Erschienen in: Experimental IR Meets Multilinguality, Multimodality, and Interaction : 8th International Conference of the CLEF Association, CLEF 2017, Dublin, Ireland, September 11-14, 2017, Proceedings / Jones, Gareth J.F. et al. (Hrsg.). - Cham : Springer, 2017. - (Lecture notes in computer science ; 10456). - S. 82-94. - ISBN 978-3-319-65813-1
https://dx.doi.org/10.1007/978-3-319-65813-1_7

Evaluating and Improving the Extraction of Mathematical Identifier Definitions

Moritz Schubotz(✉), Leonard Krämer, Norman Meuschke, Felix Hamborg, and Bela Gipp

University of Konstanz, Konstanz, Germany
{moritz.schubotz,leonard.kramer,norman.meuschke,
felix.hamborg,bela.gipp}@uni.kn

Abstract. Mathematical formulae in academic texts significantly contribute to the overall semantic content of such texts, especially in the fields of Science, Technology, Engineering and Mathematics. Knowing the definitions of the identifiers in mathematical formulae is essential to understand the semantics of the formulae. Similar to the sense-making process of human readers, mathematical information retrieval systems can analyze the text that surrounds formulae to extract the definitions of identifiers occurring in the formulae. Several approaches for extracting the definitions of mathematical identifiers from documents have been proposed in recent years. So far, these approaches have been evaluated using different collections and gold standard datasets, which prevented comparative performance assessments. To facilitate future research on the task of identifier definition extraction, we make three contributions. First, we provide an automated evaluation framework, which uses the dataset and gold standard of the NTCIR-11 Math Retrieval Wikipedia task. Second, we compare existing identifier extraction approaches using the developed evaluation framework. Third, we present a new identifier extraction approach that uses machine learning to combine the well-performing features of previous approaches. The new approach increases the precision of extracting identifier definitions from 17.85% to 48.60%, and increases the recall from 22.58% to 28.06%. The evaluation framework, the dataset and our source code are openly available at: https://ident.formulasearchengine.com.

1 Introduction

Mathematical formulae consist of *identifiers* (e.g. x, π or σ), *symbols* (e.g. $+$, $\leq$ or $\rightarrow$) and *other constituents*, such as numbers. Formally, any definition consists of three components:

1. the *definiendum*, which is the expression to be defined
2. the *definiens*, which is the phrase that defines the definiendum
3. the *definitor*, which is is the verb that links definiendum and definiens

The task in mathematical identifier definition extraction is to find definitions whose definiendum is a mathematical identifier and extract the definiens.

As an example, we use an explanation of the Planck-Einstein relation (https://en.wikipedia.org/wiki/Planck-Einstein_relation) in Wikipedia (see Fig. 1). The Planck-Einstein relation $E = hf$ consists of three identifiers: E, h, f and two symbols: '=', '$\cdot$' (times). The text surrounding the formula contains the definiens for photon energy and wave frequency. In this case, the definiens photon energy is particularly specific, since it contains an intra-wiki link to a unique concept identified by the Wikidata item Q25303639. However, the explanatory text does not give a definition for the Planck constant h. A reader or an information system must infer this missing information from elsewhere, which poses a challenge to both a reader and a system.

Identifier-definiens pairs contain semantic information that can improve mathematical information retrieval (MIR) tasks, such as formula search and recommendation, document enrichment, and author support. To increase the accessibility of this valuable semantic information, we address the automated extraction of mathematical identifiers and their definiens from documents as follows. In Sect. 2, we review existing approaches for mathematical identifier definition extraction. In Sect. 3, we describe the development of an automated evaluation framework that allows for objective and comparable performance evaluations of extraction approaches. Furthermore, we describe how we used machine learning to create a new extraction approach by combining the well-performing features of existing approaches. In Sect. 4, we present the results of evaluating the extraction performance of existing and our newly developed extraction approach. In Sect. 5, we summarize the findings of our comparative performance evaluations and present suggestions for future research.

