1.(i)

i. 点积注意力 (Dot Product Attention) vs. 乘性注意力 (Multiplicative Attention)

点积：$\mathbf{e}_{t,i} = \mathbf{s}_t^T \mathbf{h}_i$ 乘性：$\mathbf{e}_{t,i} = \mathbf{s}_t^T \mathbf{W} \mathbf{h}_i$

- **Advantage：计算效率更高，空间占用更小。**
  - 点积注意力不需要学习额外的权重矩阵 $\mathbf{W}$。这意味着它在计算上更快（少了一次矩阵乘法），并且不需要存储额外的参数，节省了显存空间。
- **Disadvantage：受限于隐藏层维度必须相同。**
  - 点积运算要求查询向量 $\mathbf{s}_t$ 和键向量 $\mathbf{h}_i$ 的维度完全一致（即必须都是 d 维向量）。如果Encoder和Decoder的隐藏层大小不同，就无法直接使用点积注意力；而乘性注意力可以通过矩阵 $\mathbf{W}$（维度为 $d_s \times d_h$）来做线性映射，从而处理不同维度的输入。

ii. 加性注意力 (Additive Attention) vs. 乘性注意力 (Multiplicative Attention)

加性：$\mathbf{e}_{t,i} = \mathbf{v}^T \tanh(\mathbf{W}_1 \mathbf{h}_i + \mathbf{W}_2 \mathbf{s}_t)$ 乘性：$\mathbf{e}_{t,i} = \mathbf{s}_t^T \mathbf{W} \mathbf{h}_i$

- **Advantage：在高维空间下表现通常更稳健/更好。**
  - 根据 Vaswani et al. (2017) 的研究，当隐藏层的维度 (d) 非常大时，点积或乘性注意力的数值会变得很大，导致 Softmax 函数进入梯度极小的区域（梯度消失），从而影响训练。虽然乘性注意力可以通过除以 $\sqrt{d}$ (Scaled Dot-Product) 来缓解，但在不缩放的原始定义下，加性注意力因为有 $\tanh$ 非线性激活函数的存在，在高维情形下通常表现出更好的性能和稳定性。
- **Disadvantage：计算效率较低，速度慢。**
  - 乘性注意力本质上是高度优化的矩阵乘法，在现代 GPU 上非常快。
  - 而加性注意力涉及更复杂的操作序列：两个独立的线性变换、加法、$\tanh$ 激活函数，最后再跟一个向量点积。这些操作比单纯的矩阵乘法更费时，且更难在硬件上进行极致的并行优化。

2.(a)

1. **捕捉局部上下文与 N-gram 特征 (Capturing Local Context / N-grams)**

中文的一个显著特点是：单个汉字往往只是一个语素（morpheme），而完整的词义通常由两个或多个汉字组合而成。

- 例子：正如Hint所说，“电” (Electricity) 和 “脑” (Brain) 分开看是两个概念，但组合在一起 “电脑” 才表示 “Computer”。
- Conv1d 的作用：1D 卷积层通过在序列上滑动一个窗口（Kernel Size > 1），可以将**相邻**的 token（在这里是字/字符）的 Embedding 结合在一起。
- 结果：这相当于让模型自动学习到了 **Bigram（双字词）** 或 **Trigram（三字词）** 的特征。它能在进入 LSTM 编码器之前，就把“电”和“脑”的特征融合，形成一个代表“Computer”的高级语义特征。

2. **弥补分词粒度过细的问题 (Mitigating Granularity Issues)**

由于使用的是 sentencepiece 或者是基于字符的 Tokenizer，输入的粒度可能太细了（Character-level）。

- BiLSTM 虽然能处理序列，但在第一层直接面对离散的字符时，可能需要更长的步数才能整合出词义。
- 加入 Conv1d 后，它充当了一个**“特征提取器”**或**“软分词器”**，将字级别的 Embedding 预处理成了更接近词级别的表示，减轻了后续 Encoder 的负担。

2.(b)

i. 单复数错误 (Singular/Plural Error)

- 1. Identify the error:

