Identifying Used Methods and Datasets in Scientific Publications

Michael Färber, **Alexander Albers, Felix Schüber**
Karlsruhe Institute of Technology (KIT), Germany

Abstract

Although it has become common to assess publications and researchers by means of their citation count (e.g., using the h-index), measuring the impact of scientific methods and datasets (e.g., using an h-index for datasets) has been performed only to a limited extent. This is not surprising because the usage information of methods and datasets is typically not explicitly provided by the authors, but hidden in a publication's text. In this paper, we propose an approach to identifying methods and datasets in texts that have actually been used by the authors. Our approach first recognizes datasets and methods in the text by means of a domain-specific named entity recognition method with minimal human interaction. It then classifies these mentions into used vs. non-used based on the textual contexts. The obtained labels are aggregated on the document level and integrated into the Microsoft Academic Knowledge Graph modeling publications' metadata. In experiments based on the Microsoft Academic Graph, we show that both method and dataset mentions can be identified and correctly classified with respect to their usage to a high degree. Overall, our approach facilitates method and dataset recommendation, enhanced paper recommendation, and scientific impact quantification. It can be extended in such a way that it can identify mentions of any entity type (e.g., task).

1 Introduction

In the past, a huge variety of scientific methods and datasets has been proposed in the different scientific disciplines. For instance, Wikipedia lists several hundred datasets for the area of machine learning.[1] It is therefore unsurprising that researchers are often unaware of which scientific methods or data sets have already been used for a given research topic. Furthermore, in digital libraries, such information regarding usage of scientific methods and datasets can be very useful. For instance, this information allows us to measure the impact of publications and researchers in novel ways (e.g., h-index for datasets). In this way, authors providing methods and datasets can be awarded properly in the light of FAIR data principles and open research efforts.

[1] https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research.

The usage of methods and datasets is typically not given explicitly, but mentioned in publications' full texts. Identifying scientific methods and datasets in texts can be considered as domain-specific named entity recognition. In the scholarly domain, a few approaches have been proposed for identifying concepts such as datasets (Mesbah et al. 2018; Luan 2019; Luan et al. 2018; Tsai, Kundu, and Roth 2013). For instance, Tsai, Kundu, and Roth (2013) propose a method to extract concepts from scientific publications. They limit their extraction method to entities that are followed by a citation indicator and only extract all mentioned concepts, rather than ones explicitly used. Gábor et al. (2018), in contrast, proposed a method to classify entity mentions into used and non-used. However, usage relations are only considered between entities of a specific type and not with respect to the papers' authors. Overall, a state-of-the-art approach that can recognize and classify scientific methods and datasets is, to the best of our knowledge, missing so far. Moreover, no large data set has been published that allows tasks for method/dataset-centric scientific impact quantification.

In this paper, we develop a framework to recognize entities of type DATASET and METHOD in scientific publications, as well as to classify them as used vs. non-used. Our framework consists of a domain-specific named entity recognition step, a classification step for determining the actual usage, and an aggregation step for retrieving the used methods and datasets on the document level. Our approach is designed to extract information about entities from scientific publications in an automated way, requiring minimal human interaction. We provide the usage information of about 771,000 methods and 449,000 datasets online for further usage. Moreover, we integrate the information into the Microsoft Academic Knowledge Graph (MAKG), which models information of more than 120 million scientific publications, and thereby provides the basis for scientific impact quantification studies (e.g., designing "h-index"-like metrics for scientific methods and datasets).

Overall, the main contributions of this paper are as follows:

- We develop a named entity recognition approach that extracts scientific methods and datasets from texts. Our approach extends preliminary works (Mesbah et al. 2018) by using state-of-the-art embedding techniques.

- We develop novel approaches to identify in texts the methods and datasets authors have indeed used in their papers.
- We create an evaluation dataset of 1,000 sentences with annotated methods and datasets and provide it to the public.
- We perform extensive experiments and identify the best classification method for the proposed task.
- We analyze the results of applying our framework to computer science papers.
- We extend the MAKG with the usage information concerning methods and datasets mentioned in 510,027 papers and provide it to the public.

Our data and code are publicly available at https://github.com/michaelfaerber/scholarly-entity-usage-detection.

The rest of our paper is structured as follows: In Section 2, we outline related work concerning domain-specific named entity recognition and usage classification. In Section 3, we describe our methods for named entity recognition and usage classification. We present our evaluation in Section 4 and our generated dataset in Section 5, before summarizing our findings in Section 6.

2 Related Work

In the following paragraphs, we outline the most relevant works concerning named entity recognition for long-tail entities and the extraction of aspects of entities.

