1 序列标注模型

序列标注模型就是定义一个函数 $f(\cdot)$，用来描述一个序列标注任务，包括模型结构（不同元素标记之间的关联关系），解码方法以及参数学习方法。常见的序列标注模型有：线性模型、隐马尔可夫模型、最大熵马尔可夫模型、条件随机场等。

首先，我们看下序列标注模型的结构，一般用**阶数**来形容模型结构和复杂度。阶数是从马尔可夫链中借鉴来的一个概念。

马尔可夫链，简称马氏链，是由随机变量组成的一个序列 $x_1, x_2, x_3, \cdots$，x_t 的值是在时间 t 时的状态。如果 x_{t+1} 对于过去状态的条件概率分布仅是 x_t 的一个函数，即

$$P(x_{t+1}|x_1, x_2, \cdots, x_t) = P(x_{t+1}|x_t) \tag{6-2}$$

那么，这个序列就称为**一阶马尔可夫链**，简称马尔可夫链。

如果 x_{t+1} 对于过去状态的条件概率分布仅是 x_t, x_{t-1} 的一个函数，即

$$P(x_{t+1}|x_1, x_2, \cdots, x_t) = P(x_{t+1}|x_t, x_{t-1}) \tag{6-3}$$

那么，这个序列就称为**二阶马尔可夫链**。

如果 x_{t+1} 对于过去状态的条件概率分布仅是 $x_{t-m+1}, \cdots, x_t$ 的一个函数，即

$$P(x_{t+1}|x_1, x_2, \cdots, x_t) = P(x_{t+1}|x_{t-m+1}, \cdots, x_t) \tag{6-4}$$

那么，这个马尔可夫链的**阶数**就是 m。

一般阶数大于等于 2 的马尔可夫链，也叫**高阶马尔可夫链**。

图6.2给出了序列标注问题中一阶和二阶的序列标注模型。

(a) 一阶序列标注模型　　(b) 二阶序列标注模型

图 6.2: 序列标注模型

序列标注模型可以分为两大类：一种是非统计方法，另一种是统计的方法。

在非统计方法中，最有代表性的是线性分类器：

$$y = \underset{y'}{\operatorname{argmax}}\, \mathbf{w} \cdot \Phi(x, y), \tag{6-5}$$

在基于统计方法，比较主流的方法是用无向图来表示模型。

在无向图模型中，条件概率可以表示为：

$$P(y|x) = \frac{\exp(\mathbf{w} \cdot \Phi(x, y))}{Z_x}, \tag{6-6}$$

其中，Z_x 是 x 的边际概率，$Z_x = \sum_{y' \in \mathcal{Y}} \exp(\mathbf{w} \cdot \Phi(x, y'))$。

基于无向图模型的序列标注可以表示为：

$$
\begin{aligned}
y &= \underset{y'}{\operatorname{argmax}} P(y'|x) && \text{(6-7)}\\
&= \underset{y'}{\operatorname{argmax}} \frac{\exp(\mathbf{w}\cdot\Phi(x,y'))}{Z_x}, && \text{(6-8)}\\
&= \underset{y'}{\operatorname{argmax}} \exp(\mathbf{w}\cdot\Phi(x,y')), && \text{(6-9)}\\
&= \underset{y'}{\operatorname{argmax}} \mathbf{w}\cdot\Phi(x,y'), && \text{(6-10)}
\end{aligned}
$$

从公式6-5和6-10看出，基于统计和非统计的方法殊途同归，最后的解码公式是相同的。但这不意味这些模型是相同的，不同的模型的出发点不同，学习参数的目标函数也不同，导致最终的模型也存在一定差异。

6.1.2 特征生成

6.1.3 特征 v.s. 模板

For the problems with structured inputs, features are usually not explicitly defined. For instance, part-of-speech tagging in the area of NLP is to label each token with a specific tag in the given token sequence (sentence). In addition to the current token, the neighbors and their associated tags can be considered as the important features to determine the tag of the current token. All these intuitive information can be represented by feature functions called templates. In this paper, we consider the templates as some predefined rules which are used for feature extraction task.

在给定序列标注模型后，接下来一个问题是如何把一个样本 (x,y) 转换为向量表示 $\Phi(x,y)$，也就是**特征向量**。输入 $x\in\mathcal{X}$ 是观察序列，可以是一篇文档，一个句子，有字或词为单位构成的序列。$x=x_0,x_1,\cdots,x_n$。输出 $y\in\mathcal{Y}$ 是状态序列，可以 x 对应的分词、词性、实体名的标记序列。$y=y_0,y_1,\cdots,y_n$。$\Phi(x,y)$ 是将 (x,y) 映射为特征向量的函数：

$$\Phi(x,y):\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^m \tag{6-11}$$

根据序列标注模型的关联性假设，我们可以对 $\Phi(x,y)$ 进行分解（factorization）。

对于一阶模型，$\Phi(x,y)$ 可以分解为：

$$\Phi(x,y)=\sum_{i=1}^{|y|}\phi(x,y_{i-1},y_i) \tag{6-12}$$

对于 k 阶模型，$\Phi(x,y)$ 可以分解为：

$$\Phi(x,y)=\sum_{i=1}^{|y|}\phi(x,y_{i-k},\cdots,y_{i-1},y_i) \tag{6-13}$$

下面通过一个例子来说明如何构造特征。

对于中文分词的序列标注问题，可以定义 $y\in\{B,O\}$，这里 B 表示把当前字作为一个新词的开始，O 表示当前字与前面的字构成一个词。例如：句子“他 / 说 / 的 / 确实 / 在理”转化为下面以字为基本元素构成的序列。

x	=	他	说	的	确	实	在	理
y	=	B	B	B	B	O	B	O

对于一阶序列标注模型，$\phi(x, y_{i-1}, y_i)$ 可以定义在 x 中的任何元素以及当前位置两个相邻的标记上。比如对于下面两个特征：ϕ_j 和 ϕ_k:

$$\phi_j(x, y_{i-1}, y_i) = \begin{cases} 1 & \text{当} x_i = \text{“在”}, x_{i+1} = \text{“理”}, y_{i-1} = \text{“O”}, y_i = \text{“B”} \\ 0 & \text{otherwise} \end{cases} \tag{6-14}$$

$$\phi_k(x, y_{i-1}, y_i) = \begin{cases} 1 & \text{当} x_i = \text{“在”}, x_{i+1} = \text{“理”}, y_{i-1} = \text{“O”}, y_i = \text{“O”} \\ 0 & \text{otherwise} \end{cases} \tag{6-15}$$

显然，对于“在”后面跟着“理”，并且“在”之前的字已经构成了词的一部分，那么在“在”就是一个新词的开始。特征 ϕ_j 比 ϕ_k 更符合中文分词习惯，它们对应的特征权重 w_j 应该大于 w_k。这样，在解码时，“在”在这种情况下更倾向被标记为“B”。

6.1.4 解码问题

对于未知标记的样本 x，$\hat{y}$ 可以通过一个得分函数求得，

$$\hat{y} = \underset{\mathbf{y}}{\operatorname{argmax}} f(\mathbf{w}, \Phi(x, \mathbf{y})), \tag{6-16}$$

这里，$\mathbf{w}$ 是函数 $f(\cdot)$ 的参数.

