

Smart Multi-Modal Search: Contextual Sparse and Dense Embedding Integration in Adobe Express

Chirag Arora, Tracy Holloway King, Jayant Kumar, Yi Lu, Sanat Sharma, Arvind Srikantan, David Uvalle, Josep Valls-Vargas and Harsha Vardhan

Adobe Inc.

Abstract

As user content and queries become increasingly multi-modal, the need for effective multi-modal search systems has grown. Traditional search systems often rely on textual and metadata annotations for indexed images, while multi-modal embeddings like CLIP enable direct search using text and image embeddings. However, embedding-based approaches face challenges in integrating contextual features such as user locale and recency. Building a scalable multi-modal search system requires fine-tuning several components. This paper presents a multi-modal search architecture and a series of AB tests that optimize embeddings and multi-modal technologies in Adobe Express template search. We address considerations such as embedding model selection, the roles of embeddings in matching and ranking, and the balance between dense and sparse embeddings. Our iterative approach demonstrates how utilizing sparse, dense, and contextual features enhances short and long query search, significantly reduces null rates (over 60%), and increases click-through rates (CTR). Our findings provide insights into developing robust multi-modal search systems, thereby enhancing relevance for complex queries.

Keywords

multi-modal search, text-image embeddings, hybrid search techniques

1. Introduction

For search over images and multi-modal content, industry search systems traditionally rely on textual and metadata annotations added to indexed images. However, multi-modal embeddings like CLIP [1] enable direct search of image content using text and image embeddings, allowing for direct text-to-image and image-to-image search. While pure embedding-based approaches facilitate content understanding, they struggle with integrating contextual features like user locale and recency into retrieved results. Building a production-grade, scalable, multi-modal search system involves carefully tuning several components. This paper describes a series of AB tests conducted to leverage embeddings and other multi-modal technologies in search for Adobe Express templates. These templates are complex multi-modal (and multi-page) documents, containing images, text and rich metadata (section 2). Figure 1 shows the Adobe Express template search for a head query and a tail query, where the templates are displayed to the user as images and template metadata drives the left rail filters.

To improve text search for templates, integrating embeddings required decisions as to:

- Which embedding model(s) to use
- Whether to leverage embeddings for matching (recall), ranking, or reranking
- Whether to use dense or sparse embeddings
- Whether head and tail queries should be treated identically
- Whether embeddings should be used for null and low recovery or everywhere

Other than which embeddings to use, these decisions were driven by latency concerns and by constraints on integration with Elasticsearch, which was the existing inverted index used for Express template search. With an ever-increasing collection of ~300,000 templates, dense embeddings could not be used

MMSR2024 Workshop

✉ charora@adobe.com (C. Arora); tking@adobe.com (T.H. King); jaykumar@adobe.com (J. Kumar); yil@adobe.com (Y. Lu); sanatsha@adobe.com (S. Sharma); asrikantan@adobe.com (A. Srikantan); duvallezeped@adobe.com (D. Uvalle); jvallsvargas@adobe.com (J. Valls-Vargas); hmatadaallam@adobe.com (H. Vardhan)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

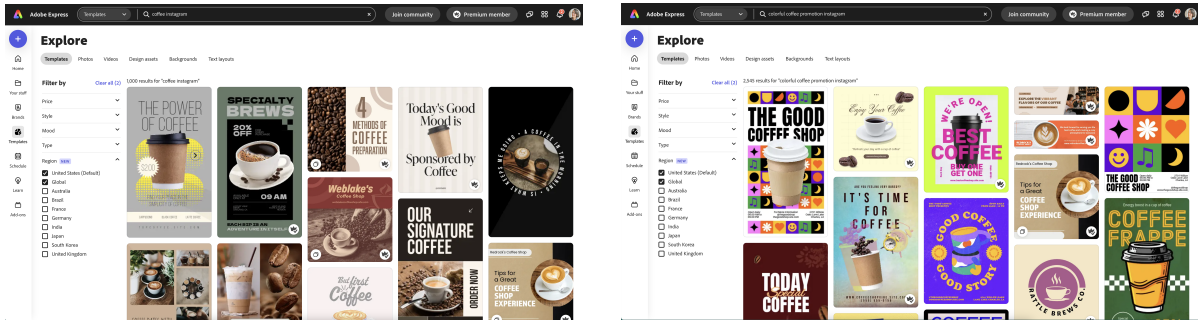


Figure 1: Examples of Express template search. Left: head query *coffee instagram*. Right: tail query *colorful coffee promotion instagram*

