计算语言学

第 1 讲　概论

刘群
中国科学院计算技术研究所
liuqun@ict.ac.cn

中国科学院研究生院 2011 年春季课程讲义

内容提要

- 计算语言学所要解决的问题
- 计算语言学的定义和特点
- 课程安排

问题驱动的学习 (1)

要了解一门学科，首先要知道这门学科所要解决的问题。只有了解了一门学科所要解决的问题，才能真正理解一门学科的内在逻辑，才能不仅知其然，而且知其所以然。

在学习一门学科之前，不妨抛开这门学科的所有知识，直接面对这门学科所面对的最基本的问题，想一想如果要你来解决这个问题，你会用什么办法。然后在学习的过程中，不断地用你学到的知识来印证你所要解决的问题，才能深刻地理解你所学到的知识，真正做到融会贯通。

以色列的“巴比通天塔”纪念币 (1)

“圣经故事”系列是以色列纪念币发行中的延续性项目之一。今年发行的是第七套，它选取的题材为招致上帝愤怒的“巴比通天塔”。

《旧约·创世纪》第 11 章讲述了“通天塔”的故事。很久很久以前，天下的人都居住在一个叫做古巴比伦的地方，那时候人们都使用同一种语言。后来，古巴比伦人计划修建一座塔，塔顶要高耸入云，直达天庭，以显示人们的团结和力量。塔越建越高，惊动了天庭的耶和华。他想，现在天下的人都是一个民族，都说一种语言，他们团结一致，什么奇迹都可以创造，那神还怎么去统治人类？于是上帝便决定要惩罚惩罚人类。他施魔法变乱了人们的口音，使他们无法沟通，高塔因此无法继续建造下去。最后，上帝还把人类驱散到地球的各个角落。

纪念币的正面图案由 Moshe Pereg 设计。只见一座高塔越过山、穿过云，直达天庭，它是由这个圣经故事的文字艺术地组合而成的。然而，每一个字又似乎是漂浮在空中，影射了缺乏沟通的语言。

“巴比塔”故事的中心是文字，因此纪念币的背面图案也主要由文字组成。它们呈环层型分布，给人一个从塔的上方向下观望的印象。

……

—— 摘自中国金币网（http://www.chinagoldcoin.net/）

以色列的“巴比通天塔”纪念币 (2)

问题之一：自动翻译 (1)

- 自动翻译问题
  - 人类最古老的问题之一：《圣经》上巴比伦塔的传说
  - 计算机发明以来，几乎是人类首先想到的计算机的应用之一
  - 困难比人们想像的大得多，至今没有找到理想的解决办法
  - 是推动计算语言学研究的永恒动力
  - 是计算语言学的终极目标
  - 自动翻译研究，几乎触及到计算语言学研究的所有方方面面

问题之一：自动翻译 (2)

- 逐字翻译的错误（一个笑话）
  - 源文： How are you?
  - 逐字翻译的译文：怎么是你？
  - 源文： How old are you?
  - 逐字翻译的译文：怎么老是你？

问题之一：自动翻译 (3)

- 演示
  - Light 英汉机器翻译系统
  - TransEasy 汉英机器翻译系统
- 目前的水平：
  - Babelfish (点击进入)
  - Google语言工具 (点击进入)
  - WordLingo (点击进入)
  - 中科院计算所机器翻译在线演示（点击进入）
  - 华建翻译中心 (点击进入)

问题之二：自动问答

- 图灵测试
  - 用于判断“机器是否具有智能”
  - 一个封闭的小屋
  - 屋外有一个人
  - 屋内有一个人或者一台计算机
  - 屋外的人并不知道屋内的是人还是计算机
  - 屋外的人向屋内的人或者计算机提出各种问题
  - 屋外的人根据回答来判断屋内是人还是计算机
  - 如果判断不出来，那么可以认为计算机具有了智能

问题之二：自动问答（例子）

User:	ELIZA:
Men are all alike.	In what way?
They're always bugging us about something or other.	Can you think of a specific example?
Well, my boyfriend made me come here.	Your boyfriend made you come here.
He says I'm depressed much of time.	I am sorry to hear you are depressed.
It's true, I am unhappy.	Do you think coming here will help you not be unhappy?
I need some help; that much seems certain.	What would it mean to you if you got some help?
Perhaps I could learn to get along with my mother.	Tell me more about your family.
My mother takes care of me.	Who else in your family takes care of you?
My father.	Your father.
You are like my father in some ways.	What resemblance do you see?

