Transformer-NMT Course Project (ZH→EN)

本项目是《人工神经网络》课程的期末大作业。
你将**手写并训练一个 Transformer** 来完成中→英机器翻译任务，并提交模型与译文用于统一测评。

> **目录结构（已给出）：**
>
> ```
> .
> ├── check_translations.py     # 译文格式自检脚本
> ├── config.yaml               # 训练/模型/数据 等统一配置
> ├── evaluate.py               # 生成译文与测评
> ├── preprocess.py             # 数据预处理
> ├── tokenizer.py              # 分词/子词编码
> ├── train.py                  # 训练入口
> ├── utils.py                  # 通用工具
> └── model
>     └── transformer.py        # ★ 待补全的核心文件 (TODO)
> ```

1. 数据准备

数据集在data目录下提供，可在 `config.yaml` 中配置：

```yaml
data:
  raw_train:      data/train_10k.jsonl
  raw_val:        data/valid.jsonl
  raw_test:       data/test.jsonl
```

2. 你需要实现的内容

打开 `model/transformer.py` ，你会看到若干 **`# TODO`** ：

模块	需要实现
`PositionalEncoding`	位置编码矩阵计算
`MultiHeadAttention`	**Q/K/V 投影、Scaled-Dot、mask等**
`EncoderLayer`	自注意力 + FFN + 残差 + LayerNorm
`DecoderLayer`	Masked Self-Attn、Cross-Attn、FFN

请按照函数签名完成，实现后保证 **`train.py` / `evaluate.py`** 能正常调用。

3. 预处理 → 训练 → 推理

3.1 预处理

```
python preprocess.py -c config.yaml
```

生成分词文件与数据缓存（路径可在 config 中修改）。

3.2 训练

```
python train.py -c config.yaml
```

- 默认保存 checkpoint 至 `runs/`
  `runs/best_model.pt` 会被 `evaluate.py` 默认调用，可根据实际改变传入checkpoint参数

> **TIP:** 根据资源情况，可修改 `config.yaml` 中：
> *`train.batch_size`，`train.max_epochs`，`model.emb_size`，`model.enc_layers/dec_layers`* 等参数。

3.3 评测 / 生成译文

```
python evaluate.py                \
    -c config.yaml                \
    --ckpt runs/best_model.pt     \
    --save_path translations.json
```

输出示例：

```
Corpus BLEU: 22.8
Translations saved to translations.json
```

- `translations.json` **必须**包含 **列表结构**，每条格式如下：

```
{
  "src": "今天天气很好",
  "ref": "It is a fine day today",
  "hyp": "The weather is great today",
  "sha": "9c9f5d229f41..."
}
```

4. 输出文件自检

在提交前，务必检查 `translations.json` 格式是否正确：

```
python check_translations.py translations.json
```

若一切 OK，会看到

文件格式无误，共 N 条记录。

5. 提交内容

请在当前目录基础上 **新增** 2 个文件后打包提交（不要包含任何数据集(data目录)）：

```
.
├── check_translations.py
├── config.yaml
├── evaluate.py
├── preprocess.py
├── tokenizer.py
├── train.py
├── utils.py
└── model
    └── transformer.py
├── runs
│   └── best_model.pt           # ← 你的最终 checkpoint
└── translations.json           # ← evaluate.py 生成的译文
```

因此，最终压缩包应至少包含：

- **源代码**（含你已补全的 `transformer.py`）
- `runs/best_model.pt`
- `translations.json`

确保你所提交的文件能够正常执行以下脚本：

```
python evaluate.py                \
    -c config.yaml                \
    --ckpt runs/best_model.pt     \
    --save_path translations.json
```