从公式6-5和6-10可知，解码问题的一般形式为：

$$\hat{y} = \underset{y}{\operatorname{argmax}} \mathbf{w} \cdot \Phi(x, y), \tag{6-17}$$

假设当前模型为一阶模型，公式6-21可以转换为：

$$\hat{y} = \underset{y}{\operatorname{argmax}} \mathbf{w} \cdot \Phi(x, y) \tag{6-18}$$

$$= \underset{y}{\operatorname{argmax}} \mathbf{w} \cdot \sum_{i=1}^{|y|} \phi(x, y_{i-1}, y_i) \tag{6-19}$$

$$= \underset{y}{\operatorname{argmax}} \sum_{i=1}^{|y|} \mathbf{w} \cdot \phi(x, y_{i-1}, y_i) \tag{6-20}$$

假设状态空间大小为 C，对于长度为 n 的 y，其可能的组合数为 C^n。因此，穷举不同的 y 已获得最佳序列是不可行的。通过观察公式6-20，我们可以用动态优化方法来快速的求解。

我们首先定义 $\alpha_{s,i}$ 是输入序列 $x_0, \cdots, x_i$ 且 $y_i = s$ 的最佳标记序列。公式6-20可以写为：

$$\hat{y} = \max_{s} \alpha_{s,n}, \tag{6-21}$$

$\alpha_{s,i}$ 可以通过下面两个递归公式来计算：

$$\alpha_{s,0} = 0, \forall s \in \mathcal{S} \tag{6-22}$$

$$\alpha_{s,i} = \max_{s'} \alpha_{s',i-1} + \mathbf{w} \cdot \phi(x, s', s) \tag{6-23}$$

这个方法也叫 **Viterbi 算法**，可以保证找到得分最高的标记序列。

6.1.5 参数学习

最后一个问题是如何学习函数 $f(\cdot)$ 的参数 $\mathbf{w}$。根据 $F(\cdot)$ 的形式不同，学习参数的方法也不同，一般为最大似然估计、最大边际距离或最小均方误差等。我们可以用传统的分类器训练算法，比如感知器、SVM、kNN 等。

这里我们介绍一种简单有效的参数学习方法，**Passive-Aggressive（PA）算法** Crammer et al. [2006]。PA 算法是一种在线学习算法，它结合了感知器和 SVM 的优点，学习速度快，效果可以和批量学习算法近似。

PA 算法

给定一个样本 (x, y)，$\hat{y}$ 定义为错误标签中得分最高的标签。

$$\hat{y} = \arg\max_{\mathbf{z} \neq y} \mathbf{w} \cdot \Phi(x, \mathbf{z}). \tag{6-24}$$

边际距离 $\gamma(\mathbf{w}; (x, y))$ 定义为：

$$\gamma(\mathbf{w}; (x, y)) = \mathbf{w} \cdot \Phi(x, y) - \mathbf{w} \cdot \Phi(x, \hat{y}). \tag{6-25}$$

损失函数定义为：**hinge loss**.

$$\ell(\mathbf{w}; (x, y)) = \begin{cases} 0, & \gamma(\mathbf{w}; (x, y)) > 1 \\ 1 - \gamma(\mathbf{w}; (x, y)), & \text{otherwise} \end{cases} \tag{6-26}$$

PA 算法用在线的方式计算更新参数。在第 t 轮, 通过下面公式计算 $\mathbf{w}_{t+1}$：

$$\begin{aligned} \mathbf{w}_{t+1} &= \arg\min_{\mathbf{w} \in \mathbb{R}^n} \tfrac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2 + \mathcal{C} \cdot \xi, \\ &\textbf{s.t. } \ell(\mathbf{w}; (x_t, y_t)) \leq \xi \textbf{ and } \xi \geq 0 \end{aligned} \tag{6-27}$$

这里，C 是正的参数来控制松弛变量的影响。

我们用 ℓ_t 来表示 $\ell(\mathbf{w_t}; (x, y))$。如果 $\ell_t = 0$，那么 w_t 满足公式6.1.5。因此我们只关心 $\ell_t > 0$ 的情况。当 $\ell_t > 0$ 时，我们通过公式求解新的参数 $\mathbf{w}$，即寻找一个 $\mathbf{w}$ 使得下面公式最小化：

$$\begin{aligned} &\min \tfrac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2 + \mathcal{C} \cdot \xi, \\ &\textbf{s.t. } \ell(\mathbf{w}; (x_t, y_t)) \leq \xi \textbf{ and } \xi \geq 0 \end{aligned} \tag{6-28}$$

我们通过拉格朗日优化方法求解，公式6-28转换为对偶形式：

$$\begin{aligned} \mathcal{L}(\mathbf{w}, (x_t, y_t), \alpha, \beta) &= \tfrac{1}{2}||\mathbf{w} - \mathbf{w}_t||^2 + \mathcal{C} \cdot \xi + \alpha(\ell(\mathbf{w}; (x_t, y_t)) - \xi) - \beta\xi, \\ &\textbf{s.t. } \alpha \geq 0 \textbf{ and } \beta \geq 0 \end{aligned} \tag{6-29}$$

将 $\mathcal{L}$ 对 $\mathbf{w}$ 求导，并令其等于 0 得：

$$0 = \nabla_{\mathbf{w}} \mathcal{L} = \mathbf{w} - \mathbf{w}_t - \alpha(\Phi(x, y) - \Phi(x, \hat{y})). \tag{6-30}$$

即，

$$\mathbf{w} = \mathbf{w}_t + \alpha(\Phi(x, y) - \Phi(x, \hat{y})). \tag{6-31}$$

将 $\mathcal{L}$ 对 ξ 求导，并令其等于 0 得：

$$0 = \nabla_{\xi}\mathcal{L} = \mathcal{C} - \alpha - \beta. \tag{6-32}$$

即

$$\beta = \mathcal{C} - \alpha. \tag{6-33}$$

将公式6-31和6-33代入公式6-29得到：

$$\mathcal{L}(\alpha)) = \frac{1}{2}||\alpha(\Phi(x,y) - \Phi(x,\hat{y}))||^2 + \alpha\mathbf{w}(\Phi(x,y) - \Phi(x,\hat{y})) - \alpha. \tag{6-34}$$

将公式6-34对 α 求导，并令其等于 0 得：

$$\alpha = \frac{1 - \mathbf{w_t}^T(\Phi(x,y) - \Phi(x,\hat{y}))}{||\Phi(x,y) - \Phi(x,\hat{y})||^2}. \tag{6-35}$$

由公式6-33可知，

$$\alpha \leq \mathcal{C}. \tag{6-36}$$

最终我们得到更新策略为：

$$\mathbf{w_{t+1}} = \mathbf{w}_t + \alpha^*(\Phi(x,y) - \Phi(x,\hat{y})). \tag{6-37}$$

