**Reference Recommendation for LLM-Generated Text Using Deep Textual Representations**

This repository implements the reference recommendation system described in the paper "Reference Recommendation to Verify Information Produced by Large Language Models". The system helps users verify information from LLM chatbots by recommending relevant references using Siamese and Triplet networks with Sentence-BERT embeddings and submodular optimization.

**Requirements**

* Python 3.7+
* PyTorch 1.8+
* Transformers 4.0+
* Sentence-Transformers
* NetworkX
* Scikit-learn
* Numpy
* Tqdm

**Data Preparation**

Before running the code, you need to prepare your data in the following format:

1. **Queries**: A list of LLM-generated texts (answers) that need references
2. **Documents**: A list of potential reference documents
3. **Citation Graph**: A NetworkX DiGraph showing citation relationships between documents

Example:

python

queries = [

"Large language models have shown remarkable capabilities in natural language understanding.",

"The transformer architecture has become fundamental to modern NLP systems."

]

documents = [

"Attention is All You Need by Vaswani et al. introduced the transformer architecture...",

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding...",

"GPT-3 demonstrated few-shot learning capabilities across diverse tasks..."

]

citation\_graph = nx.DiGraph()

citation\_graph.add\_edge(0, 1) *# Query 0 cites document 1*

citation\_graph.add\_edge(1, 2) *# Document 1 cites document 2*

**Configuration**

The system behavior can be configured by modifying the Config class in config.py:

python

class Config:

def \_\_init\_\_(self):

self.batch\_size = 16

self.learning\_rate = 2e-5

self.epochs\_siamese = {1: 30, 2: 20, 3: 10, 4: 5} *# Distance-based epochs*

self.epochs\_triplet = {1: 25, 2: 15, 3: 7, 4: 4}

self.alpha = 0.4 *# Similarity decay factor*

self.margin = 1.0 *# Triplet loss margin*

self.top\_k = [1, 3, 5] *# Evaluation metrics*

self.device = "cuda" if torch.cuda.is\_available() else "cpu"

**Expected Output**

When running the training and evaluation, you should see output similar to:

text

Training Siamese Network...

Training with distance=1 for 30 epochs

Epoch 1/30, Loss: 0.2543

Epoch 2/30, Loss: 0.1987

...

Training with distance=2 for 20 epochs

...

Evaluating Siamese Network...

Siamese Network - MRR: 0.377, F1@k: {1: 0.170, 3: 0.168, 5: 0.137}

Evaluating Triplet Network...

Triplet Network - MRR: 0.436, F1@k: {1: 0.215, 3: 0.188, 5: 0.144}

Testing Submodular Recommendation...

Query: Large language models have shown remarkable capabilities...

Recommended references:

1. BERT: Pre-training of Deep Bidirectional Transformers...

2. Attention is All You Need by Vaswani et al...

3. GPT-3 demonstrated few-shot learning capabilities...

**Pretrained Models**

You can download pretrained Sentence-BERT models from the sentence-transformers repository. The default model used is 'all-mpnet-base-v2'.