Past Paper

Main

Question 1

1. How does inflectional and derivational morpholoy affect the computational performance of NLP applications? Give an example of this issue in relation ot a particular application in your answer. Be clear to distinguish inflectional from derivational morphology
    - 屈折和派生形态如何影响 NLP 应用的计算性能？ 请在您的答案中给出与特定应用程序相关的此问题的示例。 清楚地区分屈折形态和派生形态

> **Inflectional Morphology**
>
> 屈折形态
>
> **Definition:** Inflectional morphology involves the modification of words to express different grammatical categories such as tense, case, voice, aspect, person, number, gender, and mood. Inflection does not change the part of speech of a word nor create a new word, but rather provides **different grammatical forms of the same word**.
> **定义：** 屈折形态涉及对单词进行修饰以表达不同的语法类别，例如时态、格、语态、体、人称、数字、性别和语气。 词形变化不会改变单词的词性，也不会创建新单词，而是为同一单词提供不同的语法形式。
>
> **Impact on NLP:** Inflectional morphology primarily affects NLP applications by increasing the number of word forms that a system needs to recognize as essentially the same lexical item. This can impact computational performance by inflating the vocabulary size, which in turn can increase the computational resources needed for processes like parsing, tagging, and indexing.
> **对 NLP 的影响：** 屈折形态主要通过增加系统需要识别为基本相同词汇项的词形数量来影响 NLP 应用。 这可能会通过扩大词汇量来影响计算性能，进而增加解析、标记和索引等过程所需的计算资源。
>
> **Example:** Consider a machine translation system. In languages with rich inflectional morphology, like Russian or Arabic, verbs and nouns undergo numerous inflections. For instance, the English verb "run" can be translated into Russian as "бегу" (first person), "бежит" (third person), "бежал" (past tense), etc. If the system does not appropriately handle these inflections, it might fail to recognize that these words are related, potentially leading to errors in translation where verb forms are mismatched or misunderstood.
> **示例：** 考虑一个机器翻译系统。 在具有丰富屈折形态的语言中，例如俄语或阿拉伯语，动词和名词会经历多种屈折变化。 例如，英语动词“run”可以翻译成俄语为“бегу”（第一人称）、“бежит”（第三人称）、“бежал”（过去时）等。如果系统没有正确处理这些词形变化，它可能无法识别这些单词是相> 关的，从而可能导致翻译错误，即动词形式不匹配或被误解。
>
> **Derivational Morphology**
>
> 衍生形态学
>
> **Definition:** Derivational morphology involves creating a new word from an existing one, often changing the part of speech or the basic meaning of the original word. For example, the noun "activation" is derived from the verb "activate".
> **定义：** 派生词法涉及从现有单词创建新单词，通常会改变词性或原始单词的基本含义。 例如，名词“activation”源自动词“activate”。
>
> **Impact on NLP:** Derivational morphology complicates NLP by adding layers of semantic complexity and ambiguity. New words must be understood and processed within their specific contexts, which may vary widely in meaning from the root word. This increases the lexical richness and ambiguity that NLP systems must manage, impacting their computational performance by requiring more sophisticated semantic analysis tools and larger, more complex linguistic databases.
> **对 NLP 的影响：** 派生形态学通过增加语义复杂性和歧义性使 NLP 变得复杂。 新词必须在其特定的上下文中被理解和处理，其含义可能与词根有很大不同。 这增加了 NLP 系统必须管理的词汇丰富性和歧义性，需要更复杂的语义分析工具和更大、更复杂的语言数据库，从而影响其计算性能。
>
> **Example:** In sentiment analysis, understanding the sentiment conveyed in text is crucial. Derivational morphology can affect sentiment polarity. For instance, from the verb "excite" (which may carry a positive sentiment), we derive "excitable" (often positive) and "excitation" (neutral). An NLP system must parse these derivations correctly to maintain the sentiment context in user reviews or social media posts. Misinterpreting these can lead to incorrect sentiment results, affecting the overall accuracy of sentiment analysis applications.
> **示例：** 在情感分析中，理解文本中传达的情感至关重要。 派生形态可以影响情感极性。 例如，从动词“兴奋”（可能带有积极情绪），我们派生出“兴奋”（通常是积极的）和“兴奋”（中性）。 NLP 系统必须正确解析这些推导，以维护用户评论或社交媒体帖子中的情绪上下文。 误解这些可能会导致错误的> 情绪结果，影响情绪分析应用程序的整体准确性。
>
> **Addressing Morphological Challenges in NLP**
>
> 解决 NLP 中的形态挑战
>
> **Solutions:**
>
> 1. **Lemmatization and Stemming:** These techniques reduce words to their base or root form, helping to manage the inflectional forms in text data.
>   - **词形还原和词干提取：** 这些技术将单词简化为其基本形式或词根形式，有助于管理文本数据中的屈折形式。
> 2. **Morphological Parsing:** Advanced parsing techniques can analyze the structure of words to identify root words and affixes, helping to manage derivational changes.
>   - **词法解析：** 先进的解析技术可以分析单词的结构，识别词根和词缀，帮助管理派生变化。

3. **Rich Morphological Tagging:** Using detailed part-of-speech tags that include morphological information can help in accurately capturing the syntactic and semantic properties of words.
    - **丰富的形态标记：** 使用包含形态信息的详细词性标记可以帮助准确捕获单词的句法和语义属性。
4. **Contextual Embeddings:** Modern NLP models use contextual embeddings (like BERT or ELMo) that inherently capture different word forms and their meanings from large-scale text corpora, addressing both inflectional and derivational variations.
    - **上下文嵌入：** 现代 NLP 模型使用上下文嵌入（如 BERT 或 ELMo），它本质上从大规模文本语料库中捕获不同的词形及其含义，解决词形变化和派生变化。

2. Propose two techniques to attempt to deal with the issues with morphology you have identified in part(a). Describe how the techniques could be implemented and how their performance could be tested.