其中，

$$\alpha^* = \min(\mathcal{C}, \alpha). \tag{6-38}$$

算法如6.1所示。为了避免过拟合，这里使用了平均的策略。

算法 6.1: PA 算法

输入: 训练集: $(x_n, y_n), n = 1, \cdots, N$，参数：$\mathcal{C}, K$

输出: $\mathbf{w}$

初始化：$\mathbf{cw} \leftarrow 0$,;
for $k = 0 \cdots K-1$ **do**
  $\mathbf{w}_0 \leftarrow 0$;
  for $t = 0 \cdots T-1$ **do**
    挑一个样本 (x_t, y_t);
    预测：$\hat{y}_t = \arg\max_{\mathbf{z} \neq y_t} \langle \mathbf{w}_t, \Phi(x_t, \mathbf{z}) \rangle$;
    计算 $\ell(\mathbf{w}; (x, y))$;
    用 Eq.(6-37) 更新 $\mathbf{w}_{t+1}$;
  end
  $\mathbf{cw} = \mathbf{cw} + \mathbf{w}_T$;
end
$\mathbf{w} = \mathbf{cw}/K$;

6.2 常见的序列标注模型

下面我们介绍几种常见的序列标注模型。

6.2.1 线性模型

线性模型的表示如下：

$$y = \underset{y'}{\operatorname{argmax}}\, \mathbf{w} \cdot \Phi(x, y), \tag{6-39}$$

不同线性模型的区别是如何计算参数 $\mathbf{w}$。常见的训练方法有：感知器模型、PA 模型、结构化 SVM 等。

6.2.2 隐马尔可夫模型

隐马尔可夫模型（Hidden Markov Model，HMM）[Rabiner, 1989] 是最早的序列标注模型，自 20 世纪 80 年代以来被应用于语音识别，取得重大成功，成为信号处理的一个重要方向。

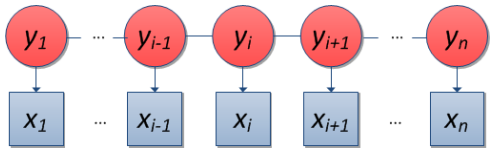


图 6.3: 隐马尔可夫模型

HMM 有三个典型问题：

1 已知模型参数，计算某一特定输出序列的概率. 通常使用 forward 算法解决.

2 已知模型参数，寻找最可能的能产生某一特定输出序列的隐含状态的序列. 通常使用 Viterbi 算法解决.

3 已知输出序列，寻找最可能的状态转移以及输出概率. 通常使用 Baum-Welch 算法以及 Reversed Viterbi 算法解决.

HMM 的三个问题可以通过动态规划方法解决，比如前向算法、后向算法、Viterbi 算法和 Baum-Welch 算法。

6.2.3 最大熵马尔可夫模型

隐马尔可夫模型是一个生成式模型，要对 $p(x, y)$ 进行建模。并且样本序列的观测值只与当前状态（标记）有关，这就限制了模型的能力。**最大熵马尔可夫模型**是一个判别式模型，直接对 $p(y|x)$ 进行建模，这样可以利用大量的冗余特征提高模型性能。最大熵马尔可夫模型如图6.4所示。

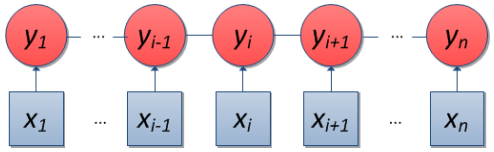


图 6.4: 最大熵马尔可夫模型

最大熵马尔可夫模型可以用 Generalized Iterative Scaling (GIS)

6.2.4 条件随机场

最大熵马尔可夫模型是用局部信息去优化全局，会有标注偏置（Label Bias）的问题。**条件随机场**（Conditional Random Fields, CRF）[Lafferty et al., 2001] 是一种无向图模型，它是在给定需要标记的观察序列 x 的条件下计算整个标记序列 y 的联合概率分布，而不是在给定当前状态条件下定义下一个状态的分布。即

$$P(y|x) = \frac{\exp(\mathbf{w} \cdot \Phi(x, y))}{Z_x}, \tag{6-40}$$

隐马尔可夫模型中存在两个假设：输出独立性假设和马尔可夫性假设。其中，输出独立性假设要求序列数据严格相互独立才能保证推导的正确性，而事实上大多数序列数据不能被表示成一系列独立事件。而条件随机场则使用一种概率图模型，条件随机场没有隐马尔可夫模型那样严格的独立性假设条件，因而可以容纳任意的上下文信息，可以灵活地设计特征。同时，条件随机场具有表达长距离依赖性和交叠性特征的能力，而且所有特征可以进行全局归一化，能够求得全局的最优解，还克服了最大熵马尔可夫模型标记偏置的缺点。

条件随机场模型作为一个整句联合标定的判别式概率模型，同时具有很强的特征融入能力，是目前解决自然语言序列标注问题最好的统计模型之一。条件随机场的缺点是训练的时间比较长。

6.2.5 最大边际距离马尔科夫网络

最大边际距离马尔科夫网络（Maximum margin Markov networks，M^3N）Taskar et al. [2003] 定义了一个对数线性马尔科夫网络。该模型的参数通过基于边际距离的优化问题进行求解。

$$\begin{aligned} \min \quad & \tfrac{1}{2}||\mathbf{w}||^2 + \mathcal{C}\textstyle\sum_{\mathbf{x}} \xi_{\mathbf{x}} \\ \text{s.t.} \quad & \mathbf{w}^T \Delta \mathrm{f}_{\mathbf{x}}(\mathbf{y}) >= \Delta \mathrm{t}_{\mathbf{x}}(\mathbf{y}) - \xi_{\mathbf{x}}, \forall \mathbf{x}, \mathbf{y} \end{aligned} \tag{6-41}$$

这里，ξ 是一个非负的松弛变量.

对于容易进行三角剖分的马尔科夫网络，可以讲二次优化问题转换为等价的多项式方程来进行高效的计算。

第七章　中文分词

词是最小的能够独立活动的有意义的语言成分，英文单词之间是以空格作为自然分界符的，而汉语是以字为基本的书写单位，词语之间没有明显的区分标记，因此，中文分词是中文信息处理的基础与关键。

中文句子是由字[1]组成的连续字符串。为了理解中文语义，首先需要将句子划分成以词为基本单位的词串，这就是**中文分词**。

中文分词最大的难点在于分词标准不明确。根据《信息处理用现代汉语分词规范 GB/T 13715-92》定义，词是最小的能独立运用的语言单位。汉语分词是从信息处理需要出发，按照特定的规范，对汉语按分词单位进行划分的过程。分词的基本原则是结合紧密、使用稳定。

从这个规范可以看出，词的定义非常灵活，不仅仅和词法、语义相关，在很大程度上也和应用场景、使用频率等其它因素相关。比如：“吃饭”是词，而“吃鱼”不是一个词。另外，像“不管三七二十一”、“由此可见”等一些常用的短语也看成一个词。

