

# Question Answering versus Named Entity Recognition for Extracting Unknown Datasets

YUSEF YOUNES <sup>1</sup>, ANSGAR SCHERP <sup>2</sup>

<sup>1</sup>GESIS – Leibniz-Institute for the Social Sciences, Cologne 50667, Germany

<sup>2</sup>University of Ulm, Ulm 89081, Germany

Corresponding authors: Yousef Younes (yousef.younes@gesis.org) and Ansgar Scherp (ansgar.scherp@uni-ulm.de).

This work was co-funded by the German Research Foundation (DFG) as part of the UnknownData Project - Grant No. 460676019.

**ABSTRACT** Dataset mention extraction is a difficult problem due to the unstructured nature of text, the sparsity of dataset mentions, and the various ways the same dataset can be mentioned. Extracting *unknown* dataset mentions which are not part of the training data of the model is even harder. We address this challenge in two ways. First, we consider a two-step approach where a binary classifier filters out positive contexts, i.e., detects sentences with a dataset mention. We consider multiple transformer-based models and strong baselines for this task. Subsequently, the dataset is extracted from the positive context. Second, we consider a one-step approach and directly aim to detect and extract a possible dataset mention. For the extraction of datasets, we consider transformer models in named entity recognition (NER) mode. We contrast NER with the transformers' capabilities for question answering (QA). We use the Coleridge Initiative "Show US the Data" dataset consisting of 14.3k scientific papers with about 35k mentions of datasets. We found that using transformers in QA mode is a better choice than NER for extracting unknown datasets. The rationale is that detecting new datasets is an out-of-vocabulary task, i.e., the dataset name has not been seen once during training. Comparing the two-step versus the one-step approach, we found contrasting strengths. A two-step dataset extraction using an MLP for filtering and RoBERTa in QA mode extracts more dataset mentions than a one-step system, but at the cost of a lower F1-score of 62.7%. A one-step extraction with DeBERTa in QA achieves the highest F1-score of 92.88% at the cost of missing dataset mentions. We recommend the one-step approach for the case when accuracy is more important, and the two-step approach when there is a postprocessing mechanism for the extracted dataset mentions, e.g., a manual check. The source code is available at [https://github.com/yousef-younes/dataset\\_mention\\_extraction](https://github.com/yousef-younes/dataset_mention_extraction).

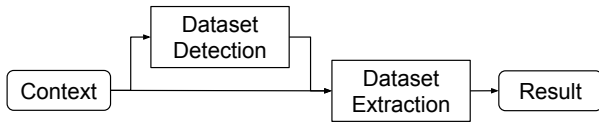
**INDEX TERMS** binary text classification, dataset mentions, named entity recognition, question answering

## I. INTRODUCTION

Datasets are very important assets in science due to their crucial role in reproducing and comparing research results. For these reasons, data reuse became one of the FAIR principles [1] agreed upon by stakeholders from academia, industry, and funding agencies in order to improve the findability, accessibility, interoperability, and reuse of digital assets. Specialized dataset search engines such as Google's Dataset Search [2] and GESIS' Datasearch [3] support the search for datasets based on metadata stored in dedicated dataset repositories and specific formats like Schema.org (<https://schema.org/>) or Dublin Core (<https://www.dublincore.org/>). This leads to a large number of undiscoverable datasets as they are newly published, i.e., mentioned in a scientific paper, and yet not

on the search engine's radar. The large amount of scientific papers published on a weekly basis makes it impossible to track the appearance of new datasets in the literature. Thus, it is desirable to automatically detect mentions of yet *unknown* datasets in scientific papers.

Detecting and extracting mentions of datasets in research papers is a challenging task for many reasons. First, there is no agreed-upon way to mention a dataset in scientific text. Although there are some suggested standards [4], researchers have different preferences for mentioning a dataset. Some use dataset names, acronyms, or a mixture of these. On top of that, there are cases where the mention is wrong, e.g., the authors may cite the primary publication of a large multi-part study but use a specific sub dataset only [5]. Second, only a



**FIGURE 1.** One-step versus two-step approach for detecting and extracting dataset mentions from some input text.

few sentences in a research paper mention a dataset, while most do not. This skewness in the classes makes dataset detection a finding-the-needle-in-the-haystack problem. Third, there are established approaches to detect already known datasets [6]–[10], but finding datasets that we are unaware of their existence makes the task more difficult. These datasets are not contained in the training data of our models.

Most of the existing works on extracting datasets considered the problem as a domain-specific named entity recognition (NER) task [6]–[9] and a few used question answering (QA) [10]. To the best of our knowledge, there are only two papers [9], [11] that handle the *unseen* dataset scenario using zero-shot learning, which is close to but different from our scenario. In our scenario, we have one class or entity type “dataset” for which many labeled samples of dataset mentions are available during training. For testing, we probe the models’ capabilities to generalize to new instances of that class i.e., unknown dataset mentions. In contrast, the assumption in zero-shot learning is that there are multiple classes and for some classes of interest there are no samples available during training.

We fill the gap on detecting and extracting the mentions of unknown datasets. We contrast the use of models using NER versus QA. Inspired by the prior works, we consider the detection and extraction of dataset mentions as a two-step approach [8] versus directly extracting the dataset mentions in a one-step approach [10], as illustrated in Figure 1. What makes our work different is the focus on a critical scenario and the use of language models in question answering mode for that purpose.

We use the Coleridge dataset from the Kaggle “Show US the Data” competition [12] to investigate our research questions. The Coleridge dataset resembles the traditional train-test scenario with known dataset mentions, so we had to preprocess its 43 unique dataset mentions for our scenario. There are about 200k contexts, i.e., sequences of 40 to 50 words in the 14.3k scientific papers in the dataset. Of those contexts, there are about 35k that have a dataset mention (positive class). This equals to only 18% of the total contexts. We split the dataset into five folds by partitioning over the set of unique dataset mentions, i.e., each fold holds contexts with a disjoint set of dataset mentions, and run a 5-fold cross-validation. This guarantees that the models are trained on dataset mentions that are different from the ones they are asked to recognize during testing.

We use six language models in different modes, i.e., classification, NER, and QA. Besides that, we use other methods

such as SVM [13] and MLP and off-the-shelf NLP tools like spaCy<sup>1</sup>. In addition, we test the effect of different techniques such as cost-sensitive learning, custom tokenization, and question engineering. Our experiments show that NER is not a good choice for extracting unknown dataset mentions, while QA performs better. This holds true for the one-step and two-step approaches. Concerning the two-step approach, detecting positive contexts in the first step, i.e., contexts containing a dataset mention, is best achieved with a recall of 93% when using BERT-mean embedding to represent contexts before feeding them into a multilayer perceptron network. We focus on the recall in this first step, as we subsequently can assume in the second step that the given context is positive, i.e., contains a dataset mention that can be extracted. Here, BERT-mean is the average of BERT tokens, instead of using the classical [CLS] token [14]. Concerning a two-step versus a one-step approach, we observe that the first extracts more dataset mentions than the latter. Using an MLP for dataset mention detection and RoBERTa in QA mode for extracting dataset mentions is overall the best combination for the two-step approach for dataset detection and extraction with 81.56% as the F1-score for the positive contexts but with the drawback of a lower F1-score of 62.7% when considering both positive and negative contexts. Regarding the one-step extraction approach, DeBERTa in QA mode achieves the highest F1-score of 92.88% but at the cost of missing dataset mentions. In summary, both the two-step and the one-step approaches have their biases and it depends on the use case what is the best. We recommend using the two-step approach for extracting dataset mentions since the goal here is to maximize discoverability of new datasets, i.e., the recall of finding unknown dataset mentions. This approach extracts more mentions with low accuracy so the results need postprocessing. If accuracy is more important, we recommend the one-step approach.

Below, we summarize the related work. Section III introduces our methods. The experimental apparatus is described in Section IV. An overview of the achieved results is reported in Section V. Section VI discusses the results, before we conclude.

## II. RELATED WORK

We review works that tackled dataset mention detection and extraction. To make the paper self-contained, we also briefly summarize the different techniques used in our methods, namely transformer models, training customized tokenizers, handling imbalanced classes, and ensemble models.