1. Morphological Parsing

形态解析

Implementation:

- **Parsing Tool Development:** Develop or utilize an existing morphological parsing tool that can decompose words into their base forms and affixes. This parser should be capable of recognizing both inflectional forms (like plurals, tenses) and derivational changes (like suffixes indicating part-of-speech changes).
  解析工具开发： 开发或利用现有的形态解析工具，可以将单词分解为其基本形式和词缀。 该解析器应该能够识别屈折形式（如复数、时态）和派生变化（如表示词性变化的后缀）。
- **Integration with NLP Systems:** Integrate the morphological parser into the preprocessing step of various NLP applications, such as text analysis tools, machine translation systems, or semantic parsing systems. For instance, before processing text for sentiment analysis, run the text through the morphological parser to normalize words to a form that reflects both their root meaning and morphological variations.
  与 NLP 系统集成： 将形态解析器集成到各种 NLP 应用程序的预处理步骤中，例如文本分析工具、机器翻译系统或语义解析系统。 例如，在处理文本进行情感分析之前，通过形态解析器运行文本，将单词规范化为既反映其根源含义又反映形态变化的形式。

Testing Performance:

- **Benchmark Testing:** Utilize standard linguistic corpora that include morphologically rich languages. Measure the parser's accuracy in identifying correct base forms and affixes across a variety of word types and morphological transformations.
  基准测试： 利用包含形态丰富的语言的标准语言语料库。 测量解析器在识别各种单词类型和形态转换中的正确基本形式和词缀方面的准确性。
- **Application-Specific Performance:** Integrate the parser into a specific NLP application (e.g., a machine translation system) and compare the system's performance with and without morphological parsing. Metrics such as translation accuracy, fluency (using BLEU scores), and the system's ability to handle morphological variance can be particularly revealing.
  特定于应用程序的性能： 将解析器集成到特定的 NLP 应用程序（例如机器翻译系统）中，并比较使用和不使用词法解析的系统性能。 翻译准确性、流畅性（使用 BLEU 分数）以及系统处理形态差异的能力等指标尤其能说明问题。

2. Contextual Word Embeddings

上下文词嵌入

Implementation:

- **Model Selection and Training:** Choose a contextual embedding model like BERT or ELMo, which are pretrained on large corpora and are known for capturing nuanced word uses in different contexts. Fine-tune these models on domain-specific corpora if necessary, especially if working with specialized jargon or less-common languages.
  模型选择和训练： 选择 BERT 或 ELMo 等上下文嵌入模型，这些模型在大型语料库上进行了预训练，以捕获不同上下文中细微的单词用法而闻名。 如有必要，请在特定领域的语料库上微调这些模型，特别是在使用专门术语或不太常见的语言时。
- **Embedding Integration:** Implement these embeddings in the NLP system's architecture. For example, in a text classification system, replace traditional word vectors with contextual embeddings to better capture the implications of morphological changes in input texts.
  嵌入集成： 在 NLP 系统架构中实现这些嵌入。 例如，在文本分类系统中，用上下文嵌入替换传统的词向量，以更好地捕获输入文本中形态变化的含义。

Testing Performance:

- **Contextual Understanding:** Test how well the embeddings handle words with multiple morphological forms in different contexts by setting up controlled experiments where the same word appears in different morphological forms and checking if the embeddings can maintain semantic consistency across forms.
  上下文理解： 通过设置受控实验（其中相同单词以不同形态形式出现）并检查嵌入是否可以保持跨形式的语义一致性，来测试嵌入在不同上下文中处理具有多种形态形式的单词的效果。
- **End-to-End System Evaluation:** Evaluate the overall impact of implementing these embeddings in specific applications. For instance, in sentiment analysis, assess whether the use of contextual embeddings leads to more accurate sentiment predictions in texts with high morphological diversity. Metrics could include precision, recall, F1-score, and user feedback.
  端到端系统评估： 评估在特定应用程序中实现这些嵌入的总体影响。 例如，在情感分析中，评估使用上下文嵌入是否可以在形态多样性较高的文本中实现更准确的情感预测。 指标可以包括精确度、召回率、F1 分数和用户反馈。

3. “The more often term x is used in a document, the more relevant that document becomes to a query containing x amongst its query terms.” Discuss the degree to which you agree with this statement.

I partially agree with the statement that “The more often term x is used in a document, the more relevant that document becomes to a query containing x amongst its query terms.” This approach, central to many information retrieval models, particularly those based on term frequency (TF), holds true under certain circumstances but has significant limitations and requires careful application.

我部分同意这样的说法："文档中使用术语 x 的频率越高，该文档与查询术语中包含 x 的查询就越相关。" 这种方法是许多信息检索模型的核心，特别是基于词频 (TF) 的模型，在某些情况下是正确的，但有很大的局限性，需要仔细应用。

Agreement

Term Frequency as a Relevance Indicator:

- **Basic Validity:** In principle, if a term appears frequently in a document, it suggests that the document might be more relevant to that term. For example, a document that mentions "chocolate" twenty times is likely more relevant to a query about chocolate than a document that mentions it once. This is the foundational idea behind the term frequency component of TF-IDF (Term Frequency-Inverse Document Frequency), which has been successfully used in search engines and document classification.
    - **基本有效性：** 原则上，如果某个术语在文档中频繁出现，则表明该文档可能与该术语更相关。 例如，提及"巧克力"二十次的文档可能比提及一次的文档与有关巧克力的查询更相关。 这是 TF-IDF（词频-逆文档频率）词频组件背后的基本思想，已成功应用于搜索引擎和文档分类。

Empirical Effectiveness:

- **Information Retrieval Success:** Many traditional search algorithms that prioritize term frequency have shown good performance in standard retrieval tasks, making documents with higher occurrences of a query term more accessible.
    - **信息检索成功：** 许多优先考虑术语频率的传统搜索算法在标准检索任务中表现出了良好的性能，使得查询术语出现率较高的文档更容易访问。

Disagreement

Context and Semantics:

- **Lack of Contextual Understanding:** Term frequency alone does not account for the context or the way terms are used in the document. For instance, a document could mention "apple" frequently but be discussing the fruit, not the technology company, which might be the intended query context.
    - **缺乏上下文理解：** 术语频率本身并不能解释文档中的上下文或术语使用方式。 例如，文档可能会频繁提及"苹果"，但讨论的是水果，而不是技术公司，这可能是预期的查询上下文。
- **Synonymy and Polysemy Issues:** Simple term frequency does not handle well cases where multiple words mean the same thing (synonyms) or where a word has multiple meanings (polysemy).
    - **同义词和一词多义问题：** 简单的术语频率不能很好地处理多个单词表示同一事物（同义词）或一个单词具有多种含义（一词多义）的情况。

Quality and Spam:

- **Keyword Stuffing:** High frequency of a term can sometimes result from manipulative practices such as keyword stuffing, where terms are unnaturally repeated to boost document relevance in search results.
    - **关键字堆砌：** 术语的高频率有时可能是由于诸如关键字堆砌之类的操纵行为造成的，其中术语不自然地重复以提高搜索结果中的文档相关性。
- **Irrelevance Despite High Frequency:** A document might repetitively use a term without actually providing valuable content about it.
    - **尽管高频但不相关：** 文档可能会重复使用某个术语，但实际上并未提供有关该术语的有价值的内容。

Better Alternatives:

- **Semantic Search Technologies:** Modern advancements like NLP and machine learning models (e.g., BERT, Word2Vec) provide ways to understand the meaning of text better and can determine relevance based on the semantic relationships and the document's overall topic, not just keyword frequency.
    - **语义搜索技术：** NLP 和机器学习模型（例如 BERT、Word2Vec）等现代进步提供了更好地理解文本含义的方法，并且可以根据语义关系和文档的整体主题确定相关性，而不仅仅是 关键词频率。
- **Weighting with Inverse Document Frequency:** TF-IDF improves upon simple TF by diminishing the weight of terms that occur very commonly across many documents, helping to balance term frequency with term uniqueness.
    - **使用逆文档频率进行加权：** TF-IDF 通过减少许多文档中经常出现的术语的权重来改进简单的 TF，从而有助于平衡术语频率与术语唯一性。

4. What are the advantages and disadvantages of using Mean Recriprocal Rank as an evaluation method for an Information Retrieval system? What are the risks of solely using this as an evaluation method?

Advantages of Mean Reciprocal Rank:

1. **Simplicity and Interpretability: 简单性和可解释性：**
    - MRR is straightforward to calculate and easy to understand. It focuses on the rank of the first correct answer in a list of responses, which simplifies its application and interpretation in scenarios where finding an initial correct answer is critical.
        - MRR 计算简单且易于理解。 它侧重于响应列表中第一个正确答案的排名，这简化了其在寻找初始正确答案至关重要的场景中的应用和解释。
2. **Emphasis on Top Results: 强调最佳结果：**
    - MRR inherently values the performance at the top of the result set. Since it is based on the reciprocal of the rank of the first relevant answer, it effectively emphasizes the importance of being accurate at the beginning of the search results, which aligns well with user satisfaction where early results are more likely to be noticed and used.

    - MRR 本质上重视结果集顶部的性能。 由于它基于第一个相关答案的排名的倒数，因此它有效地强调了搜索结果开始时准确的重要性，这与用户满意度非常吻合，因为早期结果更有可能被注意到和使用。
3. **Useful for Systems with One Correct Answer: 对于只有一个正确答案的系统有用：**
    - It is particularly suitable for tasks where a single correct answer is sufficient for the user, such as in question-answering systems or when a single document is needed to satisfy a query.
        - 它特别适合用户只需一个正确答案即可完成的任务，例如在问答系统中或需要单个文档来满足查询时。

Disadvantages of Mean Reciprocal Rank:

1. **Limited to the First Relevant Result: 仅限于第一个相关结果：**
    - MRR considers only the rank of the first relevant or correct answer. This limitation means it does not account for the possibility of multiple relevant documents, nor does it reflect the overall quality of all returned results beyond the first correct one.
        - MRR 仅考虑第一个相关或正确答案的排名。 这种限制意味着它没有考虑多个相关文档的可能性，也没有反映除第一个正确结果之外的所有返回结果的整体质量。
2. **Insensitive to Changes Beyond the First Hit: 对第一次点击之后的变化不敏感：**
    - If the first relevant result remains constant, any improvements or deteriorations in the ranks of subsequent relevant results do not affect the MRR. This can provide a skewed view of the system's overall effectiveness in retrieving all relevant documents.
        - 如果第一个相关结果保持不变，后续相关结果排名的任何改进或恶化都不会影响 MRR。 这可以提供系统在检索所有相关文档方面的整体有效性的倾斜视图。
3. **Not Reflective of User Experience in All Scenarios: 不能反映所有场景下的用户体验：**
    - In situations where users might benefit from multiple relevant results, MRR may not accurately reflect user satisfaction as it does not measure how many relevant results are returned or their distribution across the result set.
        - 在用户可能从多个相关结果中受益的情况下，MRR 可能无法准确反映用户满意度，因为它无法衡量返回的相关结果数量或其在结果集中的分布。

Risks of Solely Using MRR as an Evaluation Method:

1. **Overlooking Result Diversity: 忽视结果多样性：**
    - Solely using MRR might lead developers to optimize retrieval systems in a way that ensures the first result is always relevant, potentially at the cost of overall result diversity and richness. This optimization could lead to a decrease in the variety of answers or content that users are exposed to, which can be especially problematic in educational or exploratory search scenarios.
        - 单独使用 MRR 可能会导致开发人员以确保第一个结果始终相关的方式优化检索系统，但可能会牺牲整体结果的多样性和丰富性。 这种优化可能会导致用户接触到的答案或内容的多样性减少，这在教育或探索性搜索场景中尤其成问题。
2. **Ignoring Comprehensive Relevance: 忽略综合相关性：**
    - Because MRR does not account for the presence and ranking of other relevant documents beyond the first, systems evaluated solely on this metric may ignore the importance of retrieving a broader set of relevant documents. This could lead to a development focus that overly prioritizes the algorithms' ability to "guess" the first correct answer without ensuring comprehensive coverage of the topic or query.
        - 由于 MRR 不考虑第一个相关文档之外的其他相关文档的存在和排名，因此仅根据此指标评估的系统可能会忽略检索更广泛的相关文档集的重要性。 这可能会导致开发重点过度优先考虑算法“猜测”第一个正确答案的能力，而不确保主题或查询的全面覆盖。
3. **Misalignment With User Needs in Broad Queries: 广泛查询与用户需求不一致：**
    - For broad or exploratory queries where users benefit from seeing multiple viewpoints or detailed information, MRR's focus on the first result does not align well with actual user needs. Relying solely on MRR could misguide system tuning, leading to less effective user support in complex information-seeking tasks.
        - 对于广泛或探索性查询（用户可以从查看多个观点或详细信息中受益），MRR 对第一个结果的关注与实际用户需求不太相符。 仅仅依赖 MRR 可能会误导系统调整，从而导致在复杂的信息查找任务中用户支持效率较低。

Question 2

1. Give a definition of Mutual Information in the context of the text classification pipeline and explain its role.
    - **Definition of Mutual Information in Text Classification 文本分类中互信息的定义**
        - **Mutual Information (MI)** in the context of text classification is a measure of the amount of information that a particular feature (such as a word) provides about the class variable. It quantifies the statistical dependence between the feature and the class, with a higher MI indicating that knowledge of the feature's presence or absence significantly reduces uncertainty about the class.
        - 文本分类上下文中的互信息 (MI) 是对特定特征（例如单词）提供的有关类变量的信息量的度量。 它量化了特征和类别之间的统计依赖性，较高的 MI 表明对特征存在或不存在的了解可以显着降低类别的不确定性。
    - **Role in Text Classification Pipeline: 在文本分类管道中的作用：**
        - **Feature Selection:** MI is commonly used for feature selection in text classification. By evaluating the MI between each feature in the dataset and the class labels, one can identify and retain the most informative features, thereby reducing the dimensionality of the feature space and improving model performance. Features with higher MI are more relevant for classification because they have a greater impact on class determination.
        - **特征选择：** MI通常用于文本分类中的特征选择。 通过评估数据集中每个特征与类标签之间的MI，可以识别并保留信息量最大的特征，从而降低特征空间的维数并提高模型性能。 MI 较高的特征与分类更相关，因为它们对类别确定有更大的影响。
2. What makes a Naive Bayes classifier naive? Given a practical example for text classification of overcoming this naivety.

- A **Naive Bayes classifier** is considered naive because it assumes that all features (e.g., words in text classification) are conditionally independent given the class. This assumption is naive because it simplifies the computation drastically by decoupling the features, but it is often not true in practice as features can be correlated.
  - **朴素贝叶斯分类器**被认为是朴素的，因为它假设所有特征（例如，文本分类中的单词）在给定类别的情况下都是条件独立的。 这种假设很幼稚，因为它通过解耦特征极大地简化了计算，但在实践中通常并不正确，因为特征可以相互关联。

Practical Example to Overcome Naivety:
克服天真的实际例子：

- **Using Bigrams or N-grams:** Instead of using only individual words (unigrams) as features, incorporating bigrams (pairs of consecutive words) or n-grams (sequences of n consecutive words) can capture some of the relationships between words. For instance, while individual words "not" and "interesting" might not be very informative, the bigram "not interesting" captures a specific sentiment that can be very informative for sentiment analysis. This approach partially addresses the conditional independence assumption by modeling local context.
  - **使用二元组或 N 元组：** 不只使用单个单词（一元组）作为特征，合并二元组（连续单词对）或 n 元组（n 个连续单词的序列）可以捕获之间的一些关系 字。 例如，虽然单独的单词“不”和“有趣”可能提供的信息不多，但二元词“不有趣”捕获了特定的情感，这对于情感分析来说可以提供非常丰富的信息。 该方法通过对局部上下文进行建模来部分解决条件独立性假设。

3. Maximum Likelihood training in Na¨ıve Bayes for text classification is problematic. Describe the problem and give a practical solution. **Problem:**
问题：

- **Zero Frequency Problem:** When using maximum likelihood estimates for probabilities in Naive Bayes, if a feature (word) does not appear in the training data for a particular class, the estimated probability for that feature given the class becomes zero. This leads to a zero probability for the entire class when that feature is present in a new sample, skewing predictions unrealistically.
  - **零频率问题：** 当在朴素贝叶斯中使用最大似然估计概率时，如果某个特征（单词）没有出现在特定类的训练数据中，则给定该类的该特征的估计概率将变为零。 当该特征出现在新样本中时，这会导致整个类别的概率为零，从而使预测不切实际。

Practical Solution:
实际解决方案：

- **Additive Smoothing (Laplace Smoothing):** A common solution is to add a small constant (usually 1, known as Laplace smoothing) to the count of each word for each class before dividing by the total number of words in the class. This technique adjusts the maximum likelihood estimates to be more robust to unseen words, preventing any class probability from becoming zero due to the presence of a previously unseen feature in the test data.
  - **加法平滑（拉普拉斯平滑）：** 常见的解决方案是在除以该类中的单词总数之前，向每个类的每个单词的计数添加一个小常数（通常为 1，称为拉普拉斯平滑） 。 该技术调整最大似然估计，使其对未见过的单词更加稳健，从而防止由于测试数据中存在先前未见过的特征而导致任何类别概率变为零。

4. Assume the following likelihoods from Table 1 (on the following page) for each word being part of a positive or negative book review and equal prior probabilities for each class.
Show, with your workings, what class Na¨ıve Bayes will assign to the sentence “Its redeeming strength is authenticity.”