分词的方法可以概括为以下三类：

模板	特征
基于规则的方法	字典匹配等方法
无监督学习方法	比如词频统计、关联度分析等方法
监督学习方法	分类器、序列标注等方法

其中，由于基于规则的方法，尤其是字典匹配，易于理解和实现，成为分词方法中最流行的。因为字典匹配的方法不考虑具体的语言环境和语义，它最大的缺点就是不能处理多词冲突和新词情况。

多词冲突，也称为歧义分词[2]，是指一个字本身，或者与左右的相邻字组合，都可以匹配上词典中的词。表7.1 给出了多词冲突的示例。当然，词典匹配方法也有很多算法来处理这种情况，包括最大切分（包括向前、向后、以及前后相结合）、最少切分、全切分等。但这些算法都在一定程度上存在不足。

[1]“字”不只限于汉字。目前中文表述中不可避免地会包含少量的非汉字字符，本文所说的“字”，也包括外文字母、阿拉伯数字和标点符号等字符。所有这些字符都是构词的基本单元。

[2]这里的“歧义”仅仅是针对字典匹配方法而言，在人的认知上不存在歧义。因此不建议使用“歧义分词”这个定义。对于人也不能区分的真歧义（比如：“乒乓球拍卖完了”、等），这已经超出了中文分词的范畴。

表 7.1: 多词冲突示例

字典	示例句子
的、的确、确实、实在、在理	他说的确实在理
两、个、个人、人	两个人、个人问题
马、上、马上	坐在马上、马上来
大、大学、学生、大学生、生源	大学生、大学生源

新词，也成为未登录词，是指在字典或训练语料中都没有出现过的词。常见新词包括：人名、机构名、地名、产品名、商标名、简称、省略语、专业术语、网络新词等。字典匹配的方法只能靠不断在字典中增加新词来解决这个问题。

此外，在中文分词中，不用应用场景对分词性能的需求不同，比如在搜索引擎的应用中，对中文的处理是基于自动切分的单字切分，或者二元切分。

7.1 基于两类分类器的中文分词

中文句子中的相邻字符是否可以组成一个词，不能单独看这几个相邻的字符，还要看其出现的上下文环境。不同的上下文环境，同样的相邻字符串，有的可以组成一个词，有的就不能组成一个词。这也是基于字典匹配的分词方法的不足之处。

为了充分利用上下文信息，我们可以从机器学习的角度来看待中文分词问题。给定一个字符序列，中文分词问题可以看成是在每两个连续字符之间判断是否切分。如下面例子所示，我们用⋮表示一个潜在切分，其取值为 $\{0,1\}$，0 表示不切分，1 表示切分。

自	⋮	然	⋮	语	⋮	言	⋮	处	⋮	理
	0		1		0		1		0	

对于每一个“⋮”，我们用一个固定大小的窗口取得上下文信息，作为样本 x，“⋮”的取值作为类别标签 y。这样我们可以中文分词问题看成是两类分类问题。

窗口大小	样本 x	类别标签 y
2	“自⋮然”	0
	“然⋮语”	1
	“语⋮言”	0
4	“自然⋮语言”	1
	“然语⋮言处”	0
	“语言⋮处理”	1

但这样的方法有一个缺陷：在一个句子中，每个潜在切分是独立判定的。这个独立性假设并不符合人们的直观感觉。

7.2 基于字标记的中文分词

近年来，序列标注方法成为主流的中文分词方法，该方法将中文分词转化为基于字的序列标注问题，不依赖词典，极大地改进了歧义分词和新词识别的性能。Xue [2003] 将分词问题转换为基于字的序列标注问题，根据每个字在词中的位置赋予其一个标记。标记共四类：左、中、右、单字词，并用最大熵模型，获得很好的效果。Peng et al. [2004] 也将分词问题看成序列标记问题，但是使用条件随机场模型，取得了比最大熵模型更好的性能。

首先，我们需要将分词问题转换为序列标记问题。我们用 $\{B, M, E, S\}$ 分别表示当前字是词的开始，中间，结尾和单字词。例如：句子“FudanNLP 主要是为中文自然语言处理而开发的工具包，也包含为实现这些任务的机器学习算法和数据集。”转化为下面以字为基本元素构成的序列。

FudanNLP	主	要	是	为	中	文	自	然	语	言	处	理	而
S	B	E	S	S	B	E	B	E	B	E	B	E	S
开	发	的	工	具	包	，	也	包	含	为	实	现	这	些	
B	E	S	B	M	E	S	S	B	E	S	B	E	B	E	
任	务	的	机	器	学	习	算	法	和	数	据	集	。		
B	E	S	B	E	B	E	B	E	S	B	M	E	S		

7.2.1 特征模板

特征的好坏直接会影响学习的性能。在中文分词中，特征一般通过特征模板进行抽取。表7.2中列出了在中文分词常用的特征模板。

表 7.2: 特征模板

单字符特征	$x_{-2}y_0, x_{-1}y_0, x_0y_0, x_1y_0, x_2y_0$ [a]
双字符特征	$x_{-1}x_0y_0, x_0x_1y_0, x_{-1}x_1y_0,$
三字符特征	$x_{-1}x_0x_1y_0$
马氏链特征	$y_{-1}y_0$

[a] 下标 $\{-2, -1, 0, 1, 2\}$ 表示该字符位置到当前分析字符位置的相对位移。

用特征模板按先后次序在文本中每个位置上进行特征抽取，就可以抽取出所有的特征。

例如，对于句子“自然语言处理”和对应的标记序列“BEBEBE”，若当前分析字符是“语”，则 x_0y_0 抽取的特征为“语 |B”，$x_{-1}x_0x_1y_0$ 对应的特征为“然语言 |B”。表7.3给出了不同特征模板抽取的相应特征。

表 7.3: 不同模板抽取特征示例

模板	特征
x_0y_0	“自 |B”，“然 |E”，“语 |B”，“言 |E”，“处 |B”，“理 |E”
$x_{-1}x_0y_0$	“SOS 自 |B” [a]，“自然 |E”，“然语 |B”，“语言 |E”，“言处 |B”，“处理 |E”
$x_0x_1y_0$	“自然 |B”，“然语 |E”，“语言 |B”，“言处 |E”，“处理 |B”，“理 EOS|E” [b]
$y_{-1}y_0$	“BE”，“EB”，“SOS-B”，“E-EOS”

[a]SOS 表示句子的开始。
[b]EOS 表示句子的结束。

7.3 基于无监督学习的中文分词

从词的定义上看，词是稳定的字的组合，因此在上下文中，相邻的字同时出现的次数越多，就越有可能构成一个词。因此相邻字的共现频率能够较好的反映其组成词的概率。可以对语料中相邻共现的各个字的组合的频度进行统计，计算它们的互现信息。定义两个字的互现信息，计算两个汉字的相邻共现概率。互现信息体现了汉字之间结合关系的紧密程度。当紧密程度高于某一个阈值时，便可认为此字组可能构成了一个词。这种方法只需对语料中的字组频度进行统计，不需要词典，也不需要标注语料。但这种方法也有一定的局限性，会经常抽出一些共现频度高、但并不是词的常用字组，例如“这一”、“之一”、“有的”、“我的”、“许多的”等，并且对常用词的识别精度差，时空开销大。

