Natural Language Processing I

2. 基本文本处理

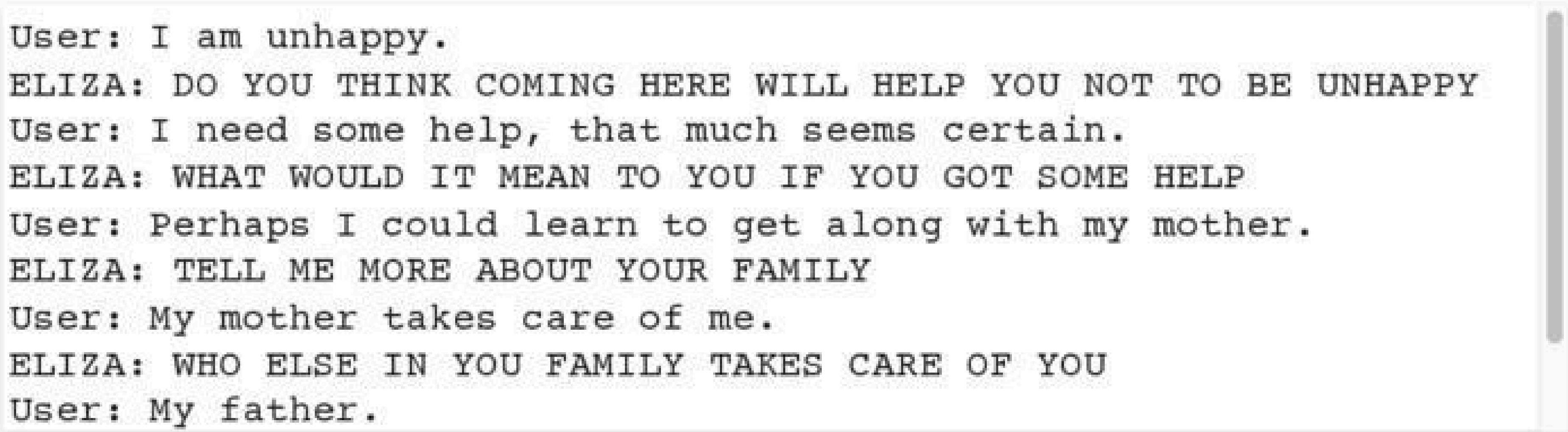
WU Xiaokun 吴晓堃

xkun.wu [at] gmail

2022/03/11

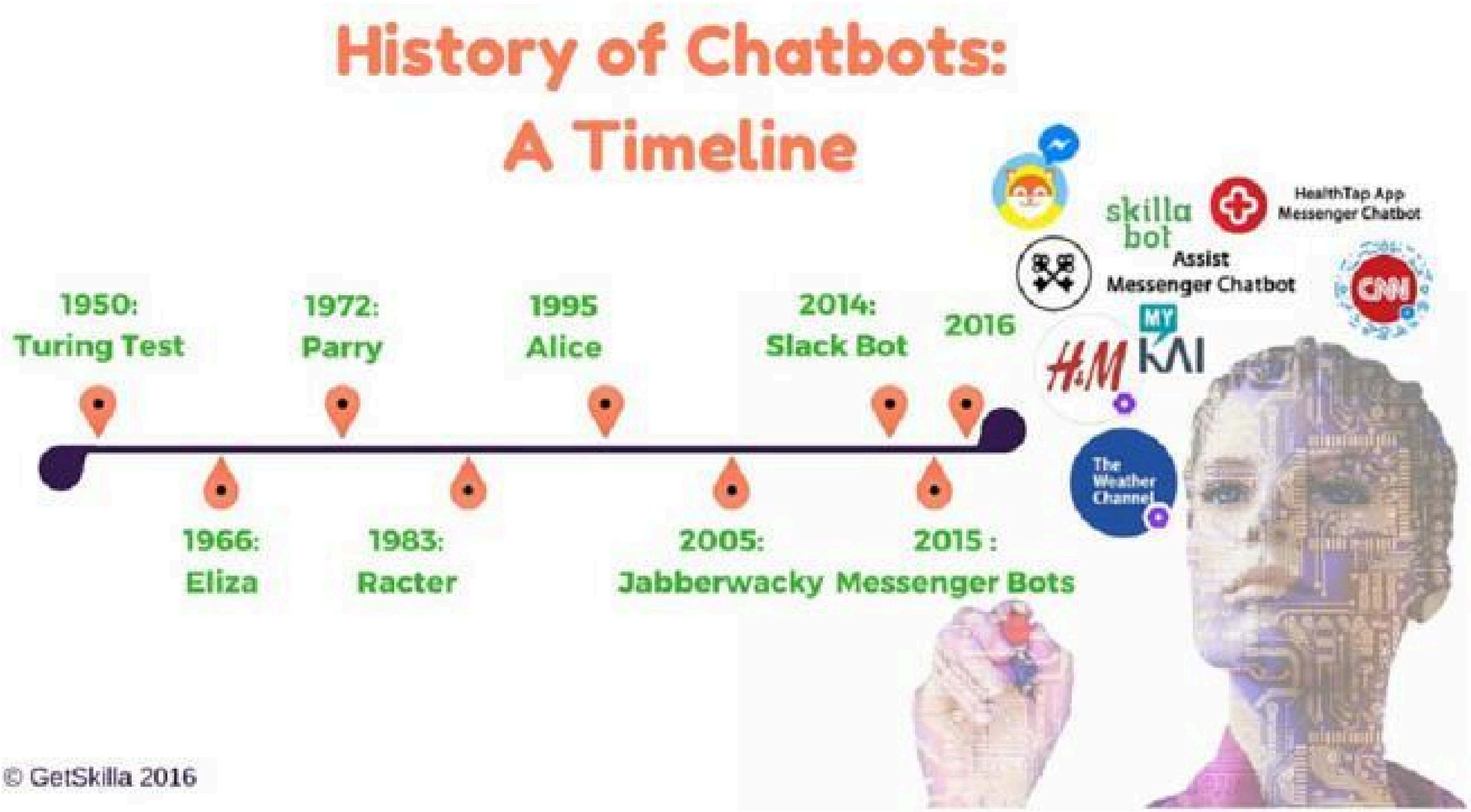
“ELIZA, I am unhappy.”

