VisRel: Media Search at Scale

Fedor Borisyuk, Siddarth Malreddy, Jun Mei, Yiqun Liu, Xiaoyi Liu
Piyush Maheshwari, Anthony Bell, Kaushik Rangadurai
Facebook Inc.

ABSTRACT

In this paper, we present *VisRel*, a deployed large-scale media search system that leverages text understanding, media understanding, and multimodal technologies to deliver a modern multimedia search experience. We share our insight on developing image and video understanding models for content retrieval, training efficient and effective media-to-query relevance models, and refining online and offline metrics to measure the success of one of the largest media search databases in the industry. We summarize our learnings gathered from hundreds of A/B test experiments and describe the most effective technical approaches. The techniques presented in this work have contributed 34% (abs.) improvement to media-to-query relevance and 10% improvement to user engagement. We believe that this work can provide practical solutions and insights for engineers who are interested in applying media understanding technologies to empower multimedia search systems that operate at Facebook scale.

CCS CONCEPTS

• **Computing methodologies** → **Image representations**; • **Information systems** → **Multimedia and multimodal retrieval**; **Image search**; **Video search**.

KEYWORDS

Image search, video search, media search, image classification, multi-modal learning

ACM Reference Format:
Fedor Borisyuk, Siddarth Malreddy, Jun Mei, Yiqun Liu, Xiaoyi Liu, Piyush Maheshwari, Anthony Bell, Kaushik Rangadurai. 2021. VisRel: Media Search at Scale. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467081

1 INTRODUCTION

Facebook connects billions of people around the world. Not only do users express themselves through posts and comments, they also come to the social platform to search and explore. Common use cases include (1) connecting with friends and communities, (2) learning about a specific topic of interest, (3) getting the latest updates on public figures and celebrities, (4) gaining a deeper perspective about a news event, (5) discovering inspirational memes, and (6) celebrating special occasions with family and friends. As photos and videos are among the most ubiquitous content types, we have developed a search surface specialized for media content to better fulfill users' diverse information needs (Fig. 1) [31].

In developing a multimedia search engine, we have encountered a number of challenges:

- *Noisy text description:* the text body of a post may not have a strong topical connection with the accompanying media content
- *Media to text representations:* relevance models need to match free-text queries with content features that are represented in different semantic spaces
- *Variety of media types:* since photos and videos can be mixed together in the search results, relevance and ranking models need to be capable of evaluating all media types with comparable performance
- *Large document corpus:* the search index comprises trillions of media documents from many popular Facebook features [27]; the search engine must be capable of sifting through large amount of data for relevant results at scale
- *Multilingual support:* our technologies need to work consistently and reliably for many languages

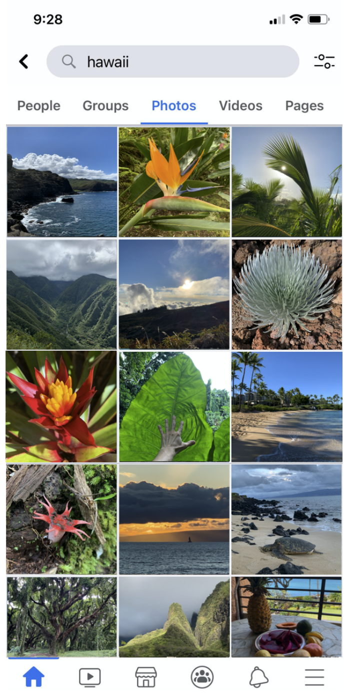

Figure 1: Example of Media Search in the Facebook app with free-text query "hawaii".

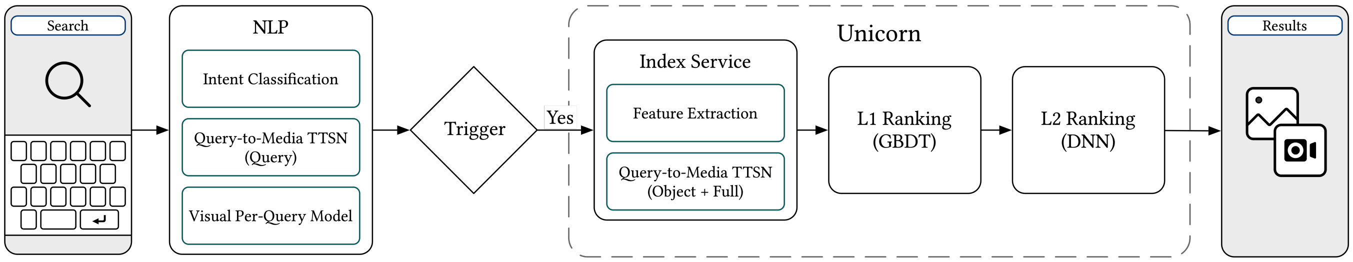

Figure 2: Sequence of information processing steps in *VisRel*: an NLP service extracts query-side features; a triggering module resolves what type of content to retrieve; an index service returns an initial set of candidates (thousands); multiple ranking stages are used to select the most relevant and engaging results (hundreds).

Solving these challenges entails a deep understanding of media content and free-text queries. We herein share our solutions for modeling media and query representations to improve search quality, methods to match query to media content representations, and techniques to interpret and handle a variety of search intents for many languages. As the *VisRel* system is deployed to a very large index of over 10^{12} entries [27], we also present the system architecture and discuss some optimization steps to help the search engine operate at scale. The contributions of this paper are highlighted below:

Practical solutions for one of the largest search systems in the industry. This work not only presents the system architecture (Fig. 2) and the building blocks for a large-scale media search system (§ 3) but also proffers practical suggestions on how each component can be tuned to improve the overall efficiency and quality of the search engine.

Experience with metrics. We share hands-on experience in working with online and offline metrics to evaluate the performance of relevance models and the output quality of the search engine. Furthermore, we offer examples wherein some metrics may fail (§ 3.1) and describe techniques to improve the alignment between online and offline metrics (§ 3.6).