7.4 小结

实际应用的中文分词系统应该是利用结合上下文进行分词，同时可以结合少量的人工规则进行错误修正。

第八章　词性标注

词性标注对句子中的每一个词附上相应的词性。

8.1　词性

词性（Part-of-Speech, POS）指作为划分词类的根据的词的特点。现代汉语的词可以粗分为 12 类。

实词 名词、动词、形容词、数词、量词和代词

虚词 副词、介词、连词、助词、拟声词和叹词

但这样的划分对于后续的语义分析远远不够。因此不同的语料构建者都会对词性进行更细的划分。比如“名词”可以再分为“一般名词”、“专用名词”、“抽象名词”和“方位名词”等。“专用名词”可用再分为“人名”、“地名”、“机构名”等。

词性标注（POS Tagging）是给句子中每个词标记出最合适的词性。中文和英语的一个重要区别是中文没有词性变化。中文的每个词会有多种词性（比如“希望”即是名词又是动词）。但特定的使用场合下，比如一个句子中，每个词都有唯一确定的词性。即在若干词性候选项中选择一个合适的词性。

8.2　词性标注规范

中文词性标注目前还没有统一明确的标注规范。不同研究者都提出了自己的标注规范以及相应的数据集。

在众多词性标注数据集中，Penn Chinese TreeBank （CTB）[1]是广泛使用的数据集。CTB 数据集中定义的词性分为 11 个大类和 33 个小类，如表8.1所示。具体规范可参考文献Xia [2000]。

比如下面例子中，“DT”，“PN”，“VV”，“AS”，“JJ”，“DEG”，“NN”和“PU”都是 CTB 定义的词性标签。

[1] http://www.cis.upenn.edu/ chinese/ctb.html

表 8.1: Chinese TreeBank 词性标记集合

类别	词性标记
动词，形容词 (4)	VA, VC, VE, VV.
名词 (3)	NR, NT, NN.
方位词 (1)	LC.
代词 (1)	PN.
限定词和数字 (3)	DT, CD, OD.
量词 (1)	M.
副词 (1)	AD.
介词 (1)	P.
连词 (2)	CC, CS.
虚词 (8)	DEC, DEG, DER, DEV, SP, AS, ETC, SP, MSP.
其他 (8)	IJ, ON, PU, JJ, FW, LB, SB, BA.

去年/DT 他/PN 取得/VV 了/AS 可喜/JJ 的/DEG 进步/NN 。/PU

在 FudanNLP 中，词性分为：动词、能愿动词、趋向动词、把动词、被动词、形谓词、形容词、副词、名词、方位词、人名、地名、机构名、时间短语、邮件、网址、型号名、实体名、疑问代词、指示代词、人称代词、量词、介词、数词、惯用词、限定词、连词、叹词、序数词、省略词、语气词、结构助词、时态词、标点、拟声词、表情词等。

8.3 词性标注

中文的词缺乏形态变化，不能直接从词的形态变化上来判别词的类别。并且大多数词是多义的，兼类现象严重。中文词性标注要更多的依赖语义，相同词在表达不同义项时，其词性往往不一致的。因此通过查词典等简单的词性标注方法效果会比较差。

目前，有效的中文词性标注方法可以分为基于规则的方法和基于统计的方法两大类。基于规则的方法的局限性在于自然语言的复杂性，建立规则库需要大量的专家知识和很高的人工成本。基于统计学习的方法的局限性在于其严重依赖数据集的规模和质量。在近几年，由于人们可以通过较低成本获得高质量的数据集，基于统计学习的词性标注方法取得了较好的效果，并成为主流方法，常用的学习算法有隐马尔科夫模型（HMM）、最大熵模型（ME）、条件随机场（CRFs）等。

8.4 基于统计学习的词性标注方法

当一个词具有多个义项时，它相应的词性和特定场合下该词的实际语义密切相关。比如：“爱”这个词，可以有“动词”和“名词”两种词性。在下面两个句子里，“爱”分别具有不同的词性。

我/人称代词爱/动词 你/人称代词。/标点

你/人称代词给/动词我/人称代词的/结构助词爱/名词 。/标点

8.5 基于序列标注的词性标注方法

8.6 中文分词和词性联合标注方法

第九章　命名实体识别

命名实体识别任务是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等。在当今世界，随着计算机的普及以及互联网的迅猛发展，大量的信息以电子文档的形式呈现在人们面前。为了应对信息爆炸带来的严重挑战，人们迫切需要一些自动化的工具帮助他们在海量的信息源中迅速找到真正重要的信息。

命名实体识别是信息提取、问答系统、句法分析、机器翻译等应用领域的重要基础工具，在自然语言处理技术走向实用化的过程中占有重要地位。

一般来说，命名实体识别的任务就是识别出待处理文本中三大类（实体类、时间类和数字类）、七小类（人名、机构名、地名、时间、日期、货币和百分比）命名实体。

命名实体识别的过程通常包括两部分：（1）实体边界识别；（2）确定实体类别（人名、地名、机构名或其他）。

英语中的命名实体具有比较明显的形式标志（即实体中的每个词的第一个字母要大写），所以实体边界识别相对容易，任务的重点是确定实体的类别。和英语相比，汉语命名实体识别任务更加复杂，而且相对于实体类别标注子任务，实体边界的识别更加困难。汉语命名实体识别的难点主要存在于：

1. 汉语文本没有类似英文文本中空格之类的显式标示词的边界标示符，命名实体识别的第一步就是确定词的边界，即分词；
2. 汉语分词和命名实体识别互相影响；
3. 除了英语中定义的实体，外国人名译名和地名译名是存在于汉语中的两类特殊实体类型；
4. 现代汉语文本，尤其是网络汉语文本，常出现中英文交替使用，这时汉语命名实体识别的任务还包括识别其中的英文命名实体；
5. 不同的命名实体具有不同的内部特征，不可能用一个统一的模型来刻画所有的实体内部特征。

在命名实体中，时间词和数量词的识别相对容易，现行通用的是基于规则的方法；实体名（人名、地名和机构名）识别是研究的焦点。本软件工具主要对汉语人名（包括人名简称）、地名（包括地名简称）和机构名识别展开专项研究，利用大颗粒度特征（词性特征）和小颗粒度特征（词类 [1] 特征）相结合、统计模型和专家知识相结合的汉语命名实体识别模型。

第十章　句法分析

对语言的深层处理过程中，句法分析处于一个十分重要的位置。基于句法分析的很多方法、模型（比如识别短语语块，基于句法树的语言模型等）在很多应用系统中被广泛的使用。

语法是语言学的一个分支，研究按确定用法来运用的词类、词的屈折变化或表示相互关系的其他手段以及词在句中的功能和关系。包括词法和句法。词法指词的构成及变化规律；句法指短语和句子的组织规律。

句法分析（Syntactic Analysis，或 Parsing），也叫语法分析[1]，是指根据给定的自然语言语法，对句子中的词语语法功能进行分析，识别出句子的语法结构。

