自然语言处理

杨海钦

2025-2026-1学期

大纲

- 主要挑战
- 系统概览
- 小试牛刀
- 温故知新

自然语言处理的主要挑战

- 一词多义 Ambiguity
- 规模化 Scale
- 稀疏性 Sparsity
- 易变性 Variation
- 表达性 Expressivity
- 未建模的变量 Unmodeled Variables
- 未知表征 Unknown Representations

自然语言处理的挑战

- **一词多义 Ambiguity**
- **规模化 Scale**
- 稀疏性 Sparsity
- 易变性 Variation
- 表达性 Expressivity
- 未建模的变量 Unmodeled Variables
- 未知表征 Unknown Representations

一词多义 Ambiguity

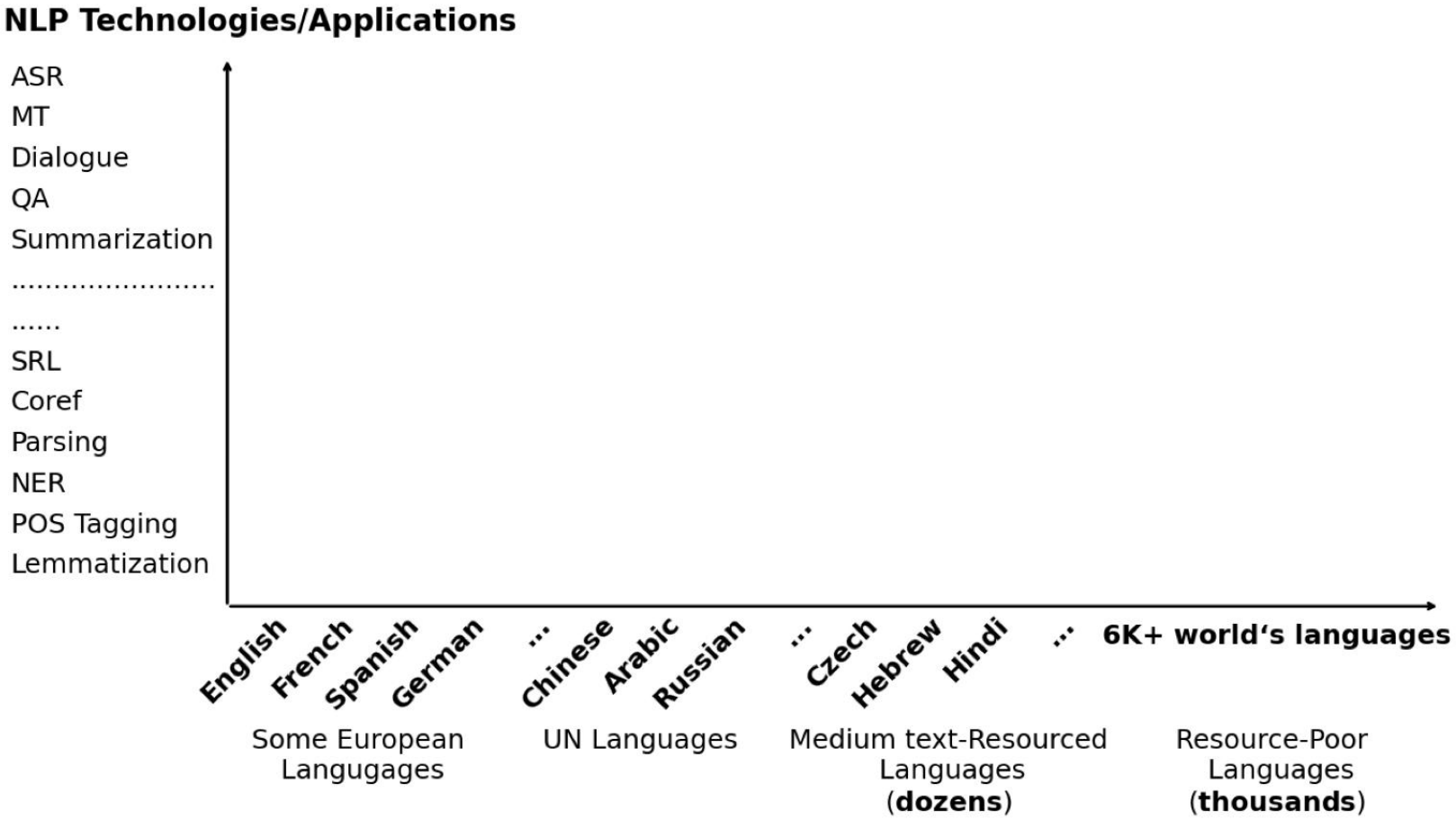
- 不同层次的多义
  - 词层面：
    - bank, apple
    - 包袱、门槛、算账、放水
    - ภาระ เกณฑ์ การตั้งถิ่นฐาน และการปลดปล่อย
    - Hebrew: נטל, סף, יישוב ושחרור
    - Arabic: العبء والعتبة والتسوية والإفراج
  - 特定领域： latex
  - 词性：
    - chair (noun or verb?)
    - 意思 (名词、形容词)