Scalable content understanding solutions for media search. *VisRel* leverages representation learning on photos and videos, multimodal modeling of textual and visual features, and free-text query understanding to render a compelling search experience. We share our experience in adapting modern computer vision technologies to information retrieval tasks (§ 4) and techniques to improve query-to-media relevance in search ranking models by aligning semantic representations from different domains (§ 5).

Experience in online experiments. Over the course of *VisRel* development, we have conducted hundreds of online A/B tests to refine and iterate on each building block in the search system. The suite of solutions discussed in this work has collectively improved query-to-media relevance rate by 34% absolute (§ 3.5). We share the results of online experiments in § 4 and § 5.

2 RELATED WORK

Media search is a powerful tool for people to learn and explore any topic of interest on the internet. Web search engines such as Google, Bing, Baidu, Yandex, and Pinterest offer the ability to search publicly accessible media content, whereas specialized search engines such as Google photos and Apple photos provide search capabilities for private image collections. By contrast, Facebook search is designed to retrieve both (i) publicly accessible posts and (ii) posts with limited visibility—e.g., friends only. To help people find and explore any content of interest on Facebook, search products are built upon a robust and scalable system called *Unicorn* [10]. With billions of active users on the social platform, our search index comprises over 10^{12} documents [27]. In this work, we share our experience in developing practical solutions to empower efficient and effective media content search with free-text queries.

Visual recognition technologies have garnered industry-wide attention in recent years [2, 14, 28–30, 33]. In particular, we are interested in adapting weakly-supervised fine-grained image recognition to search use cases. Joulin *et al.* [18] studied weakly-supervised learning on the Flickr 100M dataset, and they demonstrated that even in the absence of hand-labeled data, convolutional neural networks (CNN) can still learn good visual features and perform well in downstream tasks. Zhai and Wu [32] trained image classifiers over millions of images with user annotated data from Pinterest pin-boards and produced high quality binary embeddings using a softmax classification approach to deep metric learning. By contrast, we fine-tune the CNN from Mahajan *et al.* [23] on search logs with a large label space. We share the data preparation and model fine-tuning steps in § 4.2.

The utility of matching search query to document by their semantic representations has been demonstrated in recent adaption of representation learning for information retrieval tasks [15, 24]. Nowadays technology is moving towards incorporating advanced deep learning approaches to produce visual and textual representations that are more closely aligned in semantic space. Chen *et al.* [5] and Lu *et al.* [22] explored building complex models that achieved state of the art performance in matching natural language and visual domains. However, these models are not practical in production settings due to significant computation costs that can incur during media upload (e.g., finding bounding boxes of objects of interest) and high latency arising from late fusion of visual and textual representations. Guo *et al.* [13] used BERT-based models to encode query and document to the same semantic space and explored precomputing of document embeddings to reduce computation costs. However, their work did not explore multimodal problems. In § 5, we share our experience in applying multimodal techniques to improve query to media content matching at Facebook scale.

3 SEARCH ARCHITECTURE AND METRICS

This section first describes how we measure and evaluate changes to our search system in offline and online experiments (§ 3.1). It is followed by discussions on the four core components in *VisRel*—query preprocessing (§ 3.2), triggering (§ 3.3), retrieval (§ 3.4), and ranking (§ 3.5). Lastly, we present techniques to refine and improve the accuracy of offline evaluations with respect to online performance (§ 3.6).

3.1 Evaluation Metrics

3.1.1 Online Metrics. *User satisfaction* is an important factor that determines the success of a consumer application. User feedback surveys can be used to gauge satisfaction. However, to study an incremental modeling improvement, tens of thousands of surveys must be collected to ascertain if an observed difference between a pair of A/B test cohorts has statistical significance. Evaluation by user surveys is thus unsuitable for our day-to-day workflow due to the laborious feedback collection process and potential exposure to participation bias.

For online experiments, *user retention* may be used as a proxy for user satisfaction because it is reasonable to assume that people are much likelier to return if they have had a positive search experience in the recent past. A user is considered retained if there is search activity today and there was also a prior activity within a trailing window of l days—L7, L14, L28, etc. In practice, we find that retention metrics are often insensitive to incremental modeling changes. Consequently, we pay close attention to behavioral metrics such as *time spent*, *impressions*, *click rate*, *meaningful clicks*, and *good click rate* to gain insights into our experiments (Table 1).

Limitations of time spent. When a user wants to explore a general topic, our search engine may return thousands of media results that satisfy the query. Time spent is a meaningful application metric because one criterion of success in content discovery is the ability to continuously stimulate the user's appetite for information connected to the topic of interest. However, if ranking models are trained to optimize solely for time spent, we have found that the resulting experience favors media featuring memes and spiritual quotes. In addition, the top results become dominated by videos. We conclude that this produces a suboptimal search experience upon confirmation of retention decline in online experiments.

Table 1: Evaluation metrics for online experiments

Metric	Measurement
Time Spent	Duration of a search session
Impressions	Number of media shown per search session
Click Rate	Number of user actions per search session
Meaningful Clicks	User actions that are considered of greater application value, e.g., post shares, comments, reactions, follows, etc.
Good Click Rate	Ratio of search sessions with significant time spent or meaningful clicks to all search sessions

Table 2: Guidelines for evaluating media-to-query relevance

Score	Explanation
3	Media has a strong topical connection with the likeliest interpretation of the query
2	Media has a reasonable topical connection with a possible interpretation of the query
1	Media is vaguely connected to the query, but not the salient subject or segment
0	No meaningful topical connection between media and query

Limitations of click rate. Result clicks are commonly considered a favorable measure for search success. However, clicks can be biased by extrinsic factors unrelated to media retrieval and ranking. In one experiment, we introduced a feature to automatically adjust the application viewport toward the salient objects detected in an image. Accordingly, those objects would be prominently shown in any cropped frame regardless of the original photo's aspect ratio. However, click rate declined by 3% in this experiment, and search time spent unexpectedly increased. With cropped thumbnails, we hypothesized that poor framing forces users to click on the thumbnails more frequently to open the full image view. By contrast, when the main subjects are prominently shown in a cropped frame, it no longer entails forced actions. The increase in time spent supported our view that users experienced less friction with the application. This example shows all behavioral metrics must be evaluated together to determine whether the proposed change delivers a better user experience. Consequently, we have designed a metric called *meaningful clicks* to encapsulate user actions of greater application value. In addition, we combine time spent and meaningful clicks to a composite metric called *good clicks* to help evaluate the overall experience.