问题之二：自动问答（演示）

- 演示
  - 机器人心理医生 Frank （ Eliza 的变体）
  - Alice
    - 获 2000 Loebner Prize
    - 基于 AIML ，开放源代码，变种众多
- 目前水平
  - AskJeeves 、百度知道、新浪爱问、雅虎知识堂
  - TREC 的 QA Track
  - PowerSet ： Natural Language Search ？

其他问题

- 音字转换：语音识别、拼音输入
- 自动文摘：自动给出一篇或多篇文章的摘要
- 信息检索：在海量的信息准确找到你所需要的信息
- 信息过滤：从信息流中筛选出你所感兴趣的信息
- 信息抽取：从海量的信息中抽取出你所需要的（结构化）信息
- ……

问题驱动的学习 (2)

- 本课程采用问题驱动的学习方法
- 我们将围绕“机器翻译”和“自动问答”这两个主要问题，同时兼顾其他问题，来展开“计算语言学”这门课程的学习
- 通过大量的实例来加深对课程的理解
- 侧重汉语的处理

内容提要

- 计算语言学所要解决的问题
- 计算语言学的定义和特点
- 课程安排

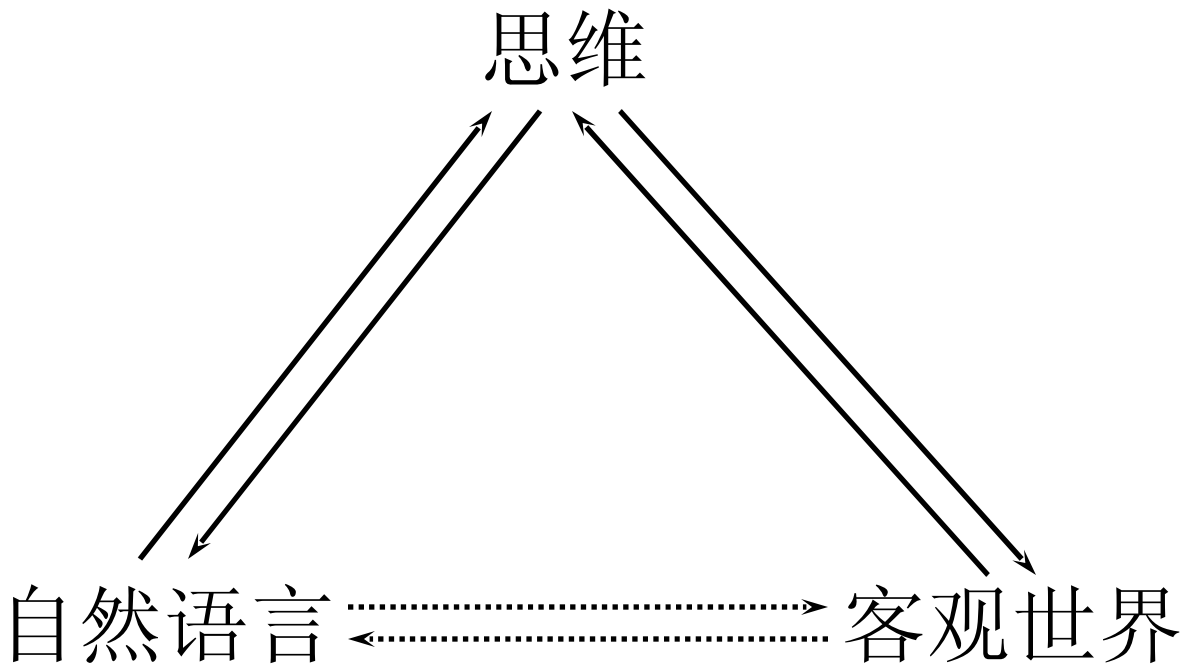
计算语言学定义

计算语言学是一门以**计算**为手段对**自然语言**进行**研究**和**处理**的科学。

计算语言学的研究对象

- 计算语言学的研究对象是自然语言
- 自然语言与形式语言的本质区别是歧义性
- 自然语言是一种符号系统
- 语言符号的特点（索绪尔）
  - 任意性：语言符号的选择是任意的
  - 线条性：语言符号的排列是线性的