- 他说：“她这个人真有意思(funny)。”
- 她说：“他这个人怪有意思的(funny)。”
- 于是人们以为他们有了意思(wish)，并让他向她意思意思(express)。
- 他火了：“我根本没有那个意思(thought)！”
- 她也生气了：“你们这么说是什么意思(intention)？”
- 事后有人说：“真有意思(funny)。”也有人说：“真没意思(nonsense)”。

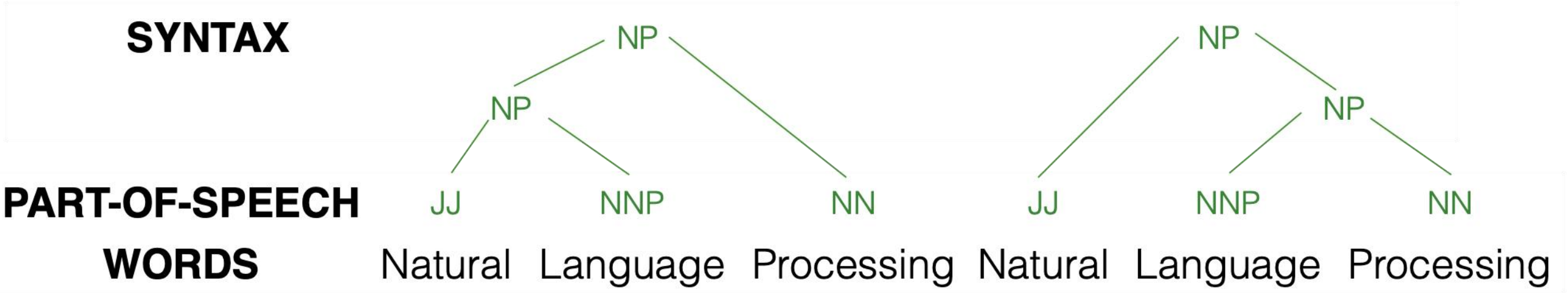
(原文见《生活报》1994.11.13.第六版) [吴尉天，1999]

规模化

- 语言众多、任务繁杂



句法多义



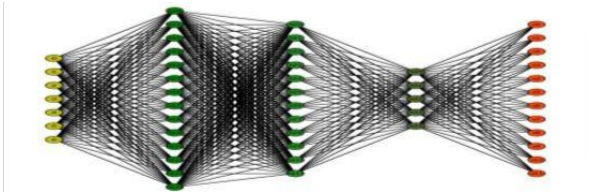
- Syntax: 句法
- Part-of-Speech tagging: 词性标注

句法和语义同时多义

- 中文例子：
  - 我们需要进口汽车。
  - 他在火车上写标语。
- 英文例子： We saw the woman with the telescope wrapped in paper.
  - 望远镜在谁那儿?
  - 谁或什么东西被纸包裹着?
  - 是某个场景、还是侵犯?

歧义性处理

- 如何为歧义建模并在上下文中选择正确的分析?
  - 非概率方法(词态学的有限状态机，语法的CKY解析器)返回所有可能的分析
  - 概率模型(隐马尔可夫模型用于词性标注， 概率上下文无关文法用于句法分析）和算法（Viterbi译码器，概率CKY） 返回最好的分析结果
  - 神经网络，预训练的语言模型现在提供端到端的解决方案



- 但“最好”的前提是概率准确。如何获得呢?

语料库

- 语料库是文本的集合
  - 通常以某种方式注释
  - 有时只是大量的文本
- 例子
  - OntoNotes 5.0：英/中/阿三语， 290万词深度语义标注库
  - 宾州数库(Penn Treebank, PTB):
    - 全球首个大规模句法标注语料库,
    - 《华尔街日报》文章，约100万词
  - 互联网：数十亿字
  - 亚马逊评论
  - …