Word	Positive	Negative
Its	0.01	0.2
redeeming	0.1	0.001
strength	0.1	0.01
weakness	0.003	0.3
is	0.05	0.005
authenticity	0.1	0.01

Given the sentence "Its redeeming strength is authenticity," we need to calculate the posterior probabilities for each class (Positive, Negative) and choose the class with the highest probability.

Likelihoods for Words:

- Its: Positive = 0.01, Negative = 0.2
- redeeming: Positive = 0.1, Negative = 0.001
- strength: Positive = 0.1, Negative = 0.01
- is: Positive = 0.05, Negative = 0.005
- authenticity: Positive = 0.1, Negative = 0.01

Calculate Posterior for Each Class (assuming equal priors, say P(Positive) = P(Negative) = 0.5):

Positive:

$$P(Positive|\text{sentence}) \propto P(Positive) \times P(\text{Its}|\text{Positive}) \times P(\text{redeeming}|\text{Positive}) \times P(\text{strength}|\text{Positive}) \times P(\text{is}|\text{Positive}) \times P(\text{authent}$$

$$= 0.5 \times 0.01 \times 0.1 \times 0.1 \times 0.05 \times 0.1 = 0.0000025$$

Negative:

$$P(Negative|\text{sentence}) \propto P(Negative) \times P(\text{Its}|\text{Negative}) \times P(\text{redeeming}|\text{Negative}) \times P(\text{strength}|\text{Negative}) \times P(\text{is}|\text{Negative}) \times P(\text{aut}$$

$$= 0.5 \times 0.2 \times 0.001 \times 0.01 \times 0.005 \times 0.01 = 0.00000005$$

Decision:

- Since (0.0000025 > 0.00000005), the Naive Bayes classifier assigns the class **Positive** to the sentence "Its redeeming strength is authenticity."

Question 3

Consider the following "document", where Word1 to Word6 are the words in the vocabulary and each line(enclosed by brackets) represents a sentence:
[Word3 Word3 Word6 Word6 Word2]
[Word1 Word2 Word3 Word5 Word2 Word4]
[Word4 Word6 Word6 Word4 Word6]
[Word2 Word6 Word5 Word1]
[Word3 Word6 Word3 Word2 Word1]

1. Given the above "document", obtain a (distributional) semantic representation of each word, based on co-occurrence. Write down the full co-occurrence matrix for the document, following the rules:

    - Two words co-occur if they appear in the same sentence
    - Ignore multiple co-occurrences of the same word pair within the same sentence
    - Co-occurrences of the same word pairs in different sentences count separately

    To create the co-occurrence matrix, we follow the rule that two words co-occur if they appear in the same sentence, ignoring multiple occurrences of the same pair within a single sentence but counting each sentence separately.

    Step-by-Step Calculation:

    - **Sentence 1:** [Word3, Word3, Word6, Word6, Word2]
        - Unique words: Word3, Word6, Word2
        - Co-occurrences: Word3 with Word6, Word2; Word6 with Word2
    - **Sentence 2:** [Word1, Word2, Word3, Word5, Word2, Word4]
        - Unique words: Word1, Word2, Word3, Word5, Word4
        - Co-occurrences: Word1 with Word2, Word3, Word5, Word4; Word2 with Word3, Word5, Word4; Word3 with Word5, Word4; Word5 with Word4
    - **Sentence 3:** [Word4, Word6, Word6, Word4, Word6]
        - Unique words: Word4, Word6
        - Co-occurrences: Word4 with Word6
    - **Sentence 4:** [Word2, Word6, Word5, Word1]
        - Unique words: Word2, Word6, Word5, Word1
        - Co-occurrences: Word2 with Word6, Word5, Word1; Word6 with Word5, Word1; Word5 with Word1
    - **Sentence 5:** [Word3, Word6, Word3, Word2, Word1]
        - Unique words: Word3, Word6, Word2, Word1
        - Co-occurrences: Word3 with Word6, Word2, Word1; Word6 with Word2, Word1; Word2 with Word1

    Co-occurrence Matrix:

	Word1	**Word2**	**Word3**	**Word4**	**Word5**	**Word6**
Word1	0	3	2	1	2	2
Word2	3	0	3	2	2	4
Word3	2	3	0	1	1	3
Word4	1	2	1	0	1	2
Word5	2	2	1	1	0	2
Word6	2	4	3	2	2	0

2. Calculate the cosine similarity between Word1 and each other word in the corpus. Which word is most similar to Word1?

    To find the cosine similarity between Word1 and the other words, use the formula:

    $$\text{Cosine similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

    Calculations:

    - **Vector for Word1**: [0, 3, 2, 1, 2, 2]
    - **Vectors for Other Words**:
        - Word2: [3, 0, 3, 2, 2, 4]
        - Word3: [2, 3, 0, 1, 1, 3]
        - Word4: [1, 2, 1, 0, 1, 2]
        - Word5: [2, 2, 1, 1, 0, 2]
        - Word6: [2, 4, 3, 2, 2, 0]

    Calculate cosine similarities manually or using a software tool.

3. Are there other ways to measure similarity in distributional space? Can you explain how they would work? With reference to the methods you have presented, outline the key differences between the method and cosine similarity?

Euclidean Distance:

- **Method**: Measures similarity based on the straight line distance between points in a vector space.

    - **方法**：根据向量空间中点之间的直线距离来测量相似度。
- **Difference**: It is sensitive to the magnitude of vector elements and better for numerical
    - **差异**：它对向量元素的大小敏感，并且对于数值更好

stability in high dimensional spaces, contrasting with cosine similarity's focus on orientation regardless of magnitude.

Key Differences: Euclidean distance provides different insights than cosine similarity, with Euclidean on absolute distances, while cosine assesses orientation and alignment, ideal for normalized vectors in text analysis.
主要区别：杰卡德距离和欧几里德距离都提供了与余弦相似性不同的见解，杰卡德距离侧重于二元存在，欧几里德距离侧重于绝对距离，而余弦评估方向和对齐，非常适合文本分析中的归一化向量。

Question 4

Intelligently discuss the components required to build a system for Automatic Fact-Checking. This discussion should be no more than 2 pages in length. The discussion should be structured around the NLP pipeline and how this framework could be applied to the task of Automatic Fact-Checking.

