Recommending Datasets for Scientific Problem Descriptions

Michael Färber
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
michael.faerber@kit.edu

Ann-Kathrin Leisinger
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
annkathrin.leisinger@gmail.com

ABSTRACT

The steadily rising number of datasets is making it increasingly difficult for researchers and practitioners to be aware of all datasets, particularly of the most relevant datasets for a given research problem. To this end, dataset search engines have been proposed. However, they are based on user’s keywords and, thus, have difficulty determining precisely fitting datasets for complex research problems. In this paper, we propose a system that recommends suitable datasets based on a given research problem description. The recommendation task is designed as a domain-specific text classification task. As shown in a comprehensive offline evaluation using various state-of-the-art models, as well as 88,000 paper abstracts and 265,000 citation contexts as research problem descriptions, we obtain an F1-score of 0.75. In an additional user study, we show that users in real-world settings are 88% satisfied in all test cases. We therefore see promising future directions for dataset recommendation.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Web search engines**; • **Computing methodologies** → **Natural language processing**; *Knowledge representation and reasoning*; **Supervised learning**.

KEYWORDS

datasets, recommendation, machine learning, text classification

ACM Reference Format:
Michael Färber and Ann-Kathrin Leisinger. 2021. Recommending Datasets for Scientific Problem Descriptions. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM ’21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482166

1 INTRODUCTION

The number of available datasets in the various scientific fields has grown vastly and is continuously on the rise [16]. For instance, OpenAIRE [20] contains the metadata of more than 23,000 datasets. In addition, in recent years, large national and international initiatives, such as the German National Research Data Infrastructure and the initiatives around the FAIR data principles [31], have been established to foster the reuse of datasets [16]. In the process of accessing and reusing datasets from repositories, identifying the most relevant datasets is often challenging for researchers and practitioners [21]. Dataset search engines, such as Google Dataset Search [2] and Zenodo [7], help users retrieve the most relevant datasets. However, existing dataset search engines using the datasets’ metadata are limited in their applicability [4]. Apart from the fact that search engines relying on metadata depend on the accuracy and maintenance of the metadata [3], existing dataset search engines are not suitable for the users’ specific and comprehensive information needs (see Table 1). Chen et al. [4] found that real data needs are most often formulated as phrases and not as keywords. The latter case constitutes only 32% of the investigated queries. Overall, to our knowledge, approaches beyond keyword search for retrieving relevant datasets for a given research problem are still missing.

Table 1: Example dataset recommendations.

Input text (e.g., paper abstract without dataset mentions)	**Rec. datasets**
This paper presents an algorithms for tagging words whose part-of-speech properties are unkown. Unlike previous work, the algorithm categorizes word tokens in context instead of word types.	Brown Corpus
Given a set of images with related captions, our goal is to show how visual features can improve the accuracy of unsupervised word sense disambiguation when the textual context is very small, as this sort of data is common in news and social media. We extend previous work in unsupervised text-only disambiguation with methods that integrate text and images. [...]	ImageNet, SemCor

In this paper, we propose a new approach to dataset search that is not based on keywords or faceted search; instead, the recommendation algorithm relies on a text classification model that predicts relevant datasets for a user’s input. The user input is a text (e.g., several sentences) that describes the research or task that the user plans to conduct (see Table 1). A classifier predicts all relevant datasets indexed in a given repository based on the entered text, ranked by relevance if needed. The hypothesis is that the quality of the dataset search can be considerably improved when using a rich formulation of the research problem in natural language, rather than relying purely on isolated keywords or attributes. In an extensive evaluation based on almost 2,000 datasets, more than 350,000 texts, and 21 different classifiers, we show that already a linear SVM can accurately predict suitable datasets for given problem descriptions to a high degree. Reaching up to 88% precision in a user study, the best-performing model illustrates clearly that text classification is a promising approach to dataset search.