大语言模型训练使用的数据

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

Training Compute-Optimal Large Language Models
https://arxiv.org/pdf/2203.15556

- 1T = 5,000,000本书
- 大学本科生知识量： 900本书约0.00018T
- 假设: 每本书0.2M tokens

自然语言处理的主要挑战

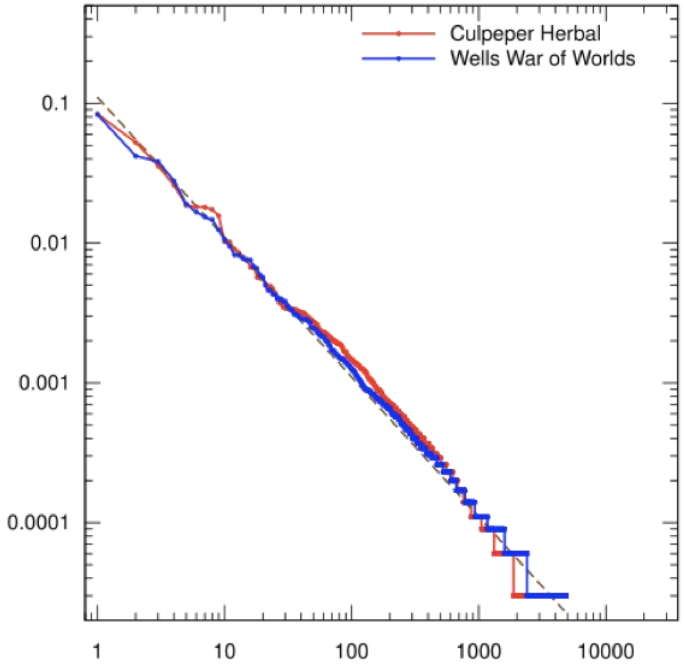
- 一词多义 Ambiguity
- 规模化 Scale
- **稀疏性 Sparsity**
- 易变性 Variation
- 表达性 Expressivity
- 未建模的变量 Unmodeled Variables
- 未知表征 Unknown Representations

稀疏性与齐夫定律

- 英语欧罗巴语料库统计结果

Frequency	Token
1,698,599	The
849,256	of
793,731	to
640,257	and
508,560	in
407,538	that
400,467	is
394,778	a
263,040	I

- 齐夫定律
  - Freq(n-th word) $\propto 1/n$



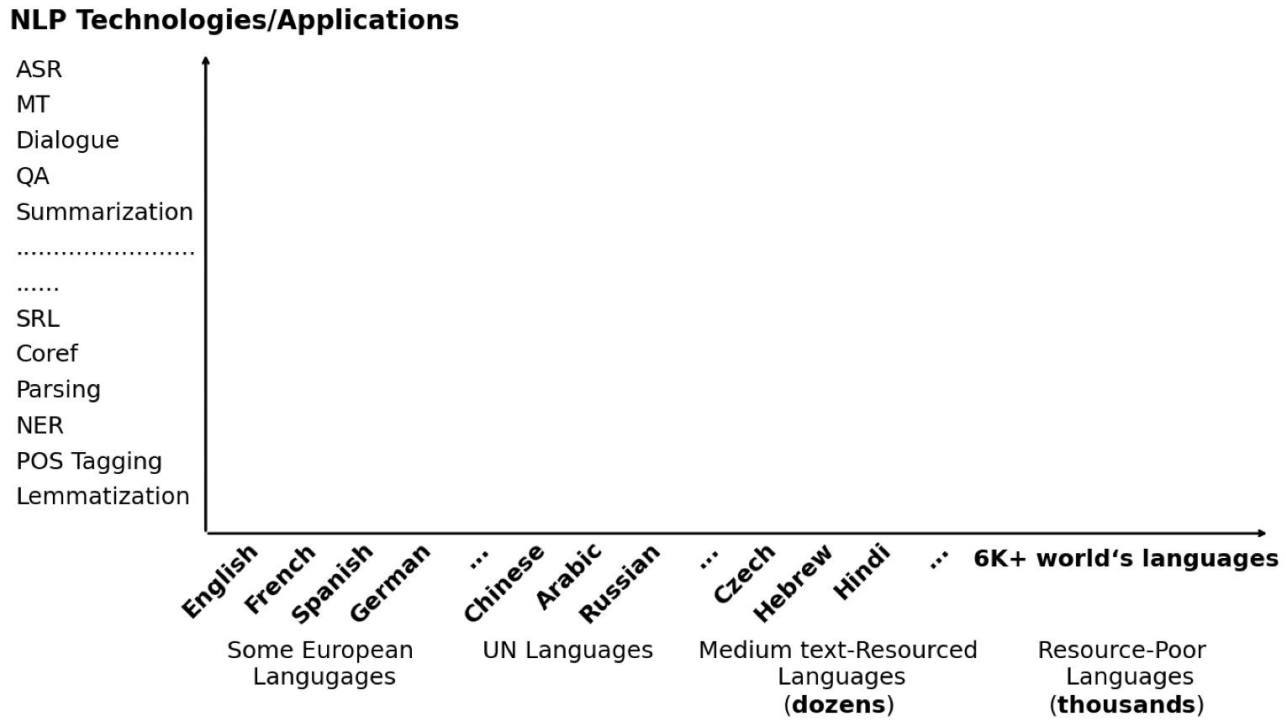
From https://en.wikipedia.org/wiki/Zipf%27s_law