ELIZA



```
User: I am unhappy.
ELIZA: DO YOU THINK COMING HERE WILL HELP YOU NOT TO BE UNHAPPY
User: I need some help, that much seems certain.
ELIZA: WHAT WOULD IT MEAN TO YOU IF YOU GOT SOME HELP
User: Perhaps I could learn to get along with my mother.
ELIZA: TELL ME MORE ABOUT YOUR FAMILY
User: My mother takes care of me.
ELIZA: WHO ELSE IN YOU FAMILY TAKES CARE OF YOU
User: My father.
```

聊天机器



History of Chatbots:
A Timeline

1950: Turing Test

1966: Eliza

1972: Parry

1983: Racter

1995 Alice

2005: Jabberwacky

2014: Slack Bot

2015 : Messenger Bots

2016

© GetSkilla 2016

ELIZA

最早的聊天机器人之一，第一个尝试挑战Turing测试的机器人

- 用于模仿**罗氏心理治疗师 Rogerian psychotherapist**
  - 倾听与引导；客户自己主导对话

ELIZA, [Weizenbaum 1966]: https://web.njit.edu/~ronkowit/eliza.html

ELIZA 基本原理

基于简单的模板匹配与替换

- “I need X” -> “What would it mean to you if you got X?”
- 不需要任何领域专家知识

ELIZA 基本原理

基于简单的模板匹配与替换

- “I need X” -> “What would it mean to you if you got X?”
- 不需要任何领域专家知识

```
User: Perhaps I could learn to get along with my mother.
ELIZA: TELL ME MORE ABOUT YOUR FAMILY
User: My mother takes care of me.
ELIZA: WHO ELSE IN YOU FAMILY TAKES CARE OF YOU
User: My father.
ELIZA: YOUR FATHER
```

ELIZA 简单评价

> *… listeners can act as if they* ***know nothing of the world****. ELIZA's mimicry of human conversation was remarkably successful: many people who interacted with ELIZA came to believe that it really* ***understood them and their problems***
>
> …
>
> *– [Weizenbaum 1976]*

ELIZA 简单评价

> *… listeners can act as if they **know nothing of the world**. ELIZA's mimicry of human conversation was remarkably successful: many people who interacted with ELIZA came to believe that it really **understood them and their problems***
>
> *…*
>
> *– [Weizenbaum 1976]*

现代聊天机器人不仅仅是用于消遣

- 语音助手、智能家居、问答、机器翻译

模板匹配与替换

Unix 工具

- `grep`：模板匹配并打印
  - “globally search for regular expression and print”
- `tr`：“translate”，字符级替换、变换
- `sort`, `uniq`
- `wc`：“word count”
- `sed`：用于过滤和转换文本的流编辑器
  - “stream editor”
- `cat`, `echo`, `head`, `tail`

问题：如何实现模板匹配与替换?

正则表达式

正则表达式

正则表达式 Regular Expressions (RE)：在文本查找中指定匹配字符串的语言

- **模式 pattern**：查找字符串
- **语料库 corpus**：目标文本

程序员的“黑话”

正则表达式

正则表达式 Regular Expressions (RE)：在文本查找中指定匹配字符串的语言

- **模式 pattern**：查找字符串
- **语料库 corpus**：目标文本

程序员的“黑话”

如何查找如下短语?

- Natural Language
- natural language
- Natural Languages
- natural languages

基本字符串

注意：Unix 正则表达式区分大小写，Windows 不区分

```
natural language
Natural Language
```

基本字符串

注意：Unix 正则表达式区分大小写，Windows 不区分

```
natural language
Natural Language
```

析取 disjunction

```
/[nN]atural [Ll]anguages/
/[abc]/
/[1234567890]/
```

注意：斜线“/”不是正则表达式的语法

- 但很多工具用斜线作为**分隔符**

范围析取

ASCII 顺序

```
/[A-Z]/
/[a-z]/
/[0-9]/
```

范围析取

ASCII 顺序

```
/[A-Z]/
/[a-z]/
/[0-9]/
```

否定 negate

```
/[^A-Z]/
/[^.]/
/[e^]/
/a^b/
```

注意：只在“^”是范围析取的第一个符号时成立

可选字符

0或1次匹配

```
[nN]atural [Ll]anguage?
/colou?r/
```

可选字符

0或1次匹配

```
[nN]atural [Ll]anguage?
/colou?r/
```

重复字符，考虑某种羊语

```
咩!
咩咩!
咩咩咩咩!
咩咩~咩~咩咩咩!
```

- 如何识别、学习羊语?

重复字符 *

```
咩！
咩咩！
咩咩咩咩！
咩咩~咩~咩咩咩！
```

Kleene *：0或多次匹配

```
/咩咩*！ /
```

重复字符 *

```
咩！
咩咩！
咩咩咩咩！
咩咩~咩~咩咩咩！
```

Kleene *：0或多次匹配

```
/咩咩*！ /
```

- 注意：如下是错误用法，匹配模板0次⇔匹配任意字符

```
/咩*！ /
```

```
汪汪汪！
```

重复字符 +

```
咩！
咩咩！
咩咩咩咩！
咩咩~咩~咩咩咩！
```

Kleene +：1或多次匹配

```
/咩+！/
/[0-9]+/
```

范围重复

Kleene 匹配可以与范围条件同时使用

```
/[0-9.][0-9.]*/
/[0-9.]+/
```

```
3.14159265357
￥199.98
```