1. **Input Processing**

- **Text Acquisition**: The first step in the pipeline is acquiring the text to be checked. This could be from news articles, social media posts, speeches, or transcripts. It involves scraping or using APIs to gather large volumes of text data from diverse sources to ensure a comprehensive analysis.
- **Preprocessing**: This involves cleaning the text data and preparing it for analysis. Common tasks include **tokenization** (breaking the text into words or phrases), **removing stop words** (common words that don't contribute to the deeper meaning, such as "and", "the", etc.), **normalization** (converting text to a standard form, like lowercasing), and **stemming/lemmatization** (reducing words to their base or root form).

2. **Natural Language Understanding**

- **Parsing and Part-of-Speech Tagging**: Analyzing the grammatical structure of sentences and identifying the part of speech for each word. This helps in understanding the grammatical relationships and roles of different words, which is crucial for interpreting the meaning of sentences.
- **Named Entity Recognition (NER)**: Identifying and classifying key elements in text into predefined categories such as names of persons, organizations, locations, dates, etc. This is critical for fact-checking as it helps in pinpointing the subjects and objects that need verification.
- **Dependency Parsing**: Understanding the dependencies between words helps in constructing the relationships and logical structure of the sentence, which is necessary for interpreting complex statements that may contain conditional or hypothetical scenarios.

3. **Claim Detection**

- **Claim Identification**: Using models to identify sentences or phrases that contain a factual claim. This involves determining whether a statement is verifiable and factual as opposed to being an opinion, a question, or an ambiguous statement.
- **Claim Classification**: Categorizing the types of claims (e.g., numerical, categorical, existence) to tailor the subsequent verification process according to the type of fact-check required.

4. **Information Retrieval**

- **Data Sources**: Establishing reliable and authoritative sources for verification, such as trusted news databases, official records, scientific journals, and verified data repositories.
- **Search Algorithms**: Implementing sophisticated search algorithms capable of fetching information from structured and unstructured data sources. This could involve keyword matching, semantic search, or more advanced AI-driven query systems.

5. **Fact Verification**

- **Evidence Gathering**: Extracting information pertinent to the claims from various sources. This includes pulling relevant documents, statistics, previous reports, or historical data that can confirm or refute the claim.
- **Consistency Checking**: Comparing the claim against the gathered evidence using logical rules, statistical models, or even deep learning models trained to evaluate the alignment and discrepancies between the claim and the data.

6. **Presentation and Visualization**

- **Result Presentation**: Displaying the fact-checking results in a user-friendly manner, categorizing statements as true, false, or partially true/false with accompanying evidence.
- **Explanation Generation**: Providing explanations and justifications for the verdict, which includes citing sources and explaining the reasoning process that led to the conclusion.

7. **Feedback Loop**

- **User Feedback**: Incorporating user feedback mechanisms to allow end-users to question or challenge the fact-check results, which can help in refining and improving the system.
- **Continuous Learning**: Using feedback and new data to continuously train and update the models to adapt to new forms of misinformation and changing information landscapes.

Evaluation and Iteration

Regularly evaluating the system's performance using standard metrics like accuracy, precision, recall, and F1-score, and making iterative improvements based on performance metrics and user feedback. This helps ensure the system remains effective and trustworthy.

Resit

Question 1

1. What is a Markov Chain? Illustrate your answer with an example with respect to Natural Language Processing A **Markov Chain** is a statistical model that describes a sequence of possible events in which the probability of each event depends only on the state attained in the previous event. This property is known as the **Markov property**.

   In NLP, Markov Chains are used to model language in a probabilistic manner. They can predict the next word in a sentence given the current word, effectively handling sequences in text data. One of the most straightforward applications of Markov Chains in NLP is in the development of simple text generators, which can generate sentences that mimic a particular style or corpus.

   Example: Generating Text Using Markov Chains

   Suppose you have a small corpus of text: **"Ask not what your country can do for you ask what you can do for your country."**

   1. **Tokenization**: First, split the text into tokens (words in this case).
   2. **State Transition Construction**: Next, construct a state transition table that counts how often each word follows another. For simplicity, consider each word as a state.
   3. **Text Generation**: To generate new text, start with a seed word (e.g., "ask") and then:
      - Use the transition probabilities to randomly select the next word (either "not" or "what" following "ask").
      - Continue selecting the next word based on the current word's transitions until a stopping condition is met (e.g., a maximum number of words or a punctuation mark that signifies the end of a sentence).

2. Propose an area of Natural Language Processing that makes use of Hidden Markov Models. Ensure you fully describe, with appropriate definitions, how Hidden Markov Models are applicable to that area.

3. One of the assumptions made with Hidden Markov Models is the 'bigram' assumption. If this was changed to a 'unigram' assumption, how would this affect the performance of the model? You may want to refer to your previous answer for part (b) to help exemplify this.

4. Suppose the transition and emission probabilities are as exemplified in the following tables:

a_{ij}	**s1 = verb**	**s2 = noun**	**s3 = adjective**	s_f **= FINISH**
s0 = START	a_{01} = 0.6	a_{02} = 0.4	a_{03} = 0.0	a_{0x} = 0.0
s1 = verb	a_{11} = 0.01	a_{12} = 0.84	a_{13} = 0.0	a_{1x} = 0.15
s2 = noun	a_{21} = 0.6	a_{22} = 0.1	a_{23} = 0.0	a_{2x} = 0.3
s3 = adjective	a_{31} = 0.0	a_{32} = 0.79	a_{33} = 0.01	a_{3x} = 0.2

Table 1: Transition Probability Matrix A

b_{it}	**fly**
s1 = verb	b_1(fly) = 0.8
s2 = noun	b_2(fly) = 0.7
s3 = adjective	b_3(fly) = 0.4

Table 2: Emission Probability Matrix B

Using the Viterbi algorithm, find the most probable tag sequence that generated the observation sequence "Fly!" and estimate its probability. Show your working.

