# Exam Concept Q&A; Review (Bilingual)

| English | Chinese (中文) |
|---|---|
| [Information Retrieval] Q: Explain stoplist removal, capitalisation, and stemming in Information Retrieval.<br><br>A: **Stoplist removal**: Removing common words (like 'the', 'is') that appear frequently but carry little meaning, saving space and improving relevance.<br>**Capitalisation**: Converting all text to lowercase so that 'Word' and 'word' are treated as the same term.<br>**Stemming**: Reducing words to their root form (e.g., 'running' -> 'run') so that different grammatical forms match the same query. | [Information Retrieval] 问: 解释信息检索中的停用词移除、大小写转换和词干提取。<br><br>答: **停用词移除**：移除那出现频率高但意义不大的常用词（如 'the', 'is'），以节省空间并提高相关性。**大小写转换**：将所有文本转换为小写，以便 'Word' 和 'word' 被视为同一个词。**词干提取**：将单词还原为其词根形式（例如 'running' -> 'run'），以便不同的语法形式能匹配同一个查询。 |
| [Information Retrieval] Q: Explain the intuition behind TF.IDF.<br><br>A: **TF (Term Frequency)**: Measures how often a word appears in a document. High TF means the word is important in that document. **IDF (Inverse Document Frequency)**: Measures how rare a word is across all documents. Words that appear everywhere (like 'the') have low IDF. **TF.IDF** combines them: it gives high weight to words that are frequent in the specific document but rare in the collection, helping to identify the document's unique topic. | [Information Retrieval] 问: 解释 TF.IDF 背后的直觉。<br><br>答: **TF（词频）**：衡量单词在文档中出现的频率。高 TF 意味着该词在该文档中很重要。**IDF（逆文档频率）**：衡量单词在所有文档中有多罕见。到处出现的词（如 'the'）具有低 IDF。**TF.IDF** 结合了两者：它赋予那些在特定文档中频繁出现但在整个集合中罕见的单词更高的权重，有助于识别文档的独特主题。 |
| [Information Retrieval] Q: Explain Precision, Recall, and F-measure. Why is Recall hard to compute for the Web?<br><br>A: **Precision**: The fraction of retrieved documents that are relevant (Quality). **Recall**: The fraction of relevant documents that were retrieved (Quantity).<br>**F-measure**: The harmonic mean of Precision and Recall, providing a single score. **Web Recall**: It is impossible to count the total number of relevant pages on the entire internet, so we cannot calculate the denominator for Recall. | [Information Retrieval] 问: 解释准确率 (Precision)、召回率 (Recall) 和 F-度量 (F-measure)。为什么在网络上计算召回率很难？<br><br>答: **准确率**：检索到的文档中相关文档的比例（质量）。**召回率**：相关文档中被检索到的比例（数量）。**F-度量**：准确率和召回率的调和平均数，提供单一评分。**网络召回率**：不可能统计整个互联网上相关页面的总数，因此我们无法计算召回率的分母。 |
| [Information Retrieval] Q: Explain the intuition behind the PageRank algorithm.<br><br>A: PageRank treats links as votes. A page is important if many other pages link to it. Furthermore, a link from an *important* page counts more than a link from an unimportant one. It simulates a 'random surfer' clicking links; pages visited more often get a higher rank. | [Information Retrieval] 问: 简述 PageRank 算法背后的直觉。<br><br>答: PageRank 将链接视为投票。如果有许多其他页面链接到一个页面，该页面就很重要。此外，来自*重要*页面的链接比来自不重要页面的链接更有分量。它模拟“随机冲浪者”点击链接；被访问次数越多的页面排名越高。 |

| | |
|---|---|
| [NLG] Q: Describe the stages of the Natural Language Generation (NLG) pipeline.<br><br>A: 1. **Content Determination**: Deciding *what* information to say. 2. **Text Structuring**: Organizing the order of information. 3. **Sentence Aggregation**: Combining small facts into sentences. 4. **Lexicalisation**: Choosing the specific words. 5. **Referring Expression Generation**: Deciding how to refer to entities (e.g., 'it', 'the dog'). 6. **Realisation**: Creating the final grammatical sentence string. | [NLG] 问: 描述自然语言生成 (NLG) 流水线的各个阶段。<br><br>答: 1. **内容确定**：决定要说*什么*信息。2. **文本结构化**：组织信息的顺序。3. **句子聚合**：将小事实组合成句子。4. **词汇化**：选择具体的词语。5. **指称表达生成**：决定如何指代实体（例如，'它'，'那只狗'）。6. **实现**：生成最终的符合语法的句子字符串。 |