通配符

通配符 wildcard

- 例如：同一詞根变形形式

```
/beg.n/
```

```
begin
beg’n
begun
```

通配符

通配符 wildcard

- 例如：同一詞根变形形式

```
/beg.n/
```

```
begin
beg’n
begun
```

与 Kleene * 搭配

```
/创新.*创新/
```

```
深入推进管党治党实践创新、理论创新、制度创新
```

锚定符

锚定符 anchors

```
/^[Tt]he .*\.$/
/^[Tt]he .* \.$/
```

锚定符

锚定符 anchors

```
/^[Tt]he .*\.$/
/^[Tt]he .* \.$/
```

注意：“\.” 用于匹配符号本身；检查语法！

锚定符

锚定符 anchors

```
/^[Tt]he .*\.$/
/^[Tt]he .* \.$/
```

注意："\." 用于匹配符号本身；检查语法！

注意："^"的三种用法

- 行起始
- 范围否定
- 字符本身

词边界

注意：分词是需要定义的，取决于编程实现的共识

```
/\bthe\b/
/\b98\b/
```

```
the Paulaner: ¥9.98, 298 bottles
other food: ¥98, 98 dishes
```

词边界

注意：分词是需要定义的，取决于编程实现的共识

```
/\bthe\b/
/\b98\b/
```

```
the Paulaner: ￥9.98, 298 bottles
other food: ￥98, 98 dishes
```

字串析取 disjunction

```
/the|other/
```

优先级 precedence

```
China, Chinese
```

```
/China|ese/
/Chin(a|ese)/
```

优先级 precedence

```
China, Chinese
```

```
/China|ese/
/Chin(a|ese)/
```

```
/Column [0-9]+ */
/(Column [0-9]+ *)+/
```

```
Column 1 Column 2 Column 3
```

优先级顺序

1. 括号：`( )`
2. 量词：`* + ? {}`
3. 字串和锚定：`^the end$`
4. 析取：`|`

优先级顺序

1. 括号：`()`
2. 量词：`* + ? {}`
3. 字串和锚定：`^the end$`
4. 析取：`|`

基本原则：**最长、最短匹配（贪心 greedy）**

```
Column 1 Column 2 Column 3
```

```
/(Column [0-9]+ *)+/
/(Column [0-9]+ *)+?/
```

Quiz：字串匹配

找出文本中所有的“the”

```
/the/
```

- 大写?

Quiz：字串匹配

找出文本中所有的“the”

```
/the/
```

- 大写?

```
/[tT]he/
```

- `these`?

Quiz：字串匹配

找出文本中所有的“the”

```
/the/
```

- 大写?

```
/[tT]he/
```

- `these`?

```
/[^a-zA-Z][tT]he[^a-zA-Z]/
```

- 能不能写简单点?

Quiz：字串匹配

找出文本中所有的“the”

```
/the/
```

- 大写?

```
/[tT]he/
```

- `these`?

```
/[^a-zA-Z][tT]he[^a-zA-Z]/
```

- 能不能写简单点?

```
/\bthe\b/
```

两类误差

假阳 False positives (Type I errors)

- 查出错误信息
  - there, then, other
- 提高**准确度 accuracy**、**精度 precision**

两类误差

假阳 False positives (Type I errors)

- 查出错误信息
    - there, then, other
- 提高**准确度 accuracy**、**精度 precision**

假阴 False negatives (Type II errors)