CIKM ’21, November 1–5, 2021, Virtual Event, QLD, Australia
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8446-9/21/11.
https://doi.org/10.1145/3459637.3482166

Overall, this paper's main contributions are as follows:

- We propose machine learning-based text classification models for retrieving relevant datasets for given research problem descriptions.[1]
- We provide a new evaluation dataset for dataset recommendation based on research problem descriptions. It consists of more than 353,000 texts with mentions of nearly 2,000 datasets. All datasets are linked to papers in the Microsoft Academic Graph [30].
- We provide an evaluation of our dataset recommendation methods and can show that besides transformer-based methods, even a linear SVM achieves promising results.[2]

2 RELATED WORK

Dataset search based on keywords and/or faceted search. A detailed overview of dataset search is provided in [3]. It shows that existing dataset search systems are mainly based on keyword or faceted search and rely on the datasets' metadata. Pietriga et al. [24] propose the dataset search engine LODAtlas. The search is realized in LODAtlas as a keyword search, whereas we allow longer texts as input. Google Dataset Search [2] relies on metadata, too. Datasets' metadata freely available on the web is often of poor quality [12]. Thus, in contrast to Google Dataset Search, in our approach we use a much cleaner data basis and links to papers. In addition, Google Dataset Search relies solely on a keyword search.

Dataset search based on text. Chen et al. [5] propose a schema label generation model that generates possible schema labels based on the content of a dataset table [5]. The schema labels are then used in a ranking model to compute the similarity between a query and a dataset. In [4], the authors show that the majority of queries are phrases or sentences, which demonstrates the suitability of our approach. Our approach of text classification for dataset retrieval is similar to the tasks of dataset mention extraction and dataset classification in [27]. However, while we use the classification for dataset search, Prasad et al. [27] use the dataset classification to identify relations between datasets and existing scientific documents.

3 APPROACH

We model the dataset recommendation as a supervised multiclass, multilabel text classification, because for each input text, one or more datasets might be relevant. Consequently, we distinguish between a training phase and a testing/application phase. The training phase can be divided into (1) training data generation, (2) text preprocessing, and (3) text classification. The testing and application phase can be divided into (1) text preprocessing, (2) text classification, and (3) ranking and metadata retrieval. In the following, we describe the steps of text preprocessing and text classification.

Text Preprocessing. We consider the following text representation methods:

- **tf-idf** [28], a text representation based on term frequencies;
- **doc2vec** [29], a way of representing an entire document;
- **fastText embeddings**, pretrained word and phrase representations achieving or even outperforming state-of-the-art results on various tasks [22];
- **SciBERT embeddings**, a widely used language model based on BERT and trained on scientific texts [1];
- **Transformer-XL embeddings** [6], based on the self-attention model Transformer-XL that models longer-term dependency.

Text Classification. We consider the following text classification methods. Note that the focus of this paper is to propose dataset recommendation based on texts using state-of-the-art methods instead of dedicated novel approaches.

(1) **Classification based on tf-idf similarity:** Datasets are selected if the cosine similarity between the tf-idf representations of dataset and problem description exceeds a threshold.
(2) **Classification based on BM25 score:** This classification method is identical to the previous one, but uses BM25 instead of the cosine similarity between tf-idf representations.
(3) **Linear SVM:** SVMs are particularly applicable to large and high-dimensional classification problems [13]. The linear SVM is especially suitable for large training datasets.
(4) **Random Forest:** The random forest classifier considers numerous decision trees and, thus, typically results in more accurate predictions than a decision tree does.
(5) **Logistic Regression:** Logistic regression is particularly suitable and efficient for complex classification tasks and outperformed other traditional classification methods [26].
(6) **Gaussian, Multinomial, and Complement Naïve Bayes:** Naïve Bayes classifiers are often accurate and fast for large datasets [14].
(7) **Convolutional Neural Network (CNN):** Liu et al. [17] stated that several forms of CNNs and RNNs count to the strongest methods for multiclass classification [18].
(8) **Recurrent Neural Networks (Simple RNN, LSTM, and BiLSTM):** RNN, LSTM, and BiLSTM are widely used and are among the strongest multiclass classifiers [17].
(9) **CNN-LSTM:** CNN-LSTM combines the strengths of CNN and RNN models and has achieved excellent performance on text classification tasks [32].
(10) **fastText classification:** fastText classifier's accuracy is often on par with deep learning classifiers, while the classifier is much faster for training and evaluation [15].
(11) **Fine-tuning of SciBERT embeddings with subsequent classification layer:** Following [1, 19], we consider fine-tuning SciBERT embeddings with an attached classification layer as a text classifier.

