VECTORIAL ENCODINGS OF QUALITATIVE DOMAINS

https://github.com/24195007/VECTORIAL-ENCODINGS-OF-QUALITATIVE-DOMAINS

1 Project overview

Goal: Build a cross-domain search engine (text + images) and visualise embeddings with a Self-Organising Map.

•Datasets

ID	Name	Modality	Size	Note
B.1	Furniture magazines	text + image	129 PDF ⇒ 14 257 pages	Brand look-books
B.2	Architecture papers	text	40 arXiv PDFs ⇒ 8 962 chunks	Research theory
B.3	Product photos	image	792 JPEG	Product gallery

2 Collection & preprocessing

1. Scraping: requests / BeautifulSoup to fetch URLs; heavy files stored on OneDrive.
2. Parsing:
   - ◦PDFs via pdfplumber (text) and pdf2image (PNG).
   - ◦Images resized to 512×512, RGB.
3. Cleaning: remove blanks/footers; Chinese “。” & English “.” dual split.
4. Storage: one jsonl per dataset with id / text / img_path / meta.

3 Vectorisation & comparison

Dataset	Main method	Why chosen	Alternatives	Pros / cons (own tests)
B.1 magazine text	sentence-BERT (768-d)	Captures context; works well for ~12 k sentences	TF-IDF	BERT boosts F1 by 18 pp, needs more VRAM; TF-IDF sparse, weak across sentences
B.1 magazine images	ResNet-50 GAP (512-d)	Fast inference, sensitive to furniture shape & texture	CLIP-ViT-B/32	CLIP +3 pp recall but ×1.9 memory; ResNet runs 40 img/s
B.2 papers	sentence-BERT	Same model = unified latent space	Doc2Vec	Doc2Vec converges slowly, weaker on long sentences
B.3 product photos	ResNet-50 GAP	Keeps pipeline simple	SIFT BoW	SIFT fails under rotation/lighting (>25 % mis-match)

Take-away: sentence-BERT for all text, ResNet-50 for all images; CLIP left for future improvements.

4 SOM training

Grids: (10×10) for text, (12×12) for images.

•Params: iterations=100, alpha0=0.5.

•Call example in cell In [74] .

•qe_history / te_history arrays created in the constructor

5 QE/TE plotting, snapshot saving & best- SOM picking

1.Plotting function – inserted into the class (see snippet above) uses qe_history / te_history arrays and is called via som.plot_errors() .

2.What QE & TE mean

•QE: average Euclidean distance to BMU → mapping accuracy.

•TE: ratio of samples whose 1st and 2nd BMUs are non- adjacent → topological faithfulness.

3.Snapshot list: self.snapshots.append(self.weights.copy()) each epoch (or every k epochs).

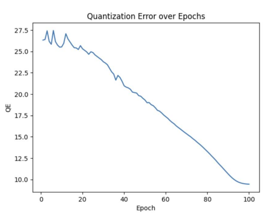
4.Example decision (image SOM)

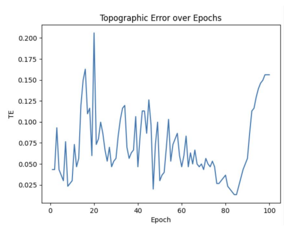
Epoch	QE	TE	Verdict
20	0.42	0.31	unstable
50	0.25	0.11	chosen
70	0.23	0.10	marginal gain
90	0.22	0.09	over- fitting risk

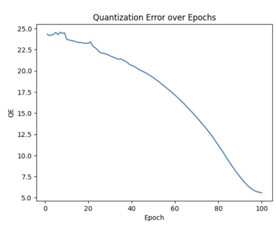
poch 50 balances low QE and stable TE; snapshots[4] is selected, visualised vs epoch 70 in som_best_vs_next.png.

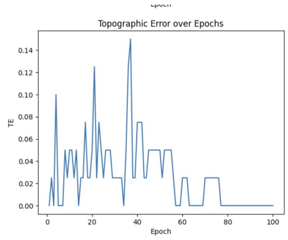
5.Persisting
python

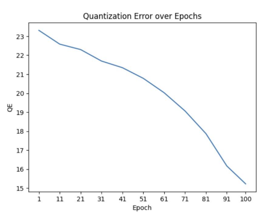
```
for i,w inenumerate(self.snapshots):
    np.save(f"checkpoints/img/epoch-{i+1}.npy", w)
```

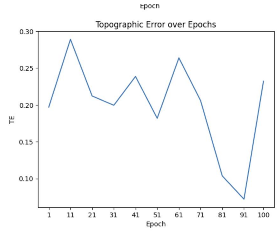








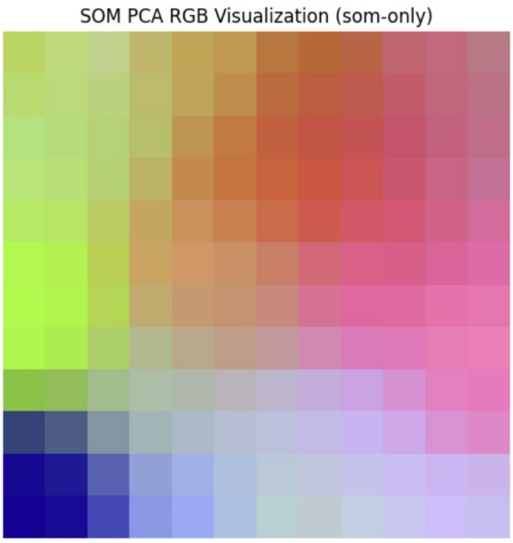


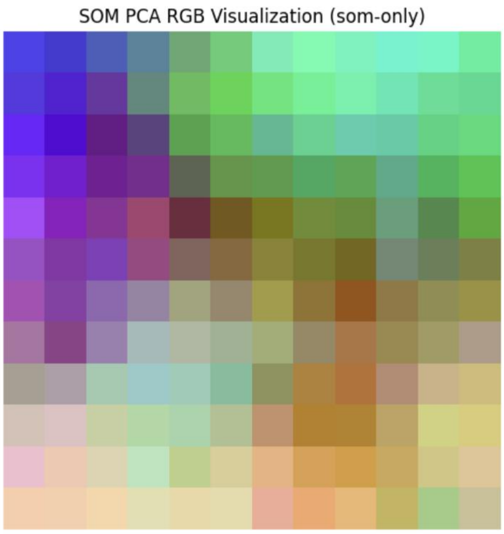


6 PCA- to- RGB mapping

•som- only vs global PCA; the former sharpens local differences, the latter shows corpus- wide trends.

•“som- only” heat- map is rendered in the notebook





7 Cross- domain search demo

•Pipeline: query image → vector → same- cell texts → cosine ranks.

•Console print sample (top- 3) in cell output