Named Entity Recognition for Long-Tail Entities. In general, existing named entity recognition (NER) approaches are of diverse nature: They utilize gazetteers, rules, parts-of-speech tagging, dependency trees, or machine learning techniques. State-of-the-art NER approaches are often based on long short-term memory networks (LSTMs) (Mysore et al. 2017), conditional random fields (CRFs) (Mesbah et al. 2018; Vliegenthart et al. 2019), or a combination of both (Lample et al. 2016; Ma and Hovy 2016; Luan 2019; Jain et al. 2020). Although many approaches to named entity recognition exist, most of them require a considerable amount of human interaction for the creation of sufficient training data. Few classification approaches take into consideration that most of the considered entities are *long-tail entities* (i.e., appearing infrequently in documents and often not represented in public knowledge repositories, such as Wikidata). To reduce the required amount of human-labeled training data, iterative and active learning techniques have been proposed, particularly for scientific publications (Tchoua et al. 2019; Mesbah et al. 2018; Vliegenthart et al. 2019; Luan et al. 2018). Mesbah et al. (2018), for instance, introduce TSE-NER, which iteratively expands a predefined seed set of terms without additional human input. The authors apply several heuristic filtering methods to automatically create positive and negative classification examples. Our approach to named entity recognition is based on TSE-NER, but extends it by using SciBERT embeddings. Vliegenthart et al. (2019) also extend the TSE-NER approach by relying on human feedback for newly added labels. Although the authors achieve a lower rate of added false positives, this semi-supervised technique reintroduces the need for human labor and thus does not meet our requirements. Tchoua et al. (2019) present a dedicated NER approach for material sciences to recognize polymer names. The approach is based on active learning to overcome the data sparsity problem. Luan et al. (2018) introduce a multi-task setup of identifying entities, relations, and coreference clusters in scientific articles. Although the approach is valuable in settings where not only named entities but facts need to be extracted from text, the authors do not specifically consider the usage of datasets and methods by the papers' authors.

Identifying Aspects of Entities. Apart from recognizing named entities, a few approaches take additional aspects of the entities, such as the actual usage of entities, into account. Gupta and Manning (2011) introduce a method to identify the focus, domain of application, and technique from computational linguistics papers, but this approach only extracts broad topics. Jain et al. (2020) focus on detecting and extracting salient information from publications. They define salient information as information (e.g., named entities) that are needed to describe the results of an article. In contrast, our goal is to find all used entities to gain enhanced insight into the general usage of methods and datasets.

3 Approach

Our framework for identifying methods and datasets authors use in a given text document is depicted in Figure 1. We can differentiate between the following steps:

1. We build a named entity recognition model to extract named entities of a given scientific paper.
2. We perform a classification of each named entity into used and non-used (i.e., merely mentioned) on a sentence level.
3. We aggregate the sentence-level classifications of all named entities in a document.

The obtained list of methods and datasets used per document can be further analyzed in various ways. For a neat alignment with papers' metadata, we extend the Microsoft Academic Knowledge Graph (MAKG) with this new data. In this way, metadata of publications, authors, venues, and research areas can be used for advanced scholarly data mining (e.g., for novel ways of research impact assessment).

In the following, we present the single steps of our pipeline in more detail.

3.1 Named Entity Recognition

For named entity recognition, we adapt the TSE-NER (Mesbah et al. 2018) to our needs. TSE-NER is based on the hypothesis that entities of the same type are mostly used in a similar context. For example, objects of the entity type DATASET may be mentioned in the documents via phrases such as "we used data set X" or "we could achieve a recall of 0.4 on data set Y." Identifying such patterns automatically in the text allows us to identify additional, unknown entities in the text – particularly long-tail entities. The contexts of these newly found entity mentions can then be mined in

