DataFinder: Scientific Dataset Recommendation from Natural Language Descriptions

Vijay Viswanathan[1]   **Luyu Gao**[1]
Tongshuang Wu[1]   **Pengfei Liu**[2,3]   **Graham Neubig**[1,3]
[1]Carnegie Mellon University   [2]Shanghai Jiao Tong University   [3]Inspired Cognition
{vijayv, luyug, sherryw, gneubig}@cs.cmu.edu   stefanpengfei@gmail.com

Abstract

Modern machine learning relies on datasets to develop and validate research ideas. Given the growth of publicly available data, finding the right dataset to use is increasingly difficult. Any research question imposes explicit and implicit constraints on how well a given dataset will enable researchers to answer this question, such as dataset size, modality, and domain. We operationalize the task of recommending datasets given a short natural language description of a research idea, to help people find relevant datasets for their needs. Dataset recommendation poses unique challenges as an information retrieval problem; datasets are hard to directly index for search and there are no corpora readily available for this task. To facilitate this task, we build *the DataFinder Dataset* which consists of a larger automatically-constructed training set (17.5K queries) and a smaller expert-annotated evaluation set (392 queries). Using this data, we compare various information retrieval algorithms on our test set and present a superior bi-encoder retriever for text-based dataset recommendation. This system, trained on *the DataFinder Dataset*, finds more relevant search results than existing third-party dataset search engines. To encourage progress on dataset recommendation, we release our dataset and models to the public.[1]

1 Introduction

Innovation in modern machine learning (ML) depends on datasets. 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).
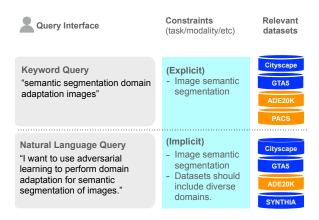

Figure 1: Queries for dataset recommendation impose constraints on the type of dataset desired. Keyword queries make these constraints explicit, while full-sentence queries impose implicit constraints. Ground truth relevant datasets for this query are colored in blue.

With the growth of research in ML and artificial intelligence (AI), there are hundreds of datasets published every year (shown in Figure 2). Knowing which to use for a given research idea can be difficult (Paullada et al., 2021). To illustrate, consider a real query from a graduate student who says, *"I want to use adversarial learning to perform domain adaptation for semantic segmentation of images."* They have implicitly issued two requirements: they need a dataset for semantic segmentation of images, and they want datasets that include diverse visual domains. A researcher may intuitively select popular, generic semantic segmentation datasets like COCO (Lin et al., 2014) or ADE20K (Zhou et al., 2019), but these are insufficient to cover the query's requirement of supporting domain adaptation. How can we infer the intent of the researcher and make appropriate recommendations?

To study this problem, we operationalize the task of "**dataset recommendation**": given *a full-sentence description* or *keywords* describing a research topic, recommend datasets to support research on this topic (§2). A concrete example is

---
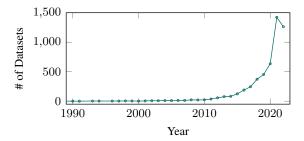
[1]Code and data: https://github.com/viswavi/datafinder

Figure 2: The number of public AI datasets has exploded in recent years. Here we show the # released from 1990 to 2022 according to Papers with Code.[2]

shown in Figure 1. This task was introduced by Färber and Leisinger (2021), who framed it as text classification. In contrast, we naturally treat this task as retrieval (Manning et al., 2005), where the *search collection* is a set of datasets represented textually with dataset descriptions[3], structured metadata, and published "citances" — references from published papers that use each dataset (Nakov et al., 2004). This framework allows us to measure performance with rigorous ranking metrics such as mean reciprocal rank (Radev et al., 2002).

To strengthen evaluation, we build a dataset, *the DataFinder Dataset*, to measure how well we can recommend datasets for a given description (§3). As a proxy for real-world queries for our dataset recommendation engine, we construct queries from *paper abstracts* to simulate researchers' historical information needs. We then identify the datasets used in a given paper, either through manual annotations (for our small test set) or using heuristic matching (for our large training set). To our knowledge, this is the first expert-annotated corpus for dataset recommendation, and we believe this can serve as a challenging testbed for researchers interested in representing and searching complex data.

We evaluate three existing ranking algorithms on our dataset and task formation, as a step towards solving this task: BM25 (Robertson and Zaragoza, 2009), nearest neighbor retrieval, and dense retrieval with neural bi-encoders (Karpukhin et al., 2020). BM-25 is a standard baseline for text search, nearest neighbor retrieval lets us measure the degree to which this task requires generalization to new queries, and bi-encoders are among the most effective search models used today (Zhong et al., 2022). Compared with third-party keyword-centric dataset search engines, a bi-encoder model trained on *DataFinder* is far more effective at finding relevant datasets. We show that finetuning the

bi-encoder on our training set is crucial for good performance. However, we observe that this model is as effective when trained and tested on keyphrase queries as on full-sentence queries, suggesting that there is room for improvement in automatically understanding full-sentence queries.

2   Dataset Recommendation Task

We establish a new task for automatically recommending relevant datasets given a description of a data-driven system. Given a query 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 1 illustrates this with a real query written by a graduate student.

