KEY PROJECTS

웹 기사 추천 AI 경진대회

[Private 7위/ 0.33491] Hybrid GNN-based Content-Aware Recommendation System

한국외국어대학교 컴퓨터전자시스템공학부 학부생

kimtoil205@gmail.com

개발환경

GOOLGE COLLAB(CPU)
PANDAS : 2.0.3
NUMPY : 1.25.2
SCIKIT-LEARN : 1.2.2

DATA_SET_INFO

1. ARTICLE_INFO
   - USER_ID : 유저 고유ID
   - ARTICLEID : 기사 고유 ID
2. VIEW_LOG
   - ARTICLEID : 기사 고유 ID
   - CONTENT : 기사의 본문
   - LANGUAGE : 기사가 작성된 언어
   - USERID : 기사를 작성한 유저의 고유 ID

USER_PREDICTED_SCORE

- BASELINE과 동일
- USER-ARTICLE 행렬
- COSINE_SIMILARITY
- 협업필터링

CONTENT_BASED_SCORE

- 각각 언어마다 불용어 처리
- 토큰화 및 전처리

```python
# 불용어 로드
stop_words_dict = {
    'en': stopwords.words('english'),
    'pt': stopwords.words('portuguese'),
    'la': ['et', 'in', 'de'],
    'es': stopwords.words('spanish')
}

# 일본어 불용어 직접 정의
japanese_stop_words = ['これ', 'それ', 'あれ', 'この', 'その', 'あの', 'ここ', 'そこ', 'あそこ', 'こちら', 'どこ', 'だれ', 'なに', 'なん']

# 전처리 함수 정의
def preprocess_text(text, language):
    # URL 제거
    text = re.sub(r'https?://₩S+', '', text)
    if language == 'en':
        text = re.sub(r'[^a-zA-Z0-9₩s]', '', text)
    elif language == 'pt':
        text = re.sub(r'[^a-zA-Z0-9áéíóúâêîôûãõçÇ₩s]', '', text)
    elif language == 'la':
        text = re.sub(r'[^a-zA-Z0-9₩s]', '', text)
    elif language == 'ja':
        text = re.sub(r'[^₩u3040-₩u309F₩u30A0-₩u30FF₩u4E00-₩u9FFF₩s]', '', text)
    elif language == 'es':
        text = re.sub(r'[^a-zA-Z0-9áéíóúñÑ₩s]', '', text)
    # 소문자 변환
    text = text.lower()
    # 토큰화 및 불용어 제거
    if language == 'ja':
        tokenizer = Tokenizer()
        tokens = [token.surface for token in tokenizer.tokenize(text)]
        tokens = [token for token in tokens if token not in japanese_stop_words]
    else:
        tokens = nltk.word_tokenize(text)
        stop_words = stop_words_dict.get(language, [])
        tokens = [[token for token in tokens if token not in stop_words]]
    return ' '.join(tokens)

# 모든 기사에 대해 전처리 적용
article_info['ProcessedContent'] = article_info.apply(lambda row: preprocess_text(row['Content'], row['Language']), axis=1)
```

GNN

- GRAPHSAGE를 기반으로 한 GNN

GRAPHSAGE 레이어:

- 입력 차원 (IN_CHANNELS): X.SIZE(1) (TF-IDF 벡터의 축소된 차원, 여기서는 64)
- 은닉 차원 (HIDDEN_CHANNELS): 64
- 레이어 수: 2 (CONV1, CONV2)

선형 투영 레이어:

- USER_PROJ: HIDDEN_CHANNELS (64) -> NUM_ARTICLES
- ARTICLE_PROJ: HIDDEN_CHANNELS (64) -> NUM_USERS

학습 관련 파라미터:

- 옵티마이저: ADAM
- 학습률 (LEARNING RATE): 0.01
- 에폭 (EPOCHS): 1000
- 차원 축소 (TRUNCATEDSVD):
- 축소된 차원 수 (N_COMPONENTS): 64

```python
class HybridGNN(torch.nn.Module):
    def __init__(self, in_channels, hidden_channels, num_users, num_articles):
        super(HybridGNN, self).__init__()
        self.conv1 = SAGEConv(in_channels, hidden_channels)
        self.conv2 = SAGEConv(hidden_channels, hidden_channels)
        self.user_proj = torch.nn.Linear(hidden_channels, num_articles)
        self.article_proj = torch.nn.Linear(hidden_channels, num_users)

    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index).relu()
        x = self.conv2(x, edge_index)
        user_emb = self.user_proj(x[:num_users])
        article_emb = self.article_proj(x[num_users:])
        return user_emb, article_emb.t()
```

FINAL_SCORE

- 최종 추천 점수 계산:
- GNN 유사도 가중치: 0.44
- 사용자-기사 행렬 가중치: 0.28
- 콘텐츠 유사도 가중치: 0.28

```
    final_scores = 0.44 * gnn_similarity.numpy() + 0.28 * user_article_matrix.values + 0.28 * content_similarity_adj

# 추천 생성
recommendations = []
for idx, user in enumerate(user_ids):
    sorted_indices = final_scores[idx].argsort()[::-1]
    top5recommend = [article_ids[i] for i in sorted_indices[:5]]
    recommendations.extend([[user, article] for article in top5recommend])
```