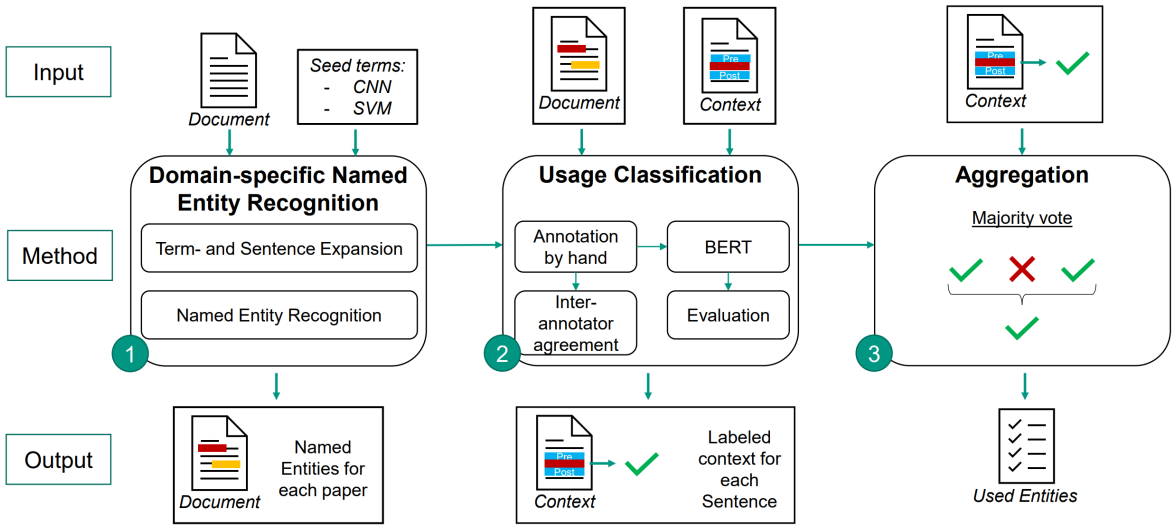


Figure 1: Overview of our framework.

another iteration, leading to additional patterns for named entity recognition.

An in-depth introduction to the original TSE-NER approach is provided by Mesbah et al. (2018). In the following, we outline the main steps of our named entity recognition approach and the main differences from the original TSE-NER approach.

1. We start with an initial set of METHOD and DATASET instances as seed terms (e.g., "SVM" and "MNIST"). These seed terms can, for instance, be gathered from existing knowledge graphs. In contrast to the original approach of Mesbah et al., we consider all computer science methods and datasets. The seed term selection is explained in Section 4.
2. We expand the list of seed terms by applying *term and sentence expansion* (TSE). In contrast to the original method, we use SciBERT as a semantic relatedness method and cluster the new entities using k-means.
3. Using the expanded set of entities, we annotate named entities in the training data. As context for each named entity we consider the current sentence as well as the preceding and subsequent sentence.
4. Using the annotated training set, we apply our NER approach and thereby identify new entity candidates. We use a CRF algorithm to learn the patterns of the data.
5. Finally, we filter the entity candidates to prevent misclassification and ensure data quality. We start with simple parts-of-speech analysis and stop-word removal methods to keep relevant nouns. Then, we use knowledge graph information and similarity scores to remove those entities with low similarity and no reference.

The output of our named entity recognition approach is a list of mentioned scientific methods and datasets with their positions in the texts.

3.2 Usage Classification

In total, we present four approaches for detecting *used* entity mentions of type METHOD or DATASET. For each model, we first apply an embedding-based method to transform the texts into a feature space, and then apply a classification algorithm to classify usage. In the following, we outline our approaches.

Model 1: TF-IDF + Random Forest As a baseline model, we use term frequency-inverse document frequency (tf-idf) to represent the words of a text as vectors. Based on preliminary evaluations of several standard classification methods, we choose a random forest classifier for classification into *used* and *non-used*.

Model 2: SciBERT + Random Forest For our second model, we make use of SciBERT (Beltagy, Lo, and Cohan 2019), a BERT-based language model pretrained on scientific publications. This embedding model has been used for various tasks, such as scientific text classification and recommendation. In our use case, we use SciBERT embeddings to create feature vectors and a random forest classifier for the binary classification.

Model 3: SciBERT + SciBERT Our third model is based on a fine-tuned SciBERT model for sequence classification. Beltagy, Lo, and Cohan (2019) show that fine-tuning SciBERT clearly improves the classification score, especially in the field of computer science. Hence, in comparison to the second model, we now also use SciBERT to make the classification by fine-tuning it to our annotated data. For the classification task, SciBERT uses a linear classification layer.

Model 4: SciBERT + CNN Our fourth model uses SciBERT embeddings as feature vectors and a convolutional neural network (CNN) for the classification task. Using the CNN approach as introduced by Kim (2014) as an advanced classification technique aims to capture the complex structures of word embeddings, which should result in a more

accurate classification score.

3.3 Document-level Aggregation

The method described above allows us to make a prediction for each occurrence of a named entity (i.e., entity-level prediction). To predict at the document level whether each unique named entity of a document is used or only mentioned or proposed, we aggregate all entity-level predictions to a document level prediction using majority vote.

3.4 Augmenting Publications' Metadata

We use our results to extend the MAKG (Färber 2019), which models publications' metadata for all scientific disciplines. Given that the MAKG is provided in the Resource Description Framework (RDF), we introduce the property `:used_methods`, which associates a paper with a used method. Because no knowledge graph contains all of the extracted methods and datasets, we refrain from linking to URIs in other knowledge graphs.