4 EVALUATION

Evaluation Setting. We use an offline evaluation and a user study to evaluate the proposed methods' performance. The offline evaluation uses labeled data in a re-prediction setting. The user study has a similar setting to the offline evaluation, but considers user feedback to decide whether a recommended dataset is relevant.

Evaluation Dataset. As a database for datasets, we use the Data Set Knowledge Graph (DSKG) [12]. This up-to-date collection published in 2021 is based on dataset entries in Wikidata and OpenAIRE. It is characterized by rich and highly accurate metadata. All datasets

[1] Our implementation and evaluation dataset is available at https://github.com/michaelfaerber/datarec.

[2] A running dataset search system based on one of the best performing models is provided online at http://data-hunter.io and described in [9].

Table 2: Offline evaluation results given (a) paper abstracts and (b) citation contexts.

		(a) paper abstracts						(b) citation contexts					
		macro			weighted			macro			weighted		
Classifier	Text represent.	P	R	F1	P	R	F1	P	R	F1	P	R	F1
tf-idf similarity	-	0.00	0.40	0.01	0.10	0.38	0.12	0.00	0.28	0.01	0.10	0.46	0.13
BM25 values	-	0.02	0.19	0.02	0.16	0.14	0.08	0.01	0.18	0.02	0.17	0.22	0.12
Linear SVM	tf-idf	**0.39**	0.17	**0.22**	**0.66**	0.42	**0.49**	**0.62**	**0.55**	**0.57**	0.75	**0.76**	**0.75**
Gaussian Naïve Bayes	tf-idf	0.02	0.04	0.03	0.12	0.29	0.16	0.52	0.29	0.34	0.62	0.40	0.45
Multinomial Naïve Bayes	tf-idf	0.00	0.00	0.00	0.15	0.10	0.11	0.15	0.04	0.05	0.53	0.43	0.35
Complement Naïve Bayes	tf-idf	0.00	0.01	0.01	0.16	0.22	0.18	0.34	0.13	0.16	0.63	0.56	0.50
CNN	tf-idf	0.00	0.00	0.00	0.02	0.13	0.03	0.00	0.00	0.00	0.02	0.15	0.04
Linear SVM	doc2vec	0.10	0.04	0.05	0.43	0.20	0.24	0.24	0.16	0.17	0.41	0.46	0.39
Gaussian Naïve Bayes	doc2vec	0.04	0.34	0.06	0.18	0.73	0.26	0.14	0.18	0.13	0.40	0.29	0.32
CNN	doc2vec	0.01	0.02	0.01	0.18	0.29	0.20	0.05	0.03	0.03	0.30	0.38	0.31
Linear SVM	fastText	0.14	0.05	0.07	0.51	0.22	0.28	0.37	0.26	0.28	0.55	0.59	0.55
Gaussian Naïve Bayes	fastText	0.03	**0.45**	0.04	0.15	0.79	0.21	0.32	0.34	0.29	0.55	0.43	0.46
CNN	fastText	0.01	0.01	0.01	0.18	0.30	0.21	0.07	0.06	0.05	0.38	0.46	0.39
Linear SVM	SciBERT	0.21	0.17	0.17	0.50	0.32	0.36	0.51	0.47	0.47	0.69	0.70	0.69
Gaussian Naïve Bayes	SciBERT	0.03	0.43	0.04	0.12	**0.82**	0.18	0.13	0.09	0.07	0.38	0.08	0.08
CNN	SciBERT	0.00	0.01	0.00	0.10	0.22	0.12	0.07	0.06	0.06	0.36	0.44	0.38
Linear SVM	Transformer-XL	0.19	0.09	0.11	0.50	0.22	0.28	0.40	0.37	0.37	0.51	0.53	0.51
Gaussian NB	Transformer-XL	0.03	0.32	0.03	0.08	0.67	0.13	0.07	0.05	0.04	0.20	0.04	0.04
CNN	Transformer-XL	0.01	0.01	0.00	0.11	0.21	0.12	0.02	0.02	0.01	0.16	0.27	0.17
FastText classifier	-	0.11	0.27	0.15	0.38	0.62	0.46	0.55	0.47	0.49	**0.76**	0.75	**0.75**
SciBERT finetuning	-	0.04	0.03	0.03	0.29	0.36	0.30	0.29	0.25	0.25	0.68	0.70	0.68