2 Related Work

This Section briefly reviews the following approaches to mathematical identifier definition extraction: (1) the statistical feature analysis of Schubotz et al. [13] (Sect. 2.1), (2) the pattern matching approach of Pagel et al. [9] (Sect. 2.2), (3) the machine learning approach of Kristianto et al. [7] (Sect. 2.3). See [10] for an extensive review of related work.

2.1 Statistical Feature Analysis (ST)

Schubotz et al. [13] proposed a *mathematical language processing (MLP)* pipeline and demonstrated the application of the pipeline for extracting the definiens of identifiers in formulae contained in Wikipedia. In summary, the MLP pipeline includes the following steps:

1. **Preprocessing:** Parse wikitext input format, perform tokenization, part-of-speech (POS) tagging and dependency parsing using an adapted version of the Stanford CoreNLP library. The modified library can handle mathematical identifiers and formulae by emitting special tokens for identifiers, definiens candidates and formulae.

The Planck-Einstein relation connects the particular photonenergy E with its associated wave frequency f

$$E = hf. \tag{1}$$

identifier	definiens
E	Q25303639 (Photon energy)
h	NIL
f	wave frequency

Fig. 1. Excerpt of a sentence from Wikipedia explaining the Planck-Einstein relation (https://en.wikipedia.org/w/index.php?title=Planck_constant\&oldid=766777932) (top) and the extractable identifiers and corresponding definiens (bottom).

2. **Find identifiers** in the text.
3. **Find candidates for identifier-definiens pairs**.
4. **Score identifier-definiens pairs** using statistical methods.
5. **Identify and extract namespaces:** Cluster documents, map the clusters to document classification schemata and determine identifier definitions specific to each identified class in the schema, i.e. specific to a namespace (NS).

To score identifier-definiens pairs, Schubotz et al. used the scoring function shown in Eq. 2. The function considers the number of words Δ between the identifier and the definiens, the number of sentences n between the first occurrence of the identifier and the sentence that connects the identifier to the definiens, and the relative term frequency of the definiens t in document d.

$$R(\Delta, n, t, d) = \frac{\alpha R_{\sigma_\alpha}(\Delta) + \beta R_{\sigma_\beta}(n) + \gamma \mathrm{tf}(t, d)}{\alpha + \beta + \gamma} \mapsto [0, 1], \tag{2}$$

The parameters α, β and γ are used to weigh the influence of the three factors by making the following assumptions:

α The definiens and the identifier appear close to each other in a sentence.
β The definiens appears close to the first occurrence of the identifier in the text.
γ The definiens is used frequently in the document.

To derive α and β, Schubotz et al. used the zero-mean Gaussian normalization function $R_\sigma(\Delta) = \exp\left(-\frac{1}{2}\frac{\Delta^2-1}{\sigma^2}\right)$ to map the infinite interval of the distances Δ and n to $[0, 1]$. The parameters σ_α and σ_β control the width of the function.

Schubotz et al. report a precision of $p = .207$ and a recall of $r = .284$ for extracting identifier definitions [13]. The weighting parameters, the Gaussians and the threshold for the overall score must be manually adjusted to the specific use case, which can be a tedious process. The major advantages of the statistical approach are its language-independence and adjustability to different document collections.

2.2 Pattern Matching (PM)

Pagel et al. employed a pattern matching approach for POS tag patterns to extract identifier-definiens pairs from Wikipedia articles [9]. The lines 1–10 in Table 1 show the patterns, which were defined by domain experts. Patterns like <identifier> denote(s?) the <definiens>, which would match the definition *'h denotes the Planck constant'* have a high probability of retrieving a true positive (tp) result. However, simpler patterns, such as <definiens> <identifier> have a high probability of producing false positive (fp). For instance, this pattern would match the apposition *'photon energy E'*, but also the phrase *'subsection a'* in the description of a law.

Pagel et al. reported a precision of $p = .911$, and recall of $r = .733$ for their approach [9]. While the recall of such a pattern matching approach can easily be increased by adding additional patterns, the precision declines if the added patterns are too broad. Therefore, Pagel et al. concluded that using more than the patterns 1–10 in Table 1 does not significantly increase the performance [9].