### A. DATASET MENTION DETECTION AND EXTRACTION

Most works on dataset extraction considered the task as domain-specific NER. The methods can be organized into those based on rules, general machine learning, and language models. Rule-based methods are the old way to tackle dataset mention extraction. They are still helpful, even with the vast

<sup>1</sup><https://spacy.io>

advancements in NLP, because of their explainability and transparency [15]. A recent example is ODDPub (Open Data Detection in Publications), a text-mining algorithm that uses keywords to screen biomedical publications and detect cases of Open Data [6]. The AllenAI approach [7] is an example of a deep learning approach and it is the winner of the first rich-context competition [16]. In this approach, King et al. extracted dataset mentions using a bidirectional long short-term memory (BI-LSTM) with a Conditional Random Field (CRF) [17] decoding layer. Similarly, Otto et al. achieved good performance using spaCy to extract the mentions [18].

Pre-trained language models achieved state-of-the-art results in many tasks, becoming the dominant NER approach. For example, Färber and others in [19] used SciBERT [20] to extract the dataset mentions. Different classifiers were used to distinguish datasets that are *used* and *unused*. The used datasets are actually analyzed in the paper, while the unused datasets are just cited. Similarly, Kumar et al. introduced a two-step system that uses SciBERT for detecting and extracting dataset mentions [8]. The first step is a classifier that selects the sentences which contain dataset mentions; the second step uses SciBERT in NER mode to extract the mentions. Alike, Heddes and others introduced a dataset and showed that SciBERT in NER mode outperforms the rule-based and traditional methods [9].

Unlike previous methods, the KAIST approach, which got the second rank in the first rich-context competition, considered the problem a question-answering one [10]. They introduce a system that uses Document QA [21] to extract dataset names. Document QA is a Machine Reading for Question-Answering (MRQA) model that selects the paragraphs most similar to the query based on TF-IDF [22]. Then it uses Bidirectional Gated Recurrent Units [23] along with self- and bi-attention mechanisms to extract multiple answers to a question. After that, they used an NER model to choose the correct answer.

In prior work, it was shown that RoBERTa with imbalance handling techniques could discover paper sections that have dataset mentions with acceptable performance (86% recall) [24]. The authors also indicated that the dataset's acronyms are being chopped into many tokens, affecting performance. In this work, we show that using custom tokenization helps identify the dataset's acronyms as one token.

Finally, we should point out that language models are not confined to plain text. For example, Starmie is a framework for discovering datasets from data lakes [25]. It uses a contrastive learning method to train column encoders from pre-trained language models in a fully unsupervised manner. It connects the representations of the same or unionable columns in the representation space while separating representations of distinct columns.

## B. UTILIZED TRANSFORMER MODELS

We will briefly mention the transformers utilized in this paper. They will be fine-tuned to do different NLP tasks such as Text Classification, Question-Answering, and NER.

BERT is a pre-trained language model which was trained on the Book Corpus and Wikipedia using two objectives Masked Language Model (MLM) and Next Sentence Prediction (NSP) [14]. RoBERTa is an optimized version of BERT trained on more data (160 GB) using dynamic masking and MLM as objectives [26]. DeBERTa (Decoding-enhanced BERT with disentangled attention) is a BERT extension that focuses on different aspects of the text using disentangled attention. It also uses decoding-enhanced training, which makes it good at generation tasks [27].

SciBERT is a specialized version of  $BERT_{base}$  trained on scientific and biomedical text with an additional objective of predicting the scientific concepts in the text [20]. MiniLM is a task-agnostic distilled version of  $BERT_{base}$ , where the transformer's last layer's self-attention distributions and value relations of  $BERT_{base}$  were used to guide its training. It managed to maintain 99% of  $BERT_{base}$  performance with double speed and fewer parameters [28].

Finally, we will use a version of the BERT model trained using SimCSE (Simple Contrastive Learning of Sentence Embeddings), a framework for learning high-quality sentence embeddings [29]. SimCSE uses a contrastive loss objective with different language models to bring similar sentences close to each other in the embedding space while pushing different sentences apart.

## C. TOKENIZATION FOR DATASET ACRONYMS

The tokenizer is the bridge between the data and the transformer model. It takes text as input and produces a numeric output for the language model. When adding domain-specific training data to the model, i.e., when fine-tuning the model, we always have the choice of using the tokenizer with which the model was pre-trained or modifying the original tokenizer. For example, the BERT tokenizer produces subtokens based on the WordPiece algorithm [14], which breaks down words into smaller subword units. The created subtokens form the tokenizer's vocabulary are dependent on the specific input text used for pre-training. When a tokenizer is faced with an input word that it is not prepared for, it will split that word into many subtokens from its vocabulary. The result is a long input sequence of many small subtokens (to form the word), which negatively affects the language model's performance. A common example where this happens is when acronyms are part of the input text [24].

We can overcome these two problems by modifying the tokenization process by adding new tokens to the tokenizer, training an existing tokenizer on our data, or building a new tokenizer [30]. While adding new tokens to the tokenizer is a suitable solution if the tokens to be added are known, training an existing tokenizer is a middle-ground solution that is less computationally expensive than building a new tokenizer. The third option is viable when there is enough data to train a model on the new tokenizer from scratch. In this work, we will make use of the first two options.

#### D. CLASS IMBALANCE IN DATASET DETECTION

There is an order of magnitude more sentences without dataset mentions than with. This requires using class balancing techniques, which can be categorized into Resampling and Cost-sensitive learning. Resampling is changing the data distribution in favor of the intended solution. This includes downsampling the majority class, upsampling the minority class, or generating new samples from the minority class(es) [31]. Sample generation is unnecessary in our case because we have enough contexts with dataset mentions, but they are few compared to the negative ones. Cost-sensitive learning techniques counter the imbalance via the loss. It changes the loss in favor of the minority samples, as in Balanced Cross Entropy [32], or in favor of the problematic samples for which the model is not confident like Focal Loss [33].

#### E. MODEL ENSEMBLES

Ensemble learning is a machine learning paradigm that combines multiple base models (a.k.a weak learners) to form a strong model with low bias and variance which can achieve better performance on a task. Ensemble models can be categorized into homogeneous which uses a single base model and heterogeneous which uses different base models. They also differ in the way the base models are combined. Here we can differentiate between three methods: Bagging, Boosting, and Stacking [34], [35]. Bagging (a.k.a Bootstrap aggregation) tries to reduce variance by operating on homogeneous models like Random Forest [36]. While Boosting operated on homogeneous models to reduce the bias, a well-known example of such a model is XGBoost [37]. These two methods use an averaging process to combine the results from the base models and are suitable for traditional machine learning algorithms. Unlike Bagging and Boosting, Stacking uses meta-models to combine the base models which can be heterogeneous, and that makes it suitable for combining deep learning models with traditional models. In this work, we try Stacking and Boosting for the detection task.

### III. MODELS

We explain how we utilize the different models and methods. First, we illustrate the techniques of imbalance handling in Section III-A and custom tokenization in Section III-B. Then we describe the models used in the experiments to perform dataset detection and extraction steps from Figure 1. Section III-C describes the models for the dataset detection and Section III-D the models involved in the dataset extraction in both NER and QA modes. The best-performing extractive model(s) that can discover contexts with dataset mentions will be considered a one-step approach. This model(s) will be combined with the best-performing detection model to form the two-step approach.

#### A. HANDLING IMBALANCE

We are only interested in the positive contexts containing dataset mentions, but our data is biased toward the negative

ones. We handle data imbalance by downsampling the negative contexts until their number equals the available positive ones. Besides that, cost-sensitive learning makes the model biased toward positive contexts. Particularly, we will be using Balanced Focal Loss (BFL) defined as  $BFL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$ , where  $p_t$  is the probability estimated by the model,  $\alpha_t$  is the balancing factor that considers the number of instances in each class, and  $\gamma$  is the focusing parameter that adjusts the values of the modulating factor  $(1 - p_t)$ . The modulating factor gives more weight to difficult examples. In the context of this paper, we will be using  $\alpha_t = 0.10$  and  $\gamma = 4$  which are chosen based on some initial experiments.

#### B. CUSTOM TOKENIZATION

In a previous work [24], it was found that the tokenizer chops the dataset acronym into many tokens, which makes the context longer with no additional information. So we customize the tokenizer by re-training it on the positive contexts that contain dataset mentions. This may help the tokenizer identify the dataset acronyms as depicted in Figure 2.

