2023 IEEE International Conference on Data Mining Workshops (ICDMW) | 979-8-3503-8164-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICDMW60847.2023.00139

# Citation Style Classification: a Comparison of Machine Learning Approaches

Artyom Kopan
*Saint-Petersburg University*
Saint-Petersburg, Russia
artyom.kopan@gmail.com

Anna Smirnova
*Universe Data*
Saint-Petersburg, Russia
anna.en.smirnova@gmail.com

Ilya Shchuckin
*Saint-Petersburg University*
Saint-Petersburg, Russia
shchuckinilya@gmail.com

Vladislav Makeev
*Saint-Petersburg University*
Saint-Petersburg, Russia
makeev.vladislav.d@gmail.com

George Chernishev
*Saint-Petersburg University,*
*Universe Data*
Saint-Petersburg, Russia
chernishev@gmail.com

***Abstract*—Citation style classification is a task that aims to detect a citation style according to which a given bibliographic entry is formatted. There are more than a hundred of recognized citation styles available, including popular ones such as ACM, IEEE, MLA, APA, and even more exotic ones. Automatic detection of citation style can be used in document linters, such as those intended for conference and term papers. Using automatic style classification enables people who assess articles in large quantities reduce cognitive load and increase efficiency. Apart from this, automatic detection of citation style can be used for reference parsing, topic classification, and reference extraction.**

**In this paper we propose two novel approaches to citation style classification using both deep and classic machine learning methods. We evaluate them using a specially designed dataset, consisting of 6 million bibliographic records and spanning 91 citation styles. Our experiments showed that the proposed approaches significantly outperform the existing state-of-the-art solution, while also supporting almost five times more citation styles.**

***Index Terms*—citation, citation classification, reference classification, citation style classification**

## I. Introduction

Bibliographic records are an integral part of research and academia. They play a crucial role in acknowledging the work of other researchers as well as tracing and attributing the sources of information correctly. Academic bibliographic records are formatted using *citation styles*, which are sets of rules that dictate the structure, capitalization, and arrangement of information in the record. Various citation styles have been developed and used throughout the years. For example, the IEEE citation style is commonly used in engineering and computer science, while the MLA style is used in humanities, such as literature and language studies. There cannot be a universal citation style, as each of them is specifically tailored for a particular discipline, containing and presenting the most pertinent information. Conforming to the selected style is therefore important, as it helps to achieve consistency and clarity.

However, when a large number of bibliographic records has to be checked for proper formatting, attempting to do so manually can be laborious and time-consuming. Thus, we consider the task of automatically identifying bibliographic styles of citations in scientific texts. A system that solves this task is an important component in applications that examine formatting of articles. Such applications include linters that check conference submissions for adherence to the given template, or tools that assist teaching staff in efficiently conducting large-scale reviews of student papers. Automating such routine tasks reduces the cognitive load and manual workload involved, and makes the assessment process significantly easier.

The task of citation style detection can be interpreted in two ways: as a *single-record* or a *multi-record* classification problem. The single-record approach focuses on classifying the citation style of a singular bibliographic record, while multi-record classification aims to identify the citation style of an ordered collection of records contained in a single document. These two tasks have different scopes of application and would require different approaches.

Apart from building linters, the single-record approach is useful [1] for reference parsing, topic classification, and reference extraction. In this paper, we consider the single-record classification task. However, it is important to note that this approach entails a significant limitation: it is incapable of correctly identifying citation styles that feature record sorting.

We propose two approaches to the task: 1) classical machine learning techniques and 2) deep learning methods. We explore classical machine learning techniques due to their suitability for low-resource environments and compatibility with self-hosted systems, which do not rely on cloud computing services. This is particularly relevant for users with limited budgets and sensitive information, such as unpublished articles. Additionally, classical solutions require less computational resources, which is important for minimizing the environmental footprint and making NLP more sustainable [2].

We then assess the quality of a deep learning-based approach, which requires a special-purpose GPU.

Our contributions include:

1) A 6-million dataset of bibliographic records.

2375-9259/23/$31.00 ©2023 IEEE
DOI 10.1109/ICDMW60847.2023.00139

2) A novel deep learning approach based on BERT, and a reimplementation of the state-of-the-art approach. All implementations and a dataset are open-source.
3) An evaluation of the proposed approaches and the existing solution.

We plan to integrate the resulting system with `Mundane Assignment Police (MAP)`[1], an open-source web app designed to assist people in writing, assessing, and grading Bachelor's and Master's theses by helping identify common mistakes, ranging from syntactic checks, such as using an undirected double quote character in the place of a left quotation mark, to more content-oriented checks, such as suggestions regarding the size and the order of the sections in the paper.

## II. Background and Related Work

The domain of *citation analysis* is a rich research area which is dedicated to studying various aspects of the usage of citations in academia, such as citation type classification, citation networks, and evaluation of citation metrics. However, to the best of our knowledge, there has not been any other work on the more "formal" task of citation *style* classification that we propose except the work of Dominika Tkaczyk [1]. Therefore, we will review several articles from the wider area of citation analysis, point out why they are not suitable for our needs, and then move on to our specific task.

The work of Charles Jochim and Hinrich Schütze [3] focuses on improving citation classification by extracting features that would capture the relationship between the citing and cited papers. However, it attempts to classify citations according to a scheme proposed in the work of Michael Moravcsik and Poovanalingam Murugesan [4]. This schema represents each citation using four facets that determine which class the citation belongs to by answering questions such as if the citation refers to an idea or a tool. Furthermore, this semantic classification partially relies on analyzing the context of citation, such as cue words surrounding the citing. Thus, this work differs from syntactical analysis of bibliographic references proposed in this paper.

A paper by Cailing Dong and Ulrich Schäfer [5] tackles a similar problem by defining a classification schema whose elements are distinguished by textual, physical, and syntactic features. This paper divides citations into categories such as use, refutation, comparison and so on. Despite taking on the same broad issue as this paper, that is, creating a system for assisting researchers using reference classification, the papers vary in both the data that is analyzed, which includes context of the citation for the paper in question [5], and by the resulting classification, which subsequently leads to differences in the applications of produced labels.

Multiple works [6]–[8] use syntactic parsing for classification of citations. They intend to aid researchers by minimizing the time needed to find relevant papers by providing labels indicating if a particular paper is a critique/rebuttal of the other paper. For example, paper [6] classifies research citations, dividing them into three types: positive, negative, and neutral. Despite those works' aim of solving this specific classification issue and consequently needing a way to process references, none of the papers provide a sophisticated way of determining the reference style of the citation. All three papers retrieve references by either parsing them as a fixed citation style (APA in [7]), differentiating between a few styles using simple predicate (APA, AMA and IEEE in [6]) or bypassing information retrieval from references using external tools (Batch Citation Matcher in [8]).

Finally, one must mention the domain of *automated peer review*, which aims to build tools that, among other things, offer various kinds of citation analysis. Unfortunately, their provided functionality is rather straightforward, e.g. detection of missed references, detection of malformed ones, and others. Next, their developers rarely publish their approaches and instead prefer to build tools, which are often commercial products. As the result, we failed to discover even a single paper on the topic of citation analysis in this domain. However, we refer an interested reader to reference [9], where a large list of such tools is presented.

As evident from this review, the aforementioned papers have addressed various aspects of citation and reference classification, but none of them directly solved the task of classifying the styles of bibliographic records. To the best of our knowledge, Tkaczyk's work [1] is the only one that addresses the task of creating a bibliographic style classifier (*single-record classification*) and provides an open-source solution. The classifier detects 17 citation styles, while classifying the remaining styles as "unknown". The full list of the used citation styles can be found in the reference. The approach uses TF-IDF feature representation and a Logistic Regression model. Their training and testing data was automatically generated using two distinct samples of Crossref metadata. Each sample included 5K original records, which were then formatted to yield 85K pairs of `(record, style)` pairs. Furthermore, 5K records of an "unknown" style were added to each sample. As a result of postprocessing, both train and test sets contained around 87K records. Their proposed solution achieved an accuracy of 94.7% on the test set. In our work, we refer to this approach as a baseline.

## III. Proposed Solution

Our first task was to construct a dataset, and for this, we had to find a sufficient number of bibliographic records (.bib files) and bibliographic style files (.bst files). We primarily obtained bibliographic records from DBLP, acquiring more than six million records in total. The LaTeX bibliographic styles were sourced from CTAN, where we downloaded 91 relevant `.bst` files. Next, we have generated PDF documents from these bib and bst files using pdfLatex[2]. Thus, we acquired 6 million PDF documents. Finally, we parsed them using the

[1]https://github.com/Darderion/map

[2]Code can be found at `https://github.com/ArtyomKopan/CitationStyleClassifier`

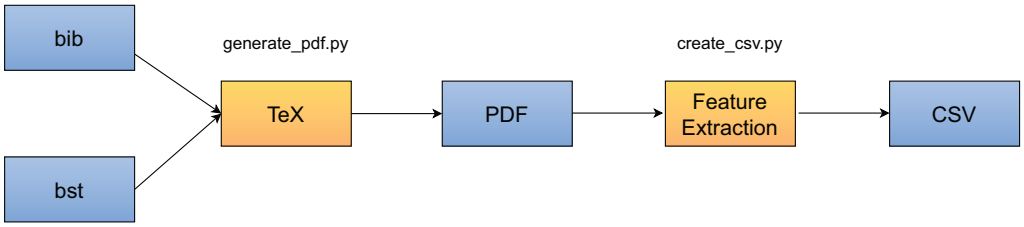


Fig. 1. Dataset generation pipeline

PDFium[3] library. The overall workflow with employed scripts is presented in Figure 1.

### A. Feature Extraction

As the actual words of the bibliographic records are not relevant to the classification task, and it is instead only their format that matters, we need to pre-process the data to extract format-related features. In this work, we have modified the baseline approach [1] to feature extraction.

In this approach, words in the bibliographic entry are replaced with the following tokens:

TABLE I
TOKEN REPLACEMENTS

| Original token type | Replacement token |
|---|---|
| uppercase word | `upword` |
| capitalized word | `capword` |
| any other word | `othword` |
| capital letter | `caplet` |
| lowercase letter | `smallet` |
| year (1900 to 2100) | `year` |
| other number | `num` |
| space | `sp` |
| punctuation symbols | unchanged |

Our modifications are as follows. First, in the baseline approach, spaces are not counted as features, and second, we did not include special tokens for "start" and "end", since they are not needed due to the structure of our dataset. Thus, we converted each entry into a string consisting of the above tokens. Next, we vectorized them with `CountVectorizer` from sklearn, extracting bigrams. Larger n-grams were not used because the dataset contains too many bibliographic records and building them would take too much time.

We have also explored a different approach to feature extraction, which consisted in counting simple numeric and categorical features that reflect the presence of various elements in a bibliographic record. These elements comprise punctuation and terms like "Abstract", "Key", and "Annotation", among many others. The preliminary results were subpar (around 60% accuracy) and we decided not to benchmark them.

### B. Training

We have used the following classical machine learning models: SVM, Naive Bayes, Random Forest (implementations were taken from *scikit-learn*), and Decision trees with gradient boosting[4]. We used a 80/20 train/test split in all experiments.

The deep learning approach that we have selected is fine-tuning a `distilbert-base-uncased` model [10] on raw records created with token replacement described above (i.e., non-vectorized sequences). We added 8 tokens from the list above to the DistilBERT tokenizer and resized the model accordingly. The maximum length of the tokenizer was set at 150. The train-test splits are explained below.

We added the following layer in order to fine-tune the BERT-based model:

```
nn.GELU(),
nn.Linear(D_in, 300),
nn.GELU(),
nn.Dropout(0.05),
nn.Linear(300, D_out)
```

Here, `D_in` is the size of the BERT vectors (768), and `D_out` is the number of classes.

We employed the AdamW optimizer and the OneCycleLR scheduler. We set the learning rate to 2e-5 and the epsilon value to 1e-8, and the max learning rate for the scheduler was set to 2e-5 with div factor of 1e-4. The pct_start parameter was set to 0. It was trained for 2 epochs on the full dataset and for 20 epochs on the small one. We refer to this model as `BERT-based` in the Evaluation[5].

## IV. EVALUATION

### A. Experimental Setup

The classical models were trained on a server with 2xIntel(R) Xeon(R) Gold 6230 CPU 2.10GHz and an NVIDIA GP104 [GeForce GTX 1080] GPU. The BERT-based classifier was trained in Google Colab using an A100 GPU. All metrics, except accuracy, are macro-averaged, which calculates the metric independently for each class and takes the average treating all classes equally, disregarding their size or imbalance.

### B. Experiment 1: Classical methods

Our first experiment aimed to select the best classical method by repeating the experiments from [1]. We evaluated logistic regression (which is the state-of-the-art method),

[3] `pypi.org/project/pypdfium2/`

[4] https://catboost.ai/

[5] The BERT model files and the processed dataset can be found at huggingface.co/citclass

TABLE II
CLASSIC MACHINE LEARNING METHODS (91 CLASS)

| Model | Accuracy | Precision | Recall | F1 | Training time |
|---|---|---|---|---|---|
| SVM | 0.61 | 0.67 | 0.58 | 0.58 | 29 min |
| Naive Bayes | 0.55 | 0.58 | 0.55 | 0.55 | **2 min** |
| Random forest | 0.68 | 0.67 | 0.66 | 0.65 | **2 min** |
| Decision tree | **0.72** | **0.71** | **0.70** | **0.70** | 9 min |
| Logistic regression | N/A | N/A | N/A | N/A | 15+ hours |

SVM, naive Bayes, and random forest. Additionally, we have included the decision tree method.

For our experiment we used the full 6 million record dataset with an 80/20 split. Results are presented in Table II, alongside with respective training times.

This experiment resulted in the following:

1) The state-of-the-art method failed to complete training in a reasonable time on our dataset. The exact reasons are presented below, in Experiment 2 discussion.
2) The decision tree was superior to all other considered methods, while requiring moderate training time.

Based on these results, we have decided to proceed with both Decision tree and state-of-the-art method.

TABLE III
DEEP LEARNING METHODS (91 CLASS)

| Model | Acc | Pr | Rec | F1 | Training time |
|---|---|---|---|---|---|
| Small dataset | | | | | |
| Decision tree | 0.71 | 0.69 | 0.70 | 0.69 | **2 min** |
| BERT-based | **0.74** | **0.73** | **0.72** | **0.70** | 1.5 hours |
| Baseline model | 0.72 | 0.69 | 0.70 | 0.69 | 1 hour |
| Full dataset | | | | | |
| Decision tree | 0.72 | 0.71 | 0.70 | 0.70 | **9 min** |
| BERT-based | **0.76** | **0.75** | **0.74** | **0.74** | 7 hours |
| Baseline model | N/A | N/A | N/A | N/A | 15+ hours |
| Small dataset, evaluated on 5.9 million | | | | | |
| Decision tree | 0.70 | 0.68 | 0.69 | 0.68 | **5 min (inference)** |
| BERT-based | **0.74** | **0.73** | **0.71** | **0.69** | 1.5 hours (inference) |
| Baseline model | 0.72 | 0.70 | 0.70 | **0.69** | 7 min (inference) |

### C. Experiment 2: Deep learning

In this experiment we have compared deep learning approach to the classical methods: the state-of-the-art (baseline) and the Decision tree. Due to the inability of baseline approach to train on the full dataset within reasonable time, we have selected a 102k subset of records with a 80/20 train-test split and trained the baseline and our models on it. This will be referred to as the *small* dataset. Next, we trained out models on the entire 6 million dataset with a 80/20 split. This is referred to as the *full* dataset. Finally, we evaluated the models trained on the small dataset (that is, on the 80% of the 102k subset selected earlier) on the remaining 5.9 million records in order to obtain a better understanding of their results. The results are presented in Table III.

In the *small* dataset setting, the decision tree model performs similarly to the baseline and BERT-based approaches. The BERT-based model outperforms the other two, but its train time is approximately one and a half hours and it requires resources that are not always readily available and much less sustainable than a consumer grade GPU which was sufficient for Experiment 1. Furthermore, these results carry over to the full dataset quite well. However, BERT-based model's inference on 5.9 million records takes around 1.5 hours on expensive hardware, and much more on hardware that is readily accessible. This had originally motivated our use of DistilBERT, which has fewer parameters and is faster, but even that does not allow the approach to achieve the speed of the classical ML solutions (approximately two minutes). Finally, the baseline model fails to train on the full dataset, since it can neither run on GPU nor use parallelism during training. The BERT-based approach is able to train on 6 million records, but it takes around 7 hours on an expensive GPU.

Turning to the third section of Table III, it is evident that the baseline model trained on the *small* dataset fails to yield good results on the *full* dataset. The same is true for other models that were trained and evaluated in the same environment.

### D. Experiment 3: Dialect Merge

The relatively low-quality of obtained results prompted us to perform an in-depth study of the underlying data. We started by looking into per-class results and found that they vary greatly depending on a class. While some classes demonstrated acceptable results, some did not, and it turned out to be possible to group under-performing ones, see Table IV.

We have analysed the existing styles, and found out that many of them are dialects of each other, which differ very little in practice or have peculiarities that obstruct successful classification.

First of all, some dialects, e.g. ieeeannot[6] and ieeetrans[7], differ only in the set of available optional fields, making them indistinguishable from each other unless the data for said fields is provided. This issue proves to be even more pronounced when the provided optional fields are not used by standard bibliography styles (such is the case[8] with `chicago`[9] and `chicagoa`[10] styles, with `chicago` being a subset of `chicagoa` without the *annotation* field).

Another issue that we have encountered is sorting. Many styles contain dialects that employ sorting of records, e.g.

[6]https://www.bibtex.com/s/bibliography-style-misc-ieeeannot/
[7]https://www.bibtex.com/s/bibliography-style-ieeetran-ieeetrans/
[8]https://mirror.truenetwork.ru/CTAN/macros/latex/contrib/biblatex/doc/biblatex.pdf
[9]https://www.bibtex.com/s/bibliography-style-chicago-chicago/
[10]https://www.bibtex.com/s/bibliography-style-misc-chicagoa/

TABLE IV
BEFORE DIALECT MERGE

| Dialect | Precision | Recall | F1 |
|---|---|---|---|
| ama | 0.94 | 0.94 | 0.94 |
| apa | 0.84 | 0.88 | 0.86 |
| ... | ... | ... | ... |
| IEEEannot | 0.36 | 0.26 | 0.3 |
| IEEEtran | 0.34 | 0.42 | 0.37 |
| IEEEtranN | 0.46 | 0.45 | 0.46 |
| IEEEtranS | 0.35 | 0.22 | 0.27 |
| IEEEtranSA | 0.86 | 0.91 | 0.88 |
| IEEEtranSN | 0.51 | 0.51 | 0.51 |
| ... | ... | ... | ... |
| bbs | 0.85 | 0.87 | 0.86 |
| hum2 | 0.92 | 0.94 | 0.93 |

TABLE V
AFTER DIALECT MERGE

| Dialect | Precision | Recall | F1 |
|---|---|---|---|
| ama | 0.97 | 0.93 | 0.95 |
| apa | 0.94 | 0.92 | 0.93 |
| ... | ... | ... | ... |
| IEEE | 0.97 | 0.96 | 0.96 |
| ... | ... | ... | ... |
| bbs | 0.93 | 0.85 | 0.89 |
| hum2 | 0.95 | 0.94 | 0.94 |

`ieeetransa` and `ieeetransn`. Since we specifically solve the *single-record* classification problem, the sorted dialects are indistinguishable from the unsorted ones.

Consequently, we merged the found dialects (`ieee`, `gost`, `chicago`, `h-physrev`), which resulted in 71 styles in total. The resulting distribution is shown in Figure 2. The majority of classes have about 72K records, with seven classes having more due to the merging. Some of classes, e.g., `bookdb`, have slightly fewer records due to `pdfLatex` errors, as not all bibliographic records could be successfully processed due to various style issues.

We have then retrained and reevaluated all the models. The results are presented in Tables VI and VII.

As it can be seen, dialect merge allowed for a significant improvement of the quality of obtained results for all methods. Interestingly, inference time was reduced for classic ones — five times for Decision tree and about 20% for the baseline method — while training time increased. Finally, Table V demonstrates that metrics for "problematic" classes improved as well.

## V. DISCUSSION

The results we have achieved with the BERT-based model might be surprising in how little of an improvement they offer over the decision tree model, which trains 40 times faster and with much less resources. However, we believe that this is due to the nature of the task: the detection of a bibliographic style essentially boils down to detecting various punctuation features such as capital letters and punctuation marks without considering actual semantic information. While we have adopted the preprocessing approach that was shown to adequately represent citation styles by [1], it only provides 6 additional tokens (`year` and `num` are already in the dictionary) to the DistilBERT model. As described previously, this leads to the records consisting of 8 tokens and punctuation marks in total (see Figure 3 for an example).

TABLE VI
CLASSIC MACHINE LEARNING METHODS (71 CLASS)

| Model | Acc | Pr | Rec | F1 | Training time |
|---|---|---|---|---|---|
| SVM | 0.70 | 0.73 | 0.66 | 0.66 | 29 min |
| Naive Bayes | 0.55 | 0.61 | 0.54 | 0.52 | **2 min** |
| Random forest | 0.79 | 0.81 | 0.73 | 0.74 | **2 min** |
| Decision tree | **0.84** | **0.84** | **0.81** | **0.81** | 9 min |
| Logistic regression | N/A | N/A | N/A | N/A | 15+ hours |

TABLE VII
DEEP LEARNING METHODS (71 CLASS)

| Model | Acc | Pr | Rec | F1 | Training time |
|---|---|---|---|---|---|
| Small dataset | | | | | |
| Decision tree | 0.84 | **0.83** | 0.80 | **0.81** | **2 min** |
| BERT-based | **0.85** | 0.82 | **0.81** | **0.81** | 1.5 hours |
| Baseline | 0.80 | 0.78 | 0.76 | 0.76 | 1 hour |
| Full dataset | | | | | |
| Decision tree | 0.84 | 0.84 | 0.81 | 0.81 | **15 min** |
| BERT-based | **0.87** | **0.87** | **0.84** | **0.85** | 7 hours |
| Baseline model | N/A | N/A | N/A | N/A | 15+ hours |
| Small dataset, evaluated on 5.9 million | | | | | |
| Decision tree | 0.84 | **0.83** | 0.80 | **0.81** | **50 s (inference)** |
| BERT-based | **0.85** | **0.83** | **0.81** | **0.81** | 1.5 hours (inference) |
| Baseline model | 0.80 | 0.77 | 0.75 | 0.76 | 6 min (inference) |

This does not provide enough semantic information to be captured by the embeddings, making the model's results comparable to a more classical approach. However, we have consciously gotten rid of the semantic information, because it would be impossible to classify the records otherwise, as the model would depend on the thematic content of the record. A future development for this task would be to cluster citation styles by their domain and include the domain of the cited article in the classification process.

Another point to consider is the difference between the computational resources needed to train the two competing models. An A100 GPU is in a different league compared to a consumer-grade GTX1080. Unlike the high-performance A100, the GTX1080 and similar models can be easily obtained and use much less resources. This makes them a more sustainable choice for an in-house deployment in environments such as universities.

Finally, it is necessary to discuss why our results differ significantly from the ones from the original study [1], especially for the logistic regression method. Their reported accuracy is 0.94 and our initial experiments yield 0.61. Such results can be explained by the fact that in our work we used five times more citation styles while employing more uncommon ones. With increase of the number of styles the classification problem becomes significantly harder for all methods and results of Experiment 3 corroborate this.

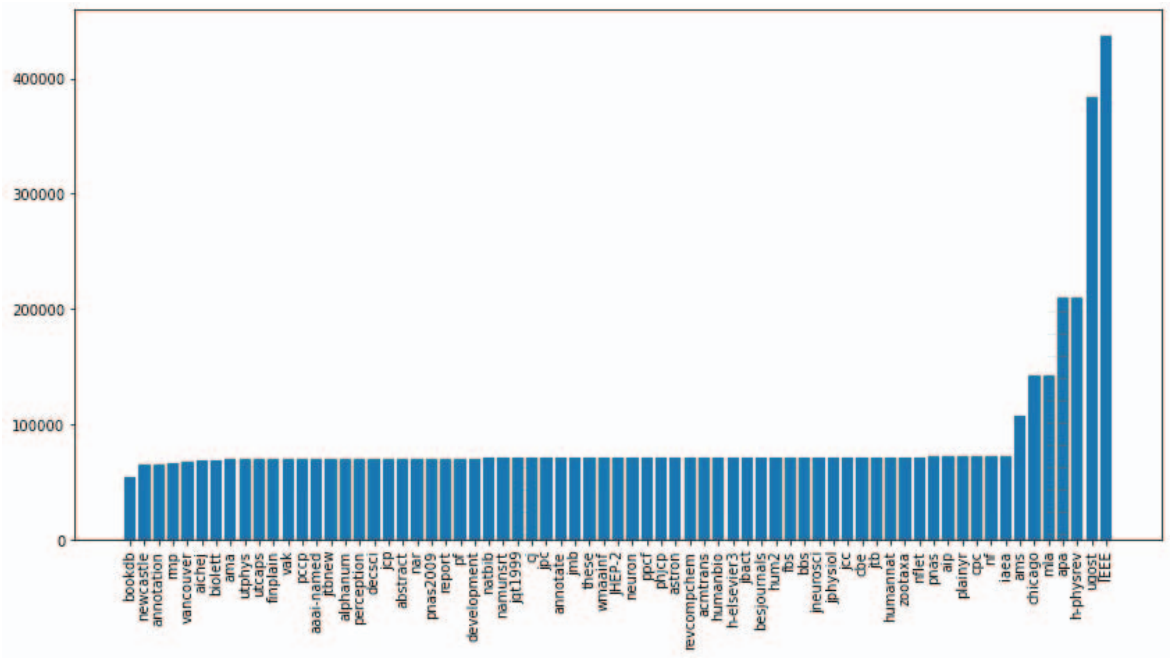


Fig. 2. The resulting class distribution (71 classes)

| tokenized_record (string) | style_name (string) |
|---|---|
| "[ num ] sp upword , sp caplet . sp caplet . sp othword sp othword . , sp othword , sp capword sp capword sp othword sp othword sp upword sp capword sp capword sp othword capword sp capword sp othword sp capword sp othword sp capword sp capword sp capword sp ( upword / upword sp year ) , sp capword , capword , sp capword sp num - num , sp year , sp othword sp num sp othword sp upword sp capword sp capword , sp upword - upword . othword , sp year ." | "iaea" |

Fig. 3. An example processed record. Punctuation mark tokens are not replaced, while the word tokens are replaced with tokens that highlight their most important feature, as per Table I.

## VI. Conclusion

In this paper, we have studied methods for classification of bibliographic record style for low-resource environments. We have proposed two methods and compared them with the state-of-the-art baseline approach. Our experiments showed that proposed approaches outperform the existing solution, while supporting almost five times more citation styles. Furthermore, we have experimentally proven that the decision tree approach is a feasible, sustainable and scalable solution for low-resource environments.

As for future work, we plan to experiment further with deep learning-based methods. Another approach that could be undertaken here is extreme multi-label classification — classification specifically tailored for a large label space. Another method we plan to explore is vector indexing. A preliminary study of indexing DistilBERT vectors with HNSW showed promising results (84 % probability of a correct answer at top-k=5 with 91 styles). However, this is a reformulation of the original task and thus needs to be explored separately.

## References


[1] D. Tkaczyk, "What's your (citations') style?" *Crossref*, 2019. [Online]. Available: https://www.crossref.org/blog/whats-your-citations-style/

[2] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Commun. ACM*, vol. 63, no. 12, p. 54–63, nov 2020. [Online]. Available: https://doi.org/10.1145/3381831

[3] C. Jochim and H. Schütze, "Towards a generic and flexible citation classifier based on a faceted classification scheme," in *Proceedings of COLING 2012*, 2012, pp. 1343–1358.

[4] M. J. Moravcsik and P. Murugesan, "Some results on the function and quality of citations," *Social studies of science*, vol. 5, no. 1, pp. 86–92, 1975.

[5] C. Dong and U. Schäfer, "Ensemble-style self-training on citation classification," in *Proceedings of 5th international joint conference on natural language processing*, 2011, pp. 623–631.

[6] B. H. Butt, M. Rafi, A. Jamal, R. S. U. Rehman, S. M. Z. Alam, and M. B. Alam, "Classification of research citations (crc)," *arXiv preprint arXiv:1506.08966*, 2015.

[7] H. Nanba, N. Kando, and M. Okumura, "Classification of research papers using citation links and citation types: Towards automatic review article generation." *Advances in Classification Research Online*, pp. 117–134, 2000.

[8] A. M. Cohen, W. R. Hersh, K. Peterson, and P.-Y. Yen, "Reducing workload in systematic review preparation using automated citation classification," *Journal of the American Medical Informatics Association*, vol. 13, no. 2, pp. 206–219, 2006.

[9] J. Lin, J. Song, Z. Zhou, Y. Chen, and X. Shi, "Automated scholarly paper review: Concepts, technologies, and challenges," *Information Fusion*, vol. 98, p. 101830, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S156625352300146X

[10] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter," *CoRR*, vol. abs/1910.01108, 2019. [Online]. Available: http://arxiv.org/abs/1910.01108