are linked to papers of the Microsoft Academic Knowledge Graph [8] containing metadata of 240 million publications. Thus, users can also see detailed information of publications using the datasets, which might be helpful for choosing an appropriate dataset. We focus on computer science, resulting in a set of 1,691 datasets.

To train and evaluate our approaches, texts representing scientific problem descriptions need to be linked to relevant datasets in the DSKG. Since pure research problem descriptions are largely not available, we use abstracts from scientific papers that contain the given datasets as in-text mentions and, thus, as target labels. Paper abstracts are very similar to problem descriptions, because they both summarize in a few sentences the examined task for which a dataset has been used or will be used. Given the diverse length of abstracts (a few words up to 1,000 words in our collection), abstracts exhibit a great variety of possible real-world queries. In addition, we consider citation contexts (i.e., sentences with in-text citations) for a supplementary evaluation. They are typically shorter than abstracts. Overall, we argue that paper abstracts and excerpts from papers, from which the dataset mentions were removed, are a valid approximation of the researchers' written information needs concerning datasets. In this way, also datasets can be recommended for already published papers.

In our evaluations, we use the paper abstracts and citation contexts from the Microsoft Academic Graph:

(1) **Paper abstracts.** We use 88,047 paper abstracts that reference 1,413 unique datasets from the field of computer science. Most paper abstracts (75,034; 85.2%) reference only one dataset. Up to 20 datasets are referenced per abstract.
(2) **Citation contexts.** We use 265,587 citation contexts that mention a computer science dataset. Each citation context contains only one dataset. Also here, the mentions were removed in the dataset creation phase. The most common dataset is mentioned in 40,236 citation contexts.

We see that a few datasets are referenced very often (up to approx. 14,500 times in the abstracts and 40,000 times in the citations), but the majority of datasets is referenced between one and 100 times. The imbalanced data makes the classification more challenging; however, particularly the recommendation of less commonly used datasets is our goal.

4.1 Offline Evaluation

4.1.1 Offline Evaluation Setting. As outlined in Section 3, we investigated the text representation methods (1) tf-idf, (2) doc2vec, (3) fastText embeddings, (4) SciBERT embeddings, and (5) TransformerXL embeddings. For text classification, we use one of the 11 methods listed in Section 3. Furthermore, we use 80% of the evaluation dataset for training the models and 20% of the data for testing. As evaluation metrics, we used precision, recall, and F1-score. To take into account the quality of small classes, we also consider the macro and weighted averages for precision, recall, and F1-score.

Cross Validation. In addition to the hold-out validation, a k-fold stratified cross validation with $k = 5$ and $k = 10$ was performed. The results were comparable to the settings with hold-out validation.

Sampling Strategies. We considered random oversampling and random undersampling. Our evaluation showed that undersampling worsens the classification quality significantly. Oversampling sometimes increased and sometimes decreased the classification quality while the training time was more than twice as long as without sampling. Thus, we apply neither oversampling nor undersampling.

Time Component. We wanted to ensure that the classification quality is not deteriorated by predicted datasets that were not available in the year in which the research problem description was written. For this reason, the false-positive predictions of several classification models were investigated. This analysis showed that no items were falsely classified due to temporal reasons or that this could not be detected because the metadata was lacking. We also compared the classification quality for temporarily sorted train and test data and for randomly shuffled train and test data, following [10]. As only a marginal difference in all investigated evaluation metrics was observed, we decided to neglect the time component.