We use the customized tokenizer in two ways. First, we replace the original tokenizer with the newly trained one. Second, we extend the original tokenizer's vocabulary and model's embedding with a new vocabulary. The new vocabulary set is obtained by the difference between the vocabulary from the tokenizer trained on our data and the original tokenizer's vocabulary set.

#### C. MODEL FOR DATASET DETECTION

The dataset detection step from Figure 1 is materialized as a binary classifier. We fine-tune the base version of the language models BERT, RoBERTa, SciBERT, and DeBERTa (cf. Section II-B) in binary classification mode. After that, the effect of tokenization is tested on the best-performing model by extending its vocabulary and replacing its tokenizer. In addition, the model is trained using BFL (see Section III-A). Furthermore, the SimCSE version of that model is used to test the impact of the contrastive loss effect.

We experiment with various text representations like TF-IDF, BERT-mean, PCA [38], and  $t$ -SNE [39] for the input contexts with different models like SVM [13] and MLP-2. MLP-2 is a two-layer perceptron with dropout and ReLU activation function as expressed in Eq. (1). It is motivated by Galke et al. [40], which showed that a wide MLP with a large hidden layer size is a strong baseline model and even outperforms text classification models like TextGCN [41].

$$y = \text{Sigmoid}(\text{Dropout}(\text{ReLU}(\mathbf{X}_{n,d} \mathbf{W}_{d,g}^1)) \mathbf{W}_{g,2}^2) \quad (1)$$

Here,  $\mathbf{X}$  is the input embedding of  $n$  contexts each represented by a vector of  $d$  dimensions. We set  $d = 768$  to match the size of the BERT embeddings, since BERT-mean is used as input. The matrices  $\mathbf{W}^1$  and  $\mathbf{W}^2$  are the weights for the first and second layers, respectively. We aim to improve generalizability by over-parametrization, so we choose  $g = 1,024$ . The expanded vectors are passed through the ReLU activation function before it is dropped out with



Original BERT Tokenization:

```
['in', 'the', 'b', '##ls', '##a', 'p1', '##b', '-', 'pet', 'study', ',', 't', '##1', '-', 'weighted', 'volume', '##tric', 'magnetic', 'resonance', 'imaging', 'scans', 'were', 'co', '-', 'registered', 'to', 'the', 'mean', 'of', 'the', 'first', '20', '-', 'min', 'dynamic', 'pet', 'images', 'with', 'the', 'mutual', 'information', 'method', 'in', 'the', 'statistical', 'para', '##metric', 'mapping', 'software', '(', 'sp', '##m', '2', ',', 'well', '##com', '##e', 'department', 'of', 'imaging', 'neuroscience', ',', '']
```

Customized BERT Tokenization:

```
['in', 'the', 'blsa', 'pib', '-', 'pet', 'study', ',', 't1', '-', 'weighted', 'volumetric', 'magnetic', 'resonance', 'imaging', 'scans', 'were', 'co', '-', 'registered', 'to', 'the', 'mean', 'of', 'the', 'first', '20', '-', 'min', 'dynamic', 'pet', 'images', 'with', 'the', 'mutual', 'information', 'method', 'in', 'the', 'statistical', 'parametric', 'mapping', 'software', '(', 'spm', '2', ',', 'welcome', 'department', 'of', 'imaging', 'neuroscience', ',', '']
```

FIGURE 2. Comparison between BERT tokenization before and after training on our data.

a probability of 0.5. After that, the second layer's weight  $W^2$  transforms the input to feed it into the Sigmoid function, which converts the logits into class probabilities, to produce the output  $y$ . During training, Balanced Focal Loss is used to counter the imbalance nature of the data.

Finally, we use two ensemble models: XGBoost and an ensemble of our suggested MLP-2 model that combines three models, each with different settings. In addition to the version just described, the other two versions have (0.5, 0.3, 2) and (0.3, None, 0) for the (dropout,  $\alpha$ ,  $\gamma$ ), respectively. These models' predictions,  $z$ , are fed into the meta-model to combine them and produce the final ensemble probabilities,  $\text{pred}$ , as expressed in Eq. (2).

$$\text{pred} = \text{Sigmoid}(\text{ReLU}(W_2 \text{Dropout}(W_1 z))) \quad (2)$$

#### D. MODELS FOR DATASET EXTRACTION

We compare two approaches to materialize the dataset extraction step from Figure 1 as a NER or QA model. We use the base version of three language models: BERT, RoBERTa, and DeBERTa (cf Section II-B) in NER mode to extract the dataset mentions. We also use the NER system of the spaCy library to verify the results. The spaCy library uses a custom word embedding strategy, a transformer architecture, and a transition-based approach to named entity parsing [42]

Since the dataset mentions are part of the input text. Extractive question-answering [30] represents a possible solution for our problem. The QA model receives two inputs, the context that might contain the dataset mention and a question whose answer is that dataset mention when it exits. An extractive QA model produces the start and end indices with the highest score to indicate the answer span. The answer is an empty string in the following three cases: the start and end indices refer to the first token, the span is invalid (start index > end index) or the selected span is from the undesirable tokens like the question or padding tokens. While the first option indicates that the model is sure the question has no answer, i.e., empty string, the other two are erroneous. Thus, they are also interpreted as empty strings by postprocessing.

We start with the two-step system. In this case, we assume that there is a detector in the first step that passes on only positive context, i.e., those containing a dataset mention. Based on that, we remove the empty string from the list of possible answers produced by the QA model. That means the system produces an empty string when the QA model extracts an invalid span or span of undesirable tokens as the

answer. Again, we use the base version of BERT, RoBERTa, DeBERTa, SciBERT, and MiniLM (cf Section II-B) in QA mode to select the best-performing model(s). We use the selected models to test the impact of further optimizations, such as fixing the question or using BM25 to select different questions for different contexts. We also investigate the effect of tokenization (cf. Section III-B) by extending the vocabulary of the selected QA model.

#### IV. EXPERIMENTAL APPARATUS

We introduce the dataset in Section IV-A and explain how the data is prepared for the unknown dataset scenario. We describe the experimental settings in Section IV-B and the metrics to be reported in Section IV-C.

TABLE 1. Dataset characteristics separated by folds. We use four folds for training and one for testing. Note there is no overlap in the datasets mentioned in the folds, i.e., the pairwise intersection of dataset IDs between folds is empty.

Fold no.	# Positive Contexts	# Negative Contexts	Unique dataset IDs in Fold	# Total Contexts
0	13,699	32,858	8	46,557
1	4,932	27,468	8	32,400
2	5,685	34,443	9	40,128
3	4,972	31,604	9	36,576
4	6,625	36,767	9	43,392
Total	35,913	163,140	43	199053

#### A. DATASET AND DATA PREPARATION

##### 1) Dataset Description

We use the Coleridge initiative dataset "Show Us the Data" [43] in our experiments.<sup>2</sup> This dataset contains research papers annotated with the datasets mentioned in them. It has 14.3k unique papers; among them, 5.3k papers have multiple dataset mentions. These papers are stored in JSON files that wrap each section's title and content in a JSON object. In addition to the JSON files, a CSV file contains basic metadata (file name, paper title, dataset title, dataset label, cleaned dataset label) about each paper.

We target unknown datasets for which no data is available, so we must prepare the data for this scenario. This is described in the following.

<sup>2</sup><https://www.kaggle.com/competitions/coleridgeinitiative-show-us-the-data/overview>

## 2) Investigating Dataset IDs and Merging of IDs

First, we analyze the title and label attributes of the datasets. Subsequently, we consolidate different references to the same dataset under one ID. We have found 133 unique pairs (dataset title, dataset label) in the data. By inspecting the titles and labels individually, we found only 130 unique labels and 45 unique titles. That means we have 45 unique datasets each has one title and different labels. To make sure that we are referencing the same dataset, we consolidated all the possible dataset mentions (title and labels) that refer to the same dataset under one ID. We went further to see if one possible mention string is contained in another across the 45 datasets. We found that one dataset mention (“Educational Longitudinal Study”) whose assigned ID is 24, is contained in another (“National Educational Longitudinal Study”), whose ID is 1. These IDs are just numbers that we assign to unique datasets. Investigating these two datasets, we found that they come from the National Center for Education Statistics (NCES) and provide information on the educational experiences and outcomes of students in the United States from 1988 and 2002<sup>3</sup>. Since the names of the datasets and their origin suggest that it is actually referring to the same data, we decided to merge them under ID 1. Similarly, the “COVID-19 Death data” dataset with ID 40 is part of the “Our World in Data COVID-19” dataset, whose ID is 37. We merged the two datasets under ID 37. This leaves us with 43 unique dataset IDs. We also noticed that for some datasets, there are acronyms such as TIMSS, ECLS, NELS, etc., that are used in the papers but not listed as possible dataset mentions in the gold standard. We add these acronyms to the list of dataset mentions. Since the acronyms refer to datasets that are already contained in our list, they do not increase the total amount of unique datasets. We found that these acronyms have increased the number of positive contexts by 2%. In some examples, the acronyms are mentioned at the end of the string in parenthesis, e.g., Aging Integrated Database (AGID). In such a case, we keep the original string and add two strings “AGID” and “Aging Integrated Database” as additional mention strings for the dataset under focus.