语言、思维与客观世界



语言的层面 (1)

- 语言研究的层面
  - 语音
  - 语法（包括词汇层和句法层）
    - 语法研究要回答的问题是：一句话为什么可以这么说而不能那么说？
  - 语义
    - 语义研究要回答的问题是：这句话说了什么？
  - 语用
    - 语用研究要回答的问题是：为什么要说这句话？

语言的层面 (2)

- 语言各层面之间的关系
  - 语言层面的划分反映了语言在不同层次上的规律性
  - 语言的各个层面是互相交织密不可分的，语言层面的划分只是为了研究方便，对任何一个层面的研究都不能忽略其他层面所起的作用

语言在不同层面的歧义性 (1)

- 语音层面：多音字，同音词
  - 施氏食狮史（赵元任）

    石室诗士施氏，嗜狮，誓食十狮。氏时时适市视狮，十时，适十狮适市，是时，适施氏适市，施氏视是十狮，拭矢试，使是十狮逝世，适石室，石室湿，氏使侍拭石室，石室拭，始食是十狮尸，始识是十狮尸，实十石狮尸，试释是事。

语言在不同层面的歧义性 (1+)

赵元任的另一篇同音古文：（全文）

羿裔熠①，邑②彝，义医，艺诣．

熠姨遗一裔伊③，伊仪迤，衣旖，异奕矣．

熠意④伊矣，易衣以贻伊，伊遗衣，衣异衣以意异熠，熠抑矣．

伊驿邑，弋一翳⑤，弈毅⑥．毅仪奕，诣弈，衣异，意逸．毅诣伊，益伊，伊怡，已臆⑦毅矣，毅亦怡伊．

翌，伊亦弈毅．毅以蜴贻伊，伊亦贻衣以毅．

伊疫，呓毅，瘪异矣，倚椅咿咿，毅亦咿咿．

毅诣熠，意以熠，议熠医伊，熠懿⑧毅，意役毅逸．毅以熠宜伊，翼逸．

熠驿邑以医伊，疑伊胰痍⑨，以蚁医伊，伊遗异，溢，伊咦．熠移伊，刈薏⑩以医，伊益矣．

伊忆毅，亦呓毅矣，熠意伊毅已逸，熠意役伊．伊异，噫，缢．

熠瘪，亦缢．

语言在不同层面的歧义性 (2)

- 语法层面
  - 词法歧义
    - 词性兼类：工作（动名兼类），在（动副兼类）
    - 词语切分歧义：乒乓球拍卖完了，鱼在长江中游
  - 句法歧义
    - 结构歧义：张三和李四的朋友
    - 组合关系歧义：观赏鱼

语言在不同层面的歧义性 (3)

- 语义层面
  - 一词多义：后门，人大，

    I can can the can in the can.
  - 结构语义歧义：吃饭，吃食堂，吃大碗
- 语用层面
  - 鸡蛋！
  - 他去修车了。

一个笑话：请客 (1)

旧时年关，有人在家设宴招待帮助过他的人，一共请了四位客人。

时近中午，还有一人未到。于是自言自语：“该来的怎么还不来？”，听到这话，其中一位客人心想：“该来的还不来，那么我是不该来了？”，于是起身告辞而去。其人很后悔自己说错了话，说：“不该走的又走了”，另一位客人心想：“不该走的走了，看来我是该走的！”，也告辞而去。主人见因自己言语不慎，把客人气走了，十分懊悔。妻子也埋怨他不会说话，于是辩解道：“我说的不是他们。”最后一位客人一听这话，心想“不是他们！ 那只有是我了！”，于是叹了口气，也走了。