自然语言处理的主要挑战

- 一词多义 Ambiguity
- 规模化 Scale
- 稀疏性 Sparsity
- **易变性 Variation**
- 表达性 Expressivity
- 未建模的变量 Unmodeled Variables
- 未知表征 Unknown Representations

易变性 Variation

- 数据来自不同领域、多样复杂
- 数据分布偏移
- 新词：
  - Covid、Niubility、润



自然语言处理的主要挑战

- 一词多义 Ambiguity
- 规模化 Scale
- 稀疏性 Sparsity
- 易变性 Variation
- **表达性 Expressivity**
- **未建模的变量 Unmodeled Variables**
- **未知表征 Unknown Representations**

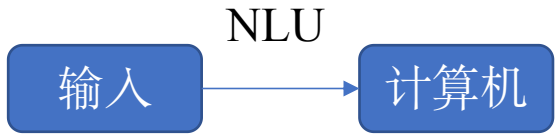
表达性 Expressivity

- 同一意思、多种表述
  - 父亲 vs. 爸爸 vs. 爹爹 vs. 家父 vs. 老爸
  - 我吃饭了 vs. 吾已用膳
  - 我打开了门 vs. 门被我打开了
  - 我饿得能吃下一头牛 vs. 我有点饿了 vs. 我饥肠辘辘

未建模的变量 Unmodeled Variables

- 世界知识
  - 我把杯子掉在地板上摔碎了
  - 我把锤子掉在玻璃上，把它打碎了

未知表征 Unknown Representations



- 如何把人类知识转化成机器中的表示?
  - 如何表示一个词或者一个句子?
  - 如何构建上下文(语境场景)?
  - 如何表示常识?

- 水果或手机
  - 他买了一个苹果
    - He bought an apple
    - He bought a new iPhone
- 上下文
  - 他没吃饭
    - 妈妈: 没吃正餐
    - 朋友: 没聚餐
    - 医生: 空腹
- 常识
  - 鸟会飞、但企鹅不会

大纲

- 主要挑战
- **系统概览**
- 小试牛刀
- 温故知新

自然语言处理系统通用算法框架

- 创建一个函数将输入X映射到输出Y，其中X和/或Y涉及语言
- 任务

输入X	输出Y	任务
文本	连续文本	语言模型 Language modeling
文本	其他语言的文本	机器翻译 Machine Translation
文本	概要	自动摘要 Automatic Summarization
文本	标签	文本分类 Text Classification
文本	语言结构	语言分析 Language Analysis
语音/ 文本	文本/ 语音	语音识别 ASR (Automatic Speech Recognition) /语音合成 TTS (Text-to-Speech)
图片/ 文本	文本/ 图片(视频)	图片描述 Image Captioning /文生图 T2I (文生视频 T2V)

创建NLP系统的方法

- 规则：手动创建规则

```
def classify(x: str) -> str:
    sports_keywords = ["baseball", "soccer", "football", "tennis"]
    if any(keyword in x for keyword in sports_keywords):
        return "sports"
    else:
        return "other"
```

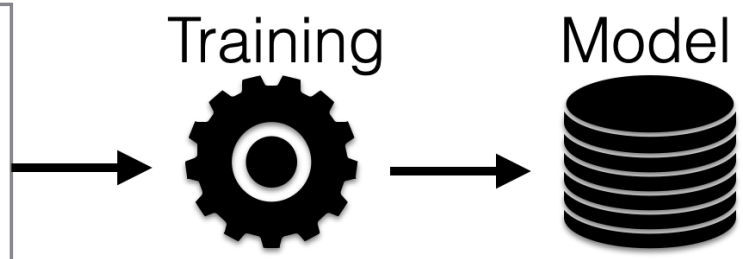
- 提示词：无需训练语言模型，仅需写相应的提示词

If the following sentences is about “sports” reply “sports”. Otherwise reply “other”.
{X}

→ **LM**

- 微调：基于成对的标注数据<*X*, *Y*>进行机器学习

I love to play baseball.	sports
The stock price is going up.	other
He got a hat-trick yesterday.	sports
He is wearing tennis shoes.	other