## 3) Computing the Dataset Contexts

The dataset is annotated on the paper level, but we want to work on small text excerpts. The excerpt length should be longer than the longest possible dataset mention, which is 17 words, and also longer than the average sentence length, which is almost 26 words. Based on that, we choose a window size of 40 words and prepare the data for our experiments as follows: scan through the sections of each paper to extract the title and text. Then search the content for mentions of all the datasets in the gold standard. For each dataset, search for its different possible mentions ordered descendingly concerning their length because we are interested in finding the longest possible mention. If no mention is found, a span of 40 words is taken from the beginning of the section

```
{
  "id": "79061",
  "context": "Issue. However, existing surveys have limitations. For example, surveys that are able to bridge postsecondary education and employment, like the Baccalaureate and Beyond Longitudinal Study, are compiled infrequently; That study has followed groups of students who graduated in 1993 and 2000, and data collection",
  "question": "On which data is the study based?",
  "answers": [{"text": ["Baccalaureate and Beyond Longitudinal Study"], "answer_start": [145]},
  "label": 1,
  "masked_context": "Issue. However, existing surveys have limitations. For example, surveys that are able to bridge postsecondary education and employment, like the, are compiled infrequently; That study has followed groups of students who graduated in 1993 and 2000, and data collection"
```

FIGURE 3. An Example of Positive Context.

as a negative context. Otherwise, a positive context for a sample is constructed by extracting a prefix and suffix, each consisting of 20 words in length, before and after the found dataset mention. These samples have the following format: (id, context, masked\_context, question, answers, and label) as shown in Figure 3. The field "label" holds (1) to indicate that the context contains dataset mention and (0) when it does not; The "masked\_context" field contains the "context" with the dataset mention being removed. The "question" field contains the question to be sent to the question-answering model. To generate these questions, we compute the frequency of the words before and after the dataset mentions in all positive contexts. We use the most frequent words to construct the following five questions:

- 1) What data are used?
- 2) Is there any use of data collected from a survey?
- 3) Which dataset or database is used?
- 4) On which data is the study based?
- 5) Which data samples or images are used?

We use BM25 [22] with default settings to assign the best question to a context. Finally, the "answers" field holds the answers for the question and contains two subfields, "text" and "answer\_start". While the "text" subfield contains a list of dataset mentions in the context, the "answer\_start" field is another list that contains the start index of the dataset mentions in the "text" subfield. The "question" and "answers" fields are required for the question-answering task, while the "masked\_context", "label", and "context" fields will be used for the detection task.

## 4) Splitting the Dataset into Folds

Our task is to detect unknown dataset mentions. These unknown datasets must not appear during model training or a language model fine-tuning. Thus, a standard split of the dataset based on samples for each dataset ID in each fold is not possible.

Instead, we need to split the data such that the training and test data are disjoint w.r.t. the dataset mentions. This introduces a challenge regarding the distribution of the dataset mentions. We know that some datasets are more dominant than others. For example, datasets with IDs 6 and 8 have a support of 13,184 and 982, respectively. This affects the distributions of the train and test sets. As a result, reporting on one test set is unreliable. To avoid that, we use 5-fold cross-validation to make sure that our results are stable.

<sup>3</sup><https://nces.ed.gov/surveys/nels88/>

We divide the data into five disjoint folds with respect to the dataset IDs. To achieve that, we divide the 43 dataset IDs based on their support into three subsets. A subset with five IDs each with support higher than 2,000. A second subset contains 23 IDs with support in the range [100, 1,000]. A third subset contains 15 IDs with support in the range [1, 100). We use these subsets to construct five folds: for every fold, we randomly select a dataset ID from the first IDs-subset, 4-5 from the second subset, and 3 from the third subset. We remove the already used dataset IDs and repeat the process until all IDs are included in the folds. This method guarantees 8-9 unique dataset IDs in each fold, as shown in Table 1. In consequence, the resulting folds are not the same size. Finally, we add to each fold one-fifth of the negative samples, i.e., contexts that do not contain a dataset mention.

As an alternative to the imbalanced sizes of the folds, we could have removed positive contexts whose mentioned datasets' IDs have high support. Still, we decided to leave the folds with different sizes because it is more realistic to have some datasets mentioned more often than others.

### B. PROCEDURE AND HYPERPARAMETERS

The experiments in this paper are conducted using 5-fold cross-validation. We train our models on 4-folds for three epochs and use the remaining fold for testing. During training, we use AdamW [44] with default settings and a learning rate of  $2 \cdot 10^{-5}$  for language models in classification, NER, and QA modes. The batch size for classification and NER is 16, while it is 12 for the QA model. This choice is bounded by the available hardware capacity. We use a gradient accumulation of two steps to improve the effective batch size. In addition, the suggested MLP-2 classifier (cf. Section III-C) was trained for five epochs using the same optimizer with a learning rate of  $5 \cdot 10^{-5}$  and batch size of 100. The experiments were conducted on a 4-GPU machine equipped with four Geforce RTX 2080 Ti GPUs, each with 11 GB of memory. Hyperparameters such as the number of neurons in each layer in MLP-2, the learning rate, batch size, and the regularization are selected based on some initial experiments. We experimented with different values and the values with the best results were chosen.

### C. METRICS

We report F1-score, recall, and precision for the NER and binary classification tasks. For the question-answering task, we use the SQUAD 2.0 metrics [45]: exact-match and F1-score. F1-Score is the harmonic mean of precision and recall. In binary classification, the metrics are reported at the class level. The focus is on the recall of the positive contexts (P), because we are interested in identifying contexts with dataset mentions. For question-answering, we focus on the F1-score. This is because it is more aligned with human judgments since it measures the token's overlap between the true and predicted answers. As in classification, the F1-score in question answering is the harmonic mean. The difference is in how precision and recall are computed. The precision

is calculated by dividing the number of common tokens between gold and predicted answers (True Positives or TP) by the number of predicted tokens. In contrast, the recall is calculated by dividing the number of common tokens (TP) by the number of gold tokens. Since the F1-Score is computed for each prediction and then averaged, it is a Macro-average score [46]. Unlike the F1-score, the exact match is a strict metric. It is true only when there is an exact match between the predicted and true answer. In other words, one different character is enough to violate the exact match. These metrics will be calculated on positive and negative contexts individually and then combined. Note that both QA metrics normalize the strings, i.e., lowering casing and removing punctuation, articles, and extra whitespace.

Since the results are the average of five-fold cross-validation, the standard deviation is important. Results with a lower standard deviation are preferable because a lower standard deviation indicates the model's stability over the five folds.

## V. RESULTS

We first report the dataset detection results of the two-step approach in Section V-A. This is followed by the dataset extraction results using NER in Section V-B and using QA in Section V-C. Finally, a comparison between the one- and two-step approaches are reported in Section V-D.