The query q can take two forms: either a keyword query (the predominant interface for dataset search today (Chapman et al., 2019)) or a full-sentence description. Textual descriptions offer a more flexible input to the recommendation system, with the ability to implicitly specify constraints based on what a researcher wants to study, without needing to carefully construct keywords a priori.

Evaluation Metrics   Our task framing naturally leads to evaluation by information retrieval metrics that estimate search relevance. In our experiments, we use four common metrics included in the `trec_eval` package,[4] a standard evaluation tool used in the IR community:

- **P**recision@k: The proportion of relevant items in top k retrieved datasets. If P@k is 1, then every retrieved document is valuable.

- **R**ecall@k: The fraction of relevant items that are retrieved. If R@k is 1, then the search results are comprehensive.

- Mean Average Precision (**MAP**): Assuming we have m relevant datasets in total, and k_i is the rank of the ith relevant dataset, MAP is calculated as $\sum_i^m \text{P@}k_i/m$ (Manning et al., 2005). High MAP indicates strong average search quality over all relevant datasets.

- Mean Reciprocal Rank (**MRR**): The average of the inverse of the ranks at which the first relevant item was retrieved. Assuming R_i is the rank of the i-th relevant item in the retrieved result, MRR is calculated as $\sum_i^m R_i/m$. High MRR means a user sees *at least some* relevant datasets early in the search results.

---

[3]From www.paperswithcode.com

[4]https://github.com/usnistgov/trec_eval. We use the -c flag for the trec_eval command.

Collection of Datasets

Papers with Code → 1 2 3 4 5 ooo

- **Dataset Description**
- **Intro. Paper Title**
- **Struct. Metadata**
- **Citances**

Training

S2ORC (Lo et al, 2020) → Galactica (Taylor et al, 2022) → Training Queries: Qry 1, Qry 2 ... → Positive Examples

Dataset Tagger → Relevant Datasets → Negative Mining → Negative Examples

Testing

SciREX (Jain et al, 2020) → Human Annotation → Relevant Datasets → Evaluation

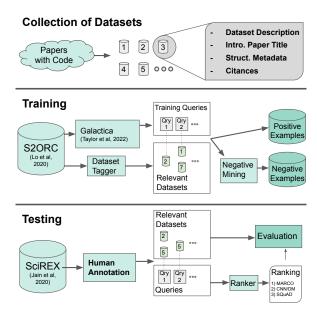Queries: Qry 1, Qry 2 ... → Ranker → Ranking: 1) MARCO 2) CNN/DM 3) SQuAD

Figure 3: We search against all datasets on Papers With Code. Our system is trained on a set of simulated queries and target datasets and evaluated on a set of expert-written queries with hand-annotated target datasets.

3 The DataFinder Dataset

To support this task, we construct a dataset called *The DataFinder Dataset* consisting of (q, R) pairs extracted from published English-language scientific proceedings, where each q is either a full-sentence description or a keyword query. We collect a large training set through an automated method (for scalability), and we collect a smaller test set using real users' annotations (for reliable and realistic model evaluation). In both cases, our data collection contains two primary steps: (1) **collecting search queries** q that a user would use to describe their dataset needs, and (2) **identifying relevant datasets** R that match the query. Our final training and test sets contain 17495 and 392 queries, respectively. Figure 3 summarizes our data collection approach. We explain the details below and provide further discussion of the limitations of our dataset in the Limitations section. We will release our data under a permissive CC-BY License.

3.1 Collection of Datasets

In our task definition, we search over the collection of datasets listed on Papers With Code, a large public index of papers which includes metadata for over 7000 datasets and benchmarks. For most datasets, Papers With Code Datasets stores a short human-written dataset description, a list of different names used to refer to the dataset (known as

"variants"), and structured metadata such as the year released, the number of papers reported as using the dataset, the tasks contained, and the the modality of data. Many datasets also include the paper that introduced the dataset. We used the dataset description, structured metadata, and the introducing paper's title to textually represent each dataset, and we analyze this design decision in §5.4.

3.2 Training Set Construction

To ensure scalability for the training set, we rely on a large corpus of scientific papers, S2ORC (Lo et al., 2020). We extract nearly 20,000 abstracts from AI papers that use datasets. To overcome the high cost of manually-annotating queries or relevant datasets, we instead simulate annotations with few-shot-learning and rule-based methods.

Query Collection We extract queries from paper abstracts because, intuitively, an abstract will contain the most salient characteristics behind a research idea or contribution. As a result, it is an ideal source for comprehensively collecting potential implicit constraints as shown in Figure 1.

We simulate query collection with the 6.7B parameter version of Galactica (Taylor et al., 2022), a large scientific language model that supports few-shot learning. In our prompt, we give the model an abstract and ask it to first extract five keyphrases: the tasks mentioned in paper, the task domain of the paper (e.g., biomedical or aerial), the modality of data required, the language of data or labels required, and the length of text required (sentence-level, paragraph-level, or none mentioned). We then ask Galactica to generate a full query containing any salient keyphrases. We perform few-shot learning using 3 examples in the prompt to guide the model. Our prompt is shown in Appendix A.