4.1.2 Offline Evaluation Results. Table 2 shows the offline evaluation results.

Basic Classification Approaches. The classifiers based on tf-idf and BM25 perform poorly and are not able to model this classification task properly. The recall rates are considerably higher than the precision scores. However, also the recall rates are outperformed by other models, such as linear SVM based on tf-idf.

Traditional Text Classification Approaches. According to our evaluation results, logistic regression and random forest are not suitable for modeling this text classification, because they often failed to converge or had very long training times. The best model of the traditional classifiers for the considered database is Linear SVM, which outperforms Naïve Bayes in terms of F1-score for all text representations. Linear SVM based on tf-idf is the model with the best overall performance for paper abstracts and citation contexts.

Deep Learning Approaches. The fastText classification model is the second most promising classifier due to the second highest F1-scores for both datasets. Besides the considerably high performance, this model is convincing because of its fast computation time that is multiple times shorter than the training time of all other models.

The SciBERT fine-tuning model with subsequent classification layer performs moderately for abstracts (weighted F1-score of 0.30), and well for citation contexts (weighted F1-score of 0.68).

We observe a superior performance of the CNN in comparison to LSTM, Simple RNN, CNN-LSTM, and BiLSTM. The results of the CNN were always on par with or better than the results of these four methods, whereas the computation time of the CNN was clearly shorter. For instance, the CNN was at least 2.8 times faster than the LSTM on all investigated text representations. This performance persists when applying extensive parameter optimization on the models. Thus, we excluded LSTM, Simple RNN, CNN-LSTM, and BiLSTM from further evaluation and focused on the CNN model with hyperparameter tuning. A reason for the superior performance of the CNN in comparison to architectures including an RNN could be that putting a strong focus on word sequences rather than the occurrence of specific words is detrimental to classifying problem descriptions [11]. The CNN’s overall performance is in most cases worse than traditional classification models (e.g., Naïve Bayes) and other deep learning methods (i.e., SciBERT classifier, fastText classifier). The CNN models were outperformed by the best models, such as Linear SVM based on tf-idf, for all text representations.

Similar to Pathak [25], we observed that Transformer-XL embeddings are particularly time consuming to train. However, the classification quality for models based on Transformer-XL embeddings is outperformed by SciBERT embeddings. Thus, the consideration of longer-term dependency does not necessarily improve the embeddings’ quality. Furthermore, the weighted average scores are generally higher than the macro average scores for all models. This behavior highlights that the classification models perform better for frequent classes than for rare classes. This issue also becomes clear when considering the confusion matrices and precision, recall, and F1-score for each present class. For the Linear SVM, the differences in performance between rare and frequent classes tend to be smaller than they are for other classifiers.

Table 3: User study results.

Model	**Precision** Abstracts	**Precision** Cit. Contexts
Linear SVM (tf-idf)	**0.643**	0.687
fastText	0.287	0.288
Linear SVM (SciBERT)	0.637	**0.881**

4.2 User Study

Our user study is based on user feedback, and it reviews the classification models’ performance in a real-world application. It focuses on the offline evaluation’s most promising classification models. We consider it as a complementary evaluation, focusing on precision.

4.2.1 User Study Setting. Similar to the offline evaluation, we used the abstracts and citation contexts as problem descriptions. Following [23], a sample size of 400 is chosen for the user study to ensure the results’ statistical significance with a confidence interval of 95%.

To create the ground truth, we let two experienced researchers judge the recommended datasets for all sample inputs (400 paper abstracts and 400 citation contexts; judged as relevant vs. nonrelevant vs. unknown). The interannotator agreement using Cohen’s kappa score is 0.443, which is moderate but acceptable.

We evaluate the classification models that were found most promising in the offline evaluation based on the macro and weighted averaged F1-scores. These are (1) Linear SVM based on tf-idf, (2) fastText classification, and (3) Linear SVM with SciBERT embeddings.