  NMT 译文使用了单数形式 "the culprit was"，而参考译文使用的是复数形式 "the culprits were"。

- 2. Possible reason(s):
  - **语言学原因 (Linguistic Construct):** 中文里的名词通常没有明显的单复数变形（不像英语加 `-s`）。虽然原文中有明确的复数标记 **“们”**，但这个字在中文语料中可能不如直接通过上下文判断复数那么常见，或者在训练数据中 “罪犯” 更多以单数含义出现。
  - **模型限制 (Model Limitation):** 模型可能没有给予字符 “们” 足够的**注意力 (Attention weight)**，或者 Word Embedding 没有学好 “名词+们” 对应英语复数的规律，导致模型倾向于输出概率更高的单数形式（Bias towards singular）。
- 3. Possible fix:
  - **Data Augmentation:** 增加包含 “们” 字及其对应英语复数形式的训练数据，强迫模型学习这个显式的复数映射。
  - **Character-level features / CNN:** 如题目上一问所述，引入字符级 CNN 可以更好地捕捉词缀（Suffix）信息，帮助模型识别 “们” 这个字所代表的语法功能。

ii. 重复与漏译 (Repetition and Omission / Coverage Issue)

- 1. Identify the error:

  NMT 译文出现了重复翻译 (Repetition)，连续输出了两次 "resources have been exhausted"，同时完全漏译 (Omission) 了前半句 “几乎已经没有地方容纳这些人” (almost no space to accommodate these people)。

- 2. Possible reason(s):
  - **注意力覆盖问题 (Attention Coverage Issue):** 这是 NMT（特别是 RNN 架构）最常见的问题之一。模型在生成译文时，注意力机制可能“忘记”了它已经翻译过 “资源已经用尽” 这部分内容，导致在这个时间步再次将注意力集中在相同的源端词语上，从而产生重复。同时，由于注意力被吸引到了句子末尾，前半句的信息被忽略了。
- 3. Possible fix:
  - **Coverage Mechanism (覆盖机制):** 在注意力机制中加入 **Coverage Penalty (覆盖惩罚)** 或维护一个 **Coverage Vector**。这会记录每个源端单词已经被“关注”了多少次。如果一个词已经被关注过了，模型在后续步骤中再关注它时会受到惩罚，从而强制模型去关注那些还没被翻译（Attention 权重低）的部分。

iii. 命名实体/罕见词错误 (Named Entity / Rare Word Error)

- 1. Identify the error:

  NMT 未能正确翻译专有名词/特定术语 “国殇日” (National Mourning Day)，而是将其错误地泛化翻译为毫无意义的 "today's day"。

- 2. Possible reason(s):
  - **Out-Of-Vocabulary (OOV) / Rare Word:** “国殇日” 在训练数据中可能非常罕见，甚至从未出现过。模型没有学到它的特定 Embedding，只能根据它认识的字——比如 “日” (Day)——进行猜测，或者直接生成了一个通用的占位符翻译。
- 3. Possible fix:
  - **Subword Units (BPE/SentencePiece):** 改进分词方式，使用更细粒度的Subword。这可能让模型将 “国殇日” 拆分为它认识的更小单元（如 “National” + “Mourning” + “Day” 的对应词根）进行组合翻译。
  - **Copy Mechanism (Pointer Network):** 引入拷贝机制，允许模型在遇到生僻词时直接查字典映射，或者如果是人名地名则直接拷贝源词。
  - **Add Dictionary:** 最简单的修复是增加一个外部的术语词典，如果在推断时遇到“国殇日”，强制替换为 "National Mourning Day"。

iv. 习语/方言误译 (Idiom / Dialect Mistranslation)

- 1. Identify the error:

  NMT 将俗语 “唔做唔错” (Cantonese/Dialect for "Do not do, do not mistake") 错误地翻译成了 "it's not wrong"。且误解了整个句子的因果逻辑。
- 2. Possible reason(s):
  - **Training Data Mismatch (Domain Issue):** 这句话包含明显的方言/文言用法。“唔” 是粤语中常见的否定词（不），而不是标准普通话。模型是基于标准普通话训练的，它可能将 “唔” 误认为是无意义字符，或者看到 “错” 字就联想到了 “不错/Wrong”，从而输出了 "It's not wrong"。模型试图按字面意思去组合（Compositional），但失败了，因为习语通常需要整体记忆。
- 3. Possible fix:
  - **Idiom/Phrase Table (短语表):** 建立一个常用习语或俗语的映射表。在翻译前先进行匹配，如果是固定搭配（Idiom），直接用预定义的正确英语短语替换，而不让模型自己生成。
  - **Transfer Learning / Fine-tuning:** 在包含更多方言、口语或文学作品的数据集上对模型进行Fine-tune，让模型见过这种非标准的表达方式。

2.(c)

i. 计算两个 Reference 下的 BLEU 分数

我们需要计算 c_1 和 c_2 相对于两个参考译文 r_1 和 r_2 的 BLEU 分数。

- 权重: $\lambda_1 = 0.5, \lambda_2 = 0.5$ (只算 1-gram 和 2-gram)。
- 公式: $BLEU = BP \times \exp(0.5 \ln p_1 + 0.5 \ln p_2) = BP \times \sqrt{p_1 \times p_2}$