3.1.2 Offline Metrics. We routinely use *ROC AUC*, *normalized entropy* [26], and *mean average precision* (mAP) to compare model quality in retrieval and ranking development. *Precision@K* and *Recall@K* [28, 32] are adapted to evaluate embeddings quality. Once model development is finalized in offline settings, we enlist the help of human raters to evaluate the correctness and quality of model predictions. Query data are de-identified and aggregated to form a validation query set, and only publicly visible photos and videos are enqueued for rating. Table 2 lists a subset of rating guidelines for media-to-query relevance. Multiple levels of granularity are necessary to derive meaningful *DCG@K* scores [16] so that search results can be compared holistically. We also keep track of *Off-Topic@K*, which measures the average fraction of validation queries where at least one irrelevant media is found among the first K search results to the query. In practice, we only proceed to start online A/B tests after determining that the candidate models perform at least on a par with the baseline with respect to DCG@K and Off-Topic@K.

Limitations of human ratings. As the human raters are instructed to assess media-to-query relevance only, the evaluation results are generally interpretable, and they give insight into the performance of candidate models. We have found that DCG score tends to positively correlate with online metrics for queries that refer to sufficiently specific topics—e.g., "underwater wedding", "origami butterfly", "round dining table". Conversely, if the topicality is overly generic—"morning", "sunset", "puppies", etc.—a high DCG score does not automatically imply that the top K media are universally appealing to end users. Thus, a comprehensive review of search quality must examine both offline and online experiment results.

3.2 Query Preprocessing

When the system receives a search query, the query text is forwarded to a *natural language processing* (NLP) service to extract query-side features. Classical techniques are applied to perform language-specific word segmentation and spelling correction. Various interest-based classifiers are run to infer query intent as a map of $\langle \text{topic} \rightarrow \text{probability} \rangle$. The topics include celebrity, commerce, person name, sports, etc. For media search, we have trained a query-to-media two-tower model and deployed the query-side network to the NLP service (§ 5.1). In addition, the NLP service provides lookup for visual per-query models (§ 5.2).

To provide similar search experience to billions of users around the world, we have built solutions that work reliably for many languages. Historically in the industry text representations were separately trained for each language. However, such approaches have high maintenance costs because it would require training separate downstream models for each use case and for each language. By contrast, we have adopted multilingual fastText [8] embeddings wherein representations for different words are aligned to the same semantic space. We have also explored BERT-based representations such as XLM-R [7], where one unified model is trained for over a hundred languages. We share results of the XLM-R based model in a separate paper.

3.3 Triggering

Photos and videos are only a subset of the types of results we show to users when they search on Facebook. Since we cannot know the intent of the user using just the query itself, we use other information such a user preferences and past engagement to rank each type of result. For a query like "cricket live", videos would be the top results shown to users while for a query like "new year party", events would be the top results. For certain queries, media results have such a low rank that the user never sees it, so the resources spent in retrieving and ranking those media results were wasted.

As an optimization, we use a combination of rules and a triggering model to predict for which queries we should call the media backend. The rules are based on thresholds of probabilities from query intent computed in the NLP service. We perform offline analysis to tune to the thresholds to maximize triggering queries users interacted with in the past. We also train a wide & deep neural network [6] model using click data to predict if we should trigger given a particular query. We used the query intents, user attributes, and historical user interactions as features in the model. The model helps the trigger logic to keep up with the changes in user query trends. We found that using this triggering logic helps to keep the capacity requirements lower while maintaining high engagement in the results we do show.

3.4 Retrieval

During retrieval we need to collect a subset of media from the entire corpus in the search index based on the user's search query. To facilitate this collection, we store several terms for each media in the inverted index. The terms are derived from many sources like text descriptions of the media, the author of the media, or terms inferred from the media itself. We match these inverted index terms with the user's query and assign a score based on how many terms match and which terms match. We then pass the top N documents to the ranking stage where N is tune-able based on the available compute capacity. We use an intuitive strategy for score assignment. For example, we assign a higher score if we match multiple terms for a particular media or if we match bi-grams in addition to uni-grams in the text description. To improve recall, we also use predicted content based terms for matching like SURU labels, media tags described in § 4.

Mixing photos and videos together. In certain scenarios we show photos and videos together in a unified media search results page. In such cases we need to find the balance between different types of media. From running A/B experiments with mixed search results pages we found that if we promote only photos, or only videos, then it degraded user retention. To solve the issue, we aimed to have similar signals and technologies to augment both videos and photos. For example, content understanding features needed to be present for both photos and videos, therefore we apply techniques like *tagging* in § 4.1 for both types of media. Additionally, we found that optimization of online metrics play an important role, and optimize for good click rate, which helps to find the balance between different types of medias on the search result page.

3.5 Ranking

VisRel executes media ranking in two stages. L1 ranking is first applied to an initial set of about 10^3 candidates that are retrieved by the index service, and the top results ($\sim$100) are forwarded to L2 ranking to construct the final search experience (Fig. 2). In this section, we describe the data preparation and training steps for each ranking stage.

Data. We train our models with human-labeled data as well as user engagement data. Query and document features are computed and logged to file system as *Unicorn* completes the search request. We collect training data from aggregated and de-identified search queries. For human-labeled data, an in-house replay system enqueues these sampled queries for publicly visibly content and creates pairs of $\langle \text{query}, \text{photo/video} \rangle$ for evaluation. The raters then score each pair in accord with the relevance guidelines in Table 2. The replay system is also used to refresh query and document features of previously rated $\langle \text{query}, \text{photo/video} \rangle$ pairs by applying model inference and feature extraction with the latest system configuration.