for matching due to the number of scoring calculations, which leads to high latency. This restricted dense embeddings to (re)ranking, where only a small (<10K) number of top templates had to be scored, and to scenarios like null and low recovery and long tail queries, where the additional latency was worth the improved relevance. In addition, certain types of queries performed better with keyword search, especially those around design type (e.g. poster, Instagram reel) and format (e.g. still, animated, video).

To determine the optimal combination, we took an iterative approach with a series of evaluations and AB tests. We started with existing models with single integrations and then built on these to improve remaining relevance issues. This paper first overviews the data and models used (section 2) and then discusses the experiments and how the decisions were made for each of these (section 3).

2. Models and Data

This section describes key data and models used for the Express template recall and ranking. The templates themselves contain rich image, text, and metadata. The standard search behavioral data (e.g. impressions, clicks) are available, as well as certain application-specific behavioral data (e.g. number of edits, number of exports). These are briefly described in section 2.1. In addition, we have two types of multi-modal models: two CLIP text-image models (section 2.2) and an intent-based model (section 2.3).

2.1. Template Data

Express templates are rich objects which contain many visual layers and text boxes. These can also be viewed as images, e.g. those that are displayed in search (Figure 1). In addition, templates have titles provided by the template designers as well as filter information such as design type, style, mood, region, and price (free/premium). Additional information is inferred about each template including multi-modal embeddings, user intents, and image tags. Finally, aggregated behavioral data such as impressions, clicks, edits (number of edits users make to the template in order to personalize them) and exports (number of times the template is exported after editing) are available. An example template with its data is shown in Figure 2.

2.2. Image-Text CLIP Embeddings

CLIP [1] embeds images and text in the same space. This allows for embedding-based search of images using text queries. There are several off-the-shelf CLIP models available. However, for Express template search and for other visual asset search like Adobe Stock, we needed a model that: (1) worked on short text (queries) as well as long text (captions); (2) covered five languages (English, French, German, Japanese, Korean); (3) performed well on high-quality image data for templates, photographs and illustrations; (4) had a sparse version as well as the dense vectors. To meet these requirements, we trained a CLIP-architecture model on Adobe-licensed image-text data. The text model was particularly



Title: Pink Unicorn Birthday Party Instagram Portrait Post
 Topics: confetti, fantasy, glitter, gold, kids, sparkle, star, unicorn
 Mood: happy, joyful; Style: bright
 Region: all; Language: en-US; Date: 2023-12-12; Behavior: still; License: premium
 AI-inferred: AdobeCLIP embedding, Multi-modal CKG embedding,
 CKG symbolic intent, autotags
 Behavioral: Search impressions, clicks, edits, exports

Figure 2: Sample Express template data available for search matching and ranking

important since the training focused on Adobe vocabulary, shorter text, and multiple languages; the training architecture was inspired by [2].

There are many ways to improve the latency when using embeddings with large numbers of assets. However, the approximate methods reduce accuracy because the list of assets whose embeddings are closest to the query embedding is not exact. Once a smaller set of embeddings is selected (e.g. by using the top n embeddings from the approximate scoring), then the dense embedding can be used to get more accurate scores for a final ranking. We used a sparsification method which allows the embeddings to be used similar to keywords in the existing index.¹ An example of this is shown in Table 1.

Table 1

Dense and Sparse representations of embeddings with sample scoring for sparse embeddings. Dense embeddings are shown with 2048 dimensions. Sparse embeddings have more dimensions (here 8192) but most of the dimensions have no values.