### A. DATASET DETECTION RESULTS

The binary dataset detection step aims at filtering out the positive context in the input and resembles the first step in the two-step approach. The results for this binary detection step are shown in Table 2. In this table and other result tables, "Negative" refers to the contexts that do not contain dataset mentions, whereas "Positive" refers to those that contain dataset mentions. The results are organized into four groups. In the first group, we apply the model on the data as provided. In the subsequent groups, we use balanced data. Furthermore, in the first group, we fine-tune the base versions of four transformer models BERT, SciBERT, RoBERTa, and DeBERTa using the masked contexts that have the dataset mentions removed as described in Section IV-A3. Among these models, BERT achieved the highest recall for the positive contexts. Here, we observe that BERT without fine-tuning achieves better results. In the second group, we select the best-performing model from the first group, BERT, and optimize it using different techniques. We use a custom tokenizer trained on our dataset, extend the vocabulary of the original tokenizer, experiment with MLP-2 that uses a balanced focal loss as classification head, and test the BERT-based version trained by SimCSE, which uses contrastive loss. The SimCSE-trained model is tested with and without fine-tuning, while the others are fine-tuned. We find that BERT's results improve when using a custom tokenizer and extended vocabulary (cf. Section III-B). It performs even better with MLP-2 and BFL where it achieves 60% recall for

**TABLE 2.** Binary Detection Results. The experiments in the first group of the table are used to select the baseline, while the ones in the second group use the selected baseline BERT to test the effect of different optimization techniques. The third group uses different embeddings as input to the considered models. The last group uses ensemble models on BERT-Mean. The average percentage results of 5-fold cross-validation are reported along with their standard deviations.

Model	Precision (SD)		Recall (SD)		F1-Score (SD)	
	Negative	Positive	Negative	Positive	Negative	Positive
<i>Transformer models (Baselines)</i>						
BERT without finetuning	4.60 (7.00)	13.60 (9.00)	57.00 (44.41)	<b>42.00 (45.61)</b>	55.40 (40.97)	17.20 (0.16)
BERT w. finetuning	86.80 (6.43)	85.60 (4.59)	98.80 (0.40)	33.00 (8.65)	92.40 (3.72)	46.60 (0.09)
SciBERT w. finetuning	86.60 (6.86)	83.00 (6.26)	98.80 (0.40)	31.20 (10.61)	92.20 (4.17)	44.20 (0.13)
RoBERTa w. finetuning	82.60 (5.82)	84.40 (16.55)	100.00 (0.00)	0.00 (0.00)	90.40 (3.72)	0.40 (0.00)
DeBERTa w. finetuning	82.60 (5.82)	93.40 (13.20)	100.00 (0.00)	0.00 (0.00)	90.40 (3.72)	0.00 (0.00)
<i>Optimizations of BERT</i>						
BERT + custom tokenizer	85.20 (14.15)	34.40 (14.57)	72.40 (19.55)	52.80 (15.70)	77.60 (16.75)	39.80 (0.13)
BERT + extended vocabulary	88.80 (8.95)	49.60 (13.54)	85.80 (10.11)	55.20 (11.30)	87.00 (9.23)	51.60 (0.11)
BERT+MLP-2+BFL	85.80 (9.43)	23.00 (2.76)	57.60 (11.24)	<b>60.20 (9.06)</b>	68.60 (10.07)	32.60 (1.96)
SimCSE without fine-tuning	83.20 (4.96)	20.20 (9.15)	72.80 (9.02)	29.80 (7.78)	77.40 (6.71)	23.20 (8.18)
SimCSE with fine-tuning	83.00 (6.10)	48.80 (29.80)	99.20 (1.17)	4.60 (5.12)	90.20 (3.76)	8.60 (8.91)
<i>Baselines with BERT-mean</i>						
SVM On bert-mean	85.00 (12.55)	32.20 (8.63)	72.80 (18.71)	51.00 (8.60)	77.80 (16.17)	38.60 (7.53)
SVM On tsne 3 components	88.80 (4.62)	25.00 (8.17)	59.40 (3.14)	65.00 (6.03)	71.00 (2.10)	35.80 (7.57)
SVM On PCA 32 components	89.80 (3.31)	27.00 (8.12)	62.40 (2.06)	66.60 (6.09)	73.60 (2.58)	38.20 (8.13)
MLP-2 with BERT-mean and FL	93.60 (7.81)	23.20 (4.07)	35.40 (15.16)	<b>93.40 (1.74)</b>	50.00 (17.96)	37.60 (5.08)
MLP-2 with TF-IDF and FL	70.40 (35.27)	17.80 (5.64)	11.40 (6.62)	90.20 (7.22)	19.60 (11.04)	29.60 (8.31)
<i>Ensembles</i>						
Ensemble of 3 MLP-2 models	89.00 (13.52)	35.00 (7.13)	70.40 (15.15)	<b>73.33 (15.15)</b>	78.60 (14.53)	46.80 (9.70)
XGBoost on bert-mean	87.20 (12.12)	31.40 (6.71)	71.20 (13.44)	60.20 (14.74)	78.40 (12.45)	41.00 (8.83)

the positive contexts. On the contrary, SimCSE shows lower performance.

The third group of Table 2 shows that using BERT-mean as input for different models brings some improvement, especially when using PCA with 32 dimensions and *t*-SNE of the BERT-mean as input to SVM. Since we aim at the highest possible recall of the positive contexts, we use the MLP-2 model described in Section III-C. This model achieves a recall for the positive contexts of 93% with high stability across folds (SD=1.7) but with low values concerning almost all other measures. We also find that using TF-IDF as input to MLP-2 performs lower than using BERT-mean. The fourth and last group reports the results of using two ensemble models: XGBoost and our ensemble model of three MLP-2 models as described in Section III-C. Our ensemble model performed better than XGBoost, resulting in almost similar recall for both classes, but almost 30% of both contexts are misclassified.

In summary, we can state that the best-performing model for the recall of the positive contexts is our MLP-2 model when applied to BERT-mean.

## B. NER RESULTS

We aim to find the best model(s) that can be used for the dataset extraction step depicted in Figure 1. We use the base versions of BERT, RoBERTa, and DeBERTa (cf. Section II-B) in NER mode, and report the results in Table 3. The results show that among the used language models DeBERTa achieves the best recall (26.4%), while RoBERTa achieves the best precision (63.14%), but both are low. spaCy performs better than language models achieving a recall of 35.33%.

**TABLE 3.** The average percentage NER results of 5-fold cross-validation are reported along their standard deviations (SD).

Model	Precision (SD)	Recall (SD)	F1-Score (SD)
BERT	38.00 (23.95)	19.52 (17.12)	24.90 (20.89)
RoBERTa	66.13 (22.82)	21.51 (15.90)	30.42 (19.20)
DeBERTa	63.00 (17.62)	26.40 (18.79)	34.60 (23.16)
spaCy	46.12 (19.64)	35.33 (18.15)	39.22 (18.30)

## C. QA RESULTS

As alternative to NER, we use question-answering (QA) to extract dataset mentions. For that, we run the experiments assuming that the contexts are positive, i.e., have dataset mentions. The results in Table 4 are organized into three groups.

The first group shows the best-performing model concerning positive contexts. The experiments use the question q1 from Section IV-A3 with five language models from Section II-B. Among these models, DeBERTa and RoBERTa show competitive performance. Although DeBERTa achieves the best F1-Score (85.23%) for the positive contexts, RoBERTa outperforms DeBERTa in the negative contexts (39.51% vs. 3.81%) with only a 1% loss for the positive ones. In addition, Roberta outperforms the best NER model in Table 3 with respect to the overall F1-score. Based on that, we focus on QA and consider both DeBERTa and RoBERTa for further experiments.

The second group shows the impact of the five different questions from Section IV-A3 on both DeBERTa and RoBERTa. When fixing the question for both models, the results concerning the F1-score of the positive contexts show that questions q3 and q4 work the best for DeBERTa and RoBERTa, respectively. As before, DeBERTa outperforms



RoBERTa concerning the F1-score of the positive contexts (F1-score for the positive class) with 2.89%, and it is also more stable when comparing the standard deviations. Nevertheless, RoBERTa is much better than DeBERTa regarding the F1-score of the negative contexts (F1-score for the negative class) with 38.46% increase. We run further experiments with both models but do not fix the question for all contexts this time. Rather, we use the question that best matches the context according to BM25 (cf. Section IV-A3). The results show that using different questions for different contexts decreases the performance of both models. Thus, the best performance is achieved when fixing the question to all contexts.

The third group shows the effect of extending the vocabulary of DeBERTa and RoBERTa as described in Section III-B. The results show that the extended vocabulary affects the two models differently. It slightly affects the F1-score for the positive contexts for RoBERTa but makes it more stable, but has unfavorable effects concerning the negative contexts. For DeBERTa, it seems to have a bad effect on all measures.