一个笑话：请客 (2)

- 对比：
  - **该来的**没来 vs 他**应该**来
  - **不该走的**走了 vs 他**不该**走
  - 说的**不是他们** vs ？？？
- 分析原因：焦点错误（语用范畴）
  - 句子中一般都有一个焦点，焦点的作用是强调焦点处前景和背景的区别，焦点影响句子的言外之意
    - **他**应该来：强调“他”与其他人的区别，言外之意是别人不应该来
    - 他**应该**来：强调“应该”与“不应该”的区别，言外之意是他不应该不来
  - 句子的焦点通过位置、重音、标记词或其他方式来体现
  - 汉语句子的焦点通常在句尾
  - “ 该…的”、“是”在汉语中是焦点的标记

一个故事：曾国藩改奏折

- 晚清时期，两江总督曾国藩的下属起草了一份奏折，说最近打仗“屡战屡败”，曾国藩看后大笔一挥，把“战”和“败”二字调了个个，变成了“屡败屡战”，同样是四个字，只不过位置换了一下，看似不经意的改动却使通篇文章的精神大变。
- 分析原因：汉语句子通常焦点在句尾
  - 屡战屡败：强调“败”与“不败”的区别
  - 屡败屡战：强调“战”与“不战”的区别

汉语的特点 (1)

- 语言的分类
  - 汉语：孤立语（分析语）
  - 英语：屈折语
  - 日语：粘着语
- 基本单位
  - 汉语：汉字（单音节，不用空格分隔）
  - 英语：词（多音节，用空格分隔）
- 词语形态变化
  - 汉语：弱（重叠、离合）
  - 英语：强（屈折）

汉语的特点 (2)

- 语言的层次划分
  - 汉语：不明显：字与词、词与语、语与句、句与段，都没有明确的界限
  - 英语：明显：词、短语、子句、句子、段落之间界限分明
- 词类与句法功能的对应
  - 汉语：多对多
  - 英语：一对一

计算语言学的研究手段

- 计算语言学的研究手段是计算
- 计算的基础是冯·诺依曼结构的计算机
- 计算的表现形式是算法
- 算法：一组有穷的操作规则
  - 确定性：每一个步骤的结果都是确定的
  - 可行性：每一个步骤可在有限时间内完成
  - 输入：有输入
  - 输出：有输出
  - 有穷性：可在有限步骤内停止
- 算法和程序的联系与区别

建议的编程语言与工具

- 编程语言： Python
- 编程工具： NLTK

什么是 Python

- 一种脚本语言（擅长处理文本）
  - 解释型而非编译行，类似于 Perl，但更精巧
- 一种很高级的语言（跟接近人的思维）
  - 类似于 Java，而不像 C 或者汇编语言
- 过程型语言
  - 类似于 C、Pascal、Basic 等
- 面向对象语言
  - 类似于 C++、Java
- 函数式语言
  - 类似于 ML、Scheme、Haskel

Python 的特点

- 历史短（不到 10 年），但应用广泛
  - 大范围应用，尤其在 AI 和 Web 领域
- 非常容易学习
  - 很多学校用把 Python 作为入门语言
- 编程方便
  - 相比 C 、 C++ 、 Java 代码短得多
- 容易阅读与维护
  - 类似于自然语言和数学公式的语法

Python 的类型特点

- 强类型：数据类型要求严格，不存在隐式的类型转换
- 弱类型：数据类型要求不严格，存在隐式类型转换
- 静态类型：编译时类型检查
- 动态类型：运行时类型检查

	weak	strong
static	C, C++	Java, Pascal
dynamic	Perl, VB	Python, OCaml, Scheme

Python 编程实例

- 以下 Python 编程实例摘自 Liang Huang 的 Python 讲义，在此表示感谢！

“Hello World!”