Dense Embedding Example			Sparse Embedding Example				
Dim.	Img. 1	Img. 2	Dim.	Query = Img. 1	Img. 2	Img. 3	Img. 4
1	0.11	1.23	1	—	1.12	—	—
2	1.21	0.42	2	—	—	0.81	—
3	0.15	0.53	3	1.16	0.83	—	0.64
4	0.22	2.25	4	—	—	1.83	—
...			...				
2048	2.17	0.64	8192	0.13	—	—	0.01

Sparse Embedding Matching and Scoring
Matches for query image 1: image 2: 1 dimension (dimension 3) image 3: 0 dimensions image 4: 2 dimensions (dimension 3 & 192)
Scoring for ranking: image 2: $1.16 * 0.83 = 0.96$ image 4: $1.16 * 0.64 + 0.13 * 0.01 = 0.74$

The dense embeddings have values for every dimension (2048 dimensions in table 1). The sparse version, which is derived from the dense one, has more dimensions (8192 in table 1) but most of them have no values. For a query, only assets which match at least n dimensions are returned. In the example, n is set to 1 and so image 2 matches dimension 3 and image 4 matches dimensions 3 and 8192, while image 3 is not matched. The scoring is the sum of the matched dimension values weighted by the score of that dimension for the query. This sparse encoding for matching and scoring is extremely fast, but comes at the cost of lower accuracy.

The Adobe-specific model (AdobeCLIP) was evaluated for Adobe Express and Stock content, baselining against the off-the-shelf CLIP versions. For large scale evaluation, a stratified sample of held-out search

¹Other approaches are described in [3, 4, 5].

queries were used for semantic search against an index of CLIP and AdobeCLIP embeddings for Express and Stock content. Past clicked assets were considered relevant, non-clicked items irrelevant. In addition, we selected titles (generally 5–15 words) for Express templates and Stock images and used them as long queries, measuring the position of the asset which originally had that title. These methods provide a lower bound on performance since many of the non-clicked items are relevant and titles often matched multiple high-ranking images. These two approaches allowed us to quickly compare different versions of AdobeCLIP against one another and against CLIP. Once the AdobeCLIP model outperformed CLIP and earlier AdobeCLIP versions, we manually inspected results for a subset of held-out queries.

2.3. Multi-Modal Creative Knowledge Graph

In addition to learning representations of the content via AdobeCLIP, we found mapping the content’s intent to discrete nodes improved recall and explainability and allowed for downstream-recommendation tasks, similar to [6]. However, we discovered that self-supervised models like AdobeCLIP, which were trained on asset-caption and asset-query data like Adobe Stock and Adobe Express failed to accurately map the asset’s intent to short discrete labels. To accomplish, this we created a “Creative” Knowledge Graph (CKG) [7, 8, 9] containing over 100K nodes focusing on Adobe-specific user intents. We then trained a multi-modal transformer (MM-CKG) specializing in mapping assets to these discrete nodes using supervised contrastive training. We mined concepts for events, actions, objects, moods, canvas types, colors, and backgrounds to get a robust understanding of an asset’s content. For example, *actions* has subtypes of *run, dance, ...*; *events* has subtypes of *birthday, graduation, wedding, seasonal, ...*; in turn *events|seasonal* has subtypes of *Halloween, Thanksgiving, 4th of July, ...*

To train the model, we created sequence-wise self-attention blocks inspired by [10]. We built our model on top of a base CLIP backbone and then added a sequence-wise attention block on top that takes in the hidden states from the last layer of the CLIP backbone that runs through a couple layers of multi-headed transformer blocks. We utilized the T_{cls} and I_{cls} outputs from the sequence-wise attention heads as the final representation of the input image and text modalities.