- 没有查出正确信息
    - The
- 提高**覆盖率 coverage**、**召回率 recall**

两类误差举例

假阳：查出错误信息

Type I 医疗事故

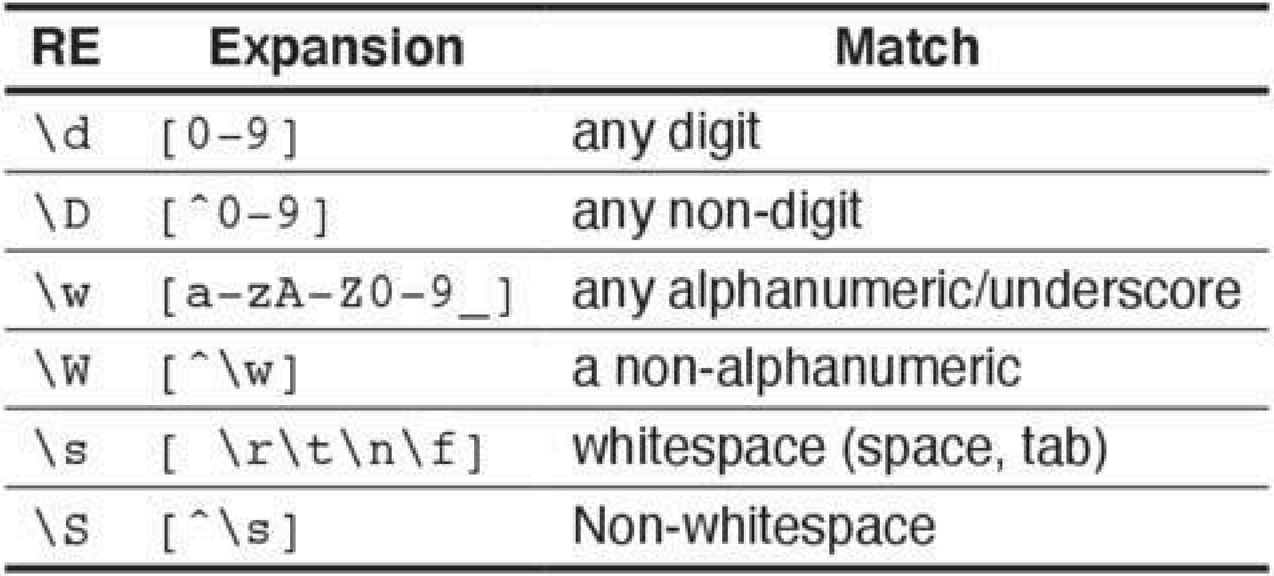
- 假阳性癌症诊断：可能会把人吓出毛病

假阴：没有查出正确信息

Type II 医疗事故

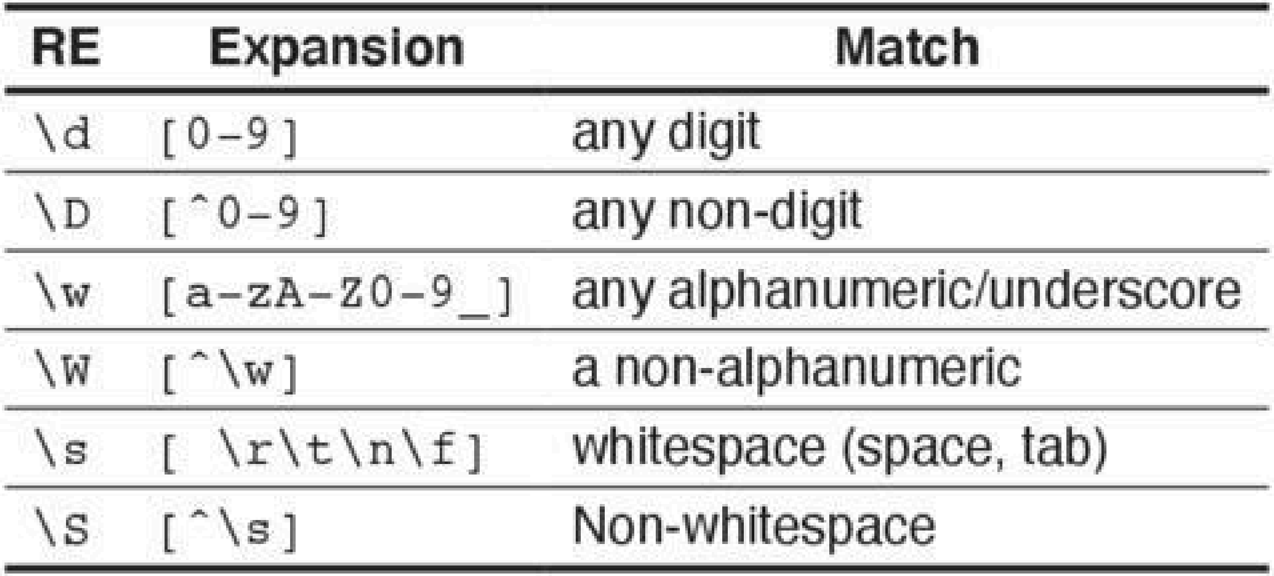
- 经验不足误诊、检测手段落后得出假阴性诊断：可能会无法及时医治

特殊别名



RE	Expansion	Match
`\d`	`[0-9]`	any digit
`\D`	`[^0-9]`	any non-digit
`\w`	`[a-zA-Z0-9_]`	any alphanumeric/underscore
`\W`	`[^\w]`	a non-alphanumeric
`\s`	`[ \r\t\n\f]`	whitespace (space, tab)
`\S`	`[^\s]`	Non-whitespace

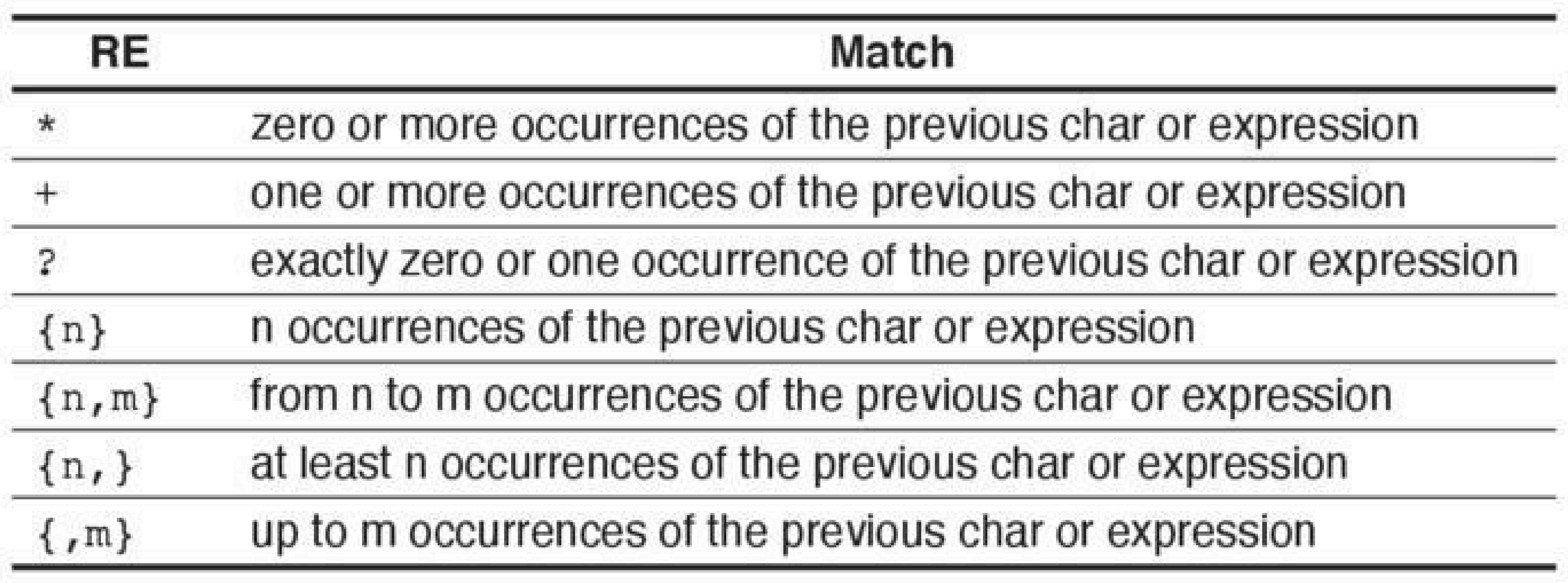
特殊别名



RE	Expansion	Match
`\d`	`[0-9]`	any digit
`\D`	`[^0-9]`	any non-digit
`\w`	`[a-zA-Z0-9_]`	any alphanumeric/underscore
`\W`	`[^\w]`	a non-alphanumeric
`\s`	`[ \r\t\n\f]`	whitespace (space, tab)
`\S`	`[^\s]`	Non-whitespace