- C

```c
#include <stdio.h>

int main(int argc, char ** argv)
{
   printf(“Hello, World!\n”);
}
```

- Java

```java
public class Hello
{
   public static void main(String argv[])
   {
      System.out.println(“Hello, World!”);
   }
}
```

- now in Python

```
print “Hello, World!”
```

Print an Array

```c
void print_array(char* a[], int len)
{
   int i;
   for (i = 0; i < len; i++)
   {
      printf("%s\n", a[i]);
   }
}
```

has to specify `len`,
and only for one type (`char*`)

C

```python
for element in list:
    print element
```

only indentations
no { ... } blocks!

for ... in ... :
    ...

no C-style for-loops!

~~`for (i = 0; i < 10; i++)`~~

Python

or even simpler:

```python
print list
```

Reverse an Array

```
static int[] reverse_array(int a[])
{
   int [] temp = new int[ a.length ];
   for (int i = 0; i < len; i++)
   {
      temp [i] = a [a.length - i - 1];
   }
   return temp;
}
```

Java

```
def rev(a):
   if a == []:
      return []
   else:
      return rev(a[1:]) + [a[0]]
```

def ...(...):
...

Python

no need to specify argument and return types! python will figure it out. (dynamically typed)

a without a[0]

singleton list

or even simpler:

```
a.reverse()
```

← built-in list-processing function

Quick Sort

```java
public void sort(int low, int high)
{
    if (low >= high) return;
    int p = partition(low, high);
    sort(low, p);
    sort(p + 1, high);
}

void swap(int i, int j)
{
    int temp = a[i];
    a[i] = a[j];
    a[j] = temp;
}
```

```java
int partition(int low, int high)
{
    int pivot = a[low];
    int i = low - 1;
    int j = high + 1;
    while (i < j)
    {
        i++; while (a[i] < pivot) i++;
        j--; while (a[j] > pivot) j--;
        if (i < j) swap(i, j);
    }
    return j;
}
```

Java

```python
def sort(a):
    if a == []:
        return []
    else:
        pivot = a[0]
        left  = [x for x in a if x < pivot ]
        right = [x for x in a[1:] if x >= pivot]
        return sort(left) + [pivot] + sort(right)
```

$$\{x \mid x \in a,\ x < pivot\}$$

Python

smaller semantic-gap!

找出文件中所有 ing 结尾的单词

```python
for line in open("file.txt"):     # for each line of the text file
    for word in line.split():     # for each word in the line
        if word.endswith('ing'): # does the word end in 'ing'?
            print word            # if so, print the word
```

Python

```c
int main(int argc, char **argv) {
   int i = 0;
   int c = 1;
   char buffer[1024];

   while (c != EOF) {
       c = fgetc(stdin);
       if ( (c >= '0' && c <= '9') || (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z') ) {
           buffer[i++] = (char) c;
           continue;
       } else {
           if (i > 2 && (strncmp(buffer+i-3, "ing", 3) == 0 || strncmp(buffer+i-3, "ING", 3) == 0 ) ) {
               buffer[i] = 0;
               puts(buffer);
           }
           i = 0;
       }
   }
   return 0;
}
```