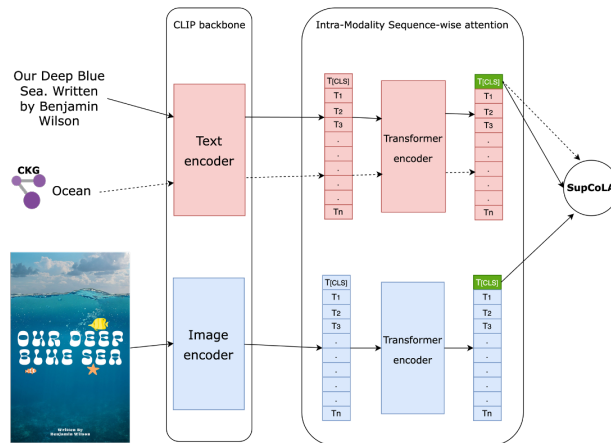


Figure 3: MM-CKG uses supervised contrastive learning with SupCoLA loss for label alignment. This allows the model to bring short labels focusing on the overall intent of the asset closer to the content embeddings.

2.3.1. Supervised Contrastive Loss (SupCoLA)

We devised our loss function with the following requirements:

1. Alignment to labels: Ensure that the image and text in the training process were close to the label embeddings.
2. Ability to handle multiple positives in a batch: Traditional contrastive learning (InfoNCE loss [11]) assumes that for a given pair in a batch, all other pairs are negatives. However, when

learning alignment with labels, multiple rows with the same label may be present in a batch. The loss function should not penalize these rows during loss computation.

3. Ability to have multiple labels per row: Some rows have multiple labels. For example, for the prompt, *boy is sitting on a beach with his dad for father’s day*, there are multiple concepts: the creative intent *father’s day*, the scene objects *boy* and *beach*, and the background *beach background*.

Our resulting loss function, Label-Aligned Supervised Contrastive Loss, is based on SupCon loss [12] where we pass image, text and label embeddings as anchor features as well as contrast features.

$$\mathcal{L}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_i^{\text{sup}} = \sum_{i \in I} \left(-\frac{1}{P(i)} \sum_{p \in P(i)} \left[\sum_{v \in j(p)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_v / \tau)}{\sum_{n \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_n / \tau)} \right] \right) \quad (1)$$

where I is the mini batch, i is the index of anchor sample in the batch, $A(i) \equiv I \setminus i$ is the set of all samples n in the batch that have distinct index than the anchor i , $j(p)$ is the set of all positives p in the batch that have the same label as anchor i and are views of p . Views of sample p denote the embeddings for the label, image and text modalities.

Why do we have two domain-specific multi-modal embeddings (AdobeCLIP and MM-CKG)? These target and excel at different use cases, both of which are important for template search relevance. MM-CKG is better at determining the underlying key intent from a query and at specific scene object detection. AdobeCLIP is better at color and layout understanding.

3. Iterative Experiments

This section describes the series of on-line experiments conducted to improve the Express template multi-modal search. We focus primarily on experiments involving multi-modal embeddings, but include one experiment that leveraged multi-modal content under the hood but text at query time.

The Express template search uses a standard architecture for relevance (Figure 4). It is built on Elasticsearch. There is an initial matching (recall) step to retrieve documents which broadly match the user’s query. This step uses keyword-style matching against text and metadata and includes an initial low latency scoring. Matching using sparse AdobeCLIP embeddings for all queries (section 3.4) and dense multi-modal CKG embeddings for long queries (section 3.5) were also added. If not enough results are found, null and low recovery occurs, including a speller (not discussed in this paper; see [13]) and the use of symbolic CKG intents (section 3.2). The top 10K templates from the initial match set are then reranked using a much broader set of features. This includes dense multi-modal embeddings as well as the usual discrete features such as BM25, locale, language, and aggregated behavioral data.

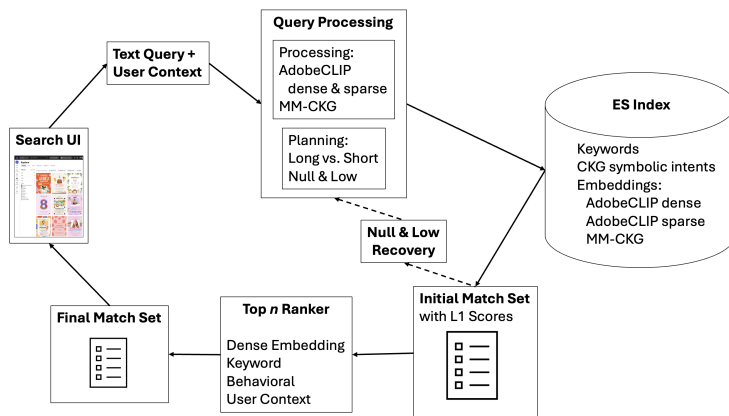


Figure 4: High-level search architecture for Express template multi-modal search

To improve relevance, and hence search click-through rate and export rate, we took an iterative approach to learn how to optimally leverage our multi-modal understanding, especially the multi-modal