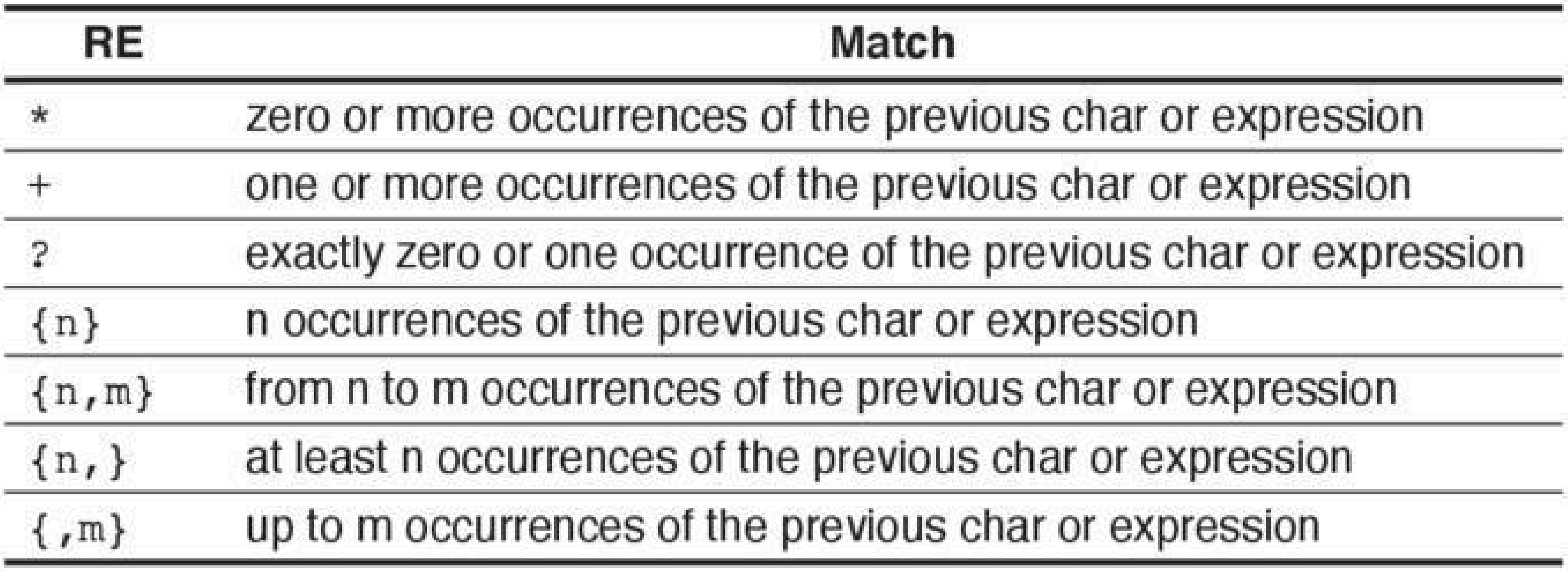
思考：如何删除每行结尾的空格?

计数符



RE	Match
`*`	zero or more occurrences of the previous char or expression
`+`	one or more occurrences of the previous char or expression
`?`	exactly zero or one occurrence of the previous char or expression
`{n}`	n occurrences of the previous char or expression
`{n,m}`	from n to m occurrences of the previous char or expression
`{n,}`	at least n occurrences of the previous char or expression
`{,m}`	up to m occurrences of the previous char or expression

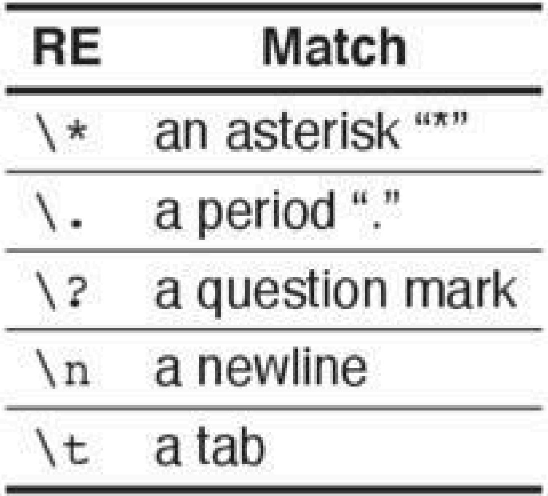
计数符



RE	Match
`*`	zero or more occurrences of the previous char or expression
`+`	one or more occurrences of the previous char or expression
`?`	exactly zero or one occurrence of the previous char or expression
`{n}`	n occurrences of the previous char or expression
`{n,m}`	from n to m occurrences of the previous char or expression
`{n,}`	at least n occurrences of the previous char or expression
`{,m}`	up to m occurrences of the previous char or expression

思考：如何查找“咩咩”?

特殊字符



RE	Match
`\*`	an asterisk “*”
`\.`	a period “.”
`\?`	a question mark
`\n`	a newline
`\t`	a tab

Quiz：Morse 密码

查找如下代码：`D: -..`

```
- .... . / --.- ..- .. -.-. -.- / -... .-. --- .-- -. / ..-. ---
-..- / .--- ..- -- .--. ... / --- ...- . .-. / - .... . / .-.. .-
--.. -.-- / -.. --- --.
```

Quiz：Morse 密码

查找如下代码：`D: -..`

```
- .... . / --.- ..- .. -.-. -.- / -... .-. --- .-- -. / ..-. ---
-..- / .--- ..- -- .--. ... / --- ...- . .-. / - .... . / .-.. .-
--.. -.-- / -.. --- --.
```

```
-\.\.
```

Quiz：Morse 密码

查找如下代码：`D: -..`

```
- .... . / --.- ..- .. -.-. -.- / -... .-. --- .-- -. / ..-. ---
-..- / .--- ..- -- .--. ... / --- ...- . .-. / - .... . / .-.. .-
--.. -.-- / -.. --- --.
```

```
-\.\.
```

```
-\.{2}
```