Relevant Datasets For our training set, relevant datasets are automatically labeled using the body text of a paper.[5] We apply a rule-based procedure to identify the dataset used in a given paper (corresponding to an abstract whose query has been auto-labeled). For each paper, we tag all datasets that satisfy two conditions: the paper must cite the paper that introduces the dataset, and the paper must mention the dataset by name twice.[6]

---

[5]Note that our queries are obtained from the abstract alone while the relevance judgements are obtained from the text body, to encourage more general queries.

[6]We apply the additional requirement that the counted dataset mentions must occur in a section with section title con-

This tagging procedure is restrictive and emphasizes precision (i.e., an identified dataset is indeed used in the paper) over recall (i.e., all the used datasets are identified). Nonetheless, using this procedure, we tag 17,495 papers from S2ORC with at least one dataset from our collection of datasets.

To estimate the quality of these tagged labels, we 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.

3.3 Test Set Construction

To accurately approximate how humans might search for datasets, we employed AI researchers and practitioners to annotate our test set. As mentioned above, the dataset collection requires both *query collection* and *relevant dataset collection*. We use SciREX (Jain et al., 2020), a human-annotated set of 438 full-text papers from major AI venues originally developed for research into full-text information extraction, as the basis of our test set. We choose this dataset because it naturally supports our dataset collection described below.

Query Collection   We collect search queries by asking annotators to digest, extract, and rephrase key information in research paper abstracts.

Annotators. To ensure domain expertise, we recruited 27 students, faculty, and recent alumni of graduate programs in machine learning, computer vision, robotics, NLP, and statistics from major US universities. We recruited 23 annotators on a voluntary basis through word of mouth; for the rest, we offered 10 USD in compensation. We sent each annotator a Google Form that contained between 10 and 20 abstracts to annotate. The instructions provided for that form are shown in Appendix B.

Annotation structure. For each abstract, we asked annotators to extract metadata regarding the abstract's task, domain, modality, language of data required, and length of data required. These metadata serve as **keyphrase queries**. Then, based on these keyphrases, we also ask the annotator to write a sentence that best reflects the dataset need of the given paper/abstract, which becomes the *full-sentence query*. Qualitatively, we found that the keyphrases helped annotators better ground and

taining "results", "experiment", "evaluation", "result", "training", or "testing", to avoid non-salient dataset mentions, such as those commonly occurring in "related work".
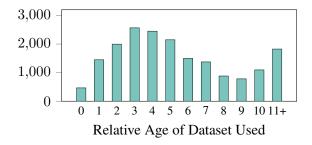

Figure 4: Distribution of relative year of datasets used across all papers that used a dataset.

concretize their queries, and the queries often contain (a subset of) these keyphrases.

Model assistance. To encourage more efficient labeling (Wang et al., 2021), we provided auto-suggestions for each field from GPT-3 (Brown et al., 2020) and Galactica 6.7B (Taylor et al., 2022) to help annotators. We note that annotators rarely applied these suggestions directly — annotators accepted the final full-sentence query generated by either large language model only 7% of the time.

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). We used these annotations as initial suggestions for the datasets used in each paper. The authors of this paper then skimmed all 438 papers in SciREX and noted the datasets used in each paper. 46 papers were omitted because they either used datasets not listed on Papers With Code or were purely theory-based papers with no relevant datasets, leaving a final set of 392 test examples.

We double-annotated 10 papers with the datasets used. The annotators labeled the exact same set of datasets for 8 out of 10 papers, with a Fleiss-Davies kappa of 0.667, suggesting that inter-annotator agreement for our "relevant dataset" annotations is substantial (Davies and Fleiss, 1982; Loper and Bird, 2002).

3.4 Dataset Analysis

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

Our final collected dataset contains 17,495 training queries and 392 test queries. The training examples usually associate queries with a single dataset much more frequently than our test set does. This is due to our rule-based tagging scheme, which emphasizes precise labels over recall. Meanwhile, the median query from our expert-annotated test set
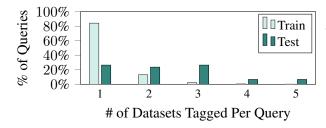
Figure 5: The distribution of the number of datasets tagged in each paper, in train and test sets

had 3 relevant datasets associated with it. We also observed interesting dataset usage patterns:

- **Researchers tend to converge towards popular datasets.** Analyzing dataset usage by community,[7] we find that in all fields, among all papers that use some publicly available dataset, more than 50% papers in our training set use at least one of the top-5 most popular datasets. Most surprisingly, nearly half of the papers tagged in the robotics community use the KITTI dataset (Geiger et al., 2013).

- **Researchers tend to rely on recent datasets.**

  In Figure 4, we see the distribution of relative ages of datasets used (i.e., the year between when a dataset is published, and when a corresponding paper uses it for experiments). In Figure 4, We observe that the average dataset used by a paper was released 5 years before the paper's publication (with a median of 5.6 years), but we also see a significant long tail of older datasets. This means that while some papers use traditional datasets, most papers exclusively use recently published datasets.