#### D. ONE-STEP VS. TWO-STEP APPROACH

In this section, we combine the best detector and extractor models in a two-step approach and compare their performance to the one-step approach. In the two-step approach, the extractor assumes that its input has a dataset mention because the detector is supposed to select the positive contexts. In the one-step approach, the extractor does not assume that the context is positive, so it has the empty string as a possible answer to indicate that there is no dataset to be extracted.

The MLP-2 classifier was the best-performing filter concerning the recall of the positive contexts, and both DeBERTa and RoBERTa models show competitive performance regarding dataset extraction. We combine MLP-2 with both RoBERTa and DeBERTa. We also consider each of RoBERTa and DeBERTa in QA mode as a one-step approach that can detect negative contexts and extract dataset mentions from the positive ones.

Table 5 shows the results of using the one-step versus two-step approaches. Although the DeBERTa model in QA mode as a one-step approach achieves an overall F1-score of 92.88%, it has only an F1-score of 69.98% for the positive contexts. In contrast, the best F1-score achieved by the two-step approach that combines MLP-2 and RoBERTa is 62.7%, but it achieves an 81.56% F1-score for the positive contexts.

## VI. DISCUSSION

We first discuss the results for the two-step approach, namely the dataset detection and dataset extraction, before reflecting on the one-step approach. Finally, we discuss on the threats to validity and limitations of our study.

#### a: Dataset Detection

Regarding the binary detection results in Table 2, data imbalance could explain the bad performance of RoBERTa

as shown by Han et al. [47]. Yet it is necessary to use imbalanced data because the model has to deal with it when put in production. Similarly, the disentangled attention seems to impact DeBERTa's performance negatively because it has different weights for the different aspects of the input. This makes DeBERTa less sensitive to the input variations, which is important in our scenario as motivated in the introduction.

The tailored tokenization improved BERT's detection performance. It helps identify dataset acronyms as one token, which enables feeding more text into the model. For example, the dataset acronym "BLSA" was chopped into three tokens by the original BERT tokenizer, whereas it was considered one token by the custom tokenizer as shown in Figure 2.

MLP-2 achieved better recall for the positive contexts than language and ensemble models. This is in line with Galke et al. [40], which showed that using BoW and TF-IDF as input to a wide MLP network is better than using language models for text classification. We investigate the performance of the MLP-2 on the different folds to get a deeper understanding of its performance on the different dataset mentions. Table 6 shows how MLP-2 works on different folds. The model shows similar performance on all folds except when Fold 0 is used for testing, which can be seen from the high number of false negatives. False negatives are the contexts that have dataset mentions but are not detected as such. Fold 0 has the highest number of positive contexts as shown in Table 1. It has 13,699 positive contexts, of which 13,184 mention the dataset with ID 6 "Alzheimer's Disease Neuroimaging Initiative (ADNI)". It turns out that this dataset is the reason for the bad performance of the model on this fold and consequently on the reported average results. The low performance on the Alzheimer dataset could be explained by the high number of occurrences and the different mentioning forms like ADNI, ANDI-1, and ADNI-GO/2 beside other long mention forms.

#### b: Dataset Extraction

Based on the results in Table 3, we can say that NER is unsuitable for extracting unknown dataset mentions. Out-Of-Vocabulary (OOV) represents the most probable reason because the model is asked to recognize instances of the dataset entity that are not seen once during fine-tuning. These could be new acronyms and vocabulary which the model is unfamiliar with. Previous works like [48], [49] have pointed that out as a challenge for NER models. Another reason is the high similarity between positive and negative contexts, which is obvious from the low performance of the binary detection task.

QA is a better option than NER for extracting dataset mentions. From Table 4, we see that DeBERTa with question three (q3) is the best-performing and most stable model. Fixing the models to use one question for all contexts proved to be better than using different questions, even if they fit the context better. We reckon that fixing the questions helps the QA model overfit the question and focus on changing contexts. Unlike its effect on the detection task, tokenization does not have a promising impact on QA.

**TABLE 4.** Unknown Dataset Mention Extraction Results Using Question Answering. BQ stands for the best matching question obtained by BM25. ext\_vocab indicates that the tokenizer and model are extended with new vocabulary. The average percentage results over the 5-fold cross-validation are reported along with their standard deviations. The lines separate different groups of experiments.

Model	Exact Match (SD)		F1-Score (SD)		Overall	Overall
	Negative	Positive	Negative	Positive	Exact Match (SD)	F1-Score (SD)
<i>Transformer models (Baselines)</i>						
Bert + q1	5.69 (2.49)	70.85 (22.29)	5.69 (2.49)	83.66 (14.96)	15.95 (1.85)	18.62 (2.55)
SciBert + q1	2.76 (2.07)	69.1 (24.33)	2.76 (2.07)	82.20 (18.60)	13.14 (2.35)	15.76 (1.95)
Minilm + q1	0.16 (0.24)	64.13 (31.89)	0.16 (0.24)	78.84 (22.76)	9.79 (4.62)	12.88 (1.61)
RoBERTa + q1	<b>39.51 (21.17)</b>	71.28 (20.17)	39.51 (21.17)	84.00 (13.81)	43.61 (17.89)	46.21 (17.76)
DeBERTa + q1	3.81 (1.53)	76.79 (12.57)	3.81 (1.53)	<b>85.23 (8.93)</b>	16.35 (1.80)	18.05 (2.49)
<i>Choice of Questions</i>						
RoBERTa						
q1	54.02 (4.51)	70.70 (21.54)	54.02 (4.51)	83.51 (15.28)	55.79 (5.54)	58.40 (4.32)
q2	52.58 (5.10)	72.10 (16.12)	52.58 (5.10)	83.92 (14.69)	55.10 (4.54)	57.18 (3.98)
q3	47.52 (20.23)	69.62 (24.39)	47.52 (20.23)	81.75 (22.21)	49.17 (13.28)	51.39 (13.70)
q4	41.32 (19.54)	<b>74.64 (15.14)</b>	41.32 (19.54)	<b>86.25 (11.28)</b>	46.16 (16.62)	48.40 (16.93)
q5	34.44 (24.33)	72.66 (17.87)	34.44 (24.33)	83.97 (14.73)	39.64 (20.16)	41.81 (20.22)
DeBERTa						
q1	3.46 (2.40)	78.89 (19.69)	3.46 (2.40)	85.89 (15.60)	15.21 (2.31)	17.24 (2.14)
q2	0.34 (0.16)	69.75 (25.44)	0.34 (0.16)	84.66 (17.06)	11.31 (2.62)	14.40 (0.97)
q3	2.86 (2.19)	<b>79.86 (10.87)</b>	2.86 (2.19)	<b>89.14 (7.49)</b>	16.07 (3.36)	17.91 (4.21)
q4	1.07 (1.21)	74.06 (26.93)	1.07 (1.21)	83.87 (20.96)	12.53 (3.39)	14.61 (1.52)
q5	1.38 (1.06)	71.67 (25.66)	1.38 (1.06)	80.47 (23.67)	12.39 (1.97)	14.05 (0.84)
RoBERTa + BQ	27.14 (32.34)	73.35 (16.60)	27.14 (32.34)	<b>86.26 (12.23)</b>	35.27 (26.94)	37.76 (26.50)
DeBERTa + BQ	2.69 (2.06)	<b>75.49 (18.13)</b>	2.69 (2.06)	85.94 (13.56)	14.55 (1.41)	16.67 (2.97)
<i>Tokenization Effect</i>						
RoBERTa + ext_vocab + q4	13.65 (13.73)	<b>68.12 (14.99)</b>	13.65 (13.73)	<b>86.51 (7.48)</b>	23.24 (12.53)	26.72 (11.10)
DeBERTa + ext_vocab + q3	1.98 (2.18)	68.55 (24.95)	1.98 (2.18)	78.52 (23.15)	12.51 (3.80)	14.32 (2.87)

**TABLE 5.** Comparison of the One- and Two-step Approaches Using Question Answering. The average percentage results over 5-fold cross-validation are reported along with their standard deviations.

