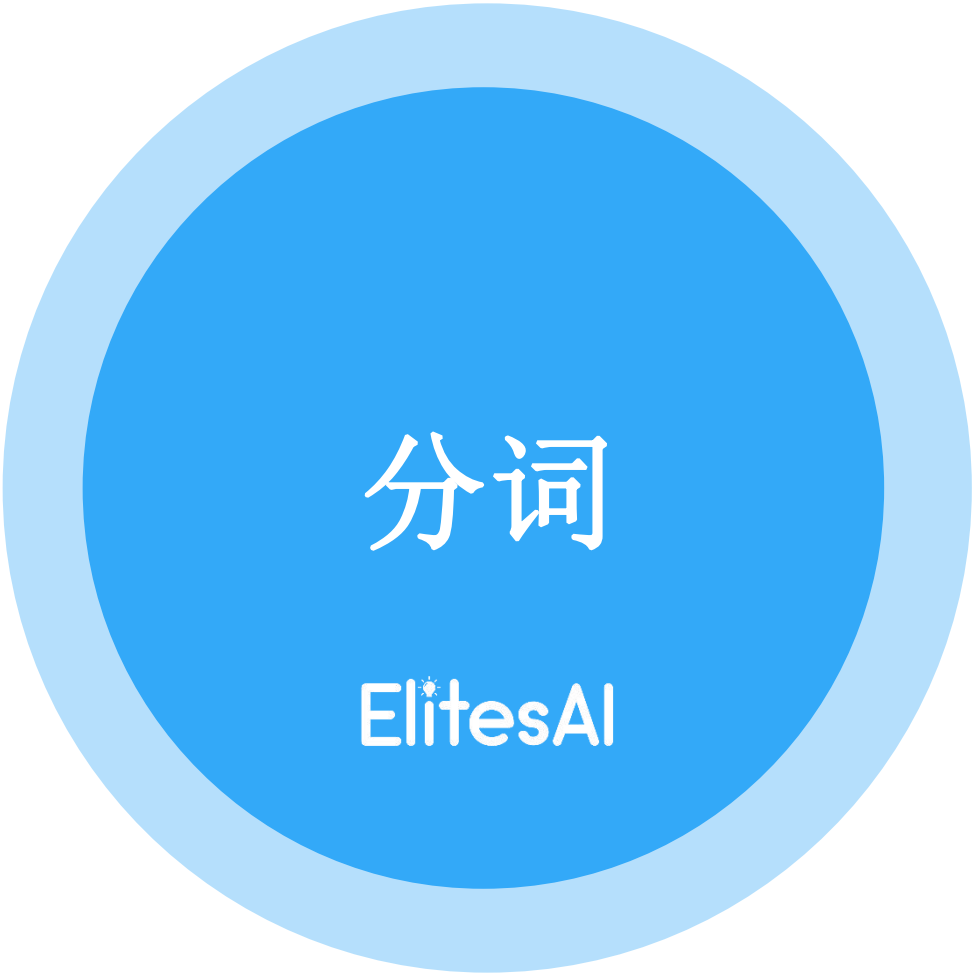
第2章：文本规范化

课件制作：王新宇、楼超、屠可伟
讲解人：王新宇

文本规范化

- 每个 NLP 任务都需要文本规范化：
  - 分词
  - 词规范化
  - 分句


分词
ElitesAI

基于空格和标点符号的分词

- 一种非常简单的分词方法
  - 对于在单词之间使用空格字符的语言
    - 阿拉伯语、西里尔语、希腊语、拉丁语等书写系统
  - 在空格（和标点符号）实例之间分割分词
- 例子
  - 句子：
    I learn natural language processing with dongshouxueNLP, too.
  - I␣learn␣natural␣language␣processing␣with dongshouxueNLP␣,␣too␣.
    - “␣”：分隔符