  These patterns hint that researchers might overlook less cited datasets that match their needs in favor of standard *status-quo* datasets. This motivates the need for nuanced dataset recommendation.

4 Experimental Setup on *DataFinder*

How do popular methods perform on our new task and new dataset? How does our new paradigm differ from existing commercial search engines? In this section, we describe a set of standard methods which we benchmark, and we consider which third-party search engines to use for comparison.

---

[7]We define "communities" by publication venues: *ACL, EMNLP, NAACL, TACL, COLING* for NLP, *CVPR, ICCV, WACV* for Vision, *IROS, ICRA, IJRR* for Robotics, and *NeurIPS, ICML ICLR* for Machine Learning. We include proceedings from associated workshops in each community.

4.1 Task Framing

We formulate dataset recommendation as a ranking task. Given a query q and a search corpus of datasets D, rank the datasets $d \in D$ based on a query-dataset similarity function $\text{sim}(q, d)$ and return the top k datasets. We compare three ways of defining $\text{sim}(q, d)$: term-based retrieval, nearest-neighbor retrieval, and neural retrieval.

4.2 Models to Benchmark

To retrieve datasets for a query, we find the nearest datasets to that query in a vector space. We represent each query and dataset in a vector space using three different approaches:

Term-Based Retrieval   We evaluated a BM25 retriever for this task, since this is a standard baseline algorithm for information retrieval. We implement BM25 (Robertson and Walker, 1999) using Pyserini (Lin et al., 2021).[8]

Nearest-Neighbor Retrieval   To understand the extent to which this task requires generalization to new queries unseen at training time, we experiment with direct k-nearest-neighbor retrieval against the training set. For a new query, we identify the most similar queries in the training set and return the relevant datasets from these training set examples. In other words, each dataset is represented by vectors corresponding to all training set queries attached to that dataset. In practice we investigate two types of feature extractors: TF-IDF (Jones, 2004) and SciBERT (Beltagy et al., 2019).

Neural Retrieval   We implement a bi-encoder retriever using the Tevatron package.[9] 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 vectors. We represent each document with the BERT embedding (Devlin et al., 2019) of its [CLS] token:

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

where $\text{cls}(\cdot)$ 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

---

[8]We run BM25 with $k_1 = 0.8$ and $b = 0.4$.
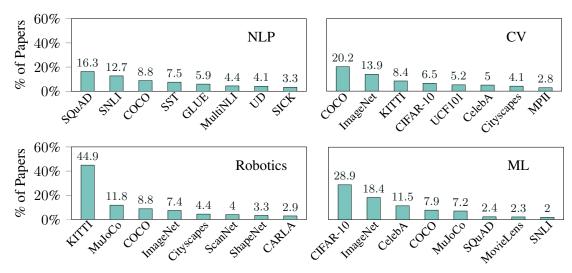[9]https://github.com/texttron/tevatron

Figure 6: We analyze the distribution of datasets used in NLP, robotics, vision, and machine learning research.

BM25 for training. We initialize the bi-encoder with SciBERT (Beltagy et al., 2019) and finetune it on our training set. This model takes 20 minutes to finetune on one 11GB Nvidia GPU.

4.3 Comparison with Search Engines

Besides benchmarking existing methods, we also compare the methods enabled by our new data recommendation task against the standard paradigm for dataset search — to use a conventional search engine with short queries (Kacprzak et al., 2019). We measured the performance of third-party dataset search engines taking as input either keyword queries or full-sentence method descriptions.

We compare on our test set with two third-party systems– *Google Dataset Search*[10] (Brickley et al., 2019) and *Papers with Code*[11] search. Google Dataset Search supports a large dataset collection, so we limit results to those from Papers with Code to allow comparison with the ground truth.

Our test set annotators frequently entered multiple keyphrases for each keyphrase type (e.g. "question answering, recognizing textual entailment" for the Task field). We constructed multiple queries by taking the Cartesian product of each set of keyphrases from each field, deduplicating tokens that occurred multiple times in each query. After running each query against a commercial search engine, results were combined using balanced interleaving (Joachims, 2002).

Model	P@5	R@5	MAP	MRR
Full-Sentence Queries				
BM25	4.7 ±0.1	11.6 ±1.7	8.0 ± 1.3	14.5 ±2.0
kNN (TF-IDF)	5.5 ±0.6	12.3 ±1.6	7.8 ±1.1	15.5 ±2.0
kNN (BERT)	7.1 ±0.7	14.2 ±1.5	9.7 ±1.2	21.3 ±2.3
Bi-Encoder	**16.0** ±1.1	**31.2** ±2.2	**23.4** ±1.9	**42.6** ±2.7
Keyphrase Queries				
BM25	6.6 ±0.5	15.3 ±1.1	11.4 ±0.8	19.9 ±1.5
kNN (TF-IDF)	2.7 ±0.4	5.9 ±1.1	3.3 ±0.7	8.2 ±1.6
kNN (BERT)	2.8 ±0.4	5.8 ±1.1	3.3 ±1.1	7.3 ±1.3
Bi-Encoder	**16.5** ±1.0	**32.4** ±2.2	**23.3** ±1.8	**42.3** ±2.6

Table 1: A comparison of methods on full-sentence and keyword search shows that the neural bi-encoder performs best by a significant margin. Standard deviations are obtained via bootstrap sampling on the test set.