The sentence patterns 3–10 in Table 1 achieved a high precision in the evaluation of Pagel et al. We consider these patterns promising candidates for inclusion in a hybrid approach that uses machine learning to combine the pattern matching approach of Pagel et al. and the statistical feature analysis of Schubotz et al. However, before applying the sentence patterns for extracting identifier definitions from a different corpus, the suitability of the patterns must be re-evaluated, since different text genres, e.g., encyclopedic article vs. scientific publication, may use different notational conventions. Furthermore, the pattern matching approach is language-dependent.

2.3 Machine Learning

Kristianto et al. proposed a machine learning approach to extract natural language descriptions for entire formulae from academic documents [7]. This extraction task is slightly different from extracting identifier-definiens pairs. Kristianto et al. associate each mathematical expression with a span of words that describes the expression. For example, for the sentence: "*..the number of permutations of length n with exactly one occurrence of* 2–31 *is* $\left(\frac{2n}{n-3}\right)$..", the gold standard of Kristianto et al. states that the correct description of "$\left(\frac{2n}{n-3}\right)$" is "*the number of permutations of length n with exactly one occurrence of* 2–31".

In contrast, the sentence contains only one identifier n, whose definiens is *'length'*. Kristianto et al. used the native Standford CoreNLP library for their analysis, whereas Schubotz et al. modified the CoreNLP library to create their MLP pipeline (cf. Sect. 2.1). To identify formulae descriptions, Kristianto et al. defined a large set of features, which they classified into three groups:

1. *pattern matching*: features similar to those of Pagel et al. (see Sect. 2.2);
2. *basic*: features that consider the POS tags between pairs of identifier and definiens, as well as the POS tags in their immediate vicinity;
3. *dependency graph (DG)*: features related to the DG of a sentence.

Kristianto et al. used a support vector machine (SVM) [2] for a combined analysis of all features. Except for the features in the *pattern matching* group, their approach is applicable to documents in all languages supported by the Stanford CoreNLP library [8]. A significant drawback of the approach is the necessity to manually annotate a portion of the dataset to train the SVM classifier.

3 Methodology

The approaches we present in Sect. 2 perform different extraction tasks (extracting identifier definitions [9,13] vs. extracting formulae descriptions [7]) using different datasets (Wikipedia articles [9,13] vs. scientific publications [7]), which so far prevented a comparison of the reported precision an recall values.

To enable comparative performance evaluations for these and other extraction approaches, we created an open evaluation framework by extending the open source MLP and MIR framework Mathosphere introduced in [13]. Section 3.1 presents the evaluation framework and explains major improvements we made to Mathosphere's MLP pipeline. Section 3.2 describes how we used the developed framework to individually evaluate the three approaches we present in Sect. 2. Section 3.3 explains how we adapted and evaluated the approach of Kristianto et al. [7] for the task of extracting identifier definitions. Section 3.4 presents how we investigated the effect of considering Namespaces (NS) as part of identifier definition extraction [13].

3.1 Evaluation Framework

Our framework uses a subset of the dataset of the NTCIR-11 Math Retrieval Wikipedia task [12] and the gold standard created by Schubotz et al. for evaluating their statistical feature analysis approach (cf. Sect. 2.1) [13]. The dataset contains 100 formulae taken from 100 unique Wikipedia articles and contains 310 identifiers [12]. Every formula in the gold standard contains: (1) a unique query-id (qID); (2) the title of the document; (3) the id of the formula within the document (fid), which corresponds to the sequential position of the formula in the document; (4) the latex representation of the formula (math_inputtex).