We log user engagement data such as *clicks*, *meaningful clicks*, and *good clicks* as defined in Table 1. Among these engagement categories, *good clicks* offer the best query-to-media relevance for a given action, followed by *meaningful clicks* and *clicks* in decreasing correlation strength. On the other hand, we have also observed that *clicks* are more abundant than *meaningful clicks* and *good clicks*. Thus, we note that signal quality varies inversely with quantity.

L1 Ranking. We use user engagement data to train a gradient boosted decision tree (GBDT) model using list-wise LambdaMART loss [4]. We adjust the training examples' weights based on their clicks and sessions, such that those with higher quality can have higher weights so as to drive the model to find the balance between engagement and quality. We firstly assign equal weight to each training example. For those ⟨query, media⟩ pairs that have received fewer than N clicks, we multiply their weights by $1 - \text{sigmoid}(\alpha \cdot \gamma_{\text{no\_click}})$, where α is a constant, and $\gamma_{\text{no\_click}}$ is the ratio of media sessions that receive no click in the past K days. We use query intent probabilities, media embeddings, user attributes as features in the GBDT model.

We also experimented with using the human rated data to train the GBDT. But because that data focuses heavily on the relevance between query text and media, the model tends to neglect engagement features that reflect more on users' interest. This leads to an overall loss in user engagement while the human rated evaluation metrics remain neutral.

L2 Ranking. We use user engagement data to train a wide & deep neural network model [6] as a second stage ranker using LambdaRank loss. In addition to all the features used in the GBDT model, we also use the query ngrams, media understanding ngrams as sparse features. The model is first trained on click data and then fine-tuned on human rated data. The human rated data helps balance the relevance vs engagement trade-off. We further fine-tune the model using labels and weights that reflect users' downstream actions—like, comment, reshare, etc.—and retention (§ 3.1.1).

3.6 Improvements in offline measurements

To reduce potential selection and positional bias in training and evaluation datasets, we have explored offline replay with randomized search results [20] and with reweighted historical examples [21]. Randomization may be implemented by shuffling the top K results. Alternatively, a small perturbation can be introduced to the ranking score (s) such that $s' = (1 - \alpha) \cdot s + \alpha \cdot \text{rand}(s)$, where α controls the strength of randomization effect. In practice, randomization is applied only to a small portion of user traffic. This is colloquially referred to as the *randomized bucket*. The decision to activate randomization is determined by a combination of user, timestamp, query, and media document. We design this procedure to be deterministic and idempotent so that it can be emulated in subsequent data processes. Although one notable drawback of randomization is that some users are exposed to a suboptimal experience, evaluation data collected from the randomized bucket can impart a stronger confidence in the online performance of candidate ranking models when they are evaluated only with traditional offline metrics such as AUC and DCG.

Reweighting of historical examples is another technique to mitigate potential bias in evaluation data. As introduced in § 3.1.1, click rate is a well-established metric to measure search success. Given a set of search sessions ($\mathcal{S}$), average click rate (CR) is computed by

$$\mathsf{CR} = \frac{1}{|\mathcal{S}|} \sum_{\langle \mathcal{X},\mathcal{D},\mathcal{C} \rangle_i \in \mathcal{S}} \sum_{d_{i,j} \in \mathcal{D}_i} \mathcal{I}\left(d_{i,j} \in \mathcal{C}_i\right) \tag{1}$$

Table 3: RMSE for traditional metrics on randomized and non-randomized data, and *log ratio* click model on non-randomized data. Measured over 44 A/B online experiments.

Randomization	ROC AUC	NDCG@3	Log Ratio
Non-randomized data	2.63	2.85	0.57
Randomized data	1.70	4.62	n.a.

where the $\langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle_i$ tuple denotes the search context ($\mathcal{X}$), retrieved documents ($\mathcal{D}$), and clicked documents ($\mathcal{C}$) in search session i; and $\mathcal{I}$ is an indicator function that evaluates to 1 when the condition is met or 0 otherwise. Instead of using only the ratio of ranking scores for sample reweighting [21], we propose a modification using logarithm to compute an estimated click rate ($\widehat{\mathsf{CR}}$):

$$\widehat{\mathsf{CR}} = \frac{1}{|\mathcal{S}|} \sum_{\langle \mathcal{X},\mathcal{D},\mathcal{C} \rangle_i \in \mathcal{S}} \sum_{d_{i,j} \in \mathcal{D}_i} \mathcal{I}\left(d_i \in \mathcal{C}_i\right) \times \text{sign}\left(\frac{s_h\left(d_i \mid \mathcal{X}_i\right)}{s_\pi\left(d_i \mid \mathcal{X}_i\right)}\right) \times \log\left(\min\left\{\left|\frac{s_h\left(d_i \mid \mathcal{X}_i\right)}{s_\pi\left(d_i \mid \mathcal{X}_i\right)}\right|, M\right\}\right) \tag{2}$$

where $s(d_i \mid \mathcal{X}_i)$ denotes the ranking score for document d_i given search context $\mathcal{X}_i$, π stands for the production ranking model, h stands for the candidate model, and M is a hyperparameter to tune the stability of offline evaluation. We normalize all scores within each search interaction session to be in range [0,1] by min-max normalization. Because we experimented with the same model type (e.g. neural network) simple score normalization within each session worked well. From practice we observed that normalization based on Isotonic regression [19] could be alternative model score normalization approach when model types are different, where each model scores are calibrated to [0,1] range.