Approach	Exact Match (SD)		F1-Score (SD)		Overall	Overall
	Negative	Positive	Negative	Positive	Exact Match (SD)	F1-Score (SD)
Two-step						
MLP + RoBERTa + q4	59.70 (8.06)	70.38 (17.35)	59.70 (8.06)	81.56 (14.36)	<b>60.62 (7.43)</b>	<b>62.70 (7.18)</b>
MLP + DeBERTa + q3	31.19 (10.24)	75.71 (14.63)	31.19 (10.24)	85.21 (10.21)	38.74 (7.77)	40.64 (7.34)
One-step						
RoBERTa + q4	99.59 (0.05)	60.13 (30.46)	99.59 (0.05)	65.59 (31.84)	90.84 (10.09)	91.68 (10.30)
DeBERTa + q3	99.84 (0.10)	66.70 (26.67)	99.84 (0.10)	69.98 (28.05)	<b>92.39 (8.83)</b>	<b>92.88 (9.03)</b>
Deberta + q3 + without ADNI	99.84 (0.10)	73.84 (13.79)	99.84 (0.10)	77.29 (14.60)	97.18 (1.55)	97.66 (1.38)

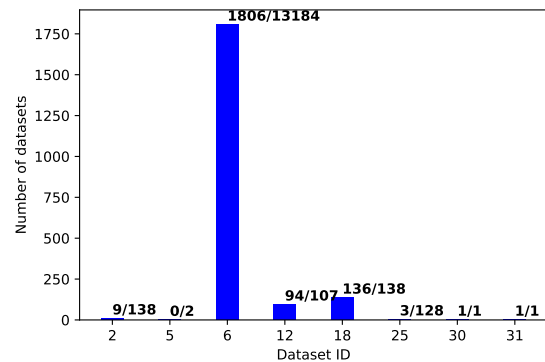
**TABLE 6.** Classification Statistics using BERT-mean as input to MLP-2 model.

fold	TP	TN	FP	FN
0	11,162	5,915	26,943	2,537
1	4,818	4,958	22,494	112
2	5,532	11,043	23,391	151
3	4,763	11,832	19,768	207
4	6,180	16,215	20,517	442
Avg.	6,491	9,992.6	22,622.6	689.8
SD	2,674.86	4,614.62	2,822.02	1,040.53

### c: One-step Approach

As shown in Table 5, DeBERTa in the one-step approach achieves an F1-score of 92.88%. Although it achieves an F1-score of 69.98% for the positive contexts over the five folds, its F1-score for the positive contexts is only 16.11% on fold 0. Again this fold is a challenge due to the skewed distribution (see above). Thus, we investigate it in more detail.

Figure 4 shows the datasets found when using DeBERTa in question answering mode on fold 0 with q3 as the question.



**FIGURE 4.** Dataset mentions Extracted from Fold 0 using DeBERTa with q3. The numbers x/y indicate that the dataset was extracted x times out of y mentions.

Like with MLP-2, again dataset with ID 6 is causing the low performance. The model could extract it in full string 1,507 times and in acronym form 299 times. In total, it

was extracted 1,806 out of 13,184 occurrences. Although the dataset with ID 6 is the reason behind the low reported result, this is still useful in terms of detecting unknown datasets. Because for an application that collects the mentions of new datasets, it is sufficient to observe a dataset mention once in order to add it to a list of known datasets. Then further training could enable the model to handle it better or even an exact or partial string matching can be used to search for it.

Noteworthy are the datasets whose IDs are 5 (“Aging Integrated Database”) and 25 (“National Assessment of Education Progress”). Dataset 25 is mentioned in 125 contexts but is only extracted three times. While dataset 5 is not extracted at all from its two occurrences. A possible explanation is that their contexts differ from those DeBERTa can recognize. Since dataset 25 has 125 contexts, it has a good chance that some of its contexts are similar to the one the model is familiar with and can extract it. But with only two contexts, the model could not extract dataset 5.

Looking at the false positives, we found that DeBERTa could extract dataset mentions that were actually true but also were not part of the gold standard. For example, we found the datasets “Korean Longitudinal Study on Health and Aging”, “Uppsala Longitudinal Study of Adult Men (ULSAM)”, and “English Longitudinal Study of Ageing (ELSA)”. This indicates the imperfectness of the used dataset. At the same time, it indicates that our method is generalizable and not confined to the dataset that we are using.

#### d: Threat to Validity and Limitations

This work focuses on a special scenario where the model is not aware of the existence of a dataset but has to detect and extract its mention. We had to prepare our study to address that scenario. We found the Coleridge “Show Us the Data” dataset to investigate the research question. The use of a single dataset could be seen as a threat to validity. We address that by using 5-fold cross-validation. Each run starts with a fresh model and uses a different test set. This makes every run an independent experiment. We choose a 5-fold over 10-fold cross-validation to reduce the computational overhead and create folds with comparable amounts of positive contexts. Using 10-fold cross-validation will generate folds with around four unique datasets. Given the different support of individual datasets, some folds will have a few positive contexts and likely produce unreliable positive or negative results.

We showed that DeBERTa in a one-step approach was able to extract dataset mentions that are not in the gold standard. This indicates that our data is not perfect. Unfortunately, there is no complete list of dataset mentions that we can use to perfect the gold standard. We also argued that one dataset “Alzheimer’s Disease Neuroimaging Initiative (ADNI)” is problematic for that model. Removing that dataset from fold 0 increases the F1-score for the positive context of DeBERTa over fold 0 from 16.11% to 52.65%. This results in a 7% increase for the F1-score of the positive contexts as shown in the last line of Table 5. It also increases the F1-score for both

contexts from 92.88% to 97.66% and makes it more stable with a standard deviation of 1.38.

## VII. CONCLUSION AND FUTURE WORK

This paper focuses on extracting unknown dataset mentions from small excerpts of scientific texts. It shows that QA works better than NER in this scenario. It also shows tokenization is a good method to detect dataset acronyms and can improve detecting whether a context contains a dataset mention. Finally, it compares a one-step approach that uses language models like DeBERTa or RoBERTa in QA mode with a two-step approach that uses MLP-2 as a dataset detector and DeBERTa or RoBERTa as an extractor. Although the two-step approach can extract more mentions than the one-step approach, it is less accurate. In conclusion, we recommend using the two-step approach for extracting datasets since the goal here is to maximize the discoverability of new datasets. If accuracy is more important, we recommend the one-step approach.

We limited our experiments to language models like BERT with admissible memory needs. This is because we plan to deploy the model as part of a service. The service will take a paper in PDF format as input, and extract contexts from its text using some heuristics taking into consideration the 40 words limit and sentence splitting of the paper’s text. For example, we search the text for a keyword like data, dataset, etc.. and take a span of at least 20 words before and after the keyword to form a context. We could take more than 40 words to avoid cutting a sentence in the middle. For each context, the model will either extract a substring of the context as a possible dataset mention indicating a positive context or will produce an empty string indicating a negative context. We hypothesize that the bigger the language model we use, the higher the chances of extracting dataset mentions. To test this hypothesis, we plan to use in future work among others some autoregressive models [50].

## REFERENCES

- [1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, et al., “The fair guiding principles for scientific data management and stewardship,” in *Scientific data*, vol. 3, no. 1, 2016, pp. 1–9.
- [2] T. Alrashed, D. Paparas, O. Benjelloun, Y. Sheng, and N. Noy, “Dataset or not? a study on the veracity of semantic markup for dataset pages,” in *The Semantic Web – ISWC 2021*, A. Hotho, E. Blomqvist, S. Dietze, et al., Eds., Cham: Springer, 2021, pp. 338–356, ISBN: 978-3-030-88361-4.
- [3] T. Krämer, C.-P. Klas, and B. Hausstein, “A data discovery index for the social sciences,” *Scientific Data*, vol. 5, no. 1, pp. 1–10, 2018.
- [4] M. Altman and G. King, “A proposed standard for the scholarly citation of quantitative data,” *D-lib Magazine*, vol. 13, no. 3/4, 2007.
- [5] K. Boland, D. Ritze, K. Eckert, and B. Mathiak, “Identifying references to datasets in publications,” in *International Conference on Theory and Practice of Digital Libraries*, Springer, 2012, pp. 150–161.
- [6] N. Riedel, M. Kip, and E. Bobrov, “Oddpub—a text-mining algorithm to detect data sharing in biomedical publications,” *bioRxiv*, pp. 2020–05, 2020.
- [7] D. King, W. Ammar, L. Beltagy, C. Betts, S. Gururangan, and M. Zuylen, “Finding datasets in publications: The allen institute for artificial intelligence approach,” in *Rich search and discovery for research datasets: Building the next generation of scholarly infrastructure*, J. Lane, I. Mulvany, and P. Nathan, Eds., Sage London, 2020, ch. 6, pp. 83–92.

