Algorithm Practice Exam (Paper-and-Pencil Mode)

1. Sentiment Analysis (Graded)

English	Chinese (中文)
Problem: Calculate the sentiment score of the tweet: 'The service was not very good.' Given Lexicon: - good: +2 - service: 0 (neutral) Rules: 1. 'very' (intensifier): increases the score of the next word by 50%. 2. 'not' (negator): multiplies the score of the following phrase by -0.8.	题目: 计算推文 Sentiemnt 分数：'The service was not very good.' 给定词典： - good: +2 - service: 0 (中性) 规则： 1. 'very' (加强词): 将下一个词的分数增加 50%。 2. 'not' (否定词): 将后续短语的分数乘以 -0.8。
Solution: 1. **Identify Sentiment Words**: 'good' (+2). 2. **Apply Intensifier**: 'very' modifies 'good'. Score('very good') = Score('good') * 1.5 = 2 * 1.5 = 3. 3. **Apply Negation**: 'not' modifies 'very good'. Score('not very good') = Score('very good') * -0.8 = 3 * -0.8 = -2.4. **Final Score**: -2.4.	解析: 1. **识别情感词**: 'good' (+2)。 2. **应用加强词**: 'very' 修饰 'good'。分数('very good') = 分数('good') * 1.5 = 2 * 1.5 = 3。 3. **应用否定词**: 'not' 修饰 'very good'。分数('not very good') = 分数('very good') * -0.8 = 3 * -0.8 = -2.4。 **最终分数**: -2.4。

2. Pointwise Mutual Information (PMI)

English	Chinese (中文)
Problem: Calculate PMI(coffee, cup). Corpus Statistics: - Total word pairs (N): 10,000 - Count('coffee') (c_x): 100 - Count('cup') (c_y): 200 - Count('coffee', 'cup') co-occurrence (c_{xy}): 20	题目: 计算 PMI(coffee, cup)。 语料库统计： - 总词对数 (N): 10,000 - Count('coffee') (c_x): 100 - Count('cup') (c_y): 200 - Count('coffee', 'cup') 共现 (c_{xy}): 20
Solution: Formula: $PMI(x, y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$ 1. **Calculate Probabilities**: $P(x) = 100 / 10000 = 0.01$ $P(y) = 200 / 10000 = 0.02$ $P(x,y) = 20 / 10000 = 0.002$ 2. **Compute Ratio**: $\frac{0.002}{0.01 \times 0.02} = \frac{0.002}{0.0002} = 10$ 3. **Logarithm**: $PMI = \log_2(10) \approx 3.32$	解析: 公式: $PMI(x, y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$ 1. **计算概率**: $P(x) = 100 / 10000 = 0.01$ $P(y) = 200 / 10000 = 0.02$ $P(x,y) = 20 / 10000 = 0.002$ 2. **计算比率**: $\frac{0.002}{0.01 \times 0.02} = \frac{0.002}{0.0002} = 10$ 3. **取对数**: $PMI = \log_2(10) \approx 3.32$

3. Naive POS Tagging

English	Chinese (中文)
Problem: Tag the word 'tweeted' in the sentence 'He tweeted.' Training Data: - 'tweet': {Verb: 10, Noun: 2} - 'tweeted': Not in training data (Unknown) Rules for Unknown Words: 1. If capitalized -> NNP (Proper Noun) 2. If ends with 'ed' -> VBD (Past Tense Verb) 3. Default -> NN (Noun)	题目: 为句子 'He tweeted' 中的 'tweeted' 标注词性。 训练数据： - 'tweet': {Verb: 10, Noun: 2} - 'tweeted': 不在训练数据中 (未知词) 未知词规则： 1. 如果首字母大写 -> NNP (专有名词) 2. 如果以 'ed' 结尾 -> VBD (过去式动词) 3. 默认 -> NN (名词)
Solution: 1. **Lookup**: Check 'tweeted' in the training dictionary. **Result**: Not found (Unknown). 2. **Apply Unknown Word Rules** (in order): - Is it capitalized? No ('t' is lowercase). - Does it end with 'ed'? **Yes**. 3. **Conclusion**: Assign tag **VBD** based on the suffix rule.	解析: 1. **查找**: 在训练字典中查找 'tweeted'。**结果**: 未找到 (未知词)。 2. **应用未知词规则** (按顺序): - 首字母大写吗？ 否 ('t' 小写)。 - 以 'ed' 结尾吗？ **是**。 3. **结论**: 根据后缀规则标记为 **VBD**。

4. Jaccard Similarity

English	Chinese (中文)
Problem: Calculate the Jaccard Similarity between Doc A and Doc B. - Doc A: 'apple banana apple' - Doc B: 'apple orange banana'	题目: 计算文档 A 和文档 B 的 Jaccard 相似度。 - Doc A: 'apple banana apple' - Doc B: 'apple orange banana'
Solution: 1. **Create Sets** (Unique words): $Set(A) = \{\text{'apple', 'banana'}\}$ $Set(B) = \{\text{'apple', 'orange', 'banana'}\}$ 2. **Intersection** ($A \cap B$): Words in both: 'apple', 'banana'. Count = 2. 3. **Union** ($A \cup B$): All unique words: 'apple', 'banana', 'orange'. Count = 3. 4. **Calculate Jaccard**: $J(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|} = \frac{2}{3} \approx 0.67$	解析: 1. **创建集合** (唯一词): $Set(A) = \{\text{'apple', 'banana'}\}$ $Set(B) = \{\text{'apple', 'orange', 'banana'}\}$ 2. **交集** ($A \cap B$): 两者都有的词: 'apple', 'banana'。数量 = 2。 3. **并集** ($A \cup B$): 所有唯一词: 'apple', 'banana', 'orange'。数量 = 3。 4. **计算 Jaccard**: $J(A,B) = \frac{\|A \cap B\|}{\|A \cup B\|} = \frac{2}{3} \approx 0.67$