We use *root-mean-square error* (RMSE) to measure the accuracy of the offline evaluation framework relative to online performance. In this work, RMSE is defined as the standard deviation of the residuals between the relative change to an offline metric (e.g. $\Delta\text{AUC}_{\pi \to h}/\text{AUC}_\pi$) and the relative change to an online metric (e.g. $\Delta\widehat{\mathsf{CR}}_{\pi \to h}/\widehat{\mathsf{CR}}_\pi$) as measured in A/B tests. In general, the smaller the error is, the more strongly the offline metrics correlate with online ones. Table 3 shows that ROC AUC becomes more effective in randomized data replay. However, offline metric based on sample reweighting with Eq. 2 has shown to even more strongly correlate with online metrics. Our intuition is that it is helpful to apply logarithm to score ratio s_h/s_π because this yields a higher penalty in cases where the new model is less confident of the clicked media, thereby penalizing candidate models that are likely to reduce performance.

4 TECHNIQUES IN MEDIA UNDERSTANDING

Even though media contains rich semantic information, most of media search is driven by keyword-based retrieval. However about 65% of media in our search index do not have text accompanying them. This causes a problem when ranking because we are essentially only working with around 35% of the data we have. To tackle this problem, we use several techniques which aim to understand the content in media and use it for retrieval and ranking.

We use a variety of heuristic, weakly-supervised, and unsupervised techniques to improve retrieval and ranking. Table 4 summarizes the metric improvements in relevance and online engagement due to media understanding. Each subsection describes a respective approach and its resulting metrics impact.

4.1 Media tagging with text

We have observed that queries with exploratory intent are often short and generic—e.g., "nails", "tattoo ideas", "interior design"—whereas many high-quality media do not contain such phrases in the textual description. A retrieval system that relies solely on query-to-description matching may be confronted with search quality issues as a result of sub-optimal recall. Thus, we apply keyword tagging to annotate media with popular and relevant search queries to improve recall of high-quality content.

The main steps of the tagging workflow are illustrated in Figure 3. First, we prepare a pool of seed media (~300M). An embedding is computed based on the media content. In addition, we collect the most frequent queries that have been used to retrieve a given media in each supported language. Second, for the target media, we apply *k-nearest neighbors* search to find the most similar seed media based on the content embeddings. Third, we infer a primary language for the target media and narrow the list of keywords to those matching the target language. Candidate keywords are weighted according to the embedding distance and engagement history:

$$w_{\star}(t) = \sum_{i=0}^{k-1} \frac{\log(1 + f_{i,t})}{\text{dist}(e_{\star}, e_i)^{\alpha} + \beta} \log \frac{N}{n_t} \tag{3}$$

This aggregation scheme is derived from tf-idf formulation [25], where

- t: the candidate keyword
- $f_{i,t}$: the number of engagement events on the ith seed media with candidate keyword t
- $\text{dist}(e_{\star}, e_i)$: a distance function for the embeddings of the target ($\star$) and the ith seed media
- N: the total number of seed media in the dataset
- n_t: the total number of seed media that contain t in their respective keyword lists
- k, α, β: hyperparameters to be tuned in offline evaluation

Finally, we assign suggested keywords that are above a predefined threshold to the target media, thereby enabling retrieval beyond direct keyword-to-description matching. Hyperparameters k, α, and β are tuned by holding out a subset of seed media and evaluating the quality of keyword suggestions for the test media. Parameter values are selected to maximize the average Recall@K [28, 32] over the evaluation instances.

We have built separate tagging datasets for photos and videos and deployed them to a large-scale retrieval system. The datasets are refreshed daily to incorporate new keywords. The tagging process is applied near real time at time of media upload to Facebook, so that tagging information is available in the search index within seconds. We ran an A/B test in retrieval by increasing retrieval score of media when predicted keywords were matched, and observed a **4.4%** improvement in DCG, **15.6%** reduction in off-topic rate, **1.68%** improvement in click rate, **1.54%** improvement in good click rate and **0.27%** improvement in time spent (Table 4).

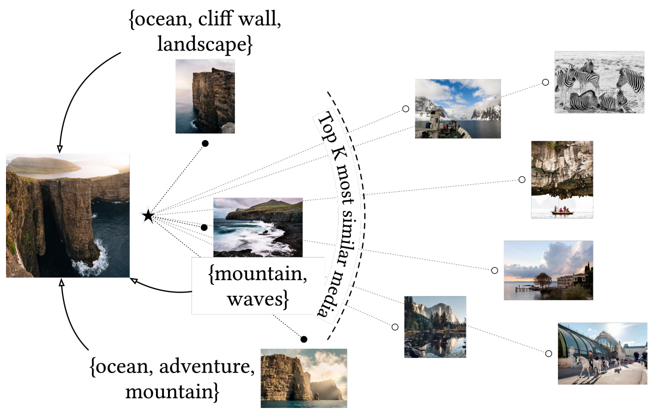

Figure 3: A hypothetical example to illustrate the 3 main steps in media tagging with text: (1) construct a data set with candidate keywords for each seed media; (2) given the target media (denoted by $\star$), find the top-K similar seed media; (3) aggregate candidate keywords

.

4.2 SURU – weakly-supervised media classifier

We have found that image embeddings from pre-trained models based on ImageNet do not adapt well to media search use cases due to significant domain shift between Facebook photos and the ImageNet dataset. To improve embeddings quality, we applied the weakly-supervised approach proposed by Tang *et al.* [28] and fine-tuned a CNN [23] using historical photo search logs. We call the resulting model *SURU*.

Data Preparation. We frame the problem as a multi-class classification task with weakly-supervised learning. Labels are derived from historical search logs in which a result click is considered a positive signal indicating relevance between the result and the search query. However, as photo results are sometimes shown along with other types of content, the absence of user action does not necessarily imply that the photo is irrelevant to the query. Furthermore, a particular photo can be retrieved by multiple search queries. Thus, we collect ⟨photo, list of queries⟩ pairs and apply weakly-supervised approach to classify photo to queries. We aggreagte search logs over multiple months to minimize seasonality bias in the data distribution. We remove sensitive and person name queries and enforce that each query should appear in the dataset at least 500 times to stabilize training. In addition, we apply deduplication using image embeddings extracted from a ResNeXt trained on hashtags [23]. The resulting dataset consisted of 10M data points and 43K labels.

Training Details. We fine-tune the last block of a ResNeXt model which was trained on 3.5B photos over 17K hashtags [23]. The model is trained using asynchronous stochastic gradient descent on 32 GPUs across 4 machines with each GPU processing of 48 photos at a time. We trained for 20 epochs. We used all the techniques described in [28] to find the best performing variant to launch to production.

Deployment. When a user uploads a media to Facebook, we run our CNN on the media and output a quantized 256-dimensional

Table 4: Summary of metric gains from each component of *VisRel*. "-" imply that the gain was not statistically significant.

Modelling Technique	Human Rated Metric Gain		Online Metric Gain		
	DCG@6	Offtopic Rate@6	Click Rate	Good Click Rate	Time Spent
SURU	+10%	-	+0.6%	+0.4%	+0.19%
Clustering	-	-	+0.82%	+0.97%	+1.1%
Two-Tower Model	+5.1%	-6.4%	+3.8%	+3.59%	+0.6%
Per Query Model	+10.85%	-13.38%	+3.76%	+0.95%	-
Media Tagging	+4.4%	-15.6%	+1.68%	+1.54%	+0.27%
OCR	+3.8%	-	+2.32%	+0.66%	+0.95%
Actual & Predicted Engagement	-	-	+3.8%	+1.88%	+0.41%

binary vector. The vector is quantized to save storage space (vide [2] for details on quantization). We also store the top 10 predicted queries for the photo along with their predicted probabilities.

Retrieval and Ranking Experiments. In the retrieval system we used the images' predicted queries as terms in the inverted index of the retrieval system. We ran an A/B test using the SURU term and observed a **0.6%** improvement in photo click rate, a **0.4%** improvement in photo good click rate and a **0.16%** improvement in photo time spent. In the ranking system we used the 256-dimensional binary vector as a ranking feature in our second stage wide & deep neural network ranking model. We ran an A/B test using this model and observed a **10%** improvement in DCG@6 (Table 4). We also experimented with using the SURU embedding in the first stage model as well, but it did not lead to statistically significant improvement in metrics.

4.3 Media clustering features

The document corpus of our search index comprises a wide spectrum of interest topics. We hypothesized that categorical data augmentation might be beneficial to ranking models. However, it is impractical to manually label each media document because of the sheer data volume. A more effective approach is k-means clustering. As a ranking feature, it is not necessary to know beforehand what each cluster represents as long as we can identify the cluster for a given media.

We collected over 6.4B engaged media from search logs. Each was represented by a 256-dimension embedding extracted from CNNs described in [12, 23]. To accommodate this large data size, we applied hierarchical clustering with Faiss k-means [17]:

(1) Distribute all the embeddings to 2000 machines which independently cluster them into 1900 clusters
(2) Collect all $2000 \times 1900 = 3.8$M cluster centers and cluster these centers into 8000 parent clusters
(3) Distribute the embeddings in each parent cluster into its own machine and further cluster them to a maximum of 3000 child clusters. Enforce that each child cluster have a minimum support of 850 embeddings to ensure sufficient popularity for each cluster.
(4) Obtain a cluster index containing ~1.6M clusters

The entire process completed in ~24 hrs end-to-end. The hyper-parameter values were chosen to ensure the embeddings fit in the machines we used. We experimented with different number of child clusters and found that 1.6M has good cluster quality (measured using the ratio of mean embedding to centroid distance and mean centroid to centroid distance).

For each media in Facebook, we assign the top 5 closest cluster ids. We use these ids as sparse features in our ranking model and train a new second stage wide & deep model. We ran an A/B test using this wide & deep neural network model and observed a **1.1%** improvement in time spent, **0.82%** improvement in click rate and **0.97%** improvement in good click rate (Table 4).

4.4 Actual and predicted engagement features

Since we define system quality by user interaction metrics such as click rate and good click rate (see § 3.1), we want to use historical interaction rates as features to our ranking model. We compute two classes of features for each media document: (1) actual engagement features and (2) predicted engagement features.

Actual engagement features are ratios of historical engagement rates of media: click rate, like rate, save rate, impression rate. In order to reduce noise, we compute these metrics every day averaged over the previous 7 days. Impression rate is defined as the number of times a media was in the application viewport normalized by a constant. Like rate is defined as the number of times a media was liked divided by the impression rate. Save rate and click rate are defined similarly.

One issue with engagement features is that they can be unreliable for fresh media (i.e. cold start problem). To mitigate this problem, we calculate *predicted engagement features*: predicted click rate, predicted like rate, predicted save rate, and predicted impression rate. To calculate predicted engagement we prepared a large pool of seed media (~100M) which has been engaged during the week. We used a different pool of seed media for photos and videos because we expect different user engagement behavior for them. For each seed media in the pool we compute its embedding and engagement features as defined above. When the new target media is deployed we extract its embedding and apply a weighted *k-nearest neighbors* search to find top-K similar seed media, and then take a weighted average of seed media engagement counters to calculate predicted engagement counter. Weights can be defined as the similarity between the target and seed media embedding. We used actual engagement features in the wide & deep neural network model (defined in § 3.5) and observed a **1.1%** improvement in good click rate (Table 4) and an additional **0.7%** improvement in good click rate when using both actual engagement and predicted engagement features.

4.5 Handling text on media - OCR

In search engines for media, special attention needs to be given to media that contains text. Common types of media containing text are memes and quotes. In such media, the picture itself only provides background or emotional support for the textual message on the media.

To handle such media we have developed an in-house optical character recognition system (OCR) called *Rosetta* [3]. *Rosetta* uses a set of detection and recognition modules; it extracts all the word boxes from the photo and then applies a CNN that translates crops of words into character sequences. We used OCR-extracted text to match to query text in retrieval. Furthermore, we computed traditional query-to-text matching features [1] and sparse word token features based on OCR-extracted text, and integrated them into our ranking models (§ 3.5), which improved search quality measured by human raters DCG@6 by **+3.8%**, click rate by **+2.32%**, good click rate by **+0.66%**, and media time spent by **+0.95%** (Table 4).