- [8] S. Kumar, T. Ghosal, and A. Ekbal, "Dataquest: An approach to automatically extract dataset mentions from scientific papers," in *Towards Open and Trustworthy Digital Societies: ICADL*, Springer, 2021, pp. 43–53.
- [9] J. Heddes, P. Meerdink, M. Pieters, and M. Marx, "The automatic detection of dataset names in scientific articles," *Data*, vol. 6, no. 8, 2021, ISSN: 2306-5729.
- [10] H. Puerto-San-Roman, G. Hong, M. Cao, and S. Myaeng, "Finding datasets in publications: The kaist approach," in *Rich search and discovery for research datasets: Building the next generation of scholarly infrastructure*, J. Lane, I. Mulvany, and P. Nathan, Eds., Sage London, 2020, ch. 7, pp. 93–104.
- [11] A. Prasad, C. Si, and M.-Y. Kan, "Dataset mention extraction and classification," in *Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications*, 2019, pp. 31–36.
- [12] E. Gimeno, J. Z. J. Lane, Maggie, and P. Culliton, *Coleridge initiative - show us the data*, 2021.
- [13] H. Xue, Q. Yang, and S. Chen, "Svm: Support vector machines," in *The top ten algorithms in data mining*, Chapman and Hall/CRC, 2009, pp. 51–74.
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, June 2019, pp. 4171–4186.
- [15] L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, and S. Vaithyanathan, "Domain adaptation of rule-based annotators for named-entity recognition tasks," in *Proceedings of the 2010 conference on empirical methods in natural language processing*, 2010, pp. 1002–1012.
- [16] J. Lane, I. Mulvany, and P. Nathan, *Rich search and discovery for research datasets: Building the next generation of scholarly infrastructure*, 2020.
- [17] H. M. Wallach, "Conditional random fields: An introduction," *Technical Reports (CIS)*, p. 22, 2004.
- [18] W. Otto, A. Zielinski, B. Ghavimi, et al., "Knowledge extraction from scholarly publications: The gesis contribution to the rich context competition," in *Rich search and discovery for research datasets: Building the next generation of scholarly infrastructure*, J. Lane, I. Mulvany, and P. Nathan, Eds., Sage London, 2020, ch. 8, pp. 107–126.
- [19] M. Färber, A. Albers, and F. Schüber, "Identifying used methods and datasets in scientific publications," in *SDU@ AAAI*, 2021.
- [20] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov 2019, pp. 3615–3620.
- [21] C. Clark and M. Gardner, "Simple and effective multi-paragraph reading comprehension," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2018, pp. 845–855.
- [22] A. Trotman, A. Paurula, and B. Burgess, "Improvements to bm25 and language models examined," in *Proceedings of the 2014 Australasian Document Computing Symposium*, 2014, pp. 58–65.
- [23] K. Cho, B. Van Merriënboer, C. Gulcehre, et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Oct 2014, pp. 1724–1734.
- [24] Y. Younes and B. Mathiak, "Handling class imbalance when detecting dataset mentions with pre-trained language models," in *ICNLSP*, 2022, pp. 79–88.
- [25] G. Fan, J. Wang, Y. Li, D. Zhang, and R. Miller, "Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning," *arXiv preprint arXiv:2210.01922*, 2022.
- [26] Y. Liu, M. Ott, N. Goyal, et al., "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
- [27] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," *arXiv preprint arXiv:2006.03654*, 2020.
- [28] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," in *Proceedings of the 34th International Conference on Neural Information Processing Systems, ser. NIPS'20*, Curran Associates Inc., 2020.
- [29] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 6894–6910.
- [30] L. Tunstall, L. Von Werra, and T. Wolf, *Natural language processing with transformers*. "O'Reilly Media, Inc.", 2022.
- [31] N. Tepper, E. Goldbraich, N. Zwerdling, G. Kour, A. Anaby Tavor, and B. Carmeli, "Balancing via Generation for Multi-Class Text Classification Improvement," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, Online: Association for Computational Linguistics, 2020, pp. 1440–1452. (visited on 01/12/2022).
- [32] T. H. Phan and K. Yamamoto, "Resolving Class Imbalance in Object Detection with Weighted Cross Entropy Losses," *arXiv:2006.01413 [cs]*, Jun. 2020, *arXiv: 2006.01413*.
- [33] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [34] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.
- [35] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, e1249, 2018.
- [36] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [37] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
- [38] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman, "An overview of principal component analysis," *Journal of Signal and Information Processing*, vol. 4, no. 3B, p. 173, 2013.
- [39] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [40] L. Galke, A. Diera, B. X. Lin, B. Khara, et al., "Are We Really Making Much Progress in Text Classification? A Comparative Review," *arXiv preprint arXiv:2204.03954*, 2023.
- [41] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI Conference on artificial intelligence*, vol. 33, 2019, pp. 7370–7377.
- [42] F. Hassan, J. Domingo-Ferrer, and J. Soria-Comas, "Anonymization of unstructured data via named-entity recognition," in *Modeling Decisions for Artificial Intelligence*, V. Torra, Y. Narukawa, I. Aguiló, and M. González-Hidalgo, Eds., Cham: Springer, 2018, pp. 296–305, ISBN: 978-3-030-00202-2.
- [43] J. Lane, E. Gimeno, E. Levitskaya, Z. Zhang, and A. Zigoni, "Data inventories for the modern age? using data science to open government data," 2022.
- [44] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," *arXiv:1711.05101 [cs, math]*, Jan. 2019, *arXiv: 1711.05101*. (visited on 01/02/2022).
- [45] P. Rajpurkar, J. Zhang, and P. Liang, "Know what you don't know: Unanswerable questions for squad," in *ACL 2018*, 2018.
- [46] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ Questions for Machine Comprehension of Text," *arXiv e-prints*, *arXiv:1606.05250*, 2016. *arXiv: 1606.05250*.
- [47] W. Han, H. Huang, and T. Han, "Finding the evidence: Localization-aware answer prediction for text visual question answering," *arXiv preprint arXiv:2010.02582*, 2020.
- [48] L. Derczynski, E. Nichols, M. Van Erp, and N. Lim-sopatham, "Results of the wnut2017 shared task on novel and emerging entity recognition," in *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 2017, pp. 140–147.
- [49] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, 2020.
- [50] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, and J. K. et al., "Language models are few-shot learners," 2020. *arXiv: 2005.14165*.





YOUSSEF YOUNES received a B.S. degree in computer systems engineering from Al-Mamoun Private University for Science and Technology Aleppo, Syria, in 2010, and an M.E. degree in software engineering from Chongqing University, China, in 2015. He is pursuing a Ph.D. in computer science at the University of Ulm.

From 2015 to 2018, he worked as a software developer at the Syrian Telecommunication company. From 2018 to 2019, he was a research assistant with the database chair at BTU-CS University, Germany. Since 2021, he has worked as a researcher at GESIS, Germany. His research interests include Quantum Logic, machine learning, deep learning, IR, and knowledge graph.



ANSGAR SCHERP is a Full Professor for Data Science and Big Data Analytics with Ulm University, Germany. Prior, he worked among others as Professor of Natural Language Processing and Data Analytics and was member of the interdisciplinary Language and Computation group with the University of Essex, England, UK. Ansgar has an excellent research reputation in Text and Graph Mining, specifically in the combination of symbolic and subsymbolic (statistical) methods

for data analysis. His research interests are in novel approaches for data analysis by combining symbolic and statistical methods. He brings together methods from Information Retrieval, Data Mining and Machine Learning, and Semantic Web. He applies his novel data analysis approaches to, e.g., very large, distributed Knowledge Graphs on the web with billions of edges or large-scale document corpora in domains like life sciences/medicine, social sciences, economics, and the web. He has won the Billion Triples Challenge at the International Semantic Web conference in 2008 and 2011. The goal of the Billion Triple Challenge is to demonstrate scalability of semantic technologies. Ansgar is elected speaker at the ACM SIGMM Rising Stars Symposium of the Special Interest Group on Multimedia (SIGMM) of the Association for Computing Machinery (ACM) that was held in October in Amsterdam honoring his 10 years of research in metadata mining and semantics. He published over 150 peer-reviewed conference papers and journal articles.

...