5 Evaluation

5.1 Time Filtering

The queries in our test set were made from papers published between 2012 and 2020[12], with median year 2017. In contrast, half the datasets in our search corpus were introduced in 2018 or later. To account for this discrepancy, for each query q, we only rank the subset of datasets $D' = \{d \in D \mid \text{year}(d) \leq \text{year}(q)\}$ that were introduced in the same year or earlier than the query.

5.2 Benchmarking and Comparisons

Benchmarking shows that DataFinder benefits from deep semantic matching. In Table 1, we report retrieval metrics on the methods described

---

[10]https://datasetsearch.research.google.com
[11]https://paperswithcode.com/datasets

[12]We could not include more recent papers in our query construction process, because SciREX was released in 2020.

Model	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*)	0.1	0.1	0.1	0.3
Google (*keywords*)	9.7	19.5	12.3	24.0
Ours (*descriptions*)	16.0	31.2	23.4	42.6
Ours (*keywords*)	16.5	32.4	23.3	42.3

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

in §4. To determine the standard deviation of each metric, we use bootstrap resampling (Koehn, 2004) over all test set queries. Term-based retrieval (BM25) performs poorly in this setting, while the neural bi-encoder model excels. This suggests our task requires capturing semantic similarity beyond what term matching can provide. Term-based kNN search is not effective, implying that generalization to new queries is necessary for this task.

Commercial Search Engines are not effective on *DataFinder*. In Table 2, we compare our proposed retrieval system against third-party dataset search engines. For each search engine, we choose the top 5 results before computing metrics.

We find these third-party search engines do not effectively support full-sentence queries. We speculate these search engines are adapted from term-based web search engines. In contrast, our neural retriever gives much better search results using both keyword search and full-sentence query search.

5.3 Qualitative Analysis

Examples in Figure 7 highlight the tradeoffs between third-party search engines and models trained on DataFinder. In the first two examples, we see keyword-based search engines struggle when dealing with terms that could apply to many datasets, such as "semantic segmentation" or "link prediction". These keywords offer a limited specification on the relevant dataset, but a system trained on simulated search queries from real papers can learn implicit filters expressed in a query.

On the final example, our system incorrectly focuses on the deep architecture described ("deep neural network architecture [...] using depthwise separable convolutions") rather than the task described by the user ("machine translation"). Improving query understanding for long queries is a key opportunity for improvement on this dataset.

Full-Sentence Query: I want to use adversarial learning to perform domain adaptation for semantic segmentation of images.
Keyword Query: semantic segmentation domain adaptation images

Actual		Google	PWC	Ours
Cityscapes	1	LoveDA	VQA	Cityscapes
GTA5	2	Office-31	RTE	GTA5
SYNTHIA	3	Dark Zurich	VQA 2.0	SYNTHIA

Full-Sentence Query: We propose a method for knowledge graph link prediction based on complex embeddings
Keyword Query: knowledge base link prediction graph

Actual		Google	PWC	Ours
FB15k	1	WN18RR	RuBQ	FB15k
WN18	2	YAGO	DRKG	WN18
	3	FB15k-237	CVL-DataBase	

Full-Sentence Query: A new deep neural network architecture for machine translation using depthwise separable convolutions.
Keyword Query: machine translation text

Actual		Google	PWC	Ours
WMT 2014	1	WMT 2014	Machine Number Sense	SQuAD
	2		UCI Datasets	WikiText-2
	3		Affective Text	WikiText-103

Figure 7: We qualitatively compare the retrieval behavior of a neural biencoder retriever (trained on *DataFinder*) and third-party dataset search engines.

5.4 More In-depth Exploration

We perform in-depth qualitative analyses to understand the trade-offs of different query formats and dataset representations.

Comparing full-sentence vs keyword queries As mentioned above, we compare two versions of the DataFinder-based system: one trained and tested with description queries and the other with keyword queries. We observe that using keyword queries offers similar performance to using full-sentence descriptions for dataset search. This suggests more work should be done on making better use of implicit requirements in full-sentence descriptions for natural language dataset search.

Key factors for successful queries What information in queries is most important for effective dataset retrieval? Using human-annotated keyphrase queries in our test set, we experiment with concealing particular information from the keyphrase query.

In Figure 8, we see task information is critical for dataset search; removing task keywords from queries reduces MAP from 23.5 to 7.5 (statistically significant with $p < 0.001$ by a paired bootstrap t-test). Removing constraints on the language of text data also causes a significant drop in MAP ($p < 0.0001$). Removing keywords for text length causes an insignificant reduction in MAP
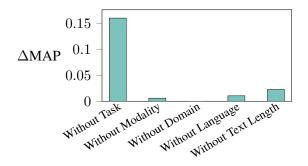
Figure 8: Comparison of the reduction in the MAP metric of the retrieval results after removing different types of query terms (e.g. keywords related to the task or language the researcher is interested in studying).