数据准备

- r_1 **(11词)**: *resources have to be sufficient and they have to be predictable*
- r_2 **(6词)**: *adequate and predictable resources are required*
- c_1 **(9词)**: *there is a need for adequate and predictable resources*
- c_2 **(6词)**: *resources be sufficient and predictable to*

Candidate 1 (c_1) 的计算

1. **长度与 BP**:
   - $len(r)$ (最佳匹配长度): 最接近的长度是 11，所以 $len(r) = 11$。
   - 由于 $len(c) < len(r)$，需要惩罚：

     $BP = \exp(1 - \frac{11}{9}) = \exp(-0.222) \approx \mathbf{0.801}$
2. **精度 (p_n)**:
   - p_1 (1-gram):
     - c_1 中的词：*there, is, a, need, for, adequate, and, predictable, resources* (共9个)
     - 匹配词：*adequate* (r_2), *and* (r_1, r_2), *predictable* (r_1, r_2), *resources* (r_1, r_2)。共 4 个匹配。
     - $p_1 = 4/9 \approx \mathbf{0.444}$
   - p_2 (2-gram):
     - c_1 Bigrams: *there is, is a, a need, need for, for adequate, adequate and, and predictable, predictable resources* (共8个)
     - 匹配 Bigrams: *adequate and* (r_2), *and predictable* (r_2), *predictable resources* (r_2)。共 3 个匹配。
     - $p_2 = 3/8 = \mathbf{0.375}$
3. 最终得分:

   $BLEU_{c1} = 0.801 \times \sqrt{0.444 \times 0.375} \approx 0.801 \times 0.408 \approx \mathbf{0.327}$

Candidate 2 (c_2) 的计算

1. **长度与 BP**:
   - $len(r)$: 最接近的长度是 6 (r_2)。所以 $len(r) = 6$。
   - 由于 $len(c) \geq len(r)$，无惩罚：

     $BP = \mathbf{1.0}$
2. **精度 (p_n)**:
   - p_1 (1-gram):
     - c_2 中的词：*resources, be, sufficient, and, predictable, to* (共6个)
     - 匹配词：全部在 r_1 中都有出现。共 6 个匹配。
     - $p_1 = 6/6 = \mathbf{1.0}$
   - p_2 (2-gram):
     - c_2 Bigrams: *resources be, be sufficient, sufficient and, and predictable, predictable to* (共5个)
     - 匹配 Bigrams: *be sufficient* (r_1), *sufficient and* (r_1), *and predictable* (r_2)。共 3 个匹配。
     - 注意：*predictable to* 不匹配，因为 r_1 是 *predictable* 结尾，r_2 是 *required*。
     - $p_2 = 3/5 = \mathbf{0.6}$
3. 最终得分:

   $BLEU_{c2} = 1.0 \times \sqrt{1.0 \times 0.6} \approx \mathbf{0.775}$

比较与结论 (i)

- **Better Translation according to BLEU:** c_2 (0.775 > 0.327)
- **Do you agree? Disagree.**
    - c_1 虽然有额外的词 ("there is a need for") 导致 p_1, p_2 较低且受到 BP 惩罚，但它是一个语法通顺且语义正确的句子。
    - c_2 虽然拼凑了很多正确的词（高 p_1）且长度合适（无 BP 惩罚），但它是语法混乱的 ("resources be... to")。这展示了 BLEU 在衡量语法连贯性上的缺陷。

ii. 只基于 r_2 计算 BLEU 分数

现在只参考 r_2: *adequate and predictable resources are required*(6词)。

Candidate 1 (c_1) 的新分数

- **长度**: $len(c) = 9, len(r) = 6$。$c > r$，所以 $BP = 1.0$。
- p_1: 匹配词 *adequate, and, predictable, resources* (4个)。$p_1 = 4/9 \approx 0.444$。
- p_2: 匹配短语 *adequate and, and predictable, predictable resources* (3个)。$p_2 = 3/8 = 0.375$。
- **BLEU**: $1.0 \times \sqrt{0.444 \times 0.375} \approx \mathbf{0.408}$