4 Evaluation

In the following, we outline our evaluations of all three steps of our pipeline. First, we compare the results of our modified TSE-NER model to the original paper. Next, we evaluate our usage classification models on our annotated test data. Finally, we apply our pipeline to full-text papers from the computer science domain to analyze trends over time in various computer science fields.

4.1 Named Entity Recognition

Evaluation Settings

(1) Training. We train our named entity recognition model on all 7 million abstracts of computer science papers given in the Microsoft Academic Graph (MAG; v2019-12-26) (Sinha et al. 2015). For the methods, we use the same 50 seed sets as the authors of the original paper. For DATASETS, we create our own set of seed terms because we were only able to expand very few sentences from our corpus using the original terms.[2] For our initial assessment, we run two iterations for each entity type, which according to the authors should already yield good results with a high precision value. Running more than two iterations increases recall at the cost of precision due to the addition of too many unrelated seed terms.

(2) Testing. To evaluate the NER approach, we use the SciREX dataset (Jain et al. 2020), which includes annotations of full-text papers from the machine learning domain for the METHOD and DATASET entity types. In this way, we can reuse existing evaluation data sets and compare our evaluation results with the evaluation results of the original TSE-NER (Mesbah et al. 2018). Although the authors of TSE-NER only apply their evaluation to triples consisting of a sentence containing the test entity, as well as the preceding and the succeeding sentence (Mesbah et al. 2018), we apply our model to full-text documents, which we regard as a more realistic setting.

[2]We extract 73 data set names from Wikipedia (https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research) and Wikidata (https://w.wiki/RrU) based on our knowledge in the machine learning domain.

Table 1: Evaluation of our modified TSE-NER model on the SciREX data set using precision, recall, and F1 score.

Training corpus	Abstracts			Full texts		
Metric	P	R	F1	P	R	F1
Method	0.44	0.14	0.21	0.26	0.45	0.33
Data set	0.33	0.27	0.30	0.20	0.29	0.25

As in the original paper, we calculate precision, recall, and F1 scores for the named entity recognition of METHOD and DATASET instances. We count partial matches as correct predictions because in most cases we do not need to cover the full span of an entity to gain meaningful insight.

Evaluation Results

(1) Study on Embeddings. The original TSE-NER approach is based on word2vec embeddings. Thus, we first analyze the difference in performance when using SciBERT token embeddings instead of word2vec embeddings for term clustering and similar terms filtering (see the steps 2 and 5 in Sec. 3.1) influences the clustering performance. We qualitatively study the clustering results of the term expansion in the first iteration for the METHOD type and find that, in general, both approaches generate very consistent clusters that differ based on various computer science fields. Given that the word2vec model had to be trained from scratch, it achieves surprisingly good results. Nevertheless, clustering based on SciBERT embeddings yields far more and richer terms, because it is not limited to just bigrams. Single clusters contain more variations of the same terms and generally contain better results. One risk of using SciBERT is that terms, such as *Netflix* or *GitHub*, are clustered together with dataset names, which is likely caused by both terms being used in the context of datasets but not being recognized jointly with neighboring terms. This may decrease the NER performance if names of other unrelated organizations are added as a result in the following iterations.

(2) NER Evaluation Results. Mesbah et al. (2018) achieve precision and recall values of 0.79 and 0.24 for the METHOD type and 0.83 and 0.10 for the DATASET type. The authors' TSE-NER model was trained based on 100 initial seed terms and the same sentence expansion and filtering strategies as our model. As shown in Table 1, we are not able to achieve a similar high precision value as the authors of the original paper, who used around 15,000 full-text papers as their corpus. The obvious reason is that publications' abstracts, as used by us, may be publicly available to a large extent and therefore may be a good data source, but seem to contain method and dataset names only to a limited degree. To improve the performance of TSE-NER, we choose to replicate a more similar corpus by using 25,060 *full-text papers* instead of 7 million abstracts from the MAG, as well

Our machine comprehension model is a hierarchical multi-stage process and consists of six layers (Figure [reference]) : Character Embedding Layer maps each word to a vector space using character - level CNNs . Word Embedding Layer maps each word to a vector space using a pre - trained word embedding model . Contextual Embedding Layer utilizes contextual cues from surrounding words to refine the embedding of the words . These first three layers are applied to both the query and context . Attention Flow Layer couples the query and context vectors and produces a set of query - aware feature vectors for each word in the context . Modeling Layer employs a Recurrent Neural Network to scan the context . Output Layer provides an answer to the query .