5 QUERY-TO-MEDIA MATCHING MODELS

To further improve the ranking quality and relevance, we develop the following two features to more accurately capture the similarity between query text and media.

5.1 Two-Tower Model

We adopt the two-tower architecture to model the similarity between query text and media (Fig. 4). The query-tower features include token/char n-grams and query text. For token/char n-grams we convert them using one-hot encoding, pass through randomly initialized embedding matrix and sum them up as the final embedding for the whole query text. For query text we apply the commonly used NLP embedding method (e.g., fastText, BERT) to convert the query text into a fixed-length vector, and then add another layer of MLP before fusing with query embedding. For contextualized embeddings (e.g., BERT), we relax the last few layers for fine-tuning. We have used multi-lingual query representation embeddings both for fastText [8] or BERT based such XLM-R [7].

The media-tower features include media embeddings trained via weakly supervision (i.e., SURU in § 4.2) followed by another layer of MLP, and media text (e.g., title, description) and OCR features, where we process them using the same models as the query tower.

We simply concatenate embeddings on each tower as the fusion method and calculate their cosine similarity as the similarity score. We collect queries and their clicked medias as the positive samples[1], and use in-batch samples or global pool sampling as negatives. We use margin rank loss for back propagation, as we find it exhibits consistently gains over the traditional cross-entropy in offline evaluations.

We re-train our media search 1st stage and 2nd stage rankers using the two-tower model similarity score as a feature, and increases their AUC by **+1.8%** and **+5.4%** respectively. Human rated evaluations show DCG@6 increases by **+5.1%** and the off-topic rate reduces by **-6.4%**. Online A/B tests show the media clicks increase by **+3.8%**, good clicks increase by **+3.59%** and the time-spent in media search increases by **+0.6%** (Table 4).

[1]We filter the training data by the number of clicks of (query, media) >= 2. This removes the random noisy clicks and improves AUC of 2nd stage ranker with the two-tower similarity score by over **2%**. We also experimented other thresholds for clicks, and we found 2 is the best option that can keep enough volume for the training data while remove the noisiest clicks for good data quality.

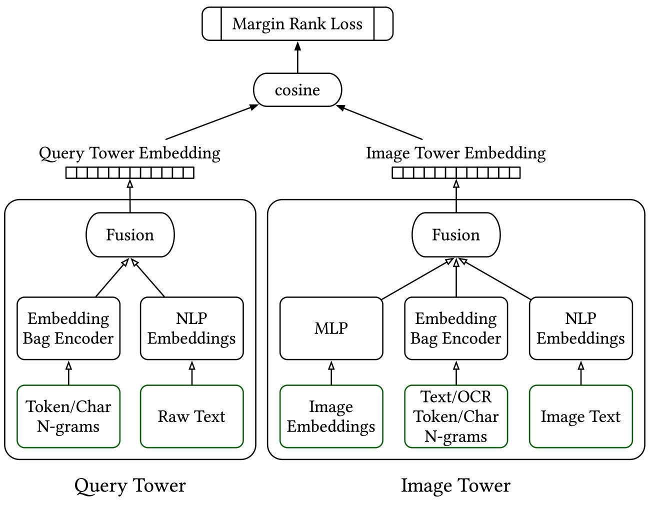

Figure 4: Two-tower model for query-to-media matching

.

5.2 Per-Query Model

We train a single two-tower model with both tail and head queries in the data to minimize the average loss. It is friendly to tail queries and provides good generality, but it also sacrifices some specialty for head queries on the other hand.

To make up for such loss, we propose to train one specific model for each head query to find its most common representation/pattern. We choose linear-SVM [9] as it is lightweight enough to run for millions of queries simultaneously and able to update/refresh frequently. We collect queries that have been searched by at least N times in the past T days and treat clicked medias as positive samples. For each positive sample, we randomly sample K media within the same batch as negative ones. We found that number of negatives play an important role to make the models effective, we use 5x more negatives during training. Therefore, for each head query s, we train linear-SVM using M clicked positive samples and $M \times K$ negatives ones with the media embedding $\boldsymbol{v}$ trained by weakly supervision as the features (i.e., SURU in § 4.2).[2]

We train the per-query model for millions of head queries on a daily/semi-weekly basis, and filter the results based on whether the evaluation results exceed the per-defined thresholds (e.g., the precision on evaluation sets ≥95%). We observed slight shift in model scores over the week due to retraining, however we did not observe significant influence on downstream models, where per query model scores are used as features. Once the training is done, we upload the linear model coefficients $\boldsymbol{p}$ and the intercept t to an online real-time lookup service with the query string as the key (§ 3.2). At serving time, we check whether the query string s has the per-query model in the lookup service. If so, we calculate the

[2]For those head queries that might have too many clicked medias (e.g., when M is too large), we further down-sample the clicked media to balance the memory constraints, training time cost and model quality.

per-query model score as $\boldsymbol{p}^T \boldsymbol{v} + t$, and use it as a feature in the ranking model.

We can interpret the per-query model coefficients as the centroid of clicked media clusters for the query, and the score as how similar this media is to other clicked media. To compare with embeddings from the two-tower model above, we run embedding-based *KNN* using FAISS [17] on head queries, and calculate Recall@k using clicked media as the golden dataset. This offline evaluation results show **>8%** absolute increase in Recall@100 and **>19%** in Recall@1000. In second stage ranker, the per-query model score also shows up as the second most important feature. Human rated evaluations show it increases DCG@30 by **+10.85%**, and reduces the off-topic rate by **-13.38%**. Online A/B test shows a **+3.76%** increase in click rate and **+0.95%** increase in good click rate.

Overall, we think two-tower and per-query complements each other and using them together can achieve both good quality and coverage in both head and tail queries.