embeddings. Each of these are discussed in detail below. Since we were redesigning the search system, including the retrieval and ranking platform, some of the experiences were evaluated extensively offline and then launched with end-over-end monitoring, while later experiences were AB tested.²

1. Reranking with External Image-Text Model (section 3.1)
2. Null and Low recovery with Symbolic Multi-Modal Intents (section 3.2)
3. Ranking with Domain-specific Image-Text Model (section 3.3)
4. Recall with Sparse Image-Text Model (section 3.4)
5. Long Query Recall and Ranking with Multi-Modal Model (section 3.5)

3.1. Reranking with External Image-Text Model

Our initial experiment used an external, English-only CLIP model in the rescore stage of the ranker. To do this, we had to determine how many items to rescore. This was largely governed by latency concerns since we wanted as many items as possible to use the CLIP multi-modal signal. By running load tests, we determined that we could use the CLIP scores for the top 10K templates, where the initial ranking was determined by the existing ranker.

We also had to determine how to weight the CLIP scores in the rescoring. Since the template ranker at this time was a non-ML, hand-tuned ranker, we determined this based on evaluations of a stratified query sample. The extreme baseline was to use only the CLIP score for the reranking. This had two drawbacks: (1) The top results were not visually diverse enough, especially for broad queries like *birthday card* or *wedding invitation*; (2) there was not enough recent content to provide a sense of freshness and seasonality. To determine a suitable weight, we used a divide-and-conquer approach by starting with a 50% balance between the first round ranker score and the CLIP score and then adjusting. This quickly converged on a weighting of basically 2/3 for CLIP and 1/3 for the first round ranker score. In AB testing, the click-through rate (CTR) and export rate improved with the CLIP-based reranking (Table 2).³ There was no change in the null rate, as was to be expected.

Table 2

AB Test Results for CLIP Reranking. All results are statistically significant.

Metric	Change
CTR	+7%
Export rate	+4%

3.2. Null and Low Recovery with Symbolic Multi-Modal Intents

Due to the broad range of user intents, the limited template collection, and the keyword based retrieval, users frequently landed on null and low result pages. When the number of results are low (<5 results), the search engagement with the results drops significantly (up to 2–3x). To reduce the null and low result rate, we incorporated recovery mechanism using the symbolic CKG intents. The CKG intents for each template were indexed and the CKG intents for the query were calculated at query time. If there were <5 results, the CKG query intents were matched against the template intents. For example, the query *hot yoga studio opening* has the intents *yoga* and would match all templates with that intent. This resulted in major improvements in CTR and null rate (table 3).

3.3. Ranking with Domain-specific Image-Text Model

The CLIP model (section 3.1) only worked for English and was not optimized for Express templates and queries. Replacing CLIP with AdobeCLIP to rerank the top 10K Express templates was expected

²The template collection is continually expanding and so the maximum recall size varied for each experiment.

³All results are shown as percentage change. We are unable to show exact metrics.

Table 3

AB Test Results for CKG Symbolic Intents Null and Low Recovery. All results are statistically significant.

Metric	Change
Null rate	-69%
Null recovery CTR	+300%
Low rate	-59%
Low recovery CTR	+30%
Overall CTR	+7%

to be on par for English queries and improve the CTR for non-English queries. Because the recall and first-round ranker constrained the result set, the core relevance, especially for head queries, was unlikely to change significantly, although the torso and tail queries, especially in non-English were expected to be significantly different. The move from CLIP to AdobeCLIP was part of a larger AB test which moved from an older search infrastructure to a newer one which, among other things, allowed for multiple embedding types. The goal of the AB test was to have no negative effects while moving to the new platform. This was borne out (table 4).