C

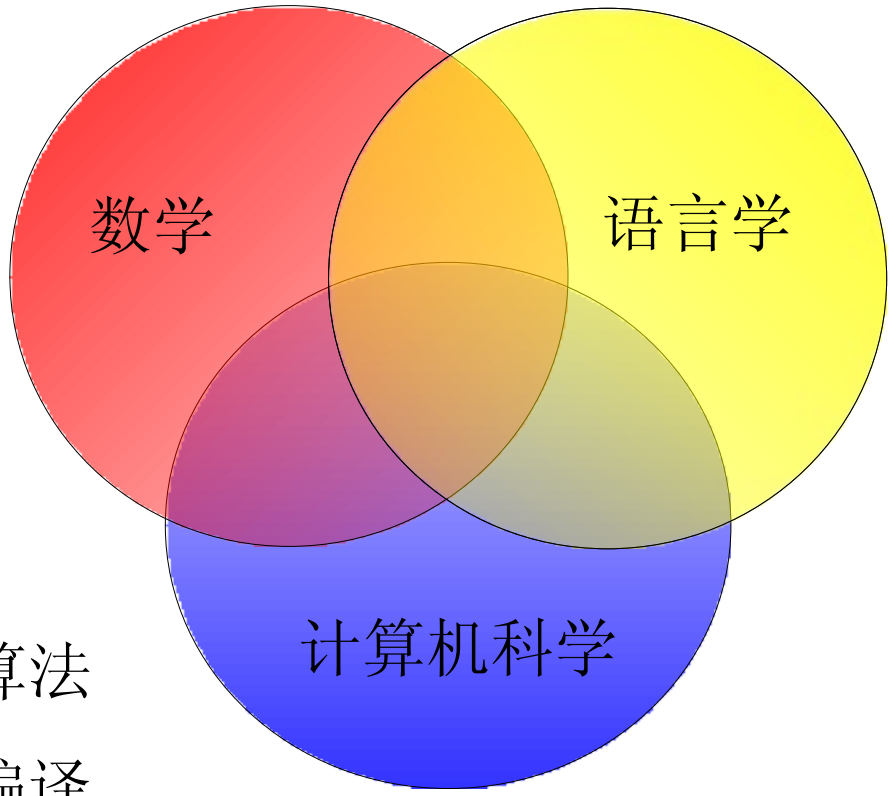
NLTK 工具

- NLTK： Natural Language Toolkit
- NLTK 是用 Python 实现的一套自然语言处理软件工具
- NLTK 包括：
  - 与 NLP 相关的基本数据类型
  - NLP 的标准函数接口：切词、标注、句法分析等
  - NLP 中常见任务的具体实现
  - NLP 任务演示（句法分析、组块分析、聊天机器人）
  - 详尽的文档、教程与参考书目
  - 随工具包发布的数据：词典、语料库等等

随 NLTK 发布的语言数据

- Australian ABC News, 2 genres, 660k words, sentence-segmented
- Brown Corpus, 15 genres, 1.15M words, tagged
- CMU Pronouncing Dictionary, 127k entries
- CoNLL 2000 Chunking Data, 270k words, tagged and chunked
- CoNLL 2002 Named Entity, 700k words, pos- and named-entity-tagged (Dutch, Spanish)
- Floresta Treebank, 9k sentences (Portuguese)
- Genesis Corpus, 6 texts, 200k words, 6 languages
- Gutenberg (sel), 14 texts, 1.7M words
- Indian POS-Tagged Corpus, 60k words pos-tagged (Bangla, Hindi, Marathi, Telugu)
- NIST 1999 Info Extr (sel), 63k words, newswire and named-entity SGML markup
- Names Corpus, 8k male and female names
- PP Attachment Corpus, 28k prepositional phrases, tagged as noun or verb modifiers
- Presidential Addresses, 485k words, formatted text
- Roget's Thesaurus, 200k words, formatted text
- SEMCOR, 880k words, part-of-speech and sense tagged
- SENSEVAL 2, 600k words, part-of-speech and sense tagged
- Shakespeare XML Corpus (sel), 8 books
- Stopwords Corpus, 2,400 stopwords for 11 languages
- Switchboard Corpus (sel), 36 phonecalls, transcribed, parsed
- Univ Decl Human Rights, 480k words, 300+ languages
- US Pres Addr Corpus, 480k words
- Penn Treebank (sel), 40k words, tagged and parsed
- TIMIT Corpus (sel), audio files and transcripts for 16 speakers
- Wordlist Corpus, 960k words and 20k affixes for 8 languages
- WordNet, 145k synonym sets

计算语言学与其他学科的关系



代数
逻辑
统计

词法
句法
语义
词汇
语料库

算法
编译

人工智能

内容提要

- 计算语言学所要解决的问题
- 计算语言学的定义和特点
- 课程安排

课程安排

第 1 讲：概论

第 2 讲：基础知识

第 3-6 讲：词法分析

第 7-9 讲：句法分析

第 10 讲：语义和语用分析

第 11-12 讲：应用

复习与考试

授课形式

- 讲课
- 讨论：每次课都留出 15-30 分钟讨论时间
- 作业
  - 项目作业：汉语词语切分系统
  - 项目作业：汉语句法分析系统
  - 翻译作业：英文论文翻译