系统构建的数据需求

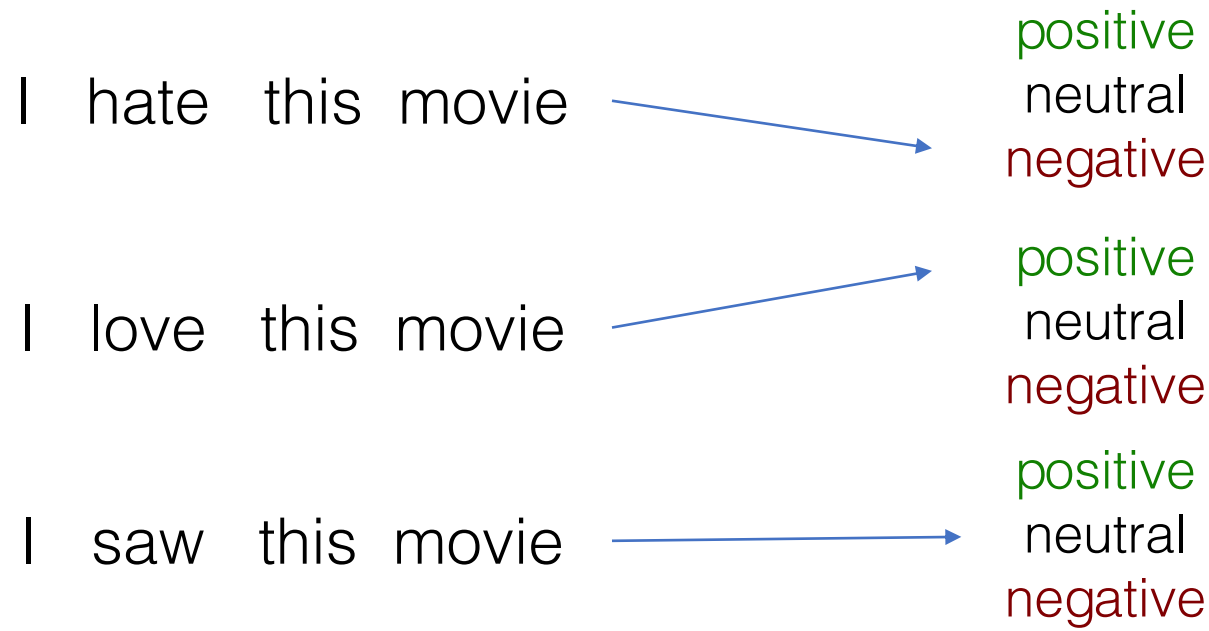
- **基于直觉的规则/提示词**：不需要训练数据，但也没有性能保证
- **基于抽查的规则/提示词**：少量数据，仅输入X
- **严格评估的规则/提示词**：输入X和输出Y的开发集（例如200-2000个示例）；有测试集更好
- **微调**：额外的训练集。越多越好—当数据量增加一倍时，通常精度会增加

大纲

- 主要挑战
- 系统概览
- **小试牛刀**
- 温故知新

示范任务：情感分析

- 给定评论网站上的评论(X)，决定其标签(Y)是positive (正面, 1)、negative (负面, -1), 还是neutral (中性, 0)



预测三步骤

- **特征提取**：从文本中提取出决策所需的显著特征
- **分数计算**：计算一个或多个可能性的分数
- **决策函数**：从几种可能性中选择一种

- **特征提取**：$\mathbf{h} = f(\mathbf{x})$
- **分数计算**：
  - 二分类：$s = \mathbf{w} \cdot \mathbf{h}$
  - 多分类：$\mathbf{s} = \mathbf{W}\mathbf{h}$
- **决策函数**：
  - $\hat{y} = \text{decide}(s/\mathbf{s})$

情感分类代码框架

- 关键5步
  - 特征抽取 Featurization
  - 得分 Scoring
  - 决策规则 Decision Rule
  - 精度计算 Accuracy Calculation
  - 错误分析 Error Analysis

Github
https://url2qr.com/fd8221

魔搭
https://url2qr.com/fd7b15

流程: 精度计算

- 精度计算 Accuracy Calculation

In [8]:

```
train_accuracy = calculate_accuracy(x_train, y_train)
test_accuracy = calculate_accuracy(x_test, y_test)
print(f'Train accuracy: {train_accuracy}')
print(f'Dev/test accuracy: {test_accuracy}')
```

In [6]:

```
def calculate_accuracy(x_data: list[str], y_data: list[int]) -> float:
    total_number = 0
    correct_number = 0
    for x, y in zip(x_data, y_data):
        y_pred = run_classifier(x)
        total_number += 1
        if y == y_pred:
            correct_number += 1
    return correct_number / float(total_number)
```

流程:决策规则 Decision Rule /得分 Scoring

In [5]:

```
def run_classifier(x: str) -> int:
    score = 0
    for feat_name, feat_value in extract_features(x).items():
        score = score + feat_value * feature_weights.get(feat_name, 0)
    if score > 0:
        return 1
    elif score < 0:
```

数据加载

In [2]:

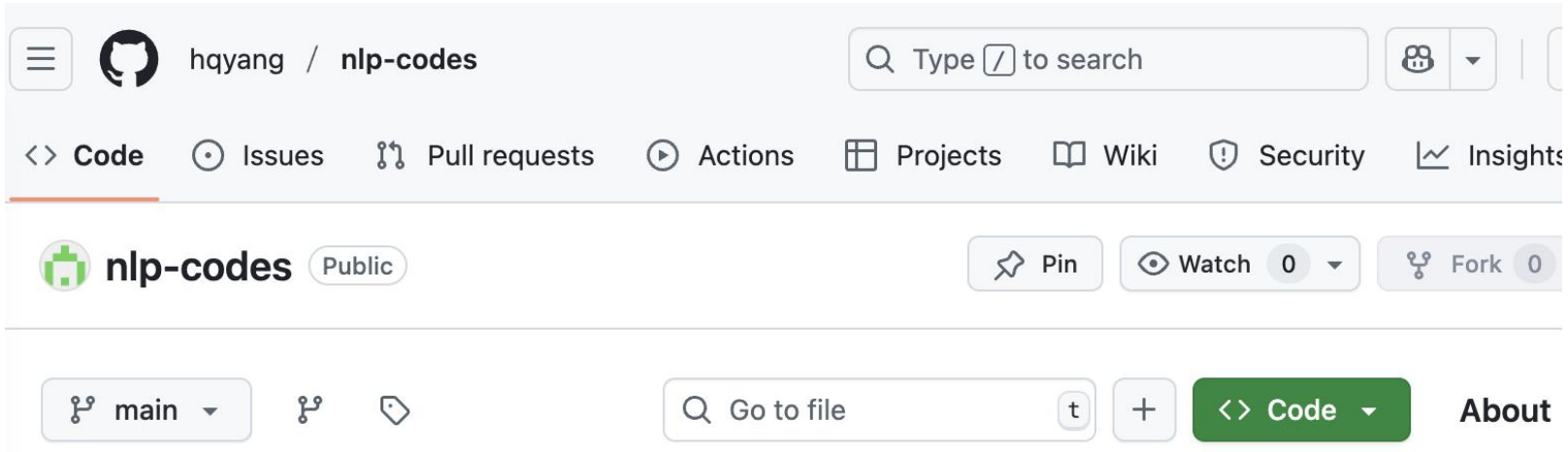
```
def read_xy_data(filename: str) -> tuple[list[str], list[int]]:
    x_data = []
    y_data = []
    with open(filename, 'r') as f:
        for line in f:
            label, text = line.strip().split(' ||| ')
            x_data.append(text)
            y_data.append(int(label))
    return x_data, y_data
```

In [3]:

```
x_train, y_train = read_xy_data('../data/sst-sentiment-text-threeclass/train.txt')
x_test, y_test = read_xy_data('../data/sst-sentiment-text-threeclass/dev.txt')
```

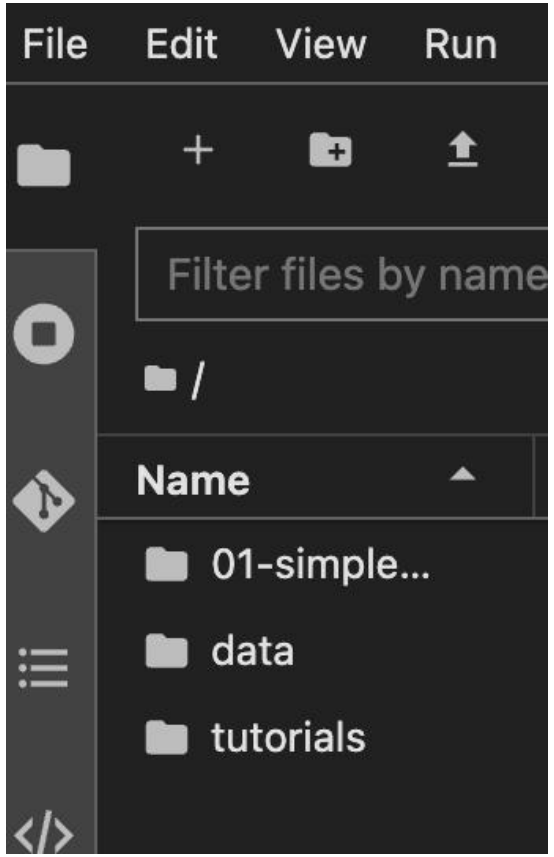
数据

https://github.com/hqyang/nlp-codes/



data/sst-sentiment-text-threeclass

魔搭
https://url2qr.com/fd7b15



流程: 特征抽取 Featurization/权重

In [1]:

```python
def extract_features(x: str) -> dict[str, float]:
    features = {}
    x_split = x.split(' ')

    # Count the number of "good words" and "bad words" in the text
    good_words = ['love', 'good', 'nice', 'great', 'enjoy', 'enjoyed']
    bad_words = ['hate', 'bad', 'terrible', 'disappointing', 'sad', 'lost', 'angry']
    for x_word in x_split:
        if x_word in good_words:
            features['good_word_count'] = features.get('good_word_count', 0) + 1
        if x_word in bad_words:
            features['bad_word_count'] = features.get('bad_word_count', 0) + 1

    # The "bias" value is always one, to allow us to assign a "default" score to the text
    features['bias'] = 1

    return features

feature_weights = {'good_word_count': 1.0, 'bad_word_count': -1.0, 'bias': 0.5}
```