The gold standard includes definiens for every identifier in a formula. In total, the gold standard includes 369 definiens for the 310 identifiers, or 575 definiens when counting wikidata links and link texts separately. However, distinguishing Wikidata links and link texts for the evaluation has a drawback. For example, the identifier c in the formula $f_c(z) = z^2 + c$ from the article on orbit portraits (https://en.wikipedia.org/w/index.php?title=Orbit_portrait\&oldid=729107245) is associated with two Wikidata concepts: parameter Q1413083 and coefficient Q50700. While this ambiguity can be interpreted as a shortcoming of insufficient concept specificity and definiteness of the Wikidata items as discussed by Corneli and Schubotz [3], we argue that current IR systems should be able to deal with such indefiniteness. In [13], each correctly extracted definition was regarded as a true positive. This can result in more

Table 1. All features used in the SVM to classify identifier-definiens pairs with rank, where possible. Feature groups 1–10: PM, 11–21: basic, 22–26: DG, 27–29: ST. The ranking was performed by comparing the merit of training with only one feature in isolation to training all features

#	Description	Merit	Rank
1	`<definiens> <identifier>` [9]	0.196	5
2	`<identifier> <definiens>` [9]	<.001	27
3	`<identifier> denote(s?) <definiens>` [9]	0.001	20
4	`<identifier> denote(s?) the <definiens>` [9]	0.001	19
5	`<identifier> (is|are) <definiens>` [9]	0.001	21
6	`<identifier> (is|are) the <definiens>` [9]	0.059	13
7	`<identifier> (is|are) denoted by <definiens>` [9]	<0.001	24
8	`<identifier> (is|are) denoted by the <definiens>` [9]	<0.001	25
9	`let <identifier> be denoted by <definiens>` [9]	<0.001	22
10	`let <identifier> be denoted by the <definiens>` [9]	<0.001	23
11	Colon between identifier and definiens [7]	0.037	15
12	Comma between identifier and definiens [7]	0.121	7
13	Other math expression or identifier between identifier and definiens [7]	0.122	6
14	Definiens is inside parentheses and identifier is outside parentheses [7]	0.016	16
15	Identifier is inside parentheses and definiens is outside parentheses [7]	0.060	12
16	Identifier appears before definiens [7]	0.015	17
17	Surface text and POS tag of two preceding and following tokens around the definiens candidate [7]	0.441	1
18	Unigram, bigram and trigram of feature 17 [7]	0.441	1
19	Surface text and POS tag of three preceding and following tokens around the identifier [7]	0.398	2
20	Unigram, bigram and trigram of feature 19 [7]	0.398	2
21	Surface text of the first verb that appears between the identifier and the definiens [7]	0.093	9
22	Distance between identifier and definiens in the shortest edge path between identifier and definiens of the dependency graph [7]	0.001	18
23	Surface text and POS tag of dependency with length 3 from definiens along the shortest path between identifier and definiens [7]	0.292	4
24	Surface text and POS tag of dependency with length 3 from identifier along the shortest path between identifier and definiens [7]	0.328	3
25	Direction of 24. Incoming to definiens or not [7]	0.064	11
26	Direction of 25. Incoming to identifier or not [7]	<0.001	26
27	Distance between the identifier and definiens in number of words [7,13]	0.064	10
28	Distance of the identifier-definiens candidate from the first appearance of the identifier in the document, in sentences [13]	0.101	8
29	Relative term frequency of the definiens [13]	0.044	14

than one correct definition for an identifier. Therefore, we evaluated using the following policy:

1. Use the number of identifiers (310) as truth.
2. True positive: at least one definition for the identifier was found.
3. Ignore: more than one correct definition was found.

4. False positive: a definition that is not in the set of possible definitions.
5. False negative: no definition was found for the identifier.

This policy assigns an optimal score $p = r = 1$ if (1) only one correct definiens is retrieved, and (2) if more than one correct definiens is retrieved. Using this policy, we could compare the precision, recall, and F_1 score of the statistical approach, the pattern matching approach, and the newly developed approach (cf. Sect. 2.3). We packaged the new evaluation method in a Java tool (https://github.com/leokraemer/mathosphere/tree/temp/evaluation) that evaluates .csv files of the form `qId`, `title`, `identifier`, `definