“Find Me a Dataset”: Scientific Dataset Recommendation from Method Descriptions

Anonymous ACL submission

Abstract

Much of modern science relies on public datasets to develop research ideas. Finding a dataset for a given task can be difficult, particularly for new researchers. We aim to improve the process of dataset discovery by introducing a system called *DatasetFinder* which recommends relevant datasets given a short natural language description of a research idea. For the new task of dataset recommendation, we construct an English-language dataset that leverages existing annotations and compare several ranking models on this dataset. We also compare our proposed models against existing commercial search engines and find evidence that leveraging natural language descriptions improves search relevance. To encourage development on this new task, we release our constructed dataset and models to the public.[1]

1 Introduction

“Data is food for AI.” (Ng, 2021)

Innovation in modern artificial intelligence (AI) research depends on the dual workhorses of methods and data. The revolution of neural network models in computer vision (Krizhevsky et al., 2012) was enabled by the ImageNet Large Scale Visual Recognition Challenge (Deng et al., 2009). Similarly, data-driven models for syntactic parsing saw rapid development after adopting the Penn Treebank (Marcus et al., 1993; Palmer and Xue, 2010).

In research using machine learning, the data collection stage of the scientific process (Crawford and Stucki, 1990) involves selecting a benchmark dataset. There are hundreds of datasets published every year in AI (shown in Figure 1) and knowing which datasets to use for a given research idea can be difficult (Paullada et al., 2021). This problem is greater for new researchers who are not intimately familiar with a subfield.

[1]Code and data: https://anonymous.4open.science/r/dataset-recommendation-75D1/

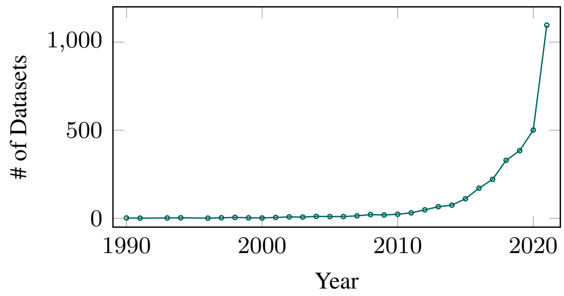

Figure 1: # of AI datasets released each year from 1990 to 2021, according to Papers with Code[2]

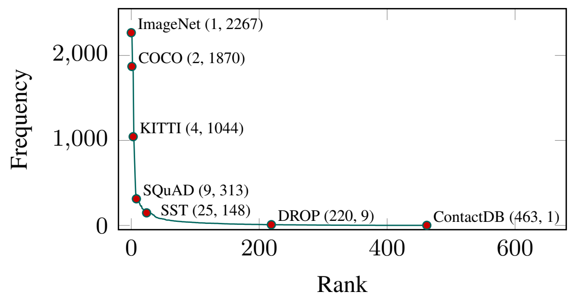

Figure 2: Usage counts of datasets in our training set show a long tail of rarely-used datasets. The rank and frequency for some example datasets are marked.

Consequently, researchers in AI typically focus their efforts on a small number of datasets they are already familiar with. The awareness of high-quality data for a task leads to an increase in published research on the task, which in turn raises awareness even further for that dataset. For illustration, for a large set of 17.5K papers obtained from the S2ORC corpus (Lo et al., 2020) (methodology details given in Section 2.4.2), we plot the frequency of datasets used in Figure 2. The dataset counts appear to follow a Zipfian distribution (Newman, 2004), with the vast majority of datasets occurring in the tail of the distribution. This “rich get richer” effect has the result of narrowing the scope of methodological development to methods that are applicable to these datasets.

In this paper, we consider that the scientific method may be improved in AI research if researchers could more easily find datasets for a given research question. Our goal is to recommend datasets by *relevance* rather than *popularity*.

Taking a step towards this goal, we introduce the task of "**dataset recommendation**": given a short *description* of an AI research idea, recommend datasets for building or testing such an idea. We show a concrete example in Figure 3. We introduce a strong baseline system, which we call *DatasetFinder*, as a step towards solving this task.

Dataset search has been studied extensively (Chapman et al., 2019) and dataset recommendation has been studied using either a set of relevant papers (Altaf et al., 2019) or an initial set of known relevant datasets (Ben Ellefi et al., 2016) as input. This is the first attempt at a natural language interface for dataset recommendation.

To operationalize this task, we first build a dataset to measure how well we can recommend datasets for a given description. As a proxy for natural method descriptions, we leverage segments from *paper abstracts* to describe a researcher's information need. We then identify the exact datasets used in a given paper, either through heuristic matching (for our large training set) or by using existing human annotations (for our small test set).

We then frame this task as a retrieval problem (Manning et al., 2005), by treating the system description as a *query* and the set of known datasets as a *search corpus*. We use standard ranking metrics such as mean reciprocal rank (Radev et al., 2002) to measure performance and also measure how well we can recommend datasets that are rare but relevant to a user.

For this ranking problem, we consider several approaches: BM25 (Robertson and Zaragoza, 2009), nearest neighbor retrieval and dense retrieval with neural bi-encoder (Karpukhin et al., 2020). Compared with the currently available keyword-centric dataset search engines, we find that our approach that leverages natural language description is far more effective at finding relevant datasets. We also show that our baseline leaves significant room for improvement, which we believe makes this an appealing task for the research community.

2 Task and Dataset

2.1 Task

We establish a new task for automatically recommending relevant datasets for a description of an

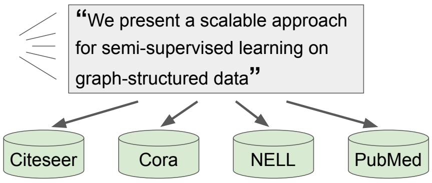

Figure 3: Example research idea with relevant datasets

AI system. Given a natural language method description q and a set of datasets D, retrieve the most relevant subset $R \subset D$ one could use to test the idea described in q. Figure 3 illustrates this with a real example which has been condensed for clarity. The query is a brief summary of a paper by Kipf and Welling (2017) and the relevant datasets shown are the actual datasets used in their study. We leverage this data to illustrate patterns in how the AI research community uses datasets.

In contrast to prior work, our input is a *method description* described briefly in natural language. We hypothesize that by defining the query as a textual description, the system is more user-friendly and will lead to better search results, compared to using a small set of keywords. In Section 5.1, we offer evidence to support this hypothesis.

To support this task, we construct a dataset consisting of (q, R) pairs extracted from published English-language scientific proceedings. Each query q in our dataset is a simulated method description constructed from published scientific abstracts and R is the set of relevant datasets used by the authors. We describe an automatic method for creating this data, summarized in Figure 4. For our test set, we leverage a small human-annotated dataset to maintain high data quality. To obtain enough training data for modern deep retrieval models, we generate training data from unlabeled papers, using information in the body of each paper for supervision. We release our data under a permissive Apache 2.0 License.

2.2 Search Corpus

Our first step in approaching this as a search problem is to construct a collection of datasets. We search against the full set of datasets listed on "Papers with Code",[3] a large public index of papers, which includes metadata for over 5000 datasets and benchmarks.[4] For most datasets, Papers with

[3] www.paperswithcode.com

[4] Not all items in our search corpus are datasets, strictly speaking. For example, the MuJoCo simulator is not a dataset but is widely used as a benchmark in reinforcement learning

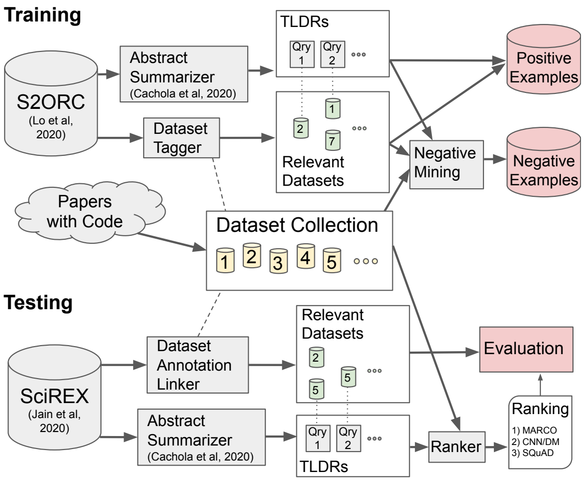

Figure 4: Architecture of the DatasetFinder system. "Qry" denotes "query".

Code stores a short human-written description and a list of different names used to refer to the dataset (known as "variants"). Many datasets are also tagged with the paper that introduced that dataset. We store this data for later processing.

2.3 Test Set Construction

2.3.1 Raw Data

Our test set is generated from a human-annotated set of AI papers, SciREX (Jain et al., 2020). SciREX is a dataset of 438 full-text papers from major AI venues whose intended use was document-level information extraction.

2.3.2 Queries

To construct simulated method descriptions from published papers, we extract the abstract from the paper then automatically summarize the abstract.

We summarize each paper's abstract using the *TLDR* system (Cachola et al., 2020). TLDR can generate very brief summaries of scientific documents. Given a scientific abstract, this model trains BART (Lewis et al., 2020) to generate both a short human-generated summary and a paper title.[5] We use the generated summaries of scientific abstracts as "method descriptions" to simulate queries for our retrieval system. Examples of generated TLDRs are shown in the "Ideas" in Figure 9.

Many of these queries did not describe the intended experiment sufficient clarity to recommend a dataset. Consider the example "We equip CNNs with a more principled pooling strategy, 'spatial pyramid pooling', to eliminate the above requirement". This query suggests a general methodological contribution, that could apply to almost any AI task, though the true label here was "Pascal VOC 2007" (Everingham et al., 2009). Our annotator[6] manually reviewed the generated natural language method descriptions in our test set. For any cases that were sufficiently ambigiuous that a trained annotator could not make an educated guess of the datasets used in the paper, we removed that example from our test set.[7]

For 17 instances in our test set, the generated TLDR explicitly mentioned one of the paper's relevant datasets. In these cases, we masked out the spans containing the dataset name with the token `[DATASET]`, to avoid label leakage.

2.3.3 Relevant Datasets

For each paper, SciREX contains annotations for mentions of all "salient" datasets, defined as datasets that "take part in the results of the article" (Jain et al., 2020). For each salient dataset in a paper, spans of all mentions of that dataset throughout the paper are provided. To link these annotations with the datasets in our search corpus, we first collect the set of mention strings used to refer to each dataset in a paper. We then check if any of these mention names matches one of the dataset variants from Papers with Code. Finally, each match was manually inspected (and corrected, if necessary) by the same annotator to ensure accurate linking.

2.4 Training Set Construction

2.4.1 Raw Data and Queries

We generate training data by automatically tagging full-text papers from S2ORC (Lo et al., 2020), a corpus of scientific papers. We use *TLDR* to summarize each abstract, to extract a short "query", in the same manner as we do for the test set (§2.3).

2.4.2 Relevant Datasets

Our training set is automatically labeled using the body text corresponding to a given abstract. We apply a rule-based procedure to identify the dataset used in a given paper. For each paper, we tag all datasets that satisfy two conditions: the paper must

research (Todorov et al., 2012)

[5]We use a maximum word length of 90 and BART length penalty parameter of 1.5 to generate detailed summaries.

[6]The annotator was one of the authors of this paper, a graduate student studying natural language processing with previous experience in vision, robotics, and ML research.

[7]Out of 402 SciREX-based method descriptions, we discarded 78 descriptions due to excessive ambiguity.

cite the paper that introduces the dataset, and the paper must mention the dataset by name twice.[8]

This tagging procedure is restrictive, and emphasizes precision over recall. Nonetheless, using this procedure, we tag 17,495 papers from S2ORC with at least one dataset from our search corpus.

To estimate the quality of these tagged labels, the annotator manually examined 200 tagged paper-dataset pairs. Each pair was labeled as correct if the paper authors would have realistically had to download the dataset in order to write the paper. 92.5% (185/200) of dataset tags were deemed correct.

2.5 Limitations

Our dataset construction methodology suffers from three key limitations:

Recency bias The ages of papers used to generate method descriptions in our train and test sets are skewed towards the present. The median years of papers in our train and test set are 2018 and 2017, respectively. This is in part because our datasets come from Papers with Code, which may not include historic datasets no longer popular today. Moreover, the rate of publication in AI has been growing rapidly in recent years (Dean, 2020).

Popular dataset bias in the test set Our test set is derived from the SciREX corpus (Jain et al., 2020). This corpus is biased towards popular works: we found the median number of citations of a paper in SciREX to be 129, compared to 19 for any computer science paper in S2ORC. Our test set method descriptions are therefore more likely to describe mainstream ideas in popular subields of AI.

Automatic tagging Our training data is generated automatically using a list of canonical dataset names from Papers with Code. This tagger will mislabel papers where a dataset is used but never referred to by one of these canonical names (e.g. non-standard abbreviations or capitalizations).

2.6 Dataset Analysis

Using this set of paper-dataset tags, what can we learn about how researchers use datasets?

2.6.1 Rank-frequency distribution of datasets

In Figure 2, we plot the frequency that each dataset is tagged in a paper in our training set. We see a distribution with a dramatic long tail. Though our data collection procedure considered all papers that use AI datasets, the most frequent datasets belong to the computer vision community. This is due to both the large volume of computer vision publications relative to other fields of AI and the popularity of computer vision datasets as benchmarks for core machine learning research.

2.6.2 Popular datasets by domain

How do different communities of AI interact with datasets in their research? We define "communities" within AI by the venues that researchers publish in.[9] We analyze the most popular datasets in each community, measuring the percentage of papers that use each dataset in NLP, Vision, Robotics, and Machine Learning in Figure 5.

The distribution of dataset usage in the NLP community is closest to uniform, suggesting a relatively broad set of datasets in use. In contrast, nearly half of the papers tagged in the robotics community use the KITTI dataset (Geiger et al., 2013), among all papers that use some publicly available dataset.

2.6.3 How old are datasets used?

In Figure 6, we show the distribution of relative ages of datasets used. We observe that the majority of datasets used are within the previous 5 years, but there is a significant long tail of older datasets.

2.6.4 Most popular datasets by year

To understand dataset trends over time, we plot the most popular computer vision datasets in 2009, 2014, and 2019 in Figure 7. We observe significantly more data from 2019 than 2014 or 2009 for reasons described in Section 2.5.

2.6.5 Dataset counts per paper

In Figure 8, we see that our training set tags associates queries with a single dataset more frequently than our test set does. This is due to our rule-based tagging scheme, which emphasizes precise labels over recall.

3 DatasetFinder

We formulate dataset recommendation as a ranking task. Given a method description q and a search corpus of datasets D, rank the datasets $d \in D$ based on

[8] We apply the additional requirement that the counted dataset mentions must occur in a section with section title containing "results", "experiment", "evaluation", "result", "training", or "testing", to avoid non-salient dataset mentions, such as those commonly occurring in "related work".

[9] *ACL*, *EMNLP*, *NAACL*, *TACL*, and *COLING* for NLP, *CVPR*, *ICCV*, and *WACV* for Computer Vision, *IROS*, *ICRA*, and *IJRR* for Robotics, and *NeurIPS*, *ICML* and *ICLR* for Machine Learning. We include proceedings from associated workshops of these conferences in our analysis.

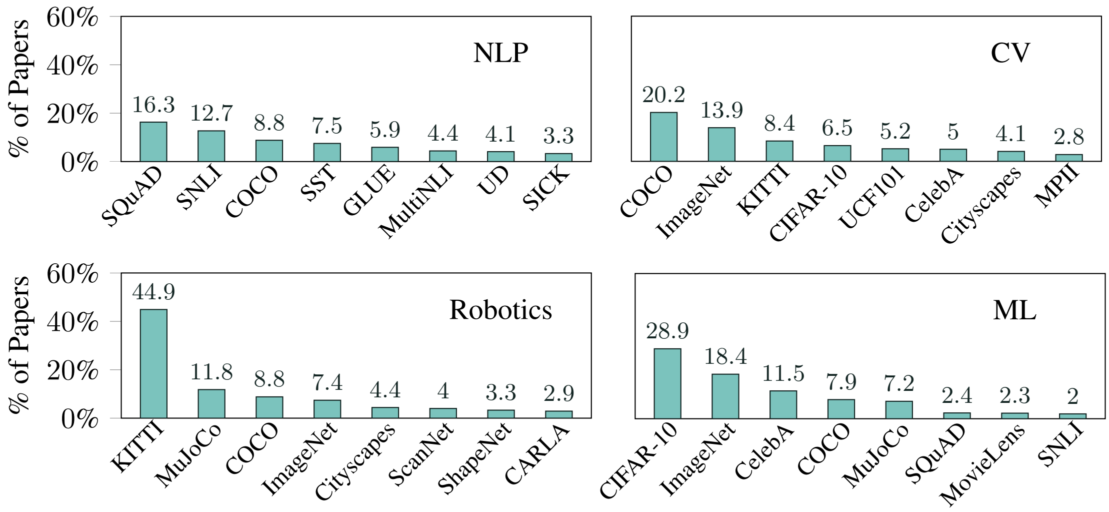

Figure 5: How many papers in each field use each dataset?

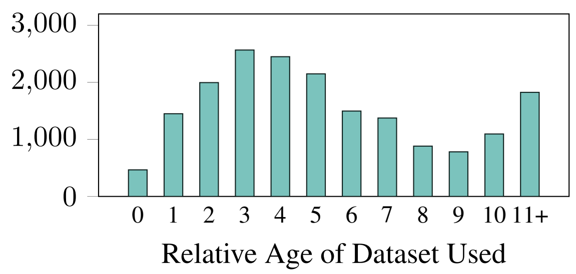

Figure 6: Distribution of relative year of datasets used across all papers that used a dataset. Median relative age of dataset is 5 and maximum is 28.

a query-dataset similarity function $\mathrm{sim}(q, d)$ and return the top k datasets. To better our understanding of this new task, we conduct a benchmark comparison of models for computing the similarity scores.

3.1 Term-Based Retrieval

We implement a BM25 retriever (Robertson and Zaragoza, 2009) using Pyserini (Lin et al., 2021).[10] We index each dataset in our search corpus with its dataset description from Papers with Code and the title of its introducing paper.

3.2 Nearest-Neighbor Retrieval

We experiment with direct k-nearest-neighbor retrieval. We map each test set query to a feature space and identify the closest training set queries in feature space using efficient similarity search (Johnson et al., 2017). We return the relevant datasets associated with these queries. In practice we investigate two types of feature extractor: TF-IDF (Jones, 2004) and SciBERT (Beltagy et al., 2019).

[10] We run BM25 with $k_1 = 0.8$ and $b = 0.4$.

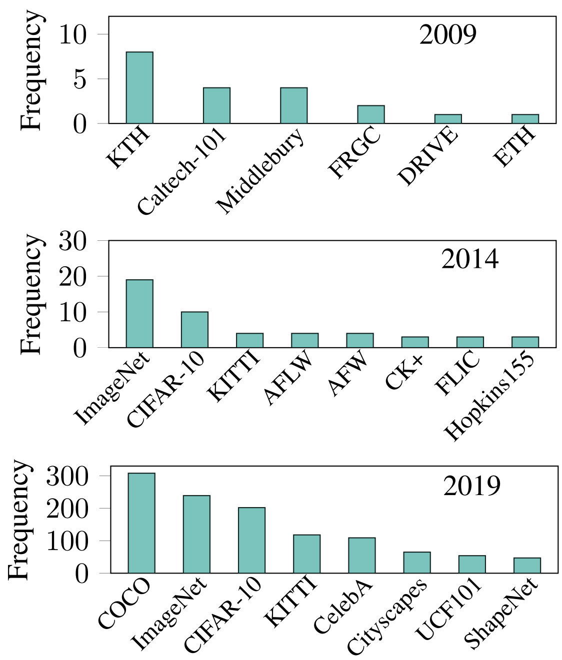

Figure 7: Popular CV datasets in 2009, 2014, and 2019.

3.3 Neural Retrieval

We implement a bi-encoder retriever using the Tevatron package.[11] In this framework, we encode each query and document into a shared vector space, and estimate similarity via the inner product between query and document representations. For each text sequence (query or document) we use the BERT embedding (Devlin et al., 2019) of that text's [CLS] token to represent the document:

$$\mathrm{sim}(q, d) = \mathrm{cls}(\mathtt{BERT}(q))^T \mathrm{cls}(\mathtt{BERT}(d))$$

[11] https://github.com/texttron/tevatron

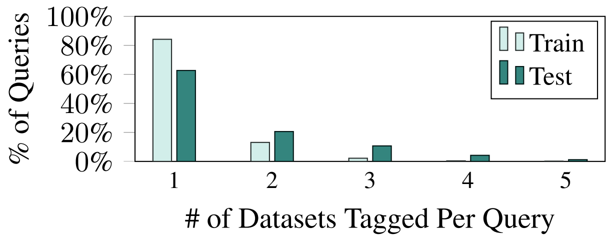

Figure 8: The distribution of the number of datasets tagged in each paper, in train and test sets

where cls(·) denotes the operation of accessing the [CLS] token representation from the contextual encoding (Gao et al., 2021). For retrieval, we separately encode all queries and documents and retrieve using efficient similarity search. Following recent work (Karpukhin et al., 2020), we minimize a contrastive loss and select hard negatives using BM25 for training. We initialize the bi-encoder with SciBERT (Beltagy et al., 2019). This model takes 20 minutes to train on one 11GB Nvidia GPU.

3.4 Commercial Search Engines

The standard paradigm for dataset search is to use a conventional search engine with short queries (Kacprzak et al., 2019). To demonstrate the impact of using natural language descriptions to find datasets, we compare with two commercial dataset search engines - *Google Dataset Search*[12] (Brickley et al., 2019) and *Papers with Code*[13] dataset search. For Google Dataset Search, we limit results to datasets from Papers with Code so retrieved results can be compared with our ground truth.

To simulate typical user behavior, we carefully constructed short keyword search queries for each natural language method description in our test set. A trained annotator[14] read each natural language method description in our test set, and assessed the dataset need underlying the method description.

Note that for the purpose of dataset search, natural language queries may convey multiple information needs. For example, the query "[..] we propose a very deep fully convolutional encoding-decoding framework for image restoration such as denoising and super-resolution" suggests two dataset needs: image denoising and image super-resolution.

Accordingly, the annotator wrote a query containing 4 or fewer keywords for each query intent conveyed by the description, using initial search results to iteratively refine the queries. After running each query against a commercial search engine, the results from all query intents were combined using balanced interleaving (Joachims, 2002).

For comparison, we measured the commercial search engines taking as input either keyword queries or natural language method descriptions.

3.5 DatasetFinder for Keyword Search

To better compare with keyword-based search systems, we train a version of our system on keyphrase inputs. We extract keyphrases from each abstract in our training set using BART (Lewis et al., 2020) finetuned on the OpenKP dataset (Xiong et al., 2019). We train our bi-encoder model with these keyphrases as a surrogate for keyword queries.

4 Evaluation

4.1 Evaluation Metrics

Information retrieval metrics estimate search relevance. These metrics count all queries equally when computing an aggregate test set metric value. We use four standard metrics using the `trec_eval` package (with the '-c' flag). Each is computed for a given test set query as follows:

Precision@k

$$P@k = \frac{\text{\# of relevant items in top } k \text{ retrieved}}{k}$$

Recall@k

$$R@k = \frac{\text{\# of relevant items in top } k \text{ retrieved}}{\text{\# of relevant items}}$$

Mean Average Precision

$$\text{MAP} = \frac{1}{m}\sum_{n=1}^{m} \text{Precision}@k_n$$

m is the total number of relevant items and k_n is the smallest integer such that the n^{th} relevant item is in the top k retrieved items (Manning et al., 2005).

Mean Reciprocal Rank

$$\text{MRR} = \frac{1}{m}\sum_{n=1}^{m} \frac{1}{\text{Rank}_n}$$

Rank_n is the rank of the n^{th} relevant item in the retrieved results (Voorhees and Harman, 1999).

4.2 Time Filtering

The queries in our test set were made between 2012 and 2020, with a median year of 2017. On the other hand, half the datasets in our search corpus were introduced in 2018 or later.

[12] https://datasetsearch.research.google.com

[13] https://paperswithcode.com/datasets

[14] A computer science graduate student with experience using both search engines.

	P@5	R@5	MAP	MRR
BM25	3.9	14.2	8.4	10.8
kNN (TF-IDF)	8.3	28.1	19.2	25.9
kNN (SciBERT)	5.7	20.7	11.5	14.9
Bi-Encoder	**11.8**	**38.3**	**27.1**	**35.3**

Table 1: Benchmarking results on standard metrics

	P@5	R@5	MAP	MRR
PwC (*descriptions*)	0.6	1.7	0.9	1.2
PwC (*keywords*)	3.5	10.0	6.5	9.1
Google (*descriptions*)	0.0	0.0	0.0	0.0
Google (*keywords*)	7.6	23.2	11.6	15.4
Ours (*descriptions*)	**11.8**	**38.3**	**27.1**	**35.3**
Ours (*keywords*)	8.9	28.6	19.1	25.5

Table 2: Comparing external search engines (*Papers with Code* and *Google Dataset Search*) against our DatasetFinder system using a bi-encoder architecture.

To account for this discrepancy, for each query q, we do not rank the full search corpus D. Rather, we consider the subset $D' = \{d \in D \mid \text{year}(d) \leq \text{year}(q)\}$ consisting of datasets introduced in the same year as the query or earlier.

4.3 Test Set Evaluation

4.3.1 Comparing Proposed Methods

In Table 1, we report performance on standard retrieval metrics of the methods described in Section 3 using a single seed when applicable. Term-based retrieval (BM25) performs very poorly in this setting, while the neural bi-encoder model excels. This suggests term matching heuristics in web search do not transfer to this task, which requires semantic matching with learned representations.

4.3.2 Comparing with Commercial Search Engines

In Table 2, we compare our proposed retrieval system against two commercial dataset search engines. For each search engine, we choose the top 5 results before computing metrics.

We find these commercial search engines do not effectively support long natural language descriptions as input. Even with hand-written keywords, which these search engines are designed to use, our neural retriever still gives better search results. With these observations, we speculate that the commercial search engines are adapted from term-based web search engines. In comparison, our neural retrievers gain a performance advantage by semantic search with neural retrievers.

Idea: We show that sequence-to-sequence method achieves state-of-the-art results on syntactic parsing, whilst making almost no assumptions about the structure of the problem.
Keywords: syntactic parsing

Actual		Google	PwC	Ours
Penn Treebank	1	GitHub-Python	AI2D	Penn Treebank
	2	English Web Treebank	PNT	SICK
	3		Spades	SST

Idea: We propose a novel ResNet-like architecture that combines multi-scale context with pixel-level accuracy for Semantic Image Segmentation.
Keywords: semantic image segmentation

Actual		Google	PwC	Ours
Cityscapes	1	Agriculture-Vision	Semantic Scholar	Cityscapes
	2	BIG	Semantic Trails	SBD
	3	PASCAL VOC	BCSS	ADE20K

Idea: We propose a dual pathway, 11-layers deep, multi-scale, three-dimensional Convolutional Neural Network for the challenging task of brain lesion segmentation.
Keywords: brain lesion segmentation

Actual		Google	PwC	Ours
BraTS 2015	1	BraTS 2017	Lesion Boundary Segmentation	DRIVE
	2	BraTS 2013	Brain US	STARE
	3	BraTS 2015	Brain-Score	LUNA

Figure 9: Qualitative comparison of the DatasetFinder system with external dataset search engines.

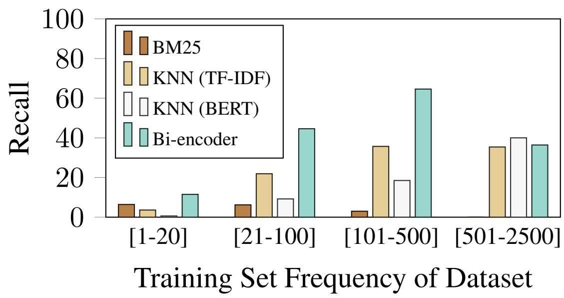

Figure 10: Examining recall on the test set on datasets with varying training set frequency.

4.3.3 Qualitative Results

We show examples in Figure 9. In the first two, we see keyword-based search engines are sensitive to ambiguous search terms, such as "semantic," unlike our system. In the final example, we see a downside of our approach: given a query for brain lesion segmentation, our system recommends data for the related (but incorrect) tasks of retinal vessel segmentation and lung nodule segmentation.

4.3.4 Evaluating Retrieval of Rare Datasets

For our retrieval task, we are particularly interested in the ability to retrieve datasets for users that they may not already be aware of. To this end, we group our search corpus into a six buckets, based on the

	P@5	R@5	MAP	MRR
DatasetFinder	11.8	38.3	27.1	35.3
w/ Tasks Hidden	11.7	38.8	26.0	34.3
w/ Methods Hidden	10.8	36.1	24.6	31.8

Table 3: Eliding mentions of methods from queries has a minor impact on search quality.

frequency that dataset is tagged as relevant to a natural language description in our training set. We then measure how often we correctly retrieve datasets in each bucket at test time.

We find that supervised system performs poorly on datasets rarely seen in the training, while BM25 performs poorly in all scenarios. Our strongest architecture, the bi-encoder, performs worse for rare datasets compared to popular datasets. Though it outperforms other systems in the rare-dataset regime, the bi-encoder may still bias users towards popular datasets. Addressing this is an important area of future work.

5 Analysis

5.1 Descriptions vs. Keywords

One defining characteristic of our recommendation system is that it uses natural language method descriptions. To what extent are natural language descriptions critical to our system's performance?

In Table 2, we compare two versions of the DatasetFinder system: one trained and tested with description queries and the other with keyword queries, as described in Section 3.5. We observe providing method descriptions leads to better search quality by a wide margin on every metric.

This supports the claim that natural language descriptions provide a richer input for dataset search. Moreover, this suggests that the performance gap between our system and the other commercial search systems reported in Table 2 cannot be explained completely by the fact that our ranker was trained using task-specific supervision.

5.2 Analysis of Successful Queries

Two types of mentions frequently seen in the input are *tasks* (e.g. "CNN") and *methods* (e.g. "image classification"). To understand how these semantic categories affect the value of natural language descriptions, we experiment with concealing task and method spans from descriptions. We extract a large list of known tasks and methods from Papers with Code and performing exact span matching. We replace task or method spans with the tokens `[TASK]` or `[METHOD]`, respectively.

We train and evaluate models on this elided data. In Table 3, we see concealing task mentions has no impact on search results, while concealing method names reduces performance slightly. This suggests our model may learn to associate method names (e.g. "CNN") with appropriate datasets. However, given these small differences, the DatasetFinder system is not relying on these surface-level lexical features; we argue it is able to understand the query to make up for missing information.

6 Related Work

Most work on scientific dataset recommendation uses a conventional information retrieval perspective (Lu et al., 2012; Kunze and Auer, 2013; Sansone et al., 2017; Chapman et al., 2019; Brickley et al., 2019; Lhoest et al., 2021). In 2019, Google Research launched *Dataset Search* (Brickley et al., 2019), offering access to over 2 million public datasets. Our work considers a subset of Google Dataset Search's search corpus - those datasets that have been posted on Papers with Code.

Some work has considered other forms of dataset recommendation. Ben Ellefi et al. (2016) presented a system for dataset recommendation where the query is a "source dataset" relevant to the user. More recently, Altaf et al. (2019) reported a system where the user's query is a set of research papers. Ours is the first to study natural language queries for dataset search, in contrast to conventional dataset search where queries are usually 3 or fewer tokens in length (Kacprzak et al., 2019).

7 Conclusion

We introduce a new task for dataset retrieval. We develop a system called *DatasetFinder* for this task with the goal of helping researchers discover new, relevant datasets for their work. Our system achieves superior search results than conventional dataset search engines, and we show evidence that natural language method descriptions are superior inputs for dataset search than traditional search keywords. We release our automatically generated dataset along with our ranking systems to the public with the hope that we spur the community to work on this task.

References

Basmah Altaf, Uchenna Akujuobi, Lu Yu, and Xiangliang Zhang. 2019. Dataset recommendation via variational graph autoencoder. *2019 IEEE International Conference on Data Mining (ICDM)*, pages 11–20.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: Pretrained language model for scientific text. In *EMNLP*.

Mohamed Ben Ellefi, Zohra Bellahsene, Stefan Dietze, and Konstantin Todorov. 2016. Dataset recommendation for data linking: An intensional approach. In *European Semantic Web Conference*.

Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference*, pages 1365–1375.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. 2020. TLDR: Extreme summarization of scientific documents. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4766–4777, Online. Association for Computational Linguistics.

Adriane P. Chapman, Elena Paslaru Bontas Simperl, Laura M. Koesten, G. Konstantinidis, Luis Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2019. Dataset search: a survey. *The VLDB Journal*, 29:251–272.

Susan Crawford and Loretta Stucki. 1990. Peer review and the changing research record. *Journal of the Association for Information Science and Technology*, 41:223–228.

Jeffrey Dean. 2020. 1.1 the deep learning revolution and its implications for computer architecture and chip design. *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, pages 8–14.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. 2009. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88:303–338.

Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. In *ECIR*.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231 – 1237.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. Scirex: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*.

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

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60:493–502.

Emilia Kacprzak, Laura M. Koesten, Luis Daniel Ibáñez, Tom Blount, Jeni Tennison, and Elena Paslaru Bontas Simperl. 2019. Characterising dataset search - an analysis of search logs and data requests. *J. Web Semant.*, 55:37–55.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90.

Sven R. Kunze and Sören Auer. 2013. Dataset retrieval. *2013 IEEE Seventh International Conference on Semantic Computing*, pages 1–8.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario vSavsko, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clement Delangue, Th'eo Matussiere, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, Franccois Lagunas, Alexander M. Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *EMNLP*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python Toolkit for Reproducible Information Retrieval Research with Sparse and Dense Representations. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2356–2362, New York, NY, USA. Association for Computing Machinery.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Meiyu Lu, Srinivas Bangalore, Graham Cormode, Marios Hadjieleftheriou, and Divesh Srivastava. 2012. A dataset search engine for the research document corpus. *2012 IEEE 28th International Conference on Data Engineering*, pages 1237–1240.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2005. Introduction to Information Retrieval.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguistics*, 19:313–330.

Mark E. J. Newman. 2004. Power laws, pareto distributions and zipf's law.

Andrew Ng. 2021. MLOps: From Model-centric to Data-centric AI.

Martha Palmer and Nianwen Xue. 2010. Linguistic annotation. *Handbook of Computational Linguistics and Natural Language Processing*, pages 238–270.

Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. 2021. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336.

Dragomir R. Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating web-based question answering systems. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Susanna-Assunta Sansone, Alejandra N. González-Beltrán, Philippe Rocca-Serra, George Alter, Jeffrey S. Grethe, Hua Xu, Ian M. Fore, Jared Lyle, Anupama E. Gururaj, Xiaoling Chen, Hyeon eui Kim, Nansu Zong, Yueling Li, Ruiling Liu, I. B. Ozyurt, and Lucila Ohno-Machado. 2017. Dats, the data tag suite to enable discoverability of datasets. *Nature Scientific Data*, 4.

Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033.

Ellen M. Voorhees and Donna K. Harman. 1999. Overview of the eighth text retrieval conference (trec-8). In *TREC*.

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Fernando Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. In *EMNLP*.