| [NLP Basics] Q: Differentiate between Inflectional and Derivational Morphology.<br><br>A: **Inflectional**: Changes the form of a word for grammar (e.g., tense, number) without changing its core meaning or class (e.g., 'cat' -> 'cats', 'walk' -> 'walked'). **Derivational**: Creates a new word, often changing the meaning or part of speech (e.g., 'happy' -> 'happiness', 'compute' -> 'computer'). | [NLP Basics] 问: 区分屈折形态学 (Inflectional) 和派生形态学 (Derivational)。<br><br>答: **屈折**：为了语法（如时态、数）改变单词的形式，而不改变其核心含义或词性（例如，'cat' -> 'cats'，'walk' -> 'walked'）。**派生**：创造一个新词，通常改变含义或词性（例如，'happy' -> 'happiness'，'compute' -> 'computer'）。 |
| [Machine Translation] Q: Explain Direct, Transfer-based, and Interlingual approaches to MT.<br><br>A: **Direct**: Word-for-word translation. Fast but poor grammar. **Transfer-based**: Converts source structure to target structure using rules. Better quality but requires specific rules for each language pair. **Interlingual**: Translates source to a universal meaning representation (Interlingua), then to target. Good for many languages, but defining a universal representation is very hard. | [Machine Translation] 问: 解释机器翻译中的直接、基于转换和中间语言方法。<br><br>答: **直接**：逐字翻译。速度快但语法差。**基于转换**：使用规则将源结构转换为目标结构。质量较好，但需要为每对语言制定特定规则。**中间语言**：将源语言翻译成通用的含义表示（中间语），然后翻译成目标语言。适用于多种语言，但定义通用表示非常困难。 |
| [Machine Translation] Q: What is the Noisy Channel Model in Machine Translation?<br><br>A: It models translation as a decoding problem. We assume the source sentence (e.g., French) is a 'distorted' version of the target sentence (e.g., English) passed through a noisy channel. The goal is to find the English sentence that is most likely to have produced the observed French sentence using probability (combining Translation Model and Language Model). | [Machine Translation] 问: 什么是机器翻译中的噪声信道模型？<br><br>答: 它将翻译建模为解码问题。我们要假设源句子（例如法语）是目标句子（例如英语）经过噪声信道后的“失真”版本。目标是利用概率（结合翻译模型和语言模型）找到最可能产生观察到的法语句子的英语句子。 |
| [Machine Translation] Q: Explain the BLEU measure.<br><br>A: BLEU (Bilingual Evaluation Understudy) automatically evaluates MT quality. It counts the overlap of N-grams (sequences of N words) between the machine translation and one or more human reference translations. It rewards matching phrases and penalizes translations that are too short (Brevity Penalty). High BLEU score means better quality. | [Machine Translation] 问: 解释 BLEU 指标。<br><br>答: BLEU（双语评估替补）自动评估机器翻译质量。它计算机器翻译与一个或多个人工参考翻译之间的 N-gram（N 个单词的序列）重叠度。它奖励匹配的短语，并惩罚过短的翻译（简短惩罚）。高 BLEU 分数意味着更好的质量。 |

| | |
|---|---|
| [Sentiment Analysis] Q: Differentiate Subjectivity from Sentiment.<br>A: **Subjectivity Classification**: Determines if a text contains an opinion/emotion or is objective (factual). **Sentiment Analysis**: Determines the polarity of that opinion (Positive, Negative, or Neutral). Usually, we first detect subjectivity, then analyze sentiment. | [Sentiment Analysis] 问: 区分主观性 (Subjectivity) 和情感 (Sentiment)。<br>答: **主观性分类**：确定文本是包含观点/情感还是客观的（事实）。**情感分析**：确定该观点的极性（积极、消极或中性）。通常，我们先检测主观性，然后分析情感。 |
| [Sentiment Analysis] Q: Explain Bing Liu's model for an opinion.<br>A: An opinion is defined as a quintuple $(e, a, s, h, t)$, where: **e (Entity)**: The target object (e.g., iPhone). **a (Aspect)**: The feature of the object (e.g., battery). **s (Sentiment)**: The sentiment value (+/-). **h (Holder)**: The person expressing the opinion. **t (Time)**: When it was expressed. | [Sentiment Analysis] 问: 解释 Bing Liu 的观点模型。<br>答: 观点被定义为一个五元组 $(e, a, s, h, t)$，其中：**e (实体)**：目标对象（例如 iPhone）。**a (方面)**：对象的特征（例如电池）。**s (情感)**：情感值（+/-）。**h (持有者)**：表达观点的人。**t (时间)**：表达的时间。 |
