.npy 文件内容与数据结构说明

本文件说明 `preprocess.ipynb` 代码生成的各个 `.npy` 文件的内容及其内部数据组织结构。

---

1. news_vert.npy

- **内容**：每条新闻的类别索引。
- **结构**：一维整型数组，长度为新闻总数+1（第0位通常为padding）。
- **示例**：

```
import numpy as np
news_vert = np.load('news_vert.npy')
print(news_vert.shape)  # (新闻数+1, )
print(news_vert[:5])    # [0 1 2 3 1]
```

---

2. news_title.npy

- **内容**：每条新闻标题的词索引序列。
- **结构**：二维整型数组，形状为 (新闻数+1, MAX_TITLE_LEN)。每一行是定长的词索引序列，不足补0，超长截断。
- **示例**：

```
news_title = np.load('news_title.npy')
print(news_title.shape)  # (新闻数+1, 16)
print(news_title[1])     # [12 45 67  0  0 ...]
```

---

3. news_body.npy

- **内容**：每条新闻正文的词索引序列。
- **结构**：二维整型数组，形状为 (新闻数+1, MAX_BODY_LEN)。每一行是正文的词索引序列，不足补0，超长截断。
- **示例**：

```
news_body = np.load('news_body.npy')
print(news_body.shape)  # (新闻数+1, 100)
print(news_body[1])     # [23 56 78 90  0 ...]
```

---

4. sources.npy

- **内容**：seq2seq模型编码器输入，每条新闻正文的词索引序列（更长）。
- **结构**：二维整型数组，形状为 (新闻数+1, MAX_CONTENT_LEN)。每一行是正文的词索引序列，不足补0，超长截断。
- **示例**：

```
sources = np.load('sources.npy')
print(sources.shape)  # (新闻数+1, 500)
print(sources[1])     # [23 56 78 ... 2 0 0 ...]  # 2为<eos>
```

---

5. target_inputs.npy

- **内容**：seq2seq模型解码器输入，每条新闻标题的词索引序列（以开头）。
- **结构**：二维整型数组，形状为 (新闻数+1, MAX_TITLE_LEN)。每一行是标题的词索引序列，不足补0，超长截断。
- **示例**：

```
target_inputs = np.load('target_inputs.npy')
print(target_inputs.shape)  # (新闻数+1, 16)
print(target_inputs[1])     # [1 12 45  0 0 ...]  # 1为<sos>
```

---

6. target_outputs.npy

- **内容**：seq2seq模型解码器输出，每条新闻标题的词索引序列（以结尾）。
- **结构**：二维整型数组，形状为 (新闻数+1, MAX_TITLE_LEN)。每一行是标题的词索引序列，不足补0，超长截断。
- **示例**：

```
target_outputs = np.load('target_outputs.npy')
print(target_outputs.shape)  # (新闻数+1, 16)
print(target_outputs[1])     # [12 45 67 2 0 ...]  # 2为<eos>
```

---

7. embedding_matrix.npy

- **内容**：词向量矩阵，每一行对应词典中一个词的预训练词向量（如GloVe）。
- **结构**：二维浮点型数组，形状为 (词数, 300)。
- **示例**：

```
embedding_matrix = np.load('embedding_matrix.npy')
print(embedding_matrix.shape)  # (词数, 300)
print(embedding_matrix[3])     # [0.12, -0.08, ...]
```

如需查看某个 `.npy` 文件的具体内容，可以用 `numpy.load` 加载后直接打印或切片查看。