评分

- 听课： 30%
- 作业： 30%
- 考试： 40%
  - 填空
  - 名词解释
  - 问答

2009 年春季学期教学日历

月份	二月		三月				四月				五月					六月
周次	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
星期一	21	28	7	14	21	28	4	11	18	25	2	9	16	23	30	6
星期二	22	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7
星期三	23	2	9	16	23	30	6	13	20	27	4	11	18	25	1	8
星期四	24	3	10	17	24	31	7	14	21	28	5	12	19	26	2	9
星期五	25	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10
星期六	26	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11
星期日	27	6	13	20	27	3	10	17	24	1	8	15	22	29	5	12
说明	每周四晚上第 5-7 节课(晚上 7:00-9:40)；上课地点：N301 教室。															

网络资源

- ACL主页　ACL文库
- NLP新闻组
- LDC（ Language Data Consortium ）
- ChineseLDC
- 中文自然语言处理开放平台
- 中科院计算所自然语言处理研究组
- 北京大学计算语言学研究所
- MIT自然语言处理课程

参考文献

朱德熙（ 1985 ）语法答问，商务印书馆
刘开瑛、郭炳炎（ 1991 ）自然语言处理，科学出版社
冯志伟（ 1995 ）自然语言机器翻译新论，语文出版社 1995 年版
冯志伟（ 1997 ）自然语言的计算机处理，上海外语教育出版社
姚天顺等（ 2002 ）自然语言理解 —— 一种让机器懂得人类语言的研究（第二版），清华大学出版社、广西科学技术出版社
翁富良、王野翊（ 1998 ）计算语言学导论，中国社会科学
俞士汶 主编（ 2003 ）计算语言学概论，商务印书馆
陈小荷（ 2000 ）现代汉语自动分析，北京语言文化大学出版社
刘颖（ 2002 ）计算语言学，清华大学出版社
宗成庆（ 2008 ）统计自然语言处理，清华大学出版社
刘群（ 2008 ）汉英机器翻译若干关键技术研究，清华大学出版社

参考文献

James Allen (1995), Natural Language Understanding (Second Edition), The Benjamin / Cummings Publishing Company, Inc.，中译本：刘群等译，自然语言理解（第二版），电子工业出版社，2005。

Christopher D. Manning and Hinrich Schutze (1999), Foundations of Statistical Natural Language Processing, The MIT Press, Cambridge, Massachusetts，中译本：苑春法等译，统计自然语言处理基础，电子工业出版社，2005。

Daniel Jurafsky, James H. Martin, Speech and Language Processing: an Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice Hall, US Ed edition, January 26, 2000, 中译本：冯志伟，孙乐译，自然语言处理综论，电子工业出版社，2005。

复习思考题

- 如果让你实现一个机器翻译系统，你会如何做？
- 如果让你实现一个问答系统，你会如何做？
- 举例说明汉语和英语在不同层面上的歧义性。
- 英语句子“ Time flies like an arrow”有多少种意思？
- 人机交流的语言和人类交流的语言有什么不同？
- 你觉得自然语言处理与人类的语言处理机制会有什么相同和不同之处？自然语言处理是否可能、是否需要模拟人类的语言处理机制？

致谢

- 本课程讲义（包括后续各节）直接引用了下面几位同行的课程讲义中的部分内容，在此深表感谢！
  - 詹卫东：《计算语言学概论》
  - 白　硕：《计算语言学》
  - 刘　颖：《计算语言学》
  - 冯志伟：《机器翻译研究的历史和现状》
    《依存语法在机器翻译中的应用》

联系方式

- 教师个人主页：
  - http://nlp.ict.ac.cn/~liuqun
- 课程邮件列表： Google Groups
  - 邮件列表：

    信箱： cl-course-at-gucas-2010@googlegroups.com

    网址： http://groups.google.com/group/cl-course-at-gucas-2010
  - 教师根据选课名单统一添加成员，不能自己申请