6 CONCLUSION

We presented approaches for building a large scale media search system called *VisRel*. The *VisRel* system is deployed to production and processes search traffic at Facebook scale. In this paper we shared practical ideas for handling large weakly-supervised training data, training media classifiers for search, provided scalable solutions for query-to-media matching, and shared experience building and improving media search over large databases confirmed with online A/B test experiments. Our success depended on the metrics definitions and how metrics interact with each other, we gave information on metrics we used and trade-offs of different approaches.

This paper showcases the state of the art for large scale media search. We envision that in the future, significant improvements in search quality will be provided by innovations in semantic understanding of media, text and authors. Therefore, in future iterations we could investigate using advanced multi-modal networks in a variety of search components from retrieval with embedding-based retrieval to media search engines based on query to media representations [11] and pushing more of the multi-modal signals into other parts of the system such as media tagging. Additionally, we believe that in the near future the search ecosystem will have to deal more with longer queries and that is why more sophisticated query to document understanding models will be necessary.

7 ACKNOWLEDGEMENTS

The authors would like to thank Yinzhe Yu, Matan Levi, Varun Nasery, Priyanka Kukreja, Yina Tang, Anirudh Sudarshan, Yumeng Tao, Mike Su, Haochen Chen, Sergey Kirshner, Christina Scheau, and others who collaborated with us.

REFERENCES

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 2011. *Modern Information Retrieval: The Concepts and Technology behind Search* (2nd ed.). Addison-Wesley Publishing Company, USA.

[2] Sean Bell, Yiqun Liu, Sami Alsheikh, Yina Tang, Edward Pizzi, M. Henning, Karun Singh, Omkar Parkhi, and Fedor Borisyuk. 2020. GrokNet: Unified Computer Vision Model Trunk and Embeddings For Commerce. In *KDD*.

[3] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. 2018. Rosetta: Large Scale System for Text Detection and Recognition in Images. In *KDD*.

[4] Chris J.C. Burges. 2010. *From RankNet to LambdaRank to LambdaMART: An Overview.* Technical Report MSR-TR-2010-82.

[5] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *ECCV*.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *RecSys*.

[7] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *ACL*.

[8] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word Translation Without Parallel Data. *arXiv preprint arXiv:1710.04087* (2017).

[9] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Mach. Learn.* 20, 3 (Sept. 1995), 273–297.

[10] Michael Curtiss, Iain Becker, Tudor Bosman, Sergey Doroshenko, Lucian Grijincu, Tom Jackson, Sandhya Kunnatur, Soren Lassen, Philip Pronin, Sriram Sankar, Guanghao Shen, Gintaras Woss, Chao Yang, and Ning Zhang. 2013. Unicorn: A System for Searching the Social Graph. In *VLDB*.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *ACL*.

[12] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. SlowFast Networks for Video Recognition. In *ICCV*.

[13] Weiwei Guo, Xiaowei Liu, Sida Wang, Huiji Gao, Ananth Sankar, Zimeng Yang, Qi Guo, Liang Zhang, Bo Long, Bee-Chung Chen, and Deepak Agarwal. 2020. DeText: A Deep Text Ranking Framework with BERT. In *CIKM*.

[14] Houdong Hu, Yan Wang, Linjun Yang, Pavel Komlev, Li Huang, Xi (Stephen) Chen, Jiapei Huang, Ye Wu, Meenaz Merchant, and Arun Sacheti. 2018. Web-Scale Responsive Visual Search at Bing. In *KDD*.

[15] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. 2020. Embedding-Based Retrieval in Facebook Search. In *KDD*.

[16] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated Gain-Based Evaluation of IR Techniques. *ACM Trans. Inf. Syst.* (2002).

[17] J. Johnson, M. Douze, and H. Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* (2019).

[18] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. 2016. Learning visual features from large weakly supervised data. In *ECCV*.

[19] S. Kullback and R. A. Leibler. 1951. On Information and Sufficiency. *The Annals of Mathematical Statistics* (1951), 79–86.

[20] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. 2011. An Unbiased Offline Evaluation of Contextual Bandit Algorithms with Generalized Linear Models. In *OTEAE*.

[21] Shuai Li, Yasin Abbasi-Yadkori, Branislav Kveton, S. Muthukrishnan, Vishwa Vinay, and Zheng Wen. 2018. Offline Evaluation of Ranking Policies with Click Models. In *KDD*.

[22] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*.

[23] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. Exploring the Limits of Weakly Supervised Pretraining. In *ECCV*.

[24] Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Hongbo Deng, and Zhenzhong Chen. 2020. A Dual Heterogeneous Graph Attention Network to Improve Long-Tail Performance for Shop Search in E-Commerce. In *KDD*.

[25] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24, 5 (1988).

[26] Claude. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal* 27, 3 (July 1948), 379–423.

[27] Tom Stocky. 2015. Search FYI: Find What the World is Saying With Facebook Search. https://about.fb.com/news/2015/10/search-fyi-find-what-the-world-is-saying-with-facebook-search/

[28] Yina Tang, Fedor Borisyuk, Siddarth Malreddy, Yixuan Li, Yiqun Liu, and Sergey Kirshner. 2019. MSURU: Large Scale E-Commerce Image Classification with Weakly Supervised Search Data. In *KDD*.

[29] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. 2017. Sampling Matters in Deep Embedding Learning. In *ICCV*.

[30] Fan Yang, Ajinkya Kale, Yury Bubnov, Leon Stein, Qiaosong Wang, Hadi Kiapour, and Robinson Piramuthu. 2017. Visual Search at EBay. In *KDD*.

[31] Shengqi Yang, Cristina Scheau, and Fei Yang. 2017. *Under the hood: Photo Search.* Facebook Inc. Retrieved December 28, 2020 from https://engineering.fb.com/2017/05/22/ml-applications/under-the-hood-photo-search/

[32] Andrew Zhai and Hao-Yu Wu. 2019. Making Classification Competitive for Deep Metric Learning. In *BMVC*.

[33] Andrew Zhai, Hao-Yu Wu, Eric Tzeng, Dong Huk Park, and Charles Rosenberg. 2019. Learning a Unified Embedding for Visual Search at Pinterest. In *KDD*.