| [Sentiment Analysis] Q: Explain the Naive Bayes classifier intuition and its 'naive' assumption.<br>A: **Intuition**: It calculates the probability that a document belongs to a class (e.g., Positive) based on the probabilities of its words appearing in that class. **Naive Assumption**: It assumes that the occurrence of each word is **independent** of all other words. This is 'naive' because in language, words are often correlated (e.g., 'not' affects the next word), but the model still works well in practice. | [Sentiment Analysis] 问: 解释朴素贝叶斯分类器的直觉及其“朴素”假设。<br>答: **直觉**：它根据单词在某类别中出现的概率，计算文档属于该类别（例如积极）的概率。**朴素假设**：它假设每个单词的出现与其他所有单词**独立**。这很“朴素”，因为在语言中，单词通常是相关的（例如 'not' 影响下一个词），但该模型在实践中仍然效果很好。 |
| [Information Extraction] Q: What is Relation Extraction? Detection vs. Classification.<br>A: **Relation Extraction**: Identifying semantic relationships between entities in text (e.g., identifying 'Steve Jobs' is the 'founder' of 'Apple'). **Relation Detection**: Deciding *if* there is a relation between two entities. **Relation Classification**: Deciding *what type* of relation helps (e.g., Founder, CEO, Spouse). | [Information Extraction] 问: 什么是关系抽取？检测与分类有何区别？<br>答: **关系抽取**：识别文本中实体之间的语义关系（例如，识别“乔布斯”是“苹果”的“创始人”）。**关系检测**：决定两个实体之间*是否*存在关系。**关系分类**：决定关系的*类型*（例如，创始人、CEO、配偶）。 |
| [Information Extraction] Q: Explain Bootstrapping for Relation Extraction.<br>A: Bootstrapping allows learning with very little data. 1. Start with a few 'seed' examples of a relation (e.g., [Barack Obama, USA]). 2. Find sentences containing these seeds to learn patterns. 3. Use patterns to find new pairs. 4. Add new pairs as seeds and repeat. **Pros**: Needs little data. **Cons**: Errors can accumulate (semantic drift). | [Information Extraction] 问: 解释用于关系抽取的 Bootstrapping（自举法）。<br>答: Bootstrapping 允许用极少的数据进行学习。1. 从关系的几个“种子”示例开始（例如 [Barack Obama, USA]）。2. 找到包含这些种子的句子以学习模式。3. 使用模式找到新的配对。4. 将新配对添加为种子并重复。**优点**：需要数据少。**缺点**：错误会累积（语义漂移）。 |

| | |
|---|---|
| [Compression] Q: Explain LZ77 compression.<br>A: LZ77 is a dictionary-based sliding window compression. It replaces repeated sequences with a pointer to where they occurred previously. The pointer is (Distance, Length), meaning 'go back D characters and copy L characters'. This exploits local repetition in data. | [Compression] 问: 解释 LZ77 压缩。<br>答: LZ77 是一种基于字典的滑动窗口压缩。它用指向先前出现位置的指针替换重复序列。指针为 (距离, 长度)，意为“向后 D 个字符并复制 L 个字符”。这利用了数据的局部重复性。 |
| [Compression] Q: Sketch the Huffman Coding algorithm.<br>A: 1. Calculate frequency of all characters. 2. Treat each character as a leaf node. 3. Repeatedly combine the two nodes with the lowest frequencies into a new parent node (summing their frequencies) until one tree remains. 4. Assign '0' and '1' to the edges. Frequent characters end up near the top (short codes), rare ones at the bottom (long codes). | [Compression] 问: 简述霍夫曼编码算法。<br>答: 1. 计算所有字符的频率。2. 将每个字符视为叶节点。3. 重复将频率最低的两个节点合并为一个新的父节点（频率求和），直到只剩一棵树。4. 为边缘分配 '0' 和 '1'。频繁字符位于顶部附近（短代码），稀有字符位于底部（长代码）。 |
| [Compression] Q: Lossy vs. Lossless Compression.<br>A: **Lossless**: Original data is perfectly reconstructed (e.g., ZIP, Text). Essential for text/code. **Lossy**: Some data is discarded to achieve higher compression (e.g., JPEG, MP3). Good for images/audio where human perception tolerates small errors. | [Compression] 问: 有损 vs. 无损压缩。<br>答: **无损**：原始数据被完美重建（如 ZIP，文本）。对文本/代码至关重要。**有损**：丢弃部分数据以实现更高压缩率（如 JPEG，MP3）。适用于人类感知容忍小错误的图像/音频。 |