句法分析可以分为两个部分：一是如何定义一套符合语言学的基本原则，并且可以在计算机中表示、处理的句法描述规范。通过抽象的符号系统，对语言进行理论上的分析和表示。二是根据给定的句法描述规范，自动分析自然语言的句子。

10.1　语法理论

句法分析首先要明确自然语言的语法结构，对自然语言的语法结构进行形式化的定义。目前主流的语法结构有三种：**成分语法**、**依存语法**和**组合范畴语法**。

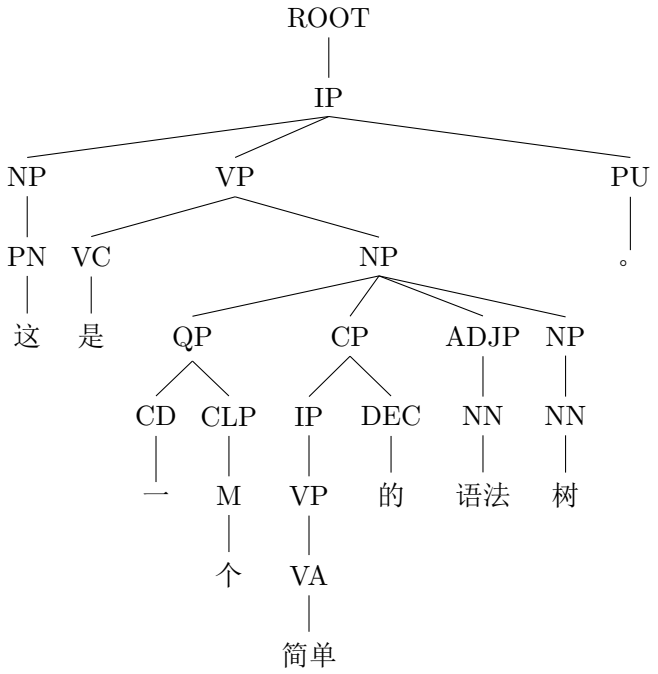
人工语言和自然语言的一个重要区别是人工语言必须是无歧义的，而自然语言中充满大量的歧义。

10.1.1　成分语法

成分语法是用层次化结构分析句子的语法成分。一般的语法成分包括：主语、谓语、宾语、定语、状语、补语。比如“我来晚了”，这里“我”是主语，“来”是谓语，“晚了”是补语。

“这是一个简单的语法树”。

[1]狭义上讲，句法是语法的一个分支，专讲句子构造的，语法则包含词、词性、句子、文法、修辞等一切现象。



事实上，我们对句法分析并不陌生。句法分析不仅仅面向自然语言，也是我们各种人工语言（比如 XML 语言、HTML 语言以及各种程序语言）中不可缺少的一部分。在自然语言的计算机处理中，成分句法主要是使用 Chomsky 的上下文无关语法。

下面我们先介绍些形式语言。

10.1.2 形式语言

形式语言是按一定规律构成的句子或符号串的有限或无限的集合.

形式文法: 是一种格式, 用来说明什么句子在该语言中是合法的，并指明把词组合成短语和句子的规则。现行的形式语法系统是 Chomsky 于 1959 年为了描述自然语言而提出的一种理论模型。

一个形式文法 G 由四个部分 $\{VN, VT, S, P\}$ 组成，其中：

1. VN 是文法 G 的非终结符号集合。VN 不出现在 G 所表示的语言集合的句子中。
2. VT 是文法 G 的终结符号集合。G 所表示的语言的句子由 VT 中的元素组成。$VN \bigcap VT = \emptyset$。$S$ 代表开始符号，$S \in VN$。
3. P 代表产生式的集合，P 中的产生式具有如下形式：$\alpha \to \beta$。
4. 产生式需要满足下面的条件：1)α 可以是 VN 和 VT 上的任意字符串，但其中必须至少包含一个非终结符, 并且不能是空字符；2)β 可以是 VN 和 VT 上的任意字符串，也可以是空字符；3)P 中至少有一个产生式中的 α 得由 S 来充当。

形式语言具有以下特点：

1. 高度的抽象化 (采用形式化的手段 -专用符号, 数学公式 -来描述语言的结构关系, 这种结构关系是抽象的)

2. 是一套演绎系统 (形式语言本身的目的就是要用有限的规则来推导语言中无限的句子, 提出形式语言的哲学基础也是想用演绎的方法来研究自然语言)

3. 具有算法的特点.(比如说句法分析中采用不同的算法来构造句子的句法推导树)

Chomsky 把文法分成 4 种类型，即 0 型，1 型，2 型，和 3 型。

1. 0 型文法也称短语文法，0 型文法的能力相当于图灵机（Turing），或者说任何 0 型语言都是递归可枚举的。

2. 1 型文法也称上下文有关法，其能力相当于线性界限自动机。

3. 2 型文法也称上下文无关法，其能力相当于非确定的下推自动机。

4. 3 型文法也称右线性文法，由于这种文法等价于正规式，所以也称正规文法。

从文法描述语言的能力来说，0 型文法最强，3 型文法最弱。

10.2 成分句法分析

在上下文无关语法的基础上，学者们提出了自顶向下分析法、自底向上分析法、左角分析法、CYK 算法、Earley 算法、线图分析法等行之有效的分析技术。但是，这些分析方法在处理自然语言的歧义时都显得无能为力。近年来对上下文无关语法的改进主要体现在两个方面：一方面是给上下文无关语法的规则加上概率，提出了概率上下文无关语法，另一方面是除了给规则加概率之外，还考虑规则的中心词对于规则概率的影响，提出了概率词汇化上下文无关语法。

这些问题的共同目标是构建这样的一个系统：对于任意的句子都能够主产生证明有用的结构，也就是要构建一个句法分析器。 句法分析的三种不同的途径可以利用概率： 1、利用概率来确定句子：一种可能的做法是将句法分析器看成是一个词语网络上的语言模型，用来确定什么样的词序列经过网络的时候会获得最大概率。 2、利用概率来加速语法分析：第二个目标是利用概率对句法分析器的搜索空间进行排序或剪枝。这使得句法分析器能够在不影响结果质量的情况下尽快找到最优的分析途径。 3、利用概率选择句法分析结果：句法分析器可以从输入句子的众多分析结果中选择可能性最大的。

第十一章　依存句法分析

理解一个句子，就是找出句子中各个词之间的所有联系。–Lucien Tesnière

依存文法在语言学上的研究一般认为开始于 Tesnière 提出的 stemma 结构。Tesnière 认为每个句子都拥有一个有组织的完整的内在结构，其基本构成成分为句子中的词，和词与词之间存在着的联系，所有的这些联系就构成了句子的结构，其中结构性的联系由可以建立词之间的依存关系，即支配词 (governer) 和从属词 (dependent) 联系起来。

11.1　依存句法

依存句法是描述句子内部各个词之间的**依存关系**。与成分句法不同，结构没有非终结点，词与词之间直接发生依存关系，构成一个依存对，其中一个是**支配词**，另一个是**从属词**，也叫**修饰词**。依存关系是非对称的，可以用有向边来表示。由从属词指向支配词。一个从属词只能依赖到一个支配词。而一个支配词可以支配多个从属词。一个从属词可以是其它词的支配词，一个支配词也可以是其它词的从属词。一个句子中不能含有循环。这样依存句法结构也可以看出是一颗依存句法树。

