# Similarity Search for Efficient Active Learning and Search of Rare Concepts

**Cody Coleman**[1]*, **Edward Chou**[2], **Sean Culatana**[2], **Peter Bailis**[1], **Alexander C. Berg**[3], **Roshan Sumbaly**[2], **Matei Zaharia**[1], **I. Zeki Yalniz**[2]
[1]Stanford University, [2]Facebook AI, [3]Facebook AI Research

## Abstract

Many active learning and search approaches are intractable for industrial settings with billions of unlabeled examples. Existing approaches search globally for the optimal examples to label, scaling linearly or even quadratically with the unlabeled data. However, in practice, data is often heavily skewed; only a small fraction of collected data will be relevant for a given learning task. For example, when identifying rare classes, detecting malicious content, or debugging model performance, positive examples can appear in less than 1% of the data. In this work, we exploit this skew in large training datasets to reduce the number of unlabeled examples considered in each selection round by only looking at the nearest neighbors to the labeled examples. Empirically, we observe that learned representations can effectively cluster unseen concepts, making active learning very effective and substantially reducing the number of viable unlabeled examples. We evaluate several active learning and search techniques in this setting on two large-scale datasets: ImageNet and OpenImages. For rare classes, active learning methods need as little as 0.31% of the labeled data to match the average precision of full supervision. By limiting active learning methods to only consider the immediate neighbors of the labeled data as candidates for labeling, we need only process as little as 1% of the unlabeled data while achieving similar reductions in labeling costs as the traditional global approach. This process of expanding the candidate pool with the nearest neighbors of the labeled set can be done efficiently and reduces the computational complexity of selection by orders of magnitude.

## 1 Introduction

Large-scale unlabeled datasets contain millions or billions of examples spread over a wide variety of underlying concepts [6, 30, 29, 26, 20, 16, 25, 1, 4, 17]. Often, these massive datasets skew towards a relatively small number of common concepts, such as cats, dogs, and people. Rare concepts, such as harbor seals, may only appear in a handful of examples. However, in many settings, performance on these rare concepts is critical [3, 26, 13, 10, 14]. For example, harmful or malicious content may comprise a small percentage of user-generated content, but it can have an outsize impact on the overall user experience [26]. Similarly, when debugging model behavior for safety-critical applications like autonomous vehicles or dealing with representational biases in models, obtaining data that captures rare concepts allows modelers to combat blind spots in model performance [13, 10, 3, 14]. Even a simple prediction task like stop sign detection can be challenging given the diversity of real-world data. Stop signs may appear in a variety of conditions (e.g., on a wall or held by a person), be heavily occluded, or have modifiers (e.g., "Except Right Turns") [14]. While large-scale datasets are core to addressing these issues, finding the relevant examples for these long-tail tasks is challenging.

*Correspondence: cody@cs.stanford.edu

NeurIPS 2020 Workshop on Dataset Curation and Security

Active learning has the potential to automate the process of identifying these rare, high value data points significantly, but existing methods become intractable at this scale. Specifically, the goal of active learning is to reduce the cost of labeling [23]. To this end, the learning algorithm is allowed to choose which data to label based on uncertainty (e.g., the entropy of predicted class probabilities) or other heuristics [22, 23, 18]. Active search is a sub-area focused on finding positive examples in skewed distributions [8]. Because of a concentrated focus on labeling costs, existing techniques, such as uncertainty sampling [18] or information density [24], perform multiple selection rounds and iterate over the entire unlabeled data to identify the optimal example or batch of examples to label and scale linearly or even quadratically with the size of the unlabeled data. Computational efficiency is becoming an impediment as the size of datasets and model complexities have increased [2]. Recent work has tried to address this problem with sophisticated methods to select larger and more diverse batches of examples in each selection round and reduce the total number of rounds needed to reach the target labeling budget [21, 15, 7, 19, 11]. Nevertheless, these approaches still scan over all of the examples in each selection round and can be intractable for large-scale unlabeled datasets.

In this work, we propose Similarity search for Efficient Active Learning and Search (SEALS) to restrict the candidates considered in each selection round and vastly reduce the computational complexity of active learning and search methods. Empirically, we find that learned representations from pre-trained models effectively cluster many unseen and rare concepts. We exploit this latent structure to improve the computational efficiency of active learning and search methods by only considering the nearest neighbors of the currently labeled examples in each selection round. Finding the nearest neighbors for each labeled example in unlabeled data can be performed efficiently with sublinear retrieval times [5] and sub-second latency on billion-scale datasets [12] for approximate approaches. While constructing the index for similarity search requires at least a linear pass over the unlabeled data, this computational cost is effectively amortized over many selection rounds or other applications. As a result, our SEALS approach enables selection to scale with the size of the labeled data rather than the size of the unlabeled data, making active learning and search tractable.

We empirically evaluated SEALS for both active learning and search on two large-scale computer vision datasets: ImageNet [20] and OpenImages [16]. We selected 603 concepts spread across these datasets that range in prevalence from 0.203% to 0.002% (1 in 50,000) of the training examples. We evaluated three selection strategies for each concept: max entropy uncertainty sampling (MaxEnt) [18], information density (ID) [24], and most-likely positive (MLP) [11]. Across datasets, selection strategies, and the vast majority of concepts, SEALS achieved similar average precision (AP) and nearly the same recall of the positive examples as the baseline approaches, while reducing the number of examples considered and the computational complexity by orders of magnitude (Figure 1).

## 2 Methods

**Active learning** is an iterative process that begins with a large pool of unlabeled data $U = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Each example is sampled from the space $\mathcal{X}$ with an unknown label from the label space $\mathcal{Y} = \{1, \ldots, C\}$ as $(\mathbf{x}_i, y_i)$. We additionally assume a feature extraction function $G_z$ to embed each $\mathbf{x}_i$ as a latent variable $G_z(\mathbf{x}_i) = \mathbf{z}_i$ and that the $C$ concepts are unequally distributed. Specifically, there are one or more valuable rare concepts $R \subset C$ that appear in less than 1% of the unlabeled data. For simplicity, we frame this as $|R|$ binary classification problems solved independently rather than 1 multi-class classification problem with $|R|$ concepts. Initially, each rare concept has a small number of positive examples and several negative examples that serve as a labeled seed set $L_r^0$. The goal of active learning is to take this seed set and select up to a budget of $T$ examples to label that produces a model $A_r^T$ that achieves low error. For each round $t$ in pool-based active learning, the most informative examples are selected according to the selection strategy $\phi$ from a pool of candidate examples $\mathcal{P}_r$ in batches of size $b$ and labeled, as shown in Algorithm 1.

**Active search** is closely related, so much of the formalism carries over. The critical difference is that rather than selecting examples to label that minimize error, the goal is to maximize the number of examples from the target concept $r$, expressed with the utility function $u(L_r) = \sum_{(\mathbf{x},y)\in L_r} \mathbb{1}\{y = r\}$). Different selection strategies are favored, but the overall algorithm is the same as Algorithm 1.

In this paper, we consider three selection strategies: **max entropy (MaxEnt)** uncertainty sampling [18] and **information density (ID)** [24] for active learning and **most-likely positive (MLP)** [28, 27, 11] for active search. Because active learning and search are similar, we evalu-

ate all the strategies in terms of both the model's error and the number of positive examples. While MaxEnt and MLP only require a linear pass over the data, ID scales quadratically because it weights the informativeness of each example by its similarity to all other examples.

**Algorithm 1** BASELINE APPROACH

**Input:** unlabeled data $U$, labeled seed set $L_r^0$, feature extractor $G_z$, selection strategy $\phi(\cdot)$, batch size $b$, labeling budget $T$

$\mathcal{L}_r = \{(G_z(\mathbf{x}), y) \mid (\mathbf{x}, y) \in L_r^0\}$
$\mathcal{P}_r = \{G_z(\mathbf{x}) \mid \mathbf{x} \in U \text{ and } (\mathbf{x}, \cdot) \notin L_r^0\}$
**repeat**
  $A_r = \text{train}(\mathcal{L}_r)$
  **for** 1 to $b$ **do**
    $\mathbf{z}^* = \arg\max_{\mathbf{z} \in \mathcal{P}_r} \phi(\mathbf{z})$
    $\mathcal{L}_r = \mathcal{L}_r \cup \{(\mathbf{z}^*, \text{label}(\mathbf{x}^*))\}$
    $\mathcal{P}_r = \mathcal{P}_r - \mathbf{z}^*$
  **end for**
**until** $|\mathcal{L}_r| = T$

**Algorithm 2** SEALS APPROACH

**Input:** unlabeled data $U$, labeled seed set $L_r^0$, feature extractor $G_z$, selection strategy $\phi(\cdot)$, batch size $b$, labeling budget $T$, $k$-nearest neighbors implementation $\mathcal{N}(\cdot, \cdot)$

$\mathcal{L}_r = \{(G_z(\mathbf{x}), y) \mid (\mathbf{x}, y) \in L_r^0\}$
$\mathcal{P}_r = \cup_{(\mathbf{z}, y) \in \mathcal{L}_r} \mathcal{N}(\mathbf{z}, k)$
**repeat**
  $A_r = \text{train}(\mathcal{L}_r)$
  **for** 1 to $b$ **do**
    $\mathbf{z}^* = \arg\max_{\mathbf{z} \in \mathcal{P}_r} \phi(\mathbf{z})$
    $\mathcal{L}_r = \mathcal{L}_r \cup \{(\mathbf{z}^*, \text{label}(\mathbf{x}^*))\}$
    $\mathcal{P}_r = \mathcal{P}_r \cup \mathcal{N}(\mathbf{z}^*, k) - \mathbf{z}^*$
  **end for**
**until** $|\mathcal{L}_r| = T$

**Similarity search for efficient active learning and search (SEALS)** accelerates the inner loop of active learning and search by restricting the candidate pool of unlabeled examples $\mathcal{P}_r$. To apply SEALS, we use an efficient method for similarity search of the embedded examples [5, 12] and make two modifications to the baseline approach, as shown in Algorithm 2: 1) $\mathcal{P}_r$ is restricted to the nearest neighbors of the labeled examples, and 2) after every example is selected, we find its $k$ nearest neighbors and update $\mathcal{P}_r$. Restricting the candidate pool $\mathcal{P}_r$ to the $k$-nearest neighbors of the labeled examples means we only apply the selection strategy to at most $k|L_r|$ examples. This can be done transparently for many strategies making it applicable to a wide range of active learning and search methods, even beyond the ones considered here. Finding the $k$ nearest neighbors for each newly labeled example adds overhead, but this can be calculated efficiently with sublinear retrieval times [5, 12] for approximate approaches. As a result, the computational complexity of each round scales with the size of the labeled data rather than the unlabeled data. Generating the embeddings and indexing the data can be expensive and slow, requiring at least a linear pass over the unlabeled data. However, this cost is effectively amortized over many rounds, concepts, or other applications.

# 3 Experiments

## 3.1 Implementation

Because we are interested in rare concepts, we kept the number of initial positive examples small, using only 5 positive examples for each concept. Negative examples were randomly selected at a ratio of 19 negative examples to every positive example to form the seed set $L_r^0$. The slightly higher number of negatives in the initial seed set improved average precision on the validation set across datasets. The batch size $b$ for each selection round was 100, and the budget $T$ was 2,000 examples. For each dataset, we split the data, selected concepts, and created embeddings as detailed below.

**ImageNet** [20] has 1.28 million training images spread almost equally over 1000-classes. To simulate rare concepts, we split the data in half, using 500 classes to train the feature extractor $G_z$ and treating the other 500 classes as unseen concepts. For $G_z$, we used ResNet-50 [9] but added a bottleneck layer before the final output to reduce the dimension of the embeddings to 256. We kept all of the other training hyperparameters the same as in He et al. [9]. We extracted features from the bottleneck layer and applied $l^2$ normalization. In total, the 500 unseen concepts had 639,906 training examples that served as the unlabeled pool. We used 50 concepts for validation, leaving the remaining 450 concepts for our final experiments. The number of examples for each concept varied slightly, ranging from 0.114-0.203% of the unlabeled pool. The validation images were treated as the test set.

**OpenImages** [16] has 7.34 million images with human-verified labels spread over 19,958 classes, taken as an unbiased sample from Flickr. However, only 6.82 million images were still available

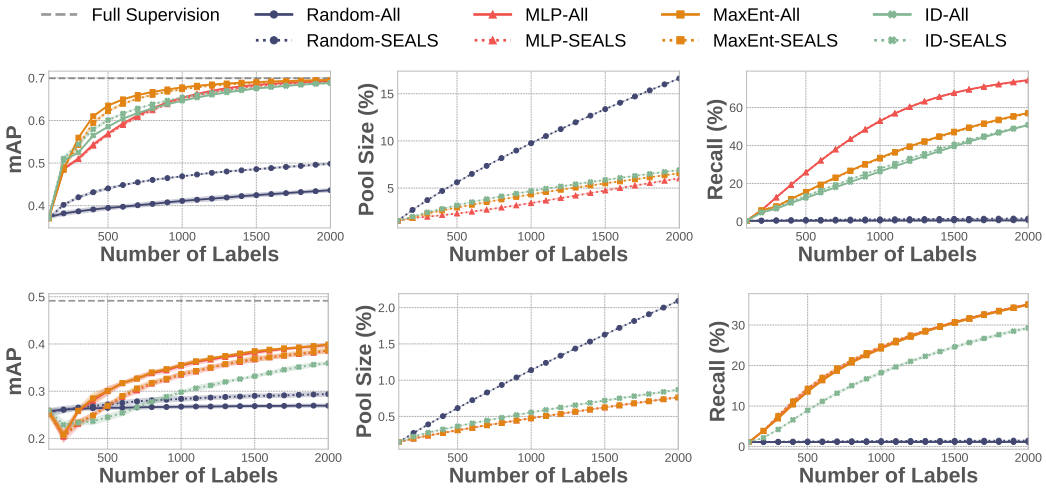


Figure 1: Active learning and search on ImageNet (top) and OpenImages (bottom). Across datasets and strategies, SEALS with $k = 100$ performed similarly to the baseline approach in terms of both the error the model achieved for active learning (left) and the recall of positive examples for active search (right), while only considering a fraction of the data $U$ (middle).

in the training set at the time of writing. As a feature extractor, we took ResNet-50 pre-trained on all of ImageNet. As rare concepts, we randomly selected 200 classes with between 100 to 6,817 positive training examples. We reviewed the selected classes and removed 47 classes that overlapped with ImageNet. The remaining classes appeared in 0.002-0.088% of the data. We used the same hyperparameters as the ImageNet experiments and the predefined test split for evaluation.

### 3.2 Results

Across selection strategies, datasets, and concepts, SEALS performed similarly to the baseline while only considering a fraction of the unlabeled data $U$ in the candidate pool $\mathcal{P}_r$, as shown in Figure 1.

**ImageNet.** For active learning, all baseline and SEALS approaches ($k = 100$) were within 0.011 mAP of the 0.699 mAP achieved with full supervision. In contrast, random sampling (Random-All) only achieved 0.436 mAP. MLP-All, MaxEnt-All, and ID-All achieved mAPs of 0.693, 0.695, and 0.688, respectively, while the SEALS equivalents were all within 0.001 mAP at 0.692, 0.695, and 0.688 respectively. The reduced skew from the nearest neighbor expansion of the initial seed set only accounted for a small part of the improvement, as demonstrated by Random-SEALS.

MLP-All and MLP-SEALS significantly outperformed all of the other selection strategies for active search. Both approaches recalled over 74% of the positive examples for each concept at 74.5% and 74.2% recall, respectively. MaxEnt-All and MaxEnt-SEALS had a similar gap of 0.4%, labeling 57.2% and 56.8% of positive examples, while ID-All and ID-SEALS were even closer with a gap of only 0.1% (50.8% vs. 50.9%). In comparison, Random-SEALS and Random-All performed poorly.

**OpenImages.** For active learning, the gap between the baseline approaches and SEALS widened slightly for OpenImages. At 2,000 labels per concept (~0.029% of $|U|$), MaxEnt-All and MLP-All achieved 0.399 and 0.398 mAP, respectively, while MaxEnt-SEALS and MLP-SEALS both achieved 0.386 mAP. Increasing $k$ to 1,000 significantly narrowed this gap for MaxEnt-SEALS and MLP-SEALS, improving mAP to 0.395. Moreover, SEALS made information density tractable on OpenImages by reducing the candidate pool to 1% of the unlabeled data, whereas ID-All ran for four days in wall-clock time without completing a single selection round.

For active search, the gap between the baseline approaches and SEALS was even closer on OpenImages despite considering a much smaller fraction of the overall unlabeled pool. MLP-All, MLP-SEALS, MaxEnt-SEALS, and MaxEnt-All were all within 0.1% with ~35% recall at 2,000 labels per concept. ID-SEALS had a lower recall of 29.3% but scaled nearly as well as the linear approaches.

## References


[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016.

[2] Dario Amodei and Danny Hernandez. Ai and compute, May 2018. URL `https://blog.openai.com/ai-and-compute/`.

[3] Khalid Ashmawy, Shouheng Yi, and Alex Chao. Searchable ground truth: Querying uncommon scenarios in self-driving car development. `https://eng.uber.com/searchable-ground-truth-atg/`, 10 2019.

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[5] Moses S Charikar. Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, 2002.

[6] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google, 2013. URL `http://arxiv.org/abs/1312.3005`.

[7] Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJg2b0VYDr`.

[8] Roman Garnett, Yamuna Krishnamurthy, Xuehan Xiong, Jeff Schneider, and Richard Mann. Bayesian optimal active search and surveying. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ICML'12, page 843–850, Madison, WI, USA, 2012. Omnipress. ISBN 9781450312851.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2016.

[10] Kenneth Holstein, Jennifer Wortman Vaughan, Hal Daumé III, Miro Dudik, and Hanna Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2019.

[11] Shali Jiang, Gustavo Malkomes, Matthew Abbott, Benjamin Moseley, and Roman Garnett. Efficient nonmyopic batch active search. In *Advances in Neural Information Processing Systems*, pages 1099–1109, 2018.

[12] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[13] Andrej Karpathy. Train ai 2018 - building the software 2.0 stack, 2018. URL `https://vimeo.com/272696002`.

[14] Andrej Karpathy. Ai for full-self driving, 2020. URL `https://youtu.be/hx7BXih7zx8`.

[15] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7024–7035, 2019.

[16] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[17] Kevin Lee, Vijay Rao, and William Christie Arnold. Accelerating facebook's infrastructure with application-specific hardware. https://engineering.fb.com/data-center-engineering/accelerating-infrastructure/, 3 2019.

[18] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. Springer-Verlag New York, Inc., 1994.

[19] Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and José Miguel Hernández-Lobato. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems*, pages 6356–6367, 2019.

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[21] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

[22] Burr Settles. From theories to queries: Active learning in practice. In Isabelle Guyon, Gavin Cawley, Gideon Dror, Vincent Lemaire, and Alexander Statnikov, editors, *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, volume 16 of *Proceedings of Machine Learning Research*, pages 1–18, Sardinia, Italy, 16 May 2011. PMLR. URL http://proceedings.mlr.press/v16/settles11a.html.

[23] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[24] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, page 1070–1079, USA, 2008. Association for Computational Linguistics.

[25] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m. *Communications of the ACM*, 59(2):64–73, Jan 2016. ISSN 1557-7317. doi: 10.1145/2812802. URL http://dx.doi.org/10.1145/2812802.

[26] Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian J. McAuley. Fine-grained spoiler detection from large-scale review corpora. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2605–2610. Association for Computational Linguistics, 2019. doi: 10.18653/v1/p19-1248. URL https://doi.org/10.18653/v1/p19-1248.

[27] Manfred K Warmuth, Jun Liao, Gunnar Rätsch, Michael Mathieson, Santosh Putta, and Christian Lemmen. Active learning with support vector machines in the drug discovery process. *Journal of chemical information and computer sciences*, 43(2):667–673, 2003.

[28] Manfred KK Warmuth, Gunnar Rätsch, Michael Mathieson, Jun Liao, and Christian Lemmen. Active learning in the drug discovery process. In *Advances in Neural information processing systems*, pages 1449–1456, 2002.

[29] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[30] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.