替换

在 UNIX 命令行工具中：

```
s/regexp/pattern/
s/colour/color/
```

替换

在 UNIX 命令行工具中：

```
s/regexp/pattern/
s/colour/color/
```

部分替换：

```
the 35 boxes
the <35> boxes
```

使用括号标注，然后使用\数字引用

```
s/the ([0-9]+) boxes/the <\1> boxes/
```

注意：Windows 是$数字

引用用于匹配

\数字的另一种用法

```
the Xer they were, the Xer they will be
```

```
/the (.*)er they were, the \ler they will be/
```

```
the bigger they were, the bigger they will be
the bigger they were, the faster they will be
```

单纯分组括号

单纯分组 non-capturing group

```
(?: pattern )
```

```
/(?:some|a few) (people|cats) like some \1/
```

```
some cats like some cats
a few cats like some cats
some people like some people
some cats like some a few
```

预先读取

预读 lookahead：有时需要分情况判断是否匹配或替换

```
^(?=some)
^(?=some).* cats$
```

注意：预读不推进光标

预先读取

预读 lookahead：有时需要分情况判断是否匹配或替换

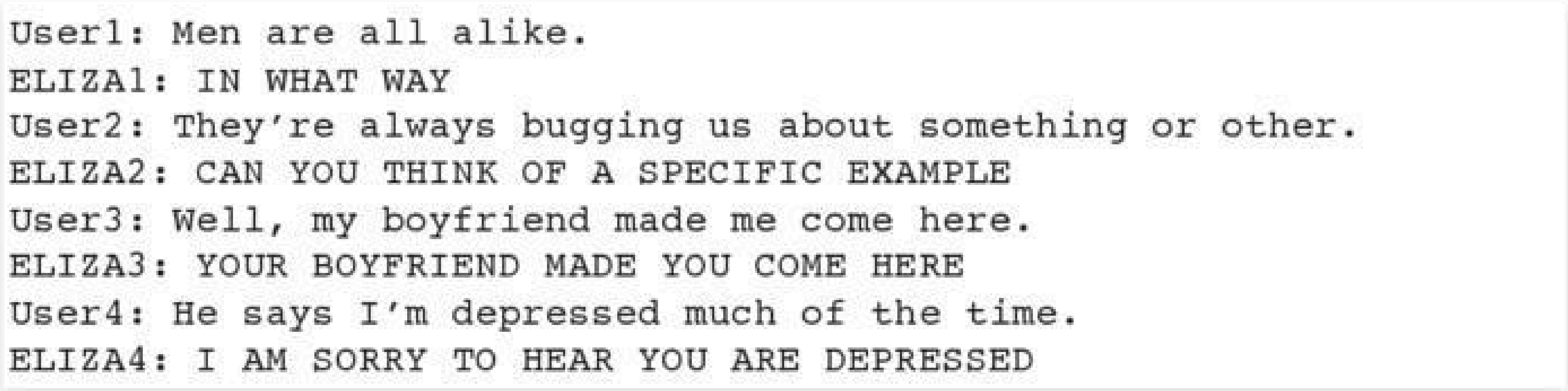
```
^(?=some)
^(?=some).* cats$
```

注意：预读不推进光标

预读否定：常用于复杂模板中排除特殊情况

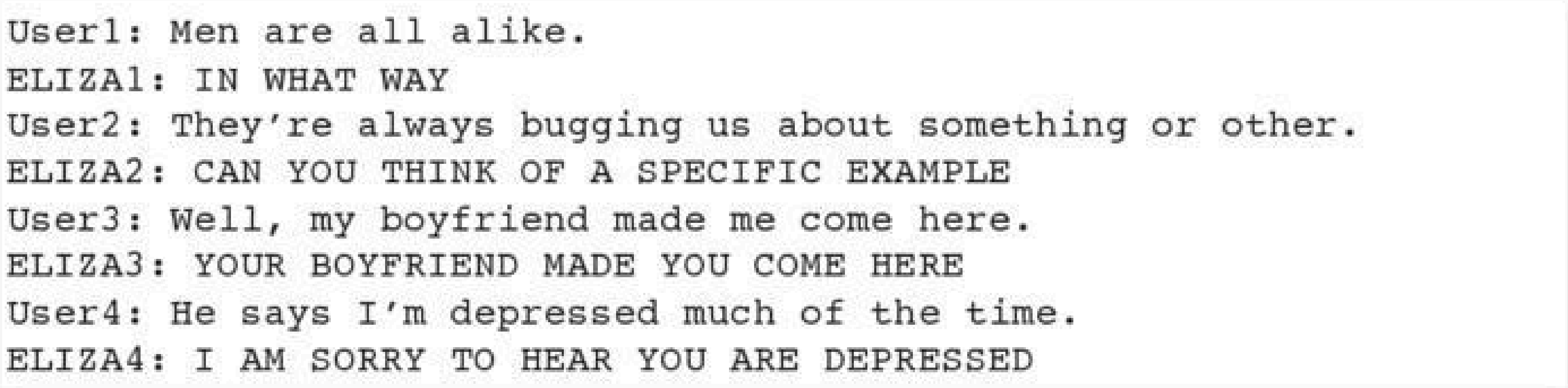
```
^(?!some)
^(?!some)a few
```

实践：ELIZA

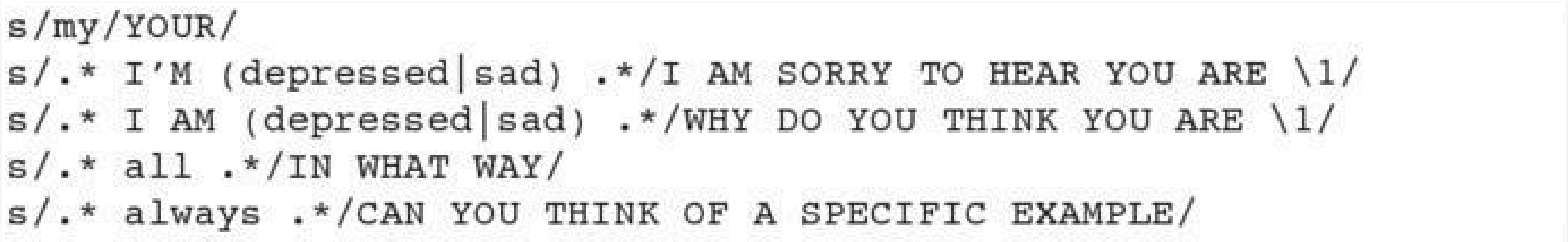


```
User1: Men are all alike.
ELIZA1: IN WHAT WAY
User2: They're always bugging us about something or other.
ELIZA2: CAN YOU THINK OF A SPECIFIC EXAMPLE
User3: Well, my boyfriend made me come here.
ELIZA3: YOUR BOYFRIEND MADE YOU COME HERE
User4: He says I'm depressed much of the time.
ELIZA4: I AM SORRY TO HEAR YOU ARE DEPRESSED
```

实践：ELIZA



```
User1: Men are all alike.
ELIZA1: IN WHAT WAY
User2: They're always bugging us about something or other.
ELIZA2: CAN YOU THINK OF A SPECIFIC EXAMPLE
User3: Well, my boyfriend made me come here.
ELIZA3: YOUR BOYFRIEND MADE YOU COME HERE
User4: He says I'm depressed much of the time.
ELIZA4: I AM SORRY TO HEAR YOU ARE DEPRESSED
```



```
s/my/YOUR/
s/.* I'M (depressed|sad) .*/I AM SORRY TO HEAR YOU ARE \1/
s/.* I AM (depressed|sad) .*/WHY DO YOU THINK YOU ARE \1/
s/.* all .*/IN WHAT WAY/
s/.* always .*/CAN YOU THINK OF A SPECIFIC EXAMPLE/
```

词、语料库

句子里有多少词?

标点、空格

- 提取语义：问好、感叹号可以**表达情绪**
- 词类标注、命名实体识别：一般不考虑

```
What's the course about?
本课程主要讲授自然语言处理。
```

句子里有多少词?

标点、空格

- 提取语义：问好、感叹号可以**表达情绪**
- 词类标注、命名实体识别：一般不考虑

```
What's the course about?
本课程主要讲授自然语言处理。