流程:错误分析 Error Analysis

In [9]:

```
import random
def find_errors(x_data, y_data):
    error_ids = []
    y_preds = []
    for i, (x, y) in enumerate(zip(x_data, y_data)):
        y_preds.append(run_classifier(x))
        if y != y_preds[-1]:
            error_ids.append(i)
    for _ in range(5):
        my_id = random.choice(error_ids)
        x, y, y_pred = x_data[my_id], y_data[my_id], y_preds[my_id]
        print(f'{x}\ntrue label: {y}\npredicted label: {y_pred}\n')
```

In [10]:

```
find_errors(x_train, y_train)
```

思考: 如何改进算法?

1. 发现问题?
   → 看错误分析
2. 修改系统（特征、评分函数等）
3. 测量精度改进，接受/拒绝变更
4. 从1开始重复
5. 最后，当对开发集精度满意时，在测试集中进行评估!

常见困难

低频词汇 Low-Frequency Words

- negative
  - The action switches between past and present , but the material link is too **tenuous** to anchor the emotional connections that **purport** to span a 125-year divide .
  - 故事情节在过去和现在之间切换，但物质上的联系过于**脆弱**，无法**支撑**起据称跨越125年鸿沟的情感联系。
  - Here 's yet another studio horror franchise **mucking** up its storyline with **glitches** casual fans could correct in their sleep .
  - 这是又一部电影公司制作的恐怖系列电影，它的故事情节被一些普通影迷在睡梦中都能纠正的**小毛病搞砸**了。

怎么解决?

- 继续添加相关词汇，直到获得全部?
- 整合外部资源，如情感词典?

词形变化 Conjugation

- positive
  - An operatic , sprawling picture that 's **entertainingly** acted , **magnificently** shot and gripping enough to sustain most of its 170-minute length .
  - 这是一幅歌剧般的、杂乱无章的画面，表演**有趣**，拍摄**精美**，扣人心弦，足以支撑其170分钟的长度。
- negative
  - It 's basically an **overlong** episode of Tales from the Crypt .
  - 它基本上是《地穴传说》的一个**过长**片段。

怎么解决?

- 使用词根形式或词性标注(Part-of-Speech Tagging)?

注：需要词法分析(Morphological Analysis)

否定修饰 Negation

- positive
  - This one is not nearly as dreadful as expected .
  - 这次并不像预期的那么可怕。
- negative
  - Serving Sara does n't serve up a whole lot of laughs .
  - 为莎拉服务并没有给她带来很多欢笑。

怎么解决?

- 如果否定修饰了一个词，忽略它?

注：需要句法分析(Syntactic Analysis)

隐喻Metaphor 或 类比Analogy

- positive
  - Puts a human face on a land most Westerners are unfamiliar with .
  - 为大多数西方人不熟悉的土地赋予人性。
- negative
  - Green might want to hang onto that ski mask , as robbery may be the only way to pay for his next project .
  - 格林可能想保留那个滑雪面罩，因为抢劫可能是支付他下一个项目的唯一方式。
  - Has all the depth of a wading pool .
  - 有一个浅水池那么深。

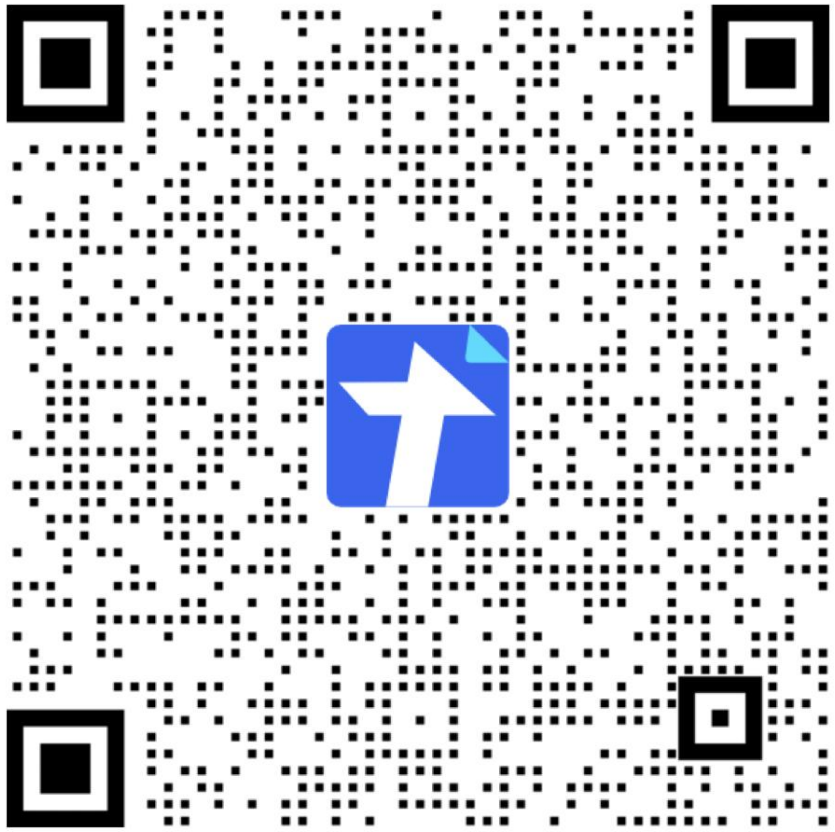
怎么解决???