4.2.2 User Study Results. The user study’s results are shown in Table 3. The Linear SVM models based on tf-idf and SciBERT embeddings perform best and reach a precision of up to 0.881. These high results show that dataset recommendation can indeed be helpful in real-world settings. The user study confirms the offline evaluation’s finding that Linear SVM with tf-idf is a well-performing model for the considered task and database. In the user study, the Linear SVM based on SciBERT embeddings is on par with this model given paper abstracts, and even outperforms it given citation contexts. For these two models, the user study’s results are on par with and partially outperform the offline evaluation’s results and therefore confirm the performance trends of these models.

5 CONCLUSION

In this paper, we developed a dataset search system that uses text classification to recommend relevant datasets for a given scientific problem description. Besides new problem descriptions, it can be used for already published research descriptions. We evaluated various state-of-the-art classification models combined with several text representation methods. The large-scale offline evaluation revealed that suitable datasets can be re-predicted to a high degree. Our user study confirmed that a linear SVM based on tf-idf or SciBERT embeddings performs particularly well in predicting datasets based on scientific problem descriptions.

REFERENCES

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (Hong Kong, China) *(EMNLP'19)*. 3613–3618.

[2] Dan Brickley, Matthew Burgess, and Natasha F. Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *Proceedings of the 28th International World Wide Web Conference* (San Francisco, CA, USA) *(WWW'19)*. 1365–1375.

[3] A. Chapman, E. Simperl, L. Koesten, G. Konstantinidis, L. D. Ibáñez, E. Kacprzak, and P. Groth. 2020. Dataset search: a survey. *VLDB J.* 29, 1 (2020), 251–272. https://doi.org/10.1007/s00778-019-00564-x

[4] Jinchi Chen, Xiaxia Wang, Gong Cheng, Evgeny Kharlamov, and Yuzhong Qu. 2019. Towards More Usable Dataset Search: From Query Characterization to Snippet Generation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM'19)*. 2445–2448.

[5] Zhiyu Chen, Haiyan Jia, Jeff Heflin, and Brian D. Davison. 2020. Leveraging Schema Labels to Enhance Dataset Search. In *Proceedings of the 42nd European Conference on IR* (Lisbon, Portugal) *(ECIR'20)*. 267–280.

[6] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics* (Florence, Italy) *(ACL'19)*. 2978–2988.

[7] European Organization For Nuclear Research and OpenAIRE. 2013. Zenodo. https://doi.org/10.25495/7GXK-RD71

[8] Michael Färber. 2019. The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data. In *Proceedings of the 18th International Semantic Web Conference* (Auckland, New Zealand) *(ISWC'19)*. 113–129.

[9] Michael Färber and Ann-Kathrin Leisinger. 2021. DataHunter: A System for Finding Datasets Based on Scientific Problem Descriptions. In *Proceedings of the 15th ACM Recommender Systems Conference* (Amsterdam, The Netherlands) *(RecSys'21)*. 749–752.

[10] Micheal Färber and Ashwath Sampath. 2020. HybridCite: A Hybrid Model for Context-Aware Citation Recommendation. In *Proceedings of the 2020 ACM/IEEE Joint Conference on Digital Libraries* (Virtual Event, China) *(JCDL'20)*. 117–126.

[11] Michael Färber, Alexander Thiemann, and Adam Jatowt. 2018. To Cite, or Not to Cite? Detecting Citation Contexts in Text. In *Proceedings of the 40th European Conference on Information Retrieval* (Grenoble, France) *(ECIR'18)*. 598–603.

[12] Färber, Michael and Lamprecht, David. 2021. The Data Set Knowledge Graph: Creating a Linked Open Data Source for Data Sets. *Quantitative Science Studies* (2021). http://dskg.org/publications/DSKG_QSS2021.pdf.

[13] Mohamed Goudjil, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali. 2018. A Novel Active Learning Method Using SVM for Text Classification. *Int. J. Autom. Comput.* 15, 3 (2018), 290–298.