```

音频断句

- 脚本翻译：一般不考虑
- 辨别讲话人：**语言特征**

```
I do uh main- mainly business data processing
```

两种计数

- **词目 word type**：每个词只记一次
  - 用于提取词汇表 **vocabulary**
- **词（词元） word token**：文本中出现的总词数

```
They picnicked by the pool, then lay back on the grass and looked
at the stars.
```

16 词，14 类：`the` 出现了3次

两种计数

- **词目 word type**：每个词只记一次
    - 用于提取词汇表 **vocabulary**
- **词（词元） word token**：文本中出现的总词数

```
They picnicked by the pool, then lay back on the grass and looked
at the stars.
```

16 词，14 类：`the` 出现了3次

词元 lemma，词形 wordform

```
The Cat in the Hat is different from other cats!
```

词汇统计

Heaps Law （语言学） = Herdan's Law （信息检索）： $|V| = kN^{\beta}$

- 通常： $.67 < \beta < .75$

	Tokens = N	**Types = $\|V\|$**
Switchboard	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13+ million

词汇统计

Heaps Law （语言学） = Herdan's Law （信息检索）： $|V| = kN^{\beta}$

- 通常： $.67 < \beta < .75$

	Tokens = N	Types = $|V|$
Switchboard	2.4 million	20 thousand
Shakespeare	884,000	31 thousand
Google N-grams	1 trillion	13+ million

用途：计算统计量（如分频词汇表）；鉴别文本风格

语料库

语料库 corpus：计算机可以读取的语言、语音集合

- 用于机器学习的**训练数据集**

语料库

语料库 corpus：计算机可以读取的语言、语音集合

- 用于机器学习的**训练数据集**

不同的文本有不同的特征

- 作者写作风格
- 时代背景
- 语言、方言
- 写作目的

注意：选用数据时一定要弄清楚数据的内容！

制作语料库

数据声明 data statement：数据使用说明书

- 动机
  - 采集者，赞助方
- 采集、标注方式
- 语言种类，内容限制等级
- 采集对象人口统计特征
- 版权

文本正则化

文本正则化

文本正则化 Text Normalization：转换成便于处理的标准格式

- **分词 Tokenizing (segmenting) words**
- **格式正则化 Normalizing word formats**
- **分句 Segmenting sentences**

通常是解决任何 NLP 任务的第一步。

基于空格的分词

使用空格分隔单词的语言

- 空格、标点符号

基于空格的分词

使用空格分隔单词的语言

- 空格、标点符号

Unix 工具

- `tr`
- 任务：给定文本，输出词及其频率

使用 UNIX 工具简单分词

`tr`；替换字符

- `-c`：除了指定字符之外
- `-s`：重复字符替换成单一字符或指定字符，例如多个空格

```
tr -sc 'A-Za-z' '\n' < shakespeare.txt
```

输出是每行一个单词

使用 UNIX 工具简单分词

`tr`；替换字符

- `-c`：除了指定字符之外
- `-s`：重复字符替换成单一字符或指定字符，例如多个空格

```
tr -sc 'A-Za-z' '\n' < shakespeare.txt
```

输出是每行一个单词

`sort | uniq -c`：排序，折叠重复并计数

```
tr -sc 'A-Za-z' '\n' < shakespeare.txt | sort | uniq -c
```

使用 UNIX 工具简单分词

通常可以忽略大小写

```
tr -sc 'A-Za-z' '\n' < shakespeare.txt | tr 'A-Z' 'a-z' | sort |
        uniq -c
```

使用 UNIX 工具简单分词

通常可以忽略大小写

```
tr -sc 'A-Za-z' '\n' < shakespeare.txt | tr 'A-Z' 'a-z' | sort |
        uniq -c
```