Our machine comprehension model is a hierarchical multi-stage process and consists of six layers (Figure [reference]) : Character Embedding Layer maps each word to a vector space using character - level CNNs . Word Embedding Layer maps each word to a vector space using a pre - trained word embedding model . Contextual Embedding Layer utilizes contextual cues from surrounding words to refine the embedding of the words . These first three layers are applied to both the query and context . Attention Flow Layer couples the query and context vectors and produces a set of query - aware feature vectors for each word in the context . Modeling Layer employs a Recurrent Neural Network to scan the context . Output Layer provides an answer to the query .

Figure 2: Example prediction of our trained TSE-NER model (top) versus ground truth (bottom) for the METHOD type after two iterations.

The images in Set14 are larger on average than those in Set5 . This selection of 14 images was proposed by Zeyde et al . [reference] . B100 is the testing set of 100 images from the Berkeley Segmentation Dataset [reference] . The images cover a large variety of real - life scenes and all have the same size of 481×321 pixels . We use them for testing as in [reference] . L20 is our newly proposed dataset .

The images in Set14 are larger on average than those in Set5 . This selection of 14 images was proposed by Zeyde et al . [reference] . B100 is the testing set of 100 images from the Berkeley Segmentation Dataset [reference] . The images cover a large variety of real - life scenes and all have the same size of 481×321 pixels . We use them for testing as in [reference] . L20 is our newly proposed dataset .

Figure 3: Example prediction of our trained TSE-NER model (top) versus ground truth (bottom) for the DATASET type after two iterations.

Table 2: TSE-NER training details using papers' abstracts as corpus. The table shows the number of words after each training step for the first and second iteration.

	i	Size of seed set	Expanded entities	Extracted entities	Filtered entities
Method	1	50	4,273	4,032	453
	2	503	3,403	7,469	1,031
Data set	1	73	354	1,450	6
	2	79	403	2,378	187

as narrowing the domain to include only machine learning papers. Although we see equal or higher recall values, this corpus does not improve the F1 scores significantly or, in the case of data sets, it even reduces the metric.

Figure 2 and Figure 3 illustrate the named entity recognition for two exemplary sections from the SciREX data set. We can observe that, in general, the approach produces decent results. The approach sometimes fails to capture the complete span of an entity mention (e.g., the first word in *character embedding layer*). Some of the false positive predictions are not too far fetched, such as *vector space*, but others, such as *query*, *answer*, and *context*, are less similar to names of methods. This indicates that there is still a potential to introduce better filtering strategies. One recurring problem for the DATASET model is that the term *dataset* is recognized without any specific names in its context.

To further compare our results with the TSE-NER publication (Mesbah et al. 2018), Table 2 shows the number of methods and datasets collected in each step based on the corpus containing papers' abstracts. While the original TSE-NER model used nearly 30,000 method names, our model is only able to use 3,403 method names as training data of the CRF. Training on the full-text corpus yields 8,355 named entities for training. This leads to more than 90,000 extracted named entities after the CRF training, compared to 7,469 named entities when training on abstracts, but still does not achieve the same results as Mesbah et al. (2018). One obvious reason for that may be that neither of our training corpora contain as many seed entities, which results in fewer found terms and sentences. Another reason may be that the found sentences contain fewer similar neighboring terms (e.g., fewer enumerations of method names or datasets), which would result in smaller cluster sizes and thus fewer added terms.

Despite the inferior evaluation results for our domain-specific named entity recognition of methods and datasets, we nevertheless believe they are sufficient for the subsequent knowledge graph expansion and trend analysis. Because we aggregate all found entities on the document level, we assume that a few missing mentions of the same entity would not affect the outcome significantly. For the subsequent tasks, we use the NER model trained on abstracts instead of full text, because we favor higher precision over recall for the knowledge graph extension.

4.2 Usage Classification

Evaluation Dataset

We needed to create a new dataset for training and evaluating our usage classification models. To this end, two authors (computer scientists) manually annotated 1,000 sentences concerning the usage of mentioned method and data sets (500 per entity type and person; see Table 4 for more statistics). We reuse a subset of the SciREX data set (Jain et al. 2020), which already contains annotated entities for the METHOD and DATASET type, and manually annotate whether an entity has been used in the given sentence and context. To reduce training bias, we also drop duplicate en-