Question 2

1. What makes Logistic Regression a discriminative classifier for NLP tasks? Compare this to another type of model used by supervised machine learning classifiers in your answer. Logistic Regression is a statistical method that models the probability of a binary outcome based on one or more predictor variables. It is termed a **discriminative classifier** because it directly models the decision boundary between classes. In other words, Logistic Regression doesn't aim to model the underlying probability distributions of each class (as generative models do). instead, it models the probability that a given input belongs to a particular class.

   Comparison with Naive Bayes (A Generative Model)

   1. **Modeling Approach**:
      - **Naive Bayes** models the joint distribution of the feature X and target Y and then uses Bayes' theorem to compute the conditional probability $P(Y|X)$. It assumes that all features are independent given the class label, which simplifies computations but can be unrealistic (hence "naive").

   - **Logistic Regression**, on the other hand, directly models the conditional probability $P(Y|X)$ without making assumptions about the independence of features. It directly learns a decision boundary from the features.

   1. **Performance in Different Scenarios**:
      - **Naive Bayes** can perform better when the independence assumption holds true or when data for each category are well-represented and abundant. It also tends to perform well with smaller datasets and can handle multiple classes naturally.

- **Logistic Regression** may outperform Naive Bayes when the relationship between features is important and dependencies exist. Logistic Regression can also be extended to non-linear boundaries using polynomial or complex feature transformations.

2. **Interpretability**:
    - Both models offer good interpretability. Naive Bayes clearly shows how prior probabilities and feature likelihoods contribute to posterior probabilities. Logistic Regression coefficients can be interpreted as the strength of the relationship between features and the probability of belonging to a class, adjusting for other features.
3. **Probabilistic Output**:
    - Both models provide probabilities as output. In Logistic Regression, the probabilities are modeled directly. In Naive Bayes, probabilities are derived using Bayes' theorem from the generative model of the data.

2. Design and give a rationale for your choice of 5 features for a Logistic Regression classifier that would be used for authorship attribution.

   Authorship attribution is the process of identifying the author of a text based on the writing style. Logistic Regression can be used effectively in this domain by leveraging various stylistic and linguistic features that capture unique aspects of an author's writing style. Here are five features that can be especially powerful for such a classifier, along with rationales for their selection:

   1. **Average Sentence Length**
      - **Feature Description**: The average number of words in sentences across a text.
      - **Rationale**: Different authors tend to have distinct preferences for sentence complexity and length. Some may use longer, more complex sentences, while others prefer shorter, clearer sentences. This feature helps capture that stylistic signature.
   2. **Lexical Diversity**
      - **Feature Description**: The ratio of unique words to the total number of words used in the text.
      - **Rationale**: Lexical diversity measures the range of different words an author uses. High lexical diversity may indicate a rich vocabulary or a formal writing style, while a lower score could suggest repetitive or simplified writing. This feature is useful in distinguishing between authors with different vocabulary usages.
   3. **Use of Function Words**
      - **Feature Description**: Frequency of function words (e.g., "the", "and", "of", "to"). This could be represented as a vector of frequencies for a curated list of function words.
   - **Rationale**: Function words are often used subconsciously and are less influenced by the topic of the text, making them good indicators of writing style. Different authors demonstrate consistent patterns in their use of function words, which can serve as reliable stylistic fingerprints.
   4. **Average Word Length**
      - **Feature Description**: The average length of words in a text.
      - **Rationale**: This feature provides insight into an author's preference for word complexity. Authors writing on technical or specific subjects might use longer words, while those focusing on broader audiences might use shorter, more common words. This metric can help differentiate authors based on the complexity of the vocabulary used.
   5. **Punctuation Usage**
      - **Feature Description**: Frequency and types of punctuation used (e.g., commas, semicolons, question marks).
      - **Rationale**: Punctuation is a crucial part of writing that affects readability and style. Some authors might use many commas and semicolons to create complex sentence structures, while others might use fewer punctuation marks. This feature helps to capture those unique stylistic choices.

3. For a given input to a binary Logistic Regression classifier for Sentiment Classification, the feature vector has the following values: [3, 2, 1, 3, 0, 4.19], the weights are [−5.0, 2.5, −1.2, 0.5, 2.0, 0.7], there is a bias of 0.1 and the class should be '1' if P (y = 1 | x) > 0.5, else it is '0'. By showing your calculations give the class that the input will be categorized as.

   The logistic regression model calculates the probability that an instance $\mathbf{x}$ belongs to the positive class $y = 1$ as follows:

   $$P(y = 1|\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

   where:

   - $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic (sigmoid) function.
   - $\mathbf{w} \cdot \mathbf{x}$ is the dot product of the weights and the feature vector.
   - b is the bias.

   Given:

   - Feature vector $\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$
   - Weights $\mathbf{w} = [-5.0, 2.5, -1.2, 0.5, 2.0, 0.7]$
   - Bias $b = 0.1$

   Dot Product Calculation

   $$\mathbf{w} \cdot \mathbf{x} = (-5.0 \times 3) + (2.5 \times 2) + (-1.2 \times 1) + (0.5 \times 3) + (2.0 \times 0) + (0.7 \times 4.19)$$

   $$= -15 + 5 - 1.2 + 1.5 + 0 + 2.933$$

   $$= -15 + 5 - 1.2 + 1.5 + 2.933$$

   $$= -8.767$$

Logistic Function Evaluation
Now, compute the logistic function:

$$z = \mathbf{w} \cdot \mathbf{x} + b = -8.767 + 0.1 = -8.667$$

$$P(y = 1|\mathbf{x}) = \sigma(-8.667) = \frac{1}{1 + e^{-(-8.667)}}$$

$$= \frac{1}{1 + e^{8.667}}$$

Sigmoid Calculation

4. The authorship classifier above has an accuracy of 0.95. Discuss the extent to which this is an appropriate method of classifier evaluation for authorship attribution. Ensure you include a comparison to other metrics in your discussion.