其他语言

- positive
  - 見事に視聴者の心を掴む作品でした。
  - 这是一部真正俘获观众心的作品。
- negative
  - モンハンの名前がついてるからとりあえずモンハン要素をちょこちょこ入れればいいだろ感が凄い。
  - 由于其中有怪物猎人的名字，感觉他们应该在这里或那里加入一些怪物猎人的元素。

怎么解决？学习日语吗？

课堂小测



NLP-Fall25群/头歌

- 【教学课堂邀请】杨海钦老师邀请您加入头歌平台教学课堂-《自然语言处理-深技大-2025秋》，您可以复制邀请码，在下方的链接中，点击“加入课堂”按钮加入该教学课堂。
- 链接：https://www.educoder.net/classrooms/mchwjqnx?code=BHTWK
- 邀请码：BHTWK

大纲

- 主要挑战
- 系统概览
- 小试牛刀
- **温故知新**

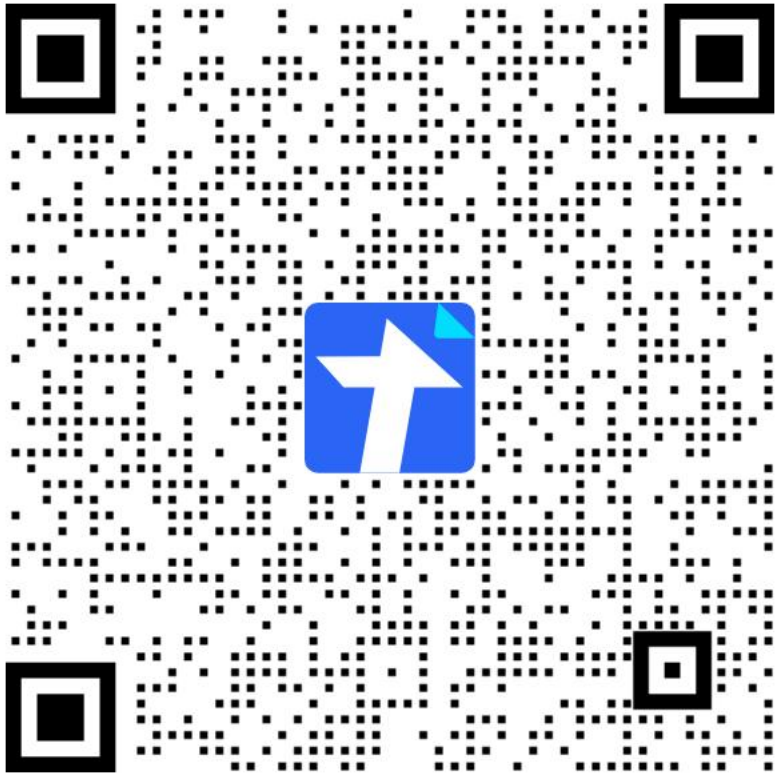
Python练习

- 集合 Collections
  - 列表 Lists
  - 元组 Tuples
  - 字典 Dictionary
- 循环 Loops
- NumPy
  - 阵列运算 Array Operations
  - 矩阵乘法 Matrix Multiplication
  - 点积 Dot Product
  - 索引 Indexing
  - 广播 Broadcasting
  - 高效NumPy代码

Github:
python_tutorial_questions.md
https://url2qr.com/108adb

一句话总结

- 自然语言处理的主要挑战
  - 一词多义、规模化、稀疏性、易变性等
    - 齐夫定律
- 系统概览
  - 通用算法框架
  - 三种实现方法及对数据的需求
- 基于规则的实现
  - 代码框架5步骤
  - 五种常见困难情形



1. 课后阅读: [SLP]第二章; python_review.pdf
2. 项目分组