此外，如果我们约束依存树中不存在交叉边，我们这颗树为**投影依存树**。否则，若依存树中存在交叉边，我们这颗树为**非投影依存树**。

一般一个句子包含一个**核心词**，也就是句子中的核心谓语动词、名词或形谓词，对主语加以陈述，说明主语“做什么”、“怎么样”或者“是什么”。核心词作为依存树的根节点。

每个依存对可以属于不同的**关系类型**，表示该依存对中的两个词之间存在什么样的依存关系。关系类型包括：主语、定语、补语等。

对于依存文法的语言学研究中，始终没有非常明确的定义什么是依存关系。因而当在一个语言结构中讨论依存关系以及如何确定中心词和修饰词时，往往只能依赖于一些标准。

Hudson [1990] 就对于一个依存关系结构 C 中如何确定支配词 H 和从属词 M 的标准总结如下：

1. H 决定了 C 的语法类型，H 常常可以取代 C。

2. H 决定了 C 的语义类型，M 在语义上对 H 作了限制。

3. H 是必需的，M 是省略的。

4. H 决定了 M 是可选的还是省略的。

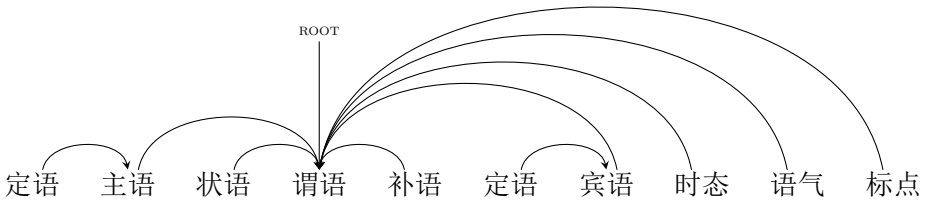
5. H 决定了 M 的形式。从属词的形式包括指示代词的单复数 (this，these)，人称代词的单复数、格等。

6. H 决定了 M 和 H 在句子中的相对位置。

这些准则涉及语法和语义这两个不同的层面，因而的确很难给出一种符合所有标准的依存关系定义。基于这点，许多理论研究者主张需要对不同类型的依存关系区别对待。Mel čuk [1988] 将依存关系分为词法的 (morphological)，句法的 (syntactic) 和语义的 (semantic)。Nikula [1986] 则认为需要区分依存关系的内在结构 (endocentric construction) 和外在结构 (exocentric construction)。在内在结构中，删去从属词并不影响句子结构，即满足第一条准则；而在外在结构中，支配词无法直接取代整个结构，从属词是不可缺少的。

11.1.1 中文依存句法

在现代汉语中，句子的一般结构为：（定）主 +[状] 谓 < 补 >+（定）宾。用依存树表示为：

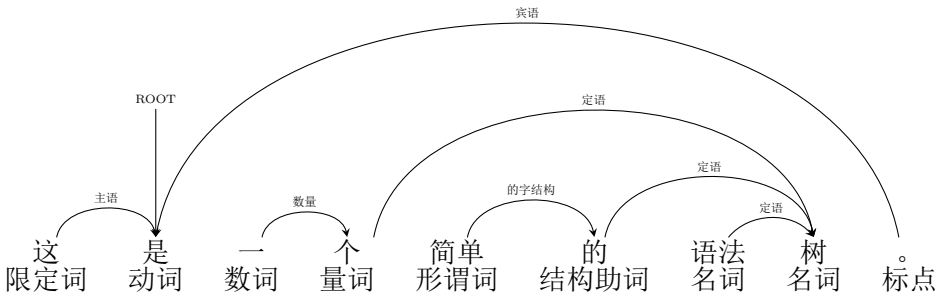


在典型的中文句子中，有些依存关系是比较明显的。比如主谓、谓宾所构成的依存关系显然是以谓语词作为支配词的外在结构，而形容词修饰名词构成的依存关系显然是内在结构。但也有一些结构，在各种依存句法研究中如何定义依存关系并不同意，这些结构主要包括两大类：

1. 包含功能词的结构，这里功能词包括介词、连词、结构助词等。比如对于助动词，有的理论将助动词作为支配词，实际动词作为从属词，而有的理论则恰恰相反。

2. 并列结构。并列结构的每个元素在语义上是并列的，因而并没有对于任意两个元素，并没有显然的依存关系存在。对于这些结构，不同的依存文法理论会使用不同的假设和处理策略。

下面是一个依存树的图形表示：



11.1.2 依存句法的优点

在目前基于统计的句法分析方法中，依存句法有着下面几个优点，并逐渐受到研究人员的重视。

1. 形式简单，容易表示。

2. 易于构造基于统计模型的高效、高精度分析工具。

3. 易于理解，具有更好的心理对照，标注难度低。

4. 更容易进行从句法到语义的转换。

11.2 依存句法分析

对于依存文法分析，也有这两种类型的方法：句法驱动的分析方法和数据驱动的分析方法。句法驱动的依存文法分析方法需要在给定文法的基础上对句子进行分析，因而这些方法经常伴随着对于依存文法形式化定义的研究。

然而依存概念的模糊性造成了依存文法理论的多样性，因而现在的依存文法分析研究主要以数据驱动。数据驱动的依存文法分析可以概括为下面 3 步：

1. 建模：如何计算输入句子的一个依存结构的分数

2. 参数估计：利用标注数据或者来标注数据估计模型中的参数

3. 推断：搜索对于当前的输入句子，分数最大的一个依存结构

目前，基于统计的依存句法分析主要分为两类：

目前基于数据驱动的依存语法分析方法又分为两类：基于图的依存句法分析 [McDonald et al., 2005] 和基于转换的依存句法分析 [Nivre and Nilsson, 2005]。其代表性成果分别为：MaltParser 和 MSTParser。

在基于图的依存句法分析中，通过动态规划或近似算法，得到一棵满足依存关系约束的全局最优的依存树。时间复杂至少为 $O(n^3)$。

在基于转换的依存句法分析中，词汇之间的依存关系是用贪婪的策略逐步决定的。由输入句子构造初始状态，在当前状态的基础上，执行某一动作，转换到一个新的状态。最终的状态包括了一棵完整的依存分析树。通过定义不同的状态和转换动作，分别由不同的分析算法。由于使用了贪婪的策略，时间复杂度一般为 $O(n)$ 或 $O(n^2)$。

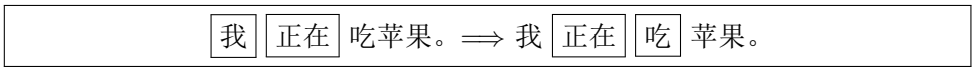
11.3 基于转换的依存句法分析

在基于转换的分析方法中，依存分析被看作是对输入句子执行若干动作，由这些动作建立起句子中词与词之间的联系。每一个动作都将当前的分析状态转换到新的状态。基于转换的分析方法并不搜索全局最优的动作序列，而是采用贪婪的策略，根据当前状态选择局部最优的动作，一个动作一旦执行就不会在更改，因而又称确定性分析方法。