再次按照词频排序

- -n：按数字排序（默认是字母序）
- -r：逆序排列

```
tr -sc 'A-Za-z' '\n' < shakespeare.txt | tr 'A-Z' 'a-z' | sort |
        uniq -c | sort -n -r
```

- 大多数语言的高频词都是比较短的功能词
- 快速得到语料库的统计信息

基于空格分词的问题

命名实体 named entity：单词的组成部分

- m.p.h., Ph.D., AT&T
- $45.55, 01/02/06
- http://www.some-uni.edu, someone@somewhere.com
- 555,500.50, 555.500,50

基于空格分词的问题

命名实体 named entity：单词的组成部分

- m.p.h., Ph.D., AT&T
- $45.55, 01/02/06
- http://www.some-uni.edu, someone@somewhere.com
- 555,500.50, 555.500,50

命名实体 named entity：专有名词

- Xi’an, New York, rock ’n’ roll

基于空格分词的问题

命名实体 named entity：单词的组成部分

- m.p.h., Ph.D., AT&T
- $45.55, 01/02/06
- http://www.some-uni.edu, someone@somewhere.com
- 555,500.50, 555.500,50

命名实体 named entity：专有名词

- Xi’an, New York, rock 'n’ roll

附着词 clitic contraction

- we’re, j’ai, l’homme

基于空格分词的问题

命名实体 named entity：单词的组成部分

- m.p.h., Ph.D., AT&T
- $45.55, 01/02/06
- http://www.some-uni.edu, someone@somewhere.com
- 555,500.50, 555.500,50

命名实体 named entity：专有名词

- Xi’an, New York, rock ’n’ roll

附着词 clitic contraction

- we’re, j’ai, l’homme

命名实体识别 Named Entity Recognition (NER)，中文分词（日语、泰语）

词正则化

词正则化 normalization：用标准形式替代有多种形式的词

- 信息可能会丢失：USA -> US

词正则化

词正则化 normalization：用标准形式替代有多种形式的词

- 信息可能会丢失：USA -> US

取消大写 case folding

- 文本分类、机器翻译等通常不进行：US $\neq$ us

词正则化

词正则化 normalization：用标准形式替代有多种形式的词

- 信息可能会丢失：USA -> US

取消大写 case folding

- 文本分类、机器翻译等通常不进行：US $\neq$ us

词形还原 lemmatization：屈折语原形替换

- am, are, is -> be

词正则化

词正则化 normalization：用标准形式替代有多种形式的词

- 信息可能会丢失：USA -> US

取消大写 case folding

- 文本分类、机器翻译等通常不进行：US ≠ us

词形还原 lemmatization：屈折语原形替换

- am, are, is -> be
- **词干 stem + 后缀 affix**
  - amaren -> amar, -en，虚拟式将来未完成时（3rd Pl.）

词正则化

词正则化 normalization：用标准形式替代有多种形式的词

- 信息可能会丢失：USA -> US

取消大写 case folding

- 文本分类、机器翻译等通常不进行：US $\neq$ us

词形还原 lemmatization：屈折语原形替换

- am, are, is -> be
- **词干 stem + 后缀 affix**
  - amaren -> amar, -en，虚拟式将来未完成时（3rd Pl.）
- **助动词 auxiliary**
  - würden lieben -> lieben，第二虚拟式第一将来时（3rd Pl.）

词干分析器

词干分析 stemming

- [Porter 1980] Porter stemmer：简单切除后缀

```
This was not the map we found in Billy Bones's chest, but
an accurate copy, complete in all things-names and heights
and soundings-with the single exception of the red crosses
and the written notes.
```

```
Thi wa not the map we found in Billi Bone s chest but an
accur copi complet in all thing name and height and sound
with the singl except of the red cross and the written note
```

词法

词法 morphology：单词是由词素构成的

- **词素 morpheme**：单词中有意义的单元

词法

词法 morphology：单词是由词素构成的

- **词素 morpheme**：单词中有意义的单元
- 汉字偏旁也可以认为是词素
  - 形声字策略：用形旁和声旁组合成汉字，“秀才念半边”

词法

词法 morphology：单词是由词素构成的

- **词素 morpheme**：单词中有意义的单元
- 汉字偏旁也可以认为是词素
    - 形声字策略：用形旁和声旁组合成汉字，“秀才念半边”

词法对很多语言的理解至关重要的

- 德语：Rechtsschutzversicherungsgesellschaften (39)
    - insurance companies that provide legal protection
    - 提供法律保护的保险公司

词法

词法 morphology：单词是由词素构成的

- **词素 morpheme**：单词中有意义的单元
- 汉字偏旁也可以认为是词素
    - 形声字策略：用形旁和声旁组合成汉字，“秀才念半边”

词法对很多语言的理解至关重要的

- 德语：Rechtsschutzversicherungsgesellschaften (39)
    - insurance companies that provide legal protection
    - 提供法律保护的保险公司
- 土耳其语（是黏着语）：Uygarlastiramadiklarimizdanmissinizcasina
    - (behaving) as if you are among those whom we could not civilize
    - 你（的行为举止表现得）仿佛是我们无法教化的那些（野蛮）人之一

```
Uygar 'civilized' + las 'become' + tir 'cause' + ama 'not able'
+ dik 'past' + lar 'plural'+ imiz 'p1pl' + dan 'abl'
```

分句

只在少数语言中涉及，如古汉语：没有标点符号

> 道可道非常道名可名非常名
> 无名万物之始有名万物之母
> --《道德经》

> 常太后后问后后云
> --《魏书·卷十三·列传第一》

分句

只在少数语言中涉及，如古汉语：没有标点符号

> 道可道非常道名可名非常名
> 无名万物之始有名万物之母
> --《道德经》

> 常太后后问后后云
> --《魏书·卷十三·列传第一》

难点：专有名词、舶来词

> 观自在菩萨行深般若波罗蜜多时照见五蕴皆空度一切苦厄
> --《心经》

> 晃后谋杀佛狸见杀
> --《南齐书·魏虏传》