Model	P@5	R@5	MAP	MRR
Full-Sentence Queries				
Description	15.3 ±1.0	30.0 ±2.1	23.0 ± 1.9	42.8 ±2.7
+ Struct. Info	**16.0** ±1.1	**31.2** ±2.2	**23.3** ±1.8	**42.4** ±2.7
+ Citances	15.8 ±1.1	30.8 ±2.2	23.1 ±1.9	42.2 ±2.7
Keyphrase Queries				
Description	13.1 ±1.0	25.6 ±2.0	17.4 ±1.6	33.1 ±2.5
+ Struct. Info	16.6 ±1.1	32.7 ±2.2	23.5 ±1.8	42.8 ±2.8
+ Citances	**16.8** ±1.0	**33.4** ±2.2	**23.6** ±1.8	**43.0** ±2.6

Table 3: Adding structured metadata for each dataset's textual representation significantly improves keyphrase search quality using a neural bi-encoder. We compute standard deviations via bootstrap resampling. We use the "Description + Struct. Info" textual representation for all other experiments in this paper.

($p = 0.15$), though it causes a statistically significant reduction on other metrics not shown in Figure 8: P@5 and R@5. Based on inspection of our test set, we speculate that domain keywords are unnecessary because the domain is typically implied by task keywords.

Comparing textual representations of datasets We represent datasets textually with a community-generated dataset description from PapersWith-Code, along with the title of the paper that introduced the dataset. We experiment with enriching this dataset representation in two ways. We first add structured metadata about each dataset (e.g. tasks, modality, number of papers that use each dataset on PapersWithCode). We cumulatively experiment with adding citances — sentences from other papers around a citation — to capture how others use the dataset. In Table 3, our neural bi-encoder achieves similar retrieval performance on all 3 representations for full-sentence search.

Keyword search is more sensitive to dataset rep-

Model	P@5	R@5	MAP	MRR
Full-Sentence Queries				
SciBERT (finetuned)	**16.0**	**31.2**	**23.3**	**42.4**
SciBERT (not finetuned)	0.0	0.0	0.0	0.0
COCO-DR (not finetuned)	6.1	14.8	8.8	15.7
Keyphrase Queries				
SciBERT (finetuned)	**16.6**	**32.7**	**23.5**	**42.8**
SciBERT (not finetuned)	0.0	0.0	0.0	0.0
COCO-DR (not finetuned)	6.2	13.9	9.6	16.8

Table 4: Finetuning for the dataset recommendation task significantly outperforms strong retrieval architectures finetuned for general search, like COCO-DR.

resentation. adding structured information to the dataset representation provides significant benefits for keyword search. This suggests keyword search requires more specific dataset metadata than full-sentence search does to be effective.

The value of finetuning Our bi-encoder retriever is finetuned on our training set. Given the effort required to construct a training set for tasks like dataset recommendation, is this step necessary?

In Table 4, we see that an off-the-shelf SciBERT encoder is ineffective. We observe that our queries, which are abstract descriptions of the user's information need (Ravfogel et al., 2023), are very far from any documents in the embedding space, making comparison difficult. Using a state-of-the-art encoder, `COCO-DR Base` — which is trained for general-purpose passage retrieval on MS MARCO (Campos et al., 2016), helps with this issue but still cannot make up for task-specific finetuning.

6 Related Work

Most work on scientific dataset recommendation uses traditional search methods, including term-based keyword search and tag search (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 the subset of datasets from their search corpus that have been posted on Papers with Code.

Some work has explored other forms of dataset recommendation. Ben Ellefi et al. (2016) study using "source datasets" as a search query, while Altaf et al. (2019) use a set of related research papers as the user's query. Färber and Leisinger (2021) are the only prior work we are aware of

that explores natural language queries for dataset recommendation. They model this task as classification, while we operationalize it as open-domain retrieval. Their dataset uses abstracts and citation contexts to simulate queries, while we use realistic short queries (with an expert-annotated test set).

7   Conclusion

We study the task of dataset recommendation from natural language queries. Our dataset supports search by either full-sentence or keyword queries, but we find that neural search algorithms trained for traditional keyword search are competitive with the same architectures trained for our proposed full-sentence search. An exciting future direction will be to make better use of natural language queries. We release our datasets along with our ranking systems to the public. We hope to spur the community to work on this task or on other tasks that can leverage the summaries, keyphrases, and relevance judgment annotations in our dataset.

Limitations

The primary limitations concern the dataset we created, which serves as the foundation of our findings. Our dataset suffers from four key limitations:

Reliance on Papers With Code Our system is trained and evaluated to retrieve datasets from Papers With Code Datasets (PwC). Unfortunately, PwC is not exhaustive. Several queries in our test set corresponded to datasets that are not in PwC, such as IWSLT 2014 (Birch et al., 2014), PASCAL VOC 2010 (Everingham et al., 2010), and CHiME-4 (Vincent et al., 2017). Papers With Code Datasets also skews the publication year of papers used in the *DataFinder Dataset* towards the present (the median years of papers in our train and test set are 2018 and 2017, respectively). For the most part, PwC only includes datasets used by another paper listed in Papers With Code, leading to the systematic omission of datasets seldom used today.