	Method			Dataset			Generalization		
Model	P	R	F1	P	R	F1	P	R	F1
Single input sentence									
Random Forest (TF-IDF)	0.56	0.83	0.67	0.56	0.83	0.67	0.57	0.89	0.70
Random Forest + SciBERT	0.75	0.76	0.75	0.71	0.81	0.76	0.57	**0.96**	0.71
SciBERT (fine-tuned)	0.73	**0.92**	0.81	0.76	0.89	**0.82**	0.68	0.93	**0.79**
SciBERT + CNN	0.76	0.79	0.77	0.52	0.95	0.67	0.58	**0.96**	0.73
With surrounding sentences for context									
Random Forest (TF-IDF)	0.69	0.76	0.72	0.69	0.76	0.72	0.54	0.92	0.68
Random Forest + SciBERT	0.75	0.76	0.75	0.73	0.84	0.78	0.57	0.95	0.71
SciBERT (fine-tuned)	0.76	0.84	0.80	0.70	**0.96**	0.81	0.64	0.95	0.76
SciBERT + CNN	0.75	**0.91**	**0.83**	0.54	0.92	0.68	0.58	**0.96**	0.72

Table 3: Precision, recall and F1 scores for our usage classification models. We train each model with a single sentence as input as well as with the preceding and succeeding sentences for both methods and data sets. Further, we show the generalization capabilities for models that have been trained on the method type and then applied on data set entities.

Table 4: Key statistics of our annotated data set.

Entity Type	# annotated sentences	# annotated entities	# used entities	# mentioned entities	# balanced entities	κ score
Method	1,000	909	508	401	802	0.858
Data set	1,000	841	595	246	492	0.909

tities. We only annotated an entity as *used* if it is obvious from reading the sentence containing the entity and its surrounding context. In any uncertain cases, we annotate the entity as *non-used*. This way, we aim to achieve high precision on the sentence level while still being able to decide for an entity on the document level using our entity aggregation step whether the entity has been used. We also label an entity as *used* if it has been used in a comparison of multiple approaches (i.e. as a baseline). In this way, we allow a thorough tracking of used methods and datasets, facilitating scientific impact quantification.

To ensure high data quality and consistency of our annotated data, we select 100 entities of the METHOD and DATASET type that were annotated to calculate the inter-annotator agreement. We achieve a satisfactory κ score of 0.86 for methods and 0.91 for datasets.

Finally, we drop invalid entity types (e.g., entities from SciREX that are classified as material type but do not make sense as a data set type) and create a training and test set. Using the same amount of used and non-used entities, we have 802 entries for the METHOD type and 492 entries for the DATASET type. For the evaluation, we split the annotated data into training and test sets with a ratio of three to one.

Evaluation Settings

Because our usage classification task constitutes a binary classification problem, we evaluate our models using precision, recall, and F1 score. As outlined in Section 3.2, we evaluate four models: (1) random forest with TF-IDF representations, (2) random forest with SciBERT embeddings, (3) a SciBERT classification model with SciBERT embeddings, and (4) a CNN model with SciBERT embeddings for text representation.

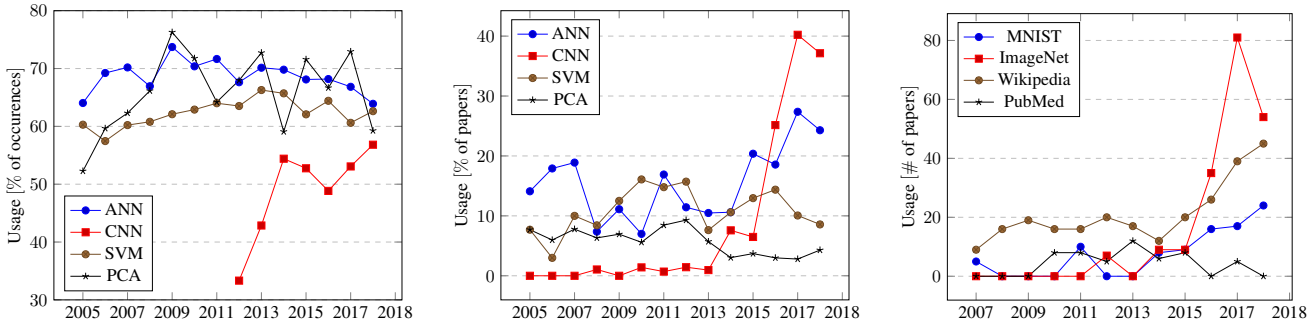
Evaluation Results

Comparison of Methods. Table 3 shows the evaluation results concerning the usage classification of method and dataset occurrences. For METHOD entities, the fine-tuned SciBERT model performs better with only a single sentence as input and achieves the best recall. The combined SciBERT and CNN model works best when the preceding and succeeding sentences are available as context. It achieves a similar high recall and slightly better precision than the fine-tuned SciBERT model.

For DATASET entities, both the fine-tuned SciBERT model and the CNN model achieve higher recall than they do for classifying METHOD entities. SciBERT still achieves relatively high precision scores but works better when neighboring sentences are available. For the CNN model, precision scores are significantly lower than they are for method entities.