Table 4

AB Test Results for Platform Move including CLIP-to-AdobeCLIP. Only the null rate change is statistically significant.

Metric	Change
CTR	+0.0%
Null rate	-7.7%
Low rate	-5.0%

3.4. Recall with Sparse Image-Text Model

None of the previous experiments leveraged the power of embeddings for augmenting the initial match set. The CLIP and AdobeCLIP models only affected the reranking of the search results. The null and low recovery with symbolic CKG multi-modal intents only affected null and low queries and leveraged symbolic intents. Dense embeddings could not be used for the initial match set due to latency constraints. So, we experimented with using the AdobeCLIP sparse embeddings in the match set to augment the existing keyword matches.⁴ This required determining how many dimensions to match in the sparse embedding (section 2.2) in order to retrieve enough new relevant documents and not too many irrelevant ones. As is well known in the literature [14, 15] determining accurate thresholds on embeddings using cosine similarity or dot product is not feasible.⁵ The sparse embedding approach allowed us to require a minimum number of asset dimension matches. Once the retrieval approach was determined, the ranking was updated to demote less relevant templates retrieved by the sparse embeddings (table 5).

Table 5

AB Test Results for AdobeCLIP Recall and Ranking. All results are statistically significant.

Metric	Change	
	English	Multilingual
CTR	+3%	+3%
Null Rate	-35%	-14%
Null and Low Recovery Rate	-54%	-54%

⁴We considered using only AdobeCLIP sparse for the match set but rejected this since the model performed badly at identifying videos, which are popular queries and important to the business. AdobeCLIP only has the image embedding to match against and the images from Express video templates are extremely similar to those of still templates.

⁵See [16] for a recent approach to determining relevance thresholds with embeddings.

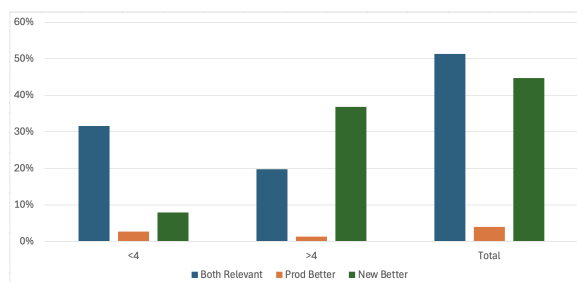
3.5. Long Query Recall and Ranking with Multi-Modal Model

The above experiments improved relevancy for head queries, both in matching the intent of the user query and in quality of the templates shown. In addition, the improved recall from CKG symbolic intents for null and low recovery (section 3.2) and the addition of sparse AdobeCLIP embeddings into the initial match set (section 3.4) resulted in a broad set of related templates being shown when there are few exact matches. However, there are often few exact match templates for more specific user queries, i.e. for tail queries.

To address this issue, we targeted longer (≥ 4 words) to use the CKG multi-modal embedding (MM-CKG section 2.3). The more specific intents of the longer queries work especially well with the domain-specific embeddings, allowing the recall and ranking to find the few templates that exactly match the user intent. The ranking combined 1/3 the weight on MM-CKG and 2/3 the weight on AdobeCLIP. The hypothesis behind this was that AdobeCLIP captures the core relevance matching the query text to the image rendition, while MM-CKG captures the underlying intent of the query and the template. The optimal query length for this experience was determined empirically by manually judging a stratified sample of queries of different lengths, comparing production to the new experience. There was a clear demarcation between queries of < 4 words and those of ≥ 4 words. Table 6 shows that for < 4 words, both production and MM-CKG largely provide relevant results, i.e. for head queries both approaches work well. However, for ≥ 4 words the new MM-CKG results are significantly better than those in production.

Table 6

Relevancy results from human annotation when leveraging MM-CKG for recall and ranking.