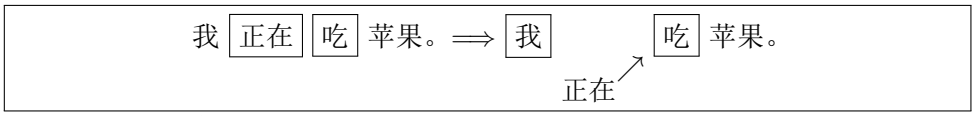
11.3.1 Yamada 句法分析

在 Yamada 分析器的每个状态中，都有一对相邻的词作为焦点词。初始状态包括输入的句子，即一个词的序列，并且设置焦点词为序列最左边的第一个和第二个词。针对这两个焦点词的依赖关系，有 SHIFT，LEFT，RIGHT 三个动作。

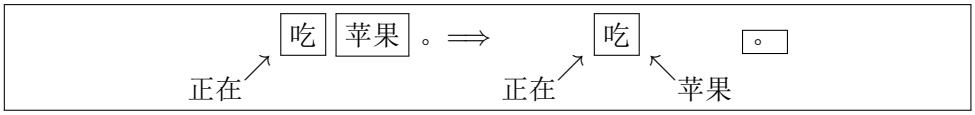
SHIFT：不建立依赖关系，焦点词向右移动一个词，如图：



RIGHT：建立一条向右的弧，即建立左焦点词依赖于右焦点词的依赖关系，并将左焦点词从序列中删去，并更新左焦点词，如图。



LEFT：建立一条向左的弧，即建立右焦点词依赖于左焦点词的依赖关系，并将右焦点词从序列中删去，并更新右焦点词，如图，需要指出的是当原右焦点词为句子的末尾时，焦点词重新返回到序列最左边的第一第二个词；



在做语法分析时，一个 3 类分类器会对当前的状态作出动作的预测。特征主要由焦点词及其上下文构成。

11.4 评测指标

依存句法的评测方法可以使用依赖准确率 DA，根词准确率 RA，全句准确率 CA 作为评测指标，分别定义如下：

依赖准确率 DA：在所有非根词 (排除标点) 中，被指定正确的中心词的词所占的比例。

$$DA = \frac{\text{正确识别的支配词个数}}{\text{识别的支配词个数}} \tag{11-1}$$

核心词准确率 RA：在所有核心词，被正确识别的比例。

$$RA = \frac{\text{正确识别的核心词个数}}{\text{核心词总数}} \tag{11-2}$$

全句准确率 CA：整句的依赖关系全部正确的句子所占的比例。

$$CA = \frac{\text{正确识别的句子数}}{\text{句子总数}} \tag{11-3}$$

在同时识别依赖关系类型的评测是，一般使用有标签依存关系的准确率 (Labeled Attachment Score，LAS) 和无标签依存关系的准确率 (Unlabeled Attachment Score，UAS) 两种评测指标。这两种评测指标都不区分根词和非根词。

$$LAS = \frac{\text{正确识别支配词和关系类型的个数}}{\text{总词数}} \tag{11-4}$$

$$UAS = \frac{\text{正确识别支配词的个数}}{\text{总词数}} \tag{11-5}$$

第十二章　关键词抽取

关键词是代表一篇文档的主要内容语。目前主流的搜索引擎也是根据关键词进行搜索的。

关键词抽取是中文信息处理的一个关键环节，在自动文摘、信息检索、文本分类、文本聚类等方面具有十分重要的作用。传统的通过人工抽取关键词方式不仅费时费力，而且在信息资源极大丰富的今天几乎是不可能完成任务，所以通过计算机对输入的文本自动抽取出相应的关键词就显得特别重要。

FudanNLP 中的关键词抽取主要参考下面文献：

R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4, pages 404–411. Barcelona: ACL, 2004

第十三章　总结

FudanNLP 是以统计机器学习为基础，并结合人工规则来处理中文自然语言的各种任务，并应用到信息检索、信息抽取、语义理解等各种实际系统中。

远景目标是实现一个更够实际应用的中文自然语言处理产品，虽然达到这个目标的距离还比较远。

参考文献

王厚峰. 指代消解的基本方法和实现技术. 中文信息学报, 16(006):9–17, 2002.

J. Allen. *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc. Redwood City, CA, USA, 1995.

N. Chomsky and G.A. Miller. *Introduction to the formal analysis of natural languages*. Wiley, 1963.

Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 2002.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge Univ Pr, 2000.

R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley, New York, 2nd edition, 2001. ISBN 0471056693.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.

R. Hudson. *English Word Grammar*. Basil Blackwell, Oxford, 1990.

T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. *Proc. of Euro. Conf. on Mach. Learn. (ECML)*, pages 137–142, 1998.

M.I. Jordan. *Learning in Graphical Models*. Kluwer Academic Publishers, 1998.

D. Jurafsky, J.H. Martin, A. Kehler, K. Vander Linden, and N. Ward. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, volume 2. Prentice Hall New Jersey, 2000.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL http://portal.acm.org/citation.cfm?id=645530.655813.

C.D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT Press, 1999.

A. McCallum, D. Freitag, and F. Pereira. Maximum entropy markov models for information extraction and segmentation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 591–598. Citeseer, 2000.

Ryan McDonald, Koby Crammer, and Fernando Pereira. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 91–98, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1219840.1219852. URL http://dx.doi.org/10.3115/1219840.1219852.

I.A. Mel čuk. *Dependency syntax: theory and practice*. State University of New York Press, 1988.

R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4, pages 404–411. Barcelona: ACL, 2004.

T.M. Mitchell. *Machine learning*. Burr Ridge, IL: McGraw Hill, 1997.

H. Nikula. *Dependensgrammatik*. LiberFörlag, 1986.

Joakim Nivre and Jens Nilsson. Pseudo-projective dependency parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 99–106, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P05/P05-1013.

F. Peng, F. Feng, and A. McCallum. Chinese segmentation and new word detection using conditional random fields. *Proceedings of the 20th international conference on Computational Linguistics*, 2004.

L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386–408, 1958.

F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys*, 34(1): 1–47, 2002.

C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. *Introduction to statistical relational learning*, page 93, 2007.

Ben Taskar, Carlos Guestrin, and Daphne Koller. Max-margin markov networks. In *In proceedings of the 17th Annual Conference on Neural Information Processing Systems*, Whistler, B.C., Canada, 2003.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning(ICML)*, 2004.

I.H. Witten, E. Frank, and M.A. Hall. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

F. Xia. *The part-of-speech tagging guidelines for the penn chinese treebank (3.0)*, 2000.

N. Xue. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48, 2003.

Y. Yang. A study of thresholding strategies for text categorization. In *Proc. of Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval(SIGIR)*, pages 137–145. ACM Press New York, NY, USA, 2001.

Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proc. of SIGIR*. ACM Press New York, NY, USA, 1999.