Neither random forest model manages to compete with the more sophisticated models, but work slightly better on the DATASET entity type. Using the SciBERT sentence embeddings instead of tf-idf consistently results in a significantly higher precision at a cost of slightly lower recall values.

On manual inspection, we identified that the SciBERT and CNN models do not work when only a single sentence is given but critical information about an entity from the preceding or succeeding sentence is needed for the decision. For instance, in the following excerpt, the usage of the method is not recognized if only the second sentence is given to the models: *"In this paper, we introduce Invariant Information*



(a) Usage of selected *machine learning* methods over time relative to total mentions in papers.

(b) Usage of selected machine learning methods over time relative to all *computer vision* papers.

(c) Usage of machine learning data sets over time.

Figure 4: Relative usage of methods and datasets over time.

*Clustering (**IIC**), a method that [...]. **IIC** is a generic clustering algorithm that directly trains [...].”*

Furthermore, it can be seen that pronouns, such as “we,” give the models a strong hint that an entity has been used. However, in some cases, such as mathematical notations, this may lead to a false positive classification: *“We can write the **joint update** for all as Restrict the update to define a contraction mapping in the Euclidean metric.”*

Generalization across Entity Types. We also evaluate how well the usage classification models generalize to other entity types. For this purpose, we apply all models trained on the METHOD entity type to DATASETS. All examined models perform slightly worse regarding the F1 score, but still achieve very high recall values. This suggests that sentences in which methods are proposed or described do not differ too much from sentences that contain information about datasets. Out of all tested models, the SciBERT model generalizes the best to another entity type.

Further Studies. We also study whether information about the current section improves the performance of our classification models. Thus, we prepend the title of the current section to the input sentence and retrain all models. Our results show negligible performance improvements from this modification.

Finally, we investigate the extent to which our created data set differs from the SciREX data set (Jain et al. 2020) containing salient information of publications. Specifically, we study the degree to which our definition of *used entities* differs from *salient entities* considered by Jain et al. Salient entities are defined as necessary to describe the results of a paper and thus are semantically similar to our definition of used entities. We find for our method annotation set that only 12 out of 1,000 entries are labeled as salient in the original paper, which results in an MCC of 0.027 with our labels. For datasets, 39 entries are labeled as salient with an MCC of 0.011. In comparison, our created annotation data contains roughly similar amounts of used and non-used (e.g., proposed, only mentioned) entities, which allows us to extract and analyze considerably more used entities than we can with the saliency approach.

4.3 Application

We apply our framework to a corpus of 25,060 full-text machine learning papers from the MAG (Sinha et al. 2015) combined with unpaywall. The publication dates range from 2005 to 2018 and for each year we draw the same number of papers to compare relative usages. We process the publications using GROBID (Lopez 2009) to extract the full text as well as the title and all section names. We extract 438,707 method and 98,276 dataset entities from our corpus. Out of all extracted entities, 56% are classified as *used* concerning the methods and 68% concerning the datasets.

Analyzing Relative Usage We first study how many publications *used* specific entities compared to the number of publications in which the same entities were only mentioned. This relative measurement allows us to perform a more granular trend analysis because irrelevant entities that are never actually used will not be over-represented in the results.

Figure 4a shows this relative usage for selected machine learning methods over time. The usage of *artificial neural networks* (ANNs) and *support vector machines* (SVMs) is mostly constant between 60 and 75 % for all papers that mention one or the other term, but a slight downward trend is discernible for plain ANNs. The relative usage of the *principal component analysis* (PCA) shows a higher variability due to fewer absolute mentions but is used up to 75 % of the time if it is mentioned. For *convolutional neural networks* (CNNs), we only show values from 2012 and later because only a few mentions of CNNs occur in earlier years. Still, a clear trend is visible, where at the beginning in 2012 only around 35 % of papers that mentioned CNNs also used them for their work, whereas in 2018 the value was greater than 55 %.

Analyzing Specific Domains For another data study, we leverage the knowledge of the MAKG to select only publications from a specific computer science domain and analyze this subset of publications over time. Figure 4b shows the usage of selected machine learning methods in the computer vision field, which is one of the most popular categories by

number of papers in our set. Here, we only analyze the relative number of publications in which an entity has been used, instead of the number of named entity occurrences. Until 2015, the most used methods were ANNs and SVMs, which together have been used in around 30% of all computer vision papers. Since 2014, the usage of CNNs has steadily grown and is now the most used computer vision method. In turn, the number of papers that use SVMs and PCA has rather declined. Compared with Figure 4a, it can be seen that the relative usage of CNNs has increased since 2016. All this demonstrates that such a study would not be possible without an approach as proposed in this paper, which determines the actual usage of mentioned entities.

