Methodology

Overview

This research employs a multi-stage computational approach to identify and analyze patterns in historical advertisement data. The methodology combines deep learning-based feature extraction, dimensionality reduction, and density-based clustering to discover latent thematic clusters in both visual and textual content. The process consists of four main stages: (1) feature extraction, (2) dimensionality reduction, (3) clustering, and (4) cluster interpretation through representative sampling.

1. Feature Extraction

1.1 Visual Feature Extraction

Visual features were extracted using DINOv2 (Facebook's self-supervised vision transformer model) [citation needed]. The `facebook/dinov2-base` model was employed to encode image content into high-dimensional feature vectors.

Implementation details:

- **Model**: DINOv2-base (Vision Transformer)
- **Feature representation**: CLS token embeddings from the model's pooler output
- **Normalization**: L2 normalization applied to all embeddings to ensure unit-norm vectors, which improves clustering performance and similarity calculations
- **Dimensionality**: 768-dimensional feature vectors

Each advertisement image was preprocessed and fed through the DINOv2 model, with the CLS token serving as a global representation of the image content. The embeddings were normalized using L2 normalization.

1.2 Textual Feature Extraction

Textual features were extracted using Qwen3-Embedding-0.6B, a Chinese language embedding model optimized for semantic understanding. The model processes text sequences and generates contextualized embeddings.

Implementation details:

- **Model**: Qwen3-Embedding-0.6B
- **Feature representation**: Last token pooling strategy, extracting the final hidden state corresponding to the last non-padding token
- **Normalization**: L2 normalization applied to embeddings
- **Dimensionality**: 1024-dimensional feature vectors
- **Tokenization**: Maximum sequence length of 8,192 tokens with left padding

The last token pooling strategy was chosen as it captures the full context of the input sequence, making it suitable for longer texts such as advertisement copy. Embeddings were normalized using L2 normalization to ensure consistent vector magnitudes.

2. Dimensionality Reduction

To enable visualization and improve clustering performance, high-dimensional embeddings were reduced to two dimensions using Uniform Manifold Approximation and Projection (UMAP) [McInnes et al., 2018]. Euclidean distance was used for visual features, while cosine distance was employed for textual features, which is appropriate for normalized embeddings.

UMAP was selected over other dimensionality reduction techniques (e.g., t-SNE, PCA) because it:

1. Preserves both local and global structure in the data
2. Scales better to large datasets
3. Produces more interpretable 2D projections for visualization

The dimensionality reduction process transformed:

- Visual embeddings: 768D → 2D
- Textual embeddings: 1024D → 2D

3. Clustering

Density-based clustering was performed using Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [Campello et al., 2013]. HDBSCAN was chosen for its ability to:

- Identify clusters of varying densities
- Automatically determine the number of clusters
- Handle noise points (outliers) explicitly
- Work effectively with normalized embeddings

Different parameter settings were used for visual and textual clustering to account for the distinct characteristics of each modality. The Excess of Mass (EOM) method was employed for cluster selection to ensure more stable results. Points not assigned to any cluster (labeled as -1) are treated as noise/outliers and excluded from cluster interpretation.

4. Cluster Interpretation

To interpret the discovered clusters, representative samples were systematically extracted from each cluster.

4.1 Sampling Strategy

- **Samples per cluster**: 10 representative items
- **Sampling method**: Random sampling with fixed random seed for reproducibility
- **Adaptive sampling**: If a cluster contains fewer than 10 items, all items are included

4.2 Interpretation Process

For each cluster:

1. **Sample extraction**: 10 representative items were randomly selected
2. **Visual inspection**: For visual clusters, images were displayed in a 2×5 grid layout for comparative analysis

3. **Textual analysis**: For textual clusters, full text content was examined for common themes, linguistic patterns, and rhetorical strategies
4. **Pattern identification**: Recurring motifs, themes, and structural elements were identified across samples
5. **Thematic labeling**: Each cluster was assigned a descriptive label based on the dominant pattern observed

The sample data was saved to CSV files (`output/clustering/samples/textual_cluster_samples.csv`) containing:

- Cluster ID and label
- Cluster size
- Sample number
- Original content (text or image path)
- 2D projection coordinates (x, y)

5. Data Pipeline Summary

The complete pipeline can be summarized as:

```
Raw Data (Images + Text)
    ↓
Feature Extraction (DINOv2 / Qwen3)
    ↓
L2 Normalization
    ↓
UMAP Dimensionality Reduction (768D/1024D → 2D)
    ↓
HDBSCAN Clustering
    ↓
Cluster Results + Representative Samples
    ↓
Interpretation & Analysis
```

6. Technical Implementation

All analysis was conducted using Python with the following key libraries:

- **Transformers**: Hugging Face transformers for DINOv2 and Qwen3 models
- **UMAP**: `umap-learn` for dimensionality reduction
- **HDBSCAN**: `hdbscan` for density-based clustering
- **NumPy/Pandas**: Data manipulation and storage
- **Visualization**: Matplotlib for image display, Cosmograph for interactive exploration

Code is organized into modular scripts:

- `visual_feature_extraction.py`: Visual embedding extraction
- `textual_feature_extraction.py`: Textual embedding extraction
- `visual_clustering.py`: Visual clustering pipeline

- `textual_clustering.py`: Textual clustering pipeline

All results are saved in structured formats (JSONL for features, CSV for clustering results) to enable reproducibility and further analysis.