分词问题

不能盲目删除标点符号：

- Ph.D.， A.M.， P.M.；
- 价格（ $45.55 ）
  - 日期（ 01/02/06 ）
  - 网址（ https://www.boyuai.com/）
  - 标签（ #nlproc ）
  - 电子邮件地址（ someone@somewhere.com）
- 不能独立存在的词：
  - we‘re中的“ are ”，法语“j’ai ”中的“je”
- 多个词有时也是词元
  - New York、rock ’n’ roll

例句

- 句子：

```
Did you spend $3.4 on arxiv.org for your pre-
print? No, it's free! It's ...
```

- 预期的分词结果：

```
Did␣you␣spend␣$3.4␣on␣arxiv.org␣for␣your␣pre-
print␣?␣No␣,␣it's␣free␣!␣It's␣...
```

  - 内部带有连字符的单词：`pre-print`
  - 网页：`arxiv.org`
  - 货币和百分比：`$3.4`
  - 省略号：`...`
  - 标点符号：`][.,;”'?():-_``

自然语言工具包中的分词（NLTK）

- 方法：
  - 编写一个匹配所有可能分词但不匹配任何非分词的模式。
  - 输出所有不重叠的匹配项。
- 工具：正则表达式（RE）
  - 带有可选内部连字符的单词：
    - RE 中的模式：`\w+ (- \w+ )*`
    - 接受的字符串：
      - pre-train、fine-tune、pre-print、...
    - 被拒绝的字符串：
      - U.S.A.、arxiv.org、$3.4、 ...
      - non-、-ly, ...

`\w` =任何单词字符

= [a-zA-Z0-9_]

`+` =匹配 1 次或多次

`*` =匹配 0 次或多次

`(…)`是一个组（后跟`+`或`*`）

自然语言工具包中的分词（NLTK）

- 缩写：
  - 模式：([A-Z] \.)+
  - 接受的字符串： U.S.A.
  - 被拒绝的字符串：UU.SA、I
- 货币和百分比：
  - 模式： \$?\d+ (\.\d+)? %?
  - 接受字符串： $12.40、20%
  - 字符串： $.4、1.4.0、1%%
- 省略：
  - 模式： \.\.\.
  - 接受的字符串： ……
- 标点符号：
  - 模式： [\]\[\.,;”’ \?\(\):-_]

[A-Z] =大写字符

\d =数字= [0-9]

在 RE 中，某些字符（例如 ^$.?+()[] ）具有特殊含义。应该对它们进行转义以匹配它们。*

\. =“.”

\$ =“$”

+ =匹配 1 次或多次

? =匹配 0 或 1 次

没有空格的语言的分词

- 许多语言（如中文、日语、泰语）不使用空格来分隔单词！
- 某些西方语言中的复合名词也是如此
  - 示例：德语：
    Freundschaftsbezeigungen （展现友谊）
  - 示例：社交媒体中的标签：
    #NaturalLanguageProcessing

没有空格的语言的分词

- 句子：姚明进入总决赛
- 有很多种分词的方法
  - 姚明/进入/总决赛
  - 姚/明/进入/总/决赛

  通常表现为序列标注+监督学习（后续的章节讨论）
  - 姚/明/进/入/总/决/赛 — 单字符切分
- 歧义性
  - 南京市长江大桥

子词分词

- 分词可以是单词的一部分，也可以是整个单词
  - 使用数据告诉我们如何分词
- 优点
  - 词汇量小得多
  - 避免词汇外（OOV）单词
  - 子词有时是有意义的
    - 前缀、后缀、词干……

子词分词

- 三种常见算法：
  - **字节对编码**（BPE）
  - Unigram **语言建模分词**
  - WordPiece
- 都有两个部分：
  - 词元**学习器**采用原始训练语料库并归纳词汇表（一组**词元**）。
  - 一个**分词器**，它采用原始测试句子并根据该词汇对其进行分词

字节对编码（BPE）词元学习器

- 令词汇表为所有单个字符的集合

  = {A，B，C，D，…，a，b，c，d…}

- 重复：
  - 选择训练语料库中最常相邻的两个符号（例如“A”、“B”）
  - 将新的合并符号“ AB ”添加到词汇表中
  - 语料库中每个相邻的“A” “B”替换为“AB”。
- 直到k次合并已经完成。