We also apply our classification pipeline to DATASET entities. Figure 4c shows the absolute amount of publications for the top four extracted datasets. A clear trend is visible for image recognition data sets, such as MNIST and ImageNet, which also correlates with the usage of CNNs in the computer vision domain. This again confirms the rising popularity of the specific domain. Another trend is visible for Wikipedia, which has become popular in research on knowledge representation and natural language processing.

5 Data Provisioning

We apply our framework to all computer science papers given both in the MAG and unpaywall (510,027 papers). Overall, we obtained 771,000 mentions of used methods and 449,000 mentions of used datasets. We provide the dataset online for further use (see our repository).

6 Conclusion

In this paper, we proposed an approach to identifying methods and datasets in texts that have actually been used by the authors. Our approach first recognizes datasets and methods in the text by means of a domain-specific named entity recognition with minimal human interaction. It then classifies these mentions into used vs. non-used. The obtained labels are aggregated on the document level and integrated into the Microsoft Academic Knowledge Graph modeling publications' metadata. In experiments based on the Microsoft Academic Graph, we showed that both method and dataset mentions can be identified and correctly classified with respect to their usage. Our approach, as well as our dataset containing the usage information of methods and datasets mentioned in 510,000 papers, can be used for research impact quantification tasks and further studies in the area of digital libraries.

In the future, we plan to use our framework with respect to other entity types, such as *task* and *evaluation metric*. Finally, a promising idea is to build a recommender system for scientific publications using our framework.

References

Beltagy, I.; Lo, K.; and Cohan, A. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP'19, 3613–3618.

Färber, M. 2019. The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data. In *Proceedings of the International Semantic Web Conference*, ISWC'19, 113–129.

Gábor, K.; Buscaldi, D.; Schumann, A.; QasemiZadeh, B.; Zargayouna, H.; and Charnois, T. 2018. SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, SemEval@NAACL-HLT'18, 679–688.

Gupta, S.; and Manning, C. D. 2011. Analyzing the Dynamics of Research by Extracting Key Aspects of Scientific Papers. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, IJCNLP'11, 1–9. The Association for Computer Linguistics.

Jain, S.; van Zuylen, M.; Hajishirzi, H.; and Beltagy, I. 2020. SciREX: A Challenge Dataset for Document-Level Information Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, ACL'20, 7506–7516.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, 1746–1751.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT'16, 260–270.

Lopez, P. 2009. GROBID: Combining Automatic Bibliographic Data Recognition and Term Extraction for Scholarship Publications. In *Proceedings of the 13th European Conference on Digital Libraries*, ECDL'09, 473–474.

Luan, Y. 2019. Information Extraction from Scientific Literature for Method Recommendation. *arXiv preprint arXiv:1901.00401* .

Luan, Y.; He, L.; Ostendorf, M.; and Hajishirzi, H. 2018. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP'18, 3219–3232.

Ma, X.; and Hovy, E. H. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL'16.

Mesbah, S.; Lofi, C.; Torre, M. V.; Bozzon, A.; and Houben, G. 2018. TSE-NER: An Iterative Approach for Long-Tail Entity Extraction in Scientific Publications. In *Proceedings of the International Semantic Web Conference*, ISWC'18, 127–143.

Mysore, S.; Kim, E.; Strubell, E.; others; and Olivetti, E. 2017. Automatically Extracting Action Graphs from Materials Science Synthesis Procedures. *CoRR* abs/1711.06872.

Sinha, A.; Shen, Z.; Song, Y.; Ma, H.; Eide, D.; Hsu, B.-J. P.; and Wang, K. 2015. An Overview of Microsoft Academic Service (MAS) and Applications. In *Proceedings of 24th International Conference on World Wide Web Companion*, WWW'15, 243–246.

Tchoua, R.; Ajith, A.; Hong, Z.; Ward, L.; Chard, K.; Audus, D.; Patel, S.; de Pablo, J.; and Foster, I. 2019. Active Learning Yields Better Training Data for Scientific Named Entity Recognition. In *Proceedings of the 15th International Conference on eScience*, eScience'19, 126–135.

Tsai, C.-T.; Kundu, G.; and Roth, D. 2013. Concept-Based Analysis of Scientific Literature. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, CIKM'13, 1733–1738.

Vliegenthart, D.; Mesbah, S.; Lofi, C.; Aizawa, A.; and Bozzon, A. 2019. Coner: A Collaborative Approach for Long-Tail Named Entity Recognition in Scientific Publications. In *Proceedings of the 23rd International Conference on Theory and Practice of Digital Libraries*, TPDL'19, 3–17.