[14] Jiawei Han, Micheline Kamber, and Jian Pei. 2012. Classification: Basic Concepts. In *Data Mining* (third ed.), Jiawei Han, Micheline Kamber, and Jian Pei (Eds.). Morgan Kaufmann, Boston, 327–391.

[15] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics* (Valencia, Spain) *(EACL'17)*. 427–431.

[16] Laura Koesten, Elena Simperl, Tom Blount, Emilia Kacprzak, and Jeni Tennison. 2020. Everything you always wanted to know about a dataset: Studies in data summarisation. *Int. J. Hum. Comput. Stud.* 135 (2020).

[17] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo). 115–124.

[18] Liqun Liu, Funan Mu, Pengyu Li, Xin Mu, Jing Tang, Xingsheng Ai, Ran Fu, Lifeng Wang, and Xing Zhou. 2019. NeuralClassifier: An Open-source Neural Hierarchical Multi-label Text Classification Toolkit. In *Proceedings of the 57th Conference of the Association for Computational Linguistics* (Florence, Italy) *(ACL'19)*. 87–92.

[19] Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. 2019. Universal Text Representation from BERT: An Empirical Study. *CoRR* abs/1910.07973 (2019).

[20] Paolo Manghi, Claudio Atzori, others, and Friedrich Summann. 2019. OpenAIRE Research Graph Dump. https://doi.org/10.5281/zenodo.3516918

[21] Yasmmin Cortes Martins, Fábio Faria da Mota, and Maria Cláudia Cavalcanti. 2016. DSCrank: A Method for Selection and Ranking of Datasets. In *Proceedings of the 10th International Conference on Metadata and Semantics Research* (Göttingen, Germany) *(MTSR'16)*. 333–344.

[22] Tomás Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation* (Miyazaki, Japan) *(LREC'18)*. 52–55.

[23] NIST/SEMATECH. 2013. Engineering Statistics Handbook. 7.2.4.2. Sample sizes required. https://www.itl.nist.gov/div898/handbook/prc/section2/prc242.htm

[24] Emmanuel Pietriga, Hande Gözükan, Caroline Appert, Marie Destandau, Sejla Cebiric, François Goasdoué, and Ioana Manolescu. 2018. Browsing Linked Data Catalogs with LODAtlas. In *Proceedings of the 17th International Semantic Web Conference* (Monterey, CA, USA) *(ISWC'18)*. 137–153.

[25] Keval Pipalia, Rahul Bhadja, and Madhu Shukla. 2020. Comparative Analysis of Different Transformer Based Architectures Used in Sentiment Analysis. In *Proceedings of the 9th International Conference System Modeling and Advancement in Research Trends* (Moradabad, India) *(SMART'20)*. 411–415.

[26] Tomas Pranckevicius and Virginijus Marcinkevicius. 2017. Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification. *Balt. J. Mod. Comput.* 5, 2 (2017).

[27] Animesh Prasad, Chenglei Si, and Min-Yen Kan. 2019. Dataset Mention Extraction and Classification. In *Proceedings of the Workshop on Extracting Structured Knowledge from Scientific Publications* (Minneapolis, Minnesota) *(ESSP'19)*. 31–36.

[28] Bruno Trstenjak, Sasa Mikac, and Dzenana Donko. 2014. KNN with TF-IDF based Framework for Text Categorization. *Procedia Engineering* 69 (2014), 1356–1364.

[29] Maria Mihaela Truşcă. 2019. Efficiency of SVM classifier with Word2Vec and Doc2Vec models. In *Proceedings of the International Conference on Applied Statistics*, Vol. 1. Sciendo, 496–503.

[30] Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. 2020. Microsoft Academic Graph: When experts are not enough. *Quant. Sci. Stud.* 1, 1 (2020), 396–413.

[31] Mark D Wilkinson, Michel Dumontier, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3, 1 (2016), 1–9.

[32] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis C. M. Lau. 2015. A C-LSTM Neural Network for Text Classification. *CoRR* abs/1511.08630 (2015).