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 or influential works: the median number of citations of a paper in SciREX is 129, compared to 19 for any computer science paper in S2ORC. The queries in our test set are therefore more likely to describe mainstream ideas in popular subfields of AI.

Automatic tagging Our training data is generated automatically using a list of canonical dataset names from Papers With Code. This tagger mislabels papers where a dataset is used but never referred to by one of these canonical names (e.g. non-standard abbreviations or capitalizations). Therefore, our training data is noisy and imperfect.

Queries in English only All queries in our training and test datasets were in English. Therefore, these datasets only support the development of dataset recommendation systems for English-language users. This is a serious limitation, as AI research is increasingly done in languages other English, such as Chinese (Chou, 2022).

Ethics Statement

Our work has the promise of improving the scientific method in artificial intelligence research, with the particular potential of being useful for younger researchers or students. We built our dataset and search systems with the intention that others could deploy and iterate on our dataset recommendation framework. However, we note that our initial dataset recommendation systems have the potential to increase inequities in two ways.

First, as mentioned in Limitations, our dataset does not support queries in languages other than English, which may exacerbate inequities in dataset access. We hope future researchers will consider the construction of multilingual dataset search queries as an area for future work.

Second, further study is required to understand how dataset recommendation systems affect the tasks, domains, and datasets that researchers choose to work on. Machine learning models are liable to amplify biases in training data (Hall et al., 2022), and inequities in which domains or tasks receive research attention could have societal consequences. We ask researchers to consider these implications when conducting work on our dataset.

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*.

Alexandra Birch, Matthias Huck, Nadir Durrani, Nikolay Bogoychev, and Philipp Koehn. 2014. Edinburgh SLT and MT system description for the iwslt 2014 evaluation. In *IWSLT*.

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268.

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.

Daniel Chou. 2022. Counting AI research: Exploring ai research output in english- and chinese-language sources.

Mark Davies and Joseph L. Fleiss. 1982. Measuring agreement for multinomial data. *Biometrics*, 38(4):1047–1051.

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. 2010. The Pascal Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 88:303–338.

Michael Färber and Ann-Kathrin Leisinger. 2021. Recommending datasets for scientific problem descriptions. In *CIKM*, pages 3014–3018.

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.

Melissa R.H. Hall, Laurens van der Maaten, Laura Gustafson, and Aaron B. Adcock. 2022. A systematic study of bias amplification. *ArXiv*, abs/2201.11706.

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*, pages 7506–7516, Online. 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*.

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.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Conference on Empirical Methods in Natural Language Processing*.

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.

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.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

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.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

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.

Preslav Nakov, Ariel S. Schwartz, and Marti A. Hearst. 2004. Citances: Citation sentences for semantic analysis of bioscience text. In *SIGIR'04 Workshop on Search and Discovery in Bioinformatics*.

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).

Shauli Ravfogel, Valentina Pyatkin, Amir D. N. Cohen, Avshalom Manevich, and Yoav Goldberg. 2023. Retrieving texts based on abstract descriptions.

Stephen E. Robertson and Steve Walker. 1999. Okapi/Keenbow at TREC-8. In *TREC*.

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.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer. 2017. An analysis of environment, microphone and data simulation mismatches in robust speech recognition. *Computer Speech and Language*, 46:535–557.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021. Want to reduce labeling cost? gpt-3 can help. In *Conference on Empirical Methods in Natural Language Processing*.

Wei Zhong, Jheng-Hong Yang, Yuqing Xie, and Jimmy J. Lin. 2022. Evaluating token-level and passage-level dense retrieval models for math information retrieval. *ArXiv*, abs/2203.11163.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2019. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321.

A Few-Shot Prompt for Generating Keyphrases and Queries

When constructing our training set, we use in-context few-shot learning with the 6.7B parameter version of Galactica (Taylor et al., 2022). We perform in-context few-shot learning with the following prompt:

Given an abstract from an artificial intelligence paper:
1) Extract keyphrases regarding the task (e.g. image classification), data modality (e.g. images or speech), domain (e.g. biomedical or aerial), training style (unsupervised, semi-supervised, fully supervised, or reinforcement learning), text length (sentence-level or paragraph-level), language required (e.g. English)
2) Write a brief, single-sentence summary containing these relevant keyphrases. This summary must describe the task studied in the paper.

Abstract:
We study automatic question generation for sentences from text passages in reading comprehension. We introduce an attention-based sequence learning model for the task and investigate the effect of encoding sentence- vs. paragraph-level information. In contrast to all previous work, our model does not rely on hand-crafted rules or a sophisticated NLP pipeline; it is instead trainable end-to-end via sequence-to-sequence learning. Automatic evaluation results show that our system significantly outperforms the state-of-the-art rule-based system. In human evaluations, questions generated by our system are also rated as being more natural (i.e., grammaticality, fluency) and as more difficult to answer (in terms of syntactic and lexical divergence from the original text and reasoning needed to answer).