Question 3

Consider the following sentence:
Julie loves the exciting section of The New York Times.

1. Given the above sentence, identify all tokens.
    - Julie
    - loves
    - the
    - exciting
    - section
    - of
    - The
    - New
    - York
    - Times
2. Given the above sentence, assign the correct part-of-speech tag to each token.
    - Julie - NNP (Proper Noun, Singular)
    - loves - VBZ (Verb, 3rd person singular present)
    - the - DT (Determiner)
    - exciting - JJ (Adjective)
    - section - NN (Noun, Singular)
    - of - IN (Preposition)
    - The - DT (Determiner)
    - New - NNP (Proper Noun, Singular)
    - York - NNP (Proper Noun, Singular)
    - Times - NNP (Proper Noun, Singular)
3. Describe three different ways of representing syntactic structure? What are the advantages and disadvantages of each of them?
    1. **Phrase Structure Trees (Constituency Parsing)**
        - **Advantages**: Visual and hierarchical representation of sentence structure that shows how words group into nested constituents. Useful for understanding the sentence structure and grammatical relationships.
    - **Disadvantages**: Can become complex and large for long sentences. It does not explicitly represent dependencies between words that are far apart in the tree.
    2. **Dependency Grammar**
        - **Advantages**: Focuses on the dependencies and semantic relationships between words, showing which words depend on others. It is less structured and more flexible, often simpler and more useful for semantic analysis and natural language understanding tasks.
    - **Disadvantages**: Does not reveal the hierarchical structure of a sentence as explicitly as phrase structure trees, which can be important in some syntactic analyses.
    3. **Feature-Based Grammar**
        - **Advantages**: Includes detailed linguistic features in the grammar, such as gender, number, case, etc. Allows for very sophisticated parsing strategies that can handle complex syntactic phenomena.
    - **Disadvantages**: Can be very complex to implement and computationally expensive. Also requires detailed linguistic knowledge to construct.
4. Can the following Context Free Grammar parse the above sentence? N.B the grammar is missing the lexicon part, you should have that if you did part of speech tagging. Punctuation is not part of the grammar. If the grammar can parse the sentence, then draw the parse tree. Otherwise, suggest a grammar that can parse the sentence.

$$S \rightarrow NP|VP$$
$$VP \rightarrow V|VNP|VPP|VPPNP$$
$$PP \rightarrow PNP$$
$$NP \rightarrow N|DetN|AdjN|DetAdjN$$

Question 4

Toxicity Detection on Social Media is the task of detecting hate speech, abuse and other toxic language.
Intelligently discuss the components required to build a system for Toxicity Detection on Social Media. This discussion should be no more than 2 pages in length. The discussion should be structured around the NLP pipeline and how this framework could be applied to the task of Toxicity Detection on Social Media.

Overview of NLP Pipeline for Toxicity Detection

An NLP pipeline for detecting toxicity in social media content generally follows these steps: data collection, preprocessing, feature extraction, model training, evaluation, and deployment. Each component must be carefully designed to handle the specifics of social media text, which often includes informal language, slang, misspellings, and emojis.

1. **Data Collection**
    - **Source Identification**: Collect data from various social media platforms like Twitter, Facebook, and Reddit, considering the platform-specific language and interactions.
    - **Data Diversity**: Ensure the dataset reflects diverse demographics to avoid bias in toxicity detection. Include various languages, dialects, and cultural contexts where possible.
2. **Preprocessing**
    - **Text Normalization**: Convert text to a uniform format, dealing with case, encoding, and removing URLs and user mentions, which are common in social media texts.
    - **Noise Removal**: Social media texts often contain hashtags, emojis, and unconventional punctuation which need to be either removed or encoded depending on their relevance to the task.
    - **Handling Slang and Abbreviations**: Use specialized dictionaries or translation techniques to handle slang and abbreviations prevalent in social media.
3. **Feature Extraction**
    - **Lexical Features**: Extract features such as n-grams, word frequencies, and presence of specific toxic keywords or phrases.
    - **Syntactic Features**: Parse trees and part-of-speech tags can help understand the structure of sentences, which might be useful for distinguishing harmful sentences from non-harmful ones.
    - **Semantic Features**: Embeddings from models like Word2Vec, GloVe, or BERT can capture the context around certain words, helping to identify subtle forms of toxicity that depend on context rather than explicit language.
4. **Model Training**
    - **Choosing a Model**: Start with baseline models like logistic regression or SVM for initial benchmarks. Gradually move to more complex models such as neural networks (CNNs, RNNs) or transformers if the task benefits from deeper semantic analysis.
    - **Handling Imbalanced Data**: Toxic comments are typically much fewer than non-toxic ones. Techniques such as oversampling, undersampling, or anomaly detection methods should be considered.
5. **Evaluation**
    - **Metrics**: Accuracy, Precision, Recall, and F1-score are standard metrics. Given the high cost of false negatives (failing to detect toxic comments), recall might be more emphasized.
    - **Validation**: Use cross-validation and external validation sets to ensure the model generalizes well across different types of social media content and demographics.
6. **Deployment and Monitoring**
    - **Real-time Processing**: Deploy models that can operate in real-time or near-real-time, given the rapid pace of social media.
    - **Continuous Learning**: Implement feedback loops where the model can be updated with new data, adapting to evolving language use and new forms of toxicity.
    - **Human in the Loop**: Integrate human review to handle ambiguous cases and to provide overrides, which can also serve as new training data for the model.
7. **Ethical Considerations**
    - **Bias and Fairness**: Monitor and mitigate any biases in the model, ensuring it does not disproportionately flag content from certain groups.
    - **Transparency**: Maintain transparency about how content is moderated, offering explanations for decisions made by AI systems.