Candidate 2 (c_2) 的新分数

- **长度**: $len(c) = 6, len(r) = 6$。$BP = 1.0$。
- p_1: 匹配词 *resources, and, predictable* (3个)。("be", "sufficient", "to" 不在 r_2 中)。$p_1 = 3/6 = 0.5$。
- p_2: 匹配短语 *and predictable* (1个)。$p_2 = 1/5 = 0.2$。
- **BLEU**: $1.0 \times \sqrt{0.5 \times 0.2} = \sqrt{0.1} \approx \mathbf{0.316}$

比较与结论 (ii)

- **Higher BLEU score:** c_1 (0.408 > 0.316)
- **Do you agree? Yes.** 在移除 r_1 后，c_2 因为使用了 "sufficient"（r_2 中没有，只在 r_1 中有）而分数大跌。

iii. 单一 Reference 的问题

- **Problem:** 使用单一参考译文的主要问题是**语言的多样性 (Linguistic Diversity)**。同一句话可以用多种不同的词汇（同义词）和句式结构正确表达。如果 NMT 生成了一个意思完全正确但用词或结构与这唯一的参考译文不同的句子，BLEU 会给予它很低的评分（如 (ii) 中的 c_2 此时因为用了 sufficient 而不是 adequate 被惩罚）。
- **Assessment Impact:**
    - **Multiple References:** BLEU 允许 NMT 译文匹配**任意一个**参考译文中的 N-gram。这增加了捕捉到正确同义词或短语的可能性，从而使评分更鲁棒。
    - **Single Reference:** NMT 必须精准“猜”中那唯一一个译员的用词习惯，这导致 BLEU 无法区分“错误的翻译”和“正确但不同措辞的翻译”。

iv. BLEU vs. Human Evaluation

Advantages:

1. **Fast & Automated:** 计算非常快且成本低，不需要像人工评估那样耗时耗力，适合在训练过程中频繁运行。
2. **Language Independent:** 不需要任何特定语言的语法知识或词典，只要有 token 后的文本就能算，通用性强。

Disadvantages:

1. **Ignores Semantics (Meaning):** 不懂语义和同义词。如果 NMT 用了 *glad* 而参考译文是 *happy*，BLEU 会认为这是错的。
2. **Insensitive to Sentence Structure:** 它是基于局部 N-gram 匹配的"词袋"变体，对全局语法结构不敏感。如题目 (i) 中所示，一堆乱序但局部匹配的词 (c_2) 可能比通顺的句子 (c_1) 得分更高。

2(d)

i. 训练过程中翻译质量的变化

The translation quality improve over the training iterations for the model:

`000200.json`:


```
"hypothesis": [
                "_it",
                "_is",
                "_also",
                "_to",
                "_be",
                "_to",
                "_be",
                "_to",
                "_be",
                "_to",
                "_be",
                "."
            ],
            "score": -29.973121643066406
```


`007000.json`:


```
"hypothesis": [
                "_i",
                "_also",
                "_clarified",
                "_the",
                "_number",
                "_of",
                "_matters",
                "_made",
                "_at",
                "_this",
```

```
                "_conference",
                "."
            ],
            "score": -7.827193260192871
```

`013800.json`:

```
"hypothesis": [
                "_i",
                "_have",
                "_also",
                "_clarified",
                "_a",
                "_number",
                "_of",
                "_matters",
                "_raised",
                "_at",
                "_this",
                "_meeting",
                "."
            ],
            "score": -7.726564407348633
```

ii. Beam Search中的选项间差异

通常 Beam Search 的结果大同小异，分数低的可能包含更多语法错误或漏词:

hypothesis#3, #6, #9 in `013800.json`:

```
{
    "hypothesis": [
        "_i",
        "_have",
        "_also",
        "_clarified",
        "_a",
        "_number",
        "_of",
        "_matters",
        "_that",
        "_have",
        "_been",
        "_made",
        "_at",
        "_this",
        "_conference",
        "."
    ],
    "score": -8.216902732849121
},
{
```

```
    "hypothesis": [
        "_i",
        "_would",
        "_also",
        "_like",
        "_to",
        "_clarify",
        "_a",
        "_number",
        "_of",
        "_matters",
        "_raised",
        "_by",
        "_this",
        "_meeting",
        "."
    ],
    "score": -8.93420696258545
},
{
    "hypothesis": [
        "_i",
        "_have",
        "_also",
        "_clarified",
        "_a",
        "_number",
        "_of",
        "_matters",
        "_that",
        "_have",
        "_been",
        "_made",
        "_at",
        "_the",
        "_conference",
        ",",
        "_which",
        "_had",
        "_been",
        "_made",
        "_at",
        "_the",
        "_conference",
        "."
    ],
    "score": -15.308104515075684
},
```