The MM-CKG embedding does not yet have a sparse version of the type available for AdobeCLIP (section 2.2). For this reason, it was not feasible from a latency perspective to use the MM-CKG matching and ranking approach used for all queries. However, since long queries are $< 10\%$ of the query traffic, we determined that the increased latency from calculating cosine similarity scores between the query and all of the templates was worth the improvements to relevance. At time of submission, this experience is in AB testing. Results will be available by the conference.

4. Conclusion

Multi-modal search experiences in industry applications traditionally depend on textual data in the index, thereby reducing the multi-modal search to a traditional keyword search. This provides a low latency experience since industry search engines are heavily optimized for keyword search. The advent of high quality multi-modal embeddings like CLIP has provided radically new capabilities. However, in an existing application, such as Adobe Express template search in this paper, the available multi-modal capabilities and the existing infrastructure including strict latency requirements, require a thoughtful, iterative approach to integrating new multi-modal technologies. This paper described five multi-modal experiments in Express template search, each of which built upon the others. This has resulted in significantly lower null and low rates, while improving click-through rates.

References

- [1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, 2021. ArXiv:2103.00020.
- [2] F. Carlsson, P. Eisen, F. Rekathati, M. Sahlgren, Cross-lingual and multilingual CLIP, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odiijk, S. Piperidis (Eds.), Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 6848–6854. URL: <https://aclanthology.org/2022.lrec-1.739>.
- [3] W. Kong, J. M. Dudek, C. Li, M. Zhang, M. Bendersky, Sparseembed: Learning sparse lexical representations with contextual embeddings for retrieval, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), 2023.
- [4] H. Jégou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 117–128.
- [5] A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain, A. Farhadi, Matryoshka representation learning, 2024. arXiv:2205.13147.
- [6] N. L. Le, M.-H. Abel, P. Gousspillou, Combining embedding-based and semantic-based models for post-hoc explanations in recommender systems, 2024. URL: <https://arxiv.org/abs/2401.04474>. arXiv:2401.04474.
- [7] J. Kumar, M. Deshmukh, M. Liu, A. Gupta, K. Suresh, C. Arora, J. Zheng, R. Sadaphule, V. Dalal, A. Stefan, 2023. URL: <https://patents.google.com/patent/US11645095B2/en>, United States Patent and Trademark Office, number US11645095B2.
- [8] S. Sharma, J. Kumar, J. Zheng, Multimodal input contextual font, 2023. URL: <https://patents.google.com/patent/US11775734B2/en>, United States Patent and Trademark Office, number US11775734B2.
- [9] S. Sharma, J. Kumar, J. Zheng, T. H. King, Contextual font recommendations based on user intent, 2023. URL: <https://arxiv.org/abs/2306.08188>. arXiv:2306.08188, presented at ECOM23 the SIGIR Workshop on e-Commerce Search.
- [10] H. Liu, S. Xu, J. Fu, Y. Liu, N. Xie, C.-C. Wang, B. Wang, Y. Sun, Cma-clip: Cross-modality attention clip for image-text classification, 2021. URL: <https://arxiv.org/abs/2112.03562>. arXiv:2112.03562.
- [11] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2019. URL: <https://arxiv.org/abs/1807.03748>. arXiv:1807.03748.
- [12] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, 2021. URL: <https://arxiv.org/abs/2004.11362>. arXiv:2004.11362.
- [13] S. Sharma, J. Valls-Vargas, T. H. King, F. Guerin, C. Arora, Contextual multilingual spellchecker for user queries, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 3395–3399. URL: <https://doi.org/10.1145/3539618.3591861>. doi:10.1145/3539618.3591861.
- [14] K. Ethayarajh, How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings, 2019. URL: <https://arxiv.org/abs/1909.00512>. arXiv:1909.00512.
- [15] C. Xiao, Y. Li, G. T. Hudson, C. Lin, N. A. Moubayed, Length is a curse and a blessing for document-level semantics, 2023. URL: <https://arxiv.org/abs/2310.16193>. arXiv:2310.16193.
- [16] N. Rossia, J. Lin, F. Liu, Z. Yang, T. Lee, A. Magnani, C. Liao, Relevance filtering for embedding-based retrieval, in: Proceedings of CIKM2024, ACM, 2024.