BPE 词元学习器

- 原始语料库：

nan nan nan nan nan nanjing nanjing beijing beijing beijing beijing beijing beijing dongbei dongbei dongbei bei bei

- 通常在空格分隔的词元内运行BPE次元学习。
- 首先添加词尾词元“_”
  - 这样我们可以区分**beijing**和**dong<u>bei</u>**中的**<u>bei</u>**
  - 结果：

```
语料：
5 ['n', 'a', 'n', '_']
2 ['n', 'a', 'n', 'j', 'i', 'n', 'g', '_']
6 ['b', 'e', 'i', 'j', 'i', 'n', 'g', '_']
3 ['d', 'o', 'n', 'g', 'b', 'e', 'i', '_']
2 ['b', 'e', 'i', '_']
词表：['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o']
```

BPE 词元分词器

```
语料:
5 ['n', 'a', 'n', '_']
2 ['n', 'a', 'n', 'j', 'i', 'n', 'g', '_']
6 ['b', 'e', 'i', 'j', 'i', 'n', 'g', '_']
3 ['d', 'o', 'n', 'g', 'b', 'e', 'i', '_']
2 ['b', 'e', 'i', '_']
词表: ['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o']
```

- 合并n g到ng

```
迭代后的语料为:
5 ['n', 'a', 'n', '_']
2 ['n', 'a', 'n', 'j', 'i', 'ng', '_']
6 ['b', 'e', 'i', 'j', 'i', 'ng', '_']
3 ['d', 'o', 'ng', 'b', 'e', 'i', '_']
2 ['b', 'e', 'i', '_']
词表: ['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o', 'ng']
```

BPE 分词学习器

```
迭代后的语料为:
5 ['n', 'a', 'n', '_']
2 ['n', 'a', 'n', 'j', 'i', 'ng', '_']
6 ['b', 'e', 'i', 'j', 'i', 'ng', '_']
3 ['d', 'o', 'ng', 'b', 'e', 'i', '_']
2 ['b', 'e', 'i', '_']
词表: ['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o', 'ng']
```

- 合并b e到be

```
迭代后的语料为:
5 ['n', 'a', 'n', '_']
2 ['n', 'a', 'n', 'j', 'i', 'ng', '_']
6 ['be', 'i', 'j', 'i', 'ng', '_']
3 ['d', 'o', 'ng', 'be', 'i', '_']
2 ['be', 'i', '_']
词表: ['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o', 'ng', 'be']
```

BPE 分词学习器

```
迭代后的语料为:
5 ['n', 'a', 'n', '_']
2 ['n', 'a', 'n', 'j', 'i', 'ng', '_']
6 ['be', 'i', 'j', 'i', 'ng', '_']
3 ['d', 'o', 'ng', 'be', 'i', '_']
2 ['be', 'i', '_']
词表: ['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o', 'ng', 'be']
```

- 合并be i到bei

```
迭代后的语料为:
5 ['n', 'a', 'n', '_']
2 ['n', 'a', 'n', 'j', 'i', 'ng', '_']
6 ['bei', 'j', 'i', 'ng', '_']
3 ['d', 'o', 'ng', 'bei', '_']
2 ['bei', '_']
词表: ['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o', 'ng', 'be', 'bei']
```

BPE 分词学习器

- 最后我们可以得到：

```
迭代后的语料为：
5 ['nan', '_']
2 ['nan', 'jing_']
6 ['beijing_']
3 ['d', 'o', 'ng', 'bei', '_']
2 ['bei', '_']
词表：['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o', 'ng', 'be', 'bei',
 'ji', 'jing', 'jing_', 'na', 'nan', 'beijing_']
```

BPE 分词器算法

- 所学词汇：
  ['_', 'a', 'b', 'd', 'e', 'g', 'i', 'j', 'n', 'o', 'ng', 'be', 'bei', 'ji', 'jing', 'jing_', 'na', 'nan', 'beijing_']