Output: (Task | Modality | Domain | Training Style | Text Length | Language Required | Single-Sentence Summary)
Task: question generation
Modality: text
Domain: N/A

Training Style: fully supervised
Text Length: paragraph-level
Language Required: N/A
Single-Sentence Summary: We propose an improved end-to-end system for automatic question generation.
–

Abstract:
We present a self-supervised approach to estimate flow in camera image and top-view grid map sequences using fully convolutional neural networks in the domain of automated driving. We extend existing approaches for self-supervised optical flow estimation by adding a regularizer expressing motion consistency assuming a static environment. However, as this assumption is violated for other moving traffic participants we also estimate a mask to scale this regularization. Adding a regularization towards motion consistency improves convergence and flow estimation accuracy. Furthermore, we scale the errors due to spatial flow inconsistency by a mask that we derive from the motion mask. This improves accuracy in regions where the flow drastically changes due to a better separation between static and dynamic environment. We apply our approach to optical flow estimation from camera image sequences, validate on odometry estimation and suggest a method to iteratively increase optical flow estimation accuracy using the generated motion masks. Finally, we provide quantitative and qualitative results based on the KITTI odometry and tracking benchmark for scene flow estimation based on grid map sequences. We show that we can improve accuracy and convergence when applying motion and spatial consistency regularization.

Output: (Task | Modality | Domain | Training Style | Text Length | Language Required | Single-Sentence Summary)
Task: optical flow estimation
Modality: images and top-view grid map sequences
Domain: autonomous driving
Training Style: unsupervised
Text Length: N/A
Language Required: N/A
Single-Sentence Summary: A system for self-supervised optical flow estimation from images and top-down maps.
–

Abstract:

In this paper, we study the actor-action semantic segmentation problem, which requires joint labeling of both actor and action categories in video frames. One major challenge for this task is that when an actor performs an action, different body parts of the actor provide different types of cues for the action category and may receive inconsistent action labeling when they are labeled independently. To address this issue, we propose an end-to-end region-based actor-action segmentation approach which relies on region masks from an instance segmentation algorithm. Our main novelty is to avoid labeling pixels in a region mask independently - instead we assign a single action label to these pixels to achieve consistent action labeling. When a pixel belongs to multiple region masks, max pooling is applied to resolve labeling conflicts. Our approach uses a two-stream network as the front-end (which learns features capturing both appearance and motion information), and uses two region-based segmentation networks as the back-end (which takes the fused features from the two-stream network as the input and predicts actor-action labeling). Experiments on the A2D dataset demonstrate that both the region-based segmentation strategy and the fused features from the two-stream network contribute to the performance improvements. The proposed approach outperforms the state-of-the-art results by more than 8

Output: (Task | Modality | Domain | Training Style | Text Length | Language Required | Single-Sentence Summary)
Task: actor-action semantic segmentation
Modality: video
Domain: N/A
Training Style: fully supervised
Text Length: N/A
Language Required: N/A
Single-Sentence Summary: I want to train a supervised model for actor-action semantic segmentation from video.
—

For a given abstract that we want to process, we then add this abstract's text to this prompt and ask the language model to generate at most 250 new tokens.

B Information on Expert Annotations

As mentioned in §3, we recruited 27 graduate students, faculty, and recent graduate program alumni for our annotation collection process. For each annotator, we received their verbal or written interest in participating in our data collection.

We then sent them a Google Form containing between 10 and 20 abstracts to annotate. An example of the form instructions is included in Figure 9.

We originally had annotators label the "Training Style" (unsupervised, semi-supervised, supervised, or reinforcement learning), in addition to Task, Modality, Domain, Text Length, and Language Required. However, this field saw excessively noisy labels so we ignore this field for our experiments.

DatasetFinder Annotation Form #21

We are developing a dataset search engine which accepts natural language descriptions of what the user wants to build. We need your help writing queries to test our search engine, and you will write each query based on a real, published research paper.

Given an abstract from an artificial intelligence paper:
1) **extract keyphrases regarding**:
   - the task (e.g. image classification)
   - data modality (e.g. images or speech)
   - domain (e.g. biomedical or aerial)
   - training style (unsupervised, semi-supervised, supervised, or reinforcement learning)
   - text length (sentence-level or paragraph-level)
   - language required (e.g. English)
2) **write a very short, single-sentence summary** that contains these relevant keyphrases, only including other information if critical to understanding the abstract. Do not include any information about model architecture or engineering decisions, beyond what is relevant to selecting a training/evaluation dataset.

Things to keep in mind:
   - We're providing you with a machine-generated "TLDR" of the abstract, as well as AI-generated suggestions for each field.
   - Feel free to skim the abstract rather than closely reading the whole thing, or even skip it if the TLDR is sufficiently informative.
   - Do not spend more than *2 minutes* in total on each example. If you find yourself taking too long to understand or tag a given abstract, just skip to the next one.
   - Do not mention any datasets by name.

Let's go through an example:
Abstract: Semantic image segmentation is an essential component of modern autonomous driving systems, as an accurate understanding of the surrounding scene is crucial to navigation and action planning. Current state-of-the-art approaches in semantic image segmentation rely on pre-trained networks that were initially developed for

Figure 9: Annotators each annotated 10-20 abstracts for our label collection using a Google Form with the instructions shown here..