- 在测试数据上，运行从训练数据中学到的每个合并：
  - 按照我们学习的顺序贪婪地进行选择
  - （测试数据的频率不影响我们选择的顺序）
- 所以：将每个n g合并到ng ，然后将b e合并到be ，等等。
- 结果：
  - 测试集“beijing_”将被分词为完整单词
  - 测试集“nanjing_”将是两个分词：“nan jing_”

BPE 分词的特性

- 通常包括频繁出现的单词和频繁出现的子词
  - 通常是像-est或-er这样的语素
- 语素是语言中最小的意义单位
  - unbreakable有 3 个语素un-、-break-和-able

词规范化

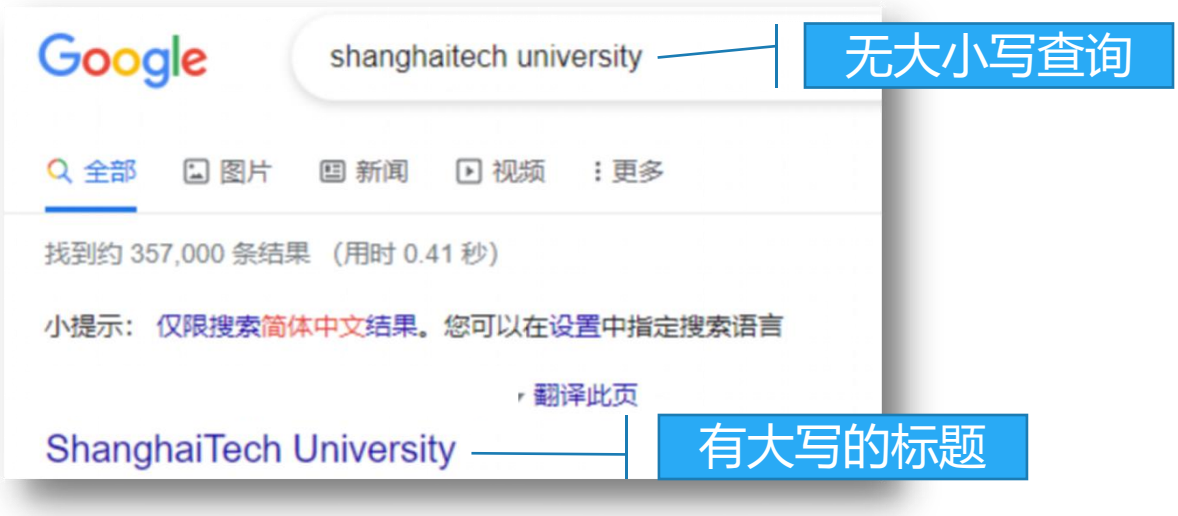
ElitesAI

词规范化

- 将单词/分词放入标准格式
  - U.S.A还是USA
  - Shanghai还是shanghai
  - Am、is、are、be

大小写折叠

- 将所有字母转为小写
- 在某些场景下效果很好
  - 信息检索：用户倾向于使用小写字母
  - 例子：



- 对别的场景不好
  - 示例：US、Fed、General Motors

词目还原

- 将所有单词表示为它们的词目也就是字典词条形式
  - am、are、is → be
  - cat, cats, cat's, cats' → cat
- 例子
  - Two dogs are chasing three cats

  → Two dog be chase three cat

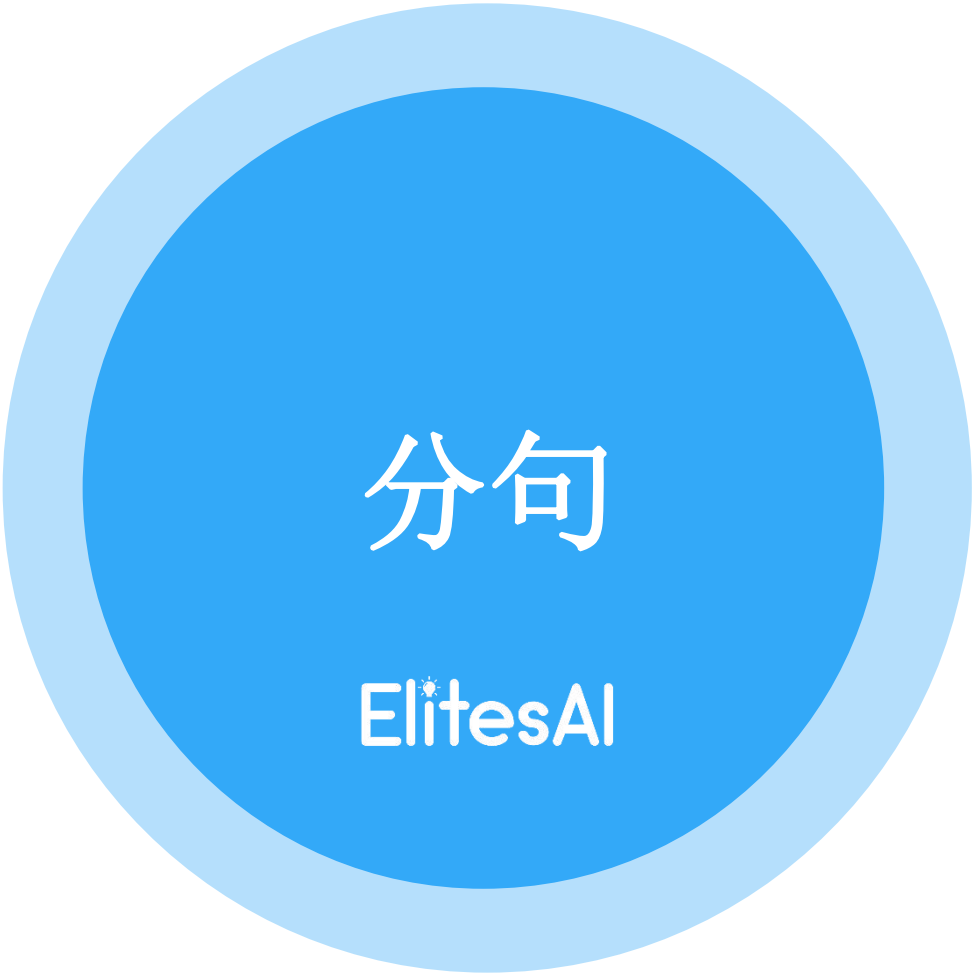
通过语素分析完成词目还原

- 语素：
  - 组成单词的有意义的最小单位
  - 词干（stem）：核心意义承载单位
  - 词缀（affix）：附着在词干上的部分，通常具有语法功能
- 语素分析器：
  - 将 *cats* 分成两个语素*cat* 和 *s*

词干提取

- 提取词干，**粗暴地砍掉词缀**
  - 词目还原的简单粗暴的替代方案
- 波特词干还原器
  - 基于一系列的重写规则
  - 一些示例规则

$$
\begin{aligned}
\text{ATIONAL} &\rightarrow \text{ATE} \quad \text{(e.g., relational} \rightarrow \text{relate)} \\
\text{ING} &\rightarrow \epsilon \quad \text{if stem contains vowel (e.g., motoring} \rightarrow \text{motor)} \\
\text{SSES} &\rightarrow \text{SS} \quad \text{(e.g., grasses} \rightarrow \text{grass)}
\end{aligned}
$$


分句
ElitesAI

分句

- !，？大多是明确的，但英文中的句号“.”很暧昧
  - 句子边界
  - 英文缩写Inc.或Dr.
  - 0.02%或4.3
- 常用算法
  - 首先分词
    - 因此，句点被分类为单词的一部分或句子边界。
  - 通常可以通过基于此分词的规则来完成分句。
    - 例如：标点符号、大小写

总结

ElitesAI

文本规范化

- 分词
  - 正则表达式、BPE
- 词规范化
  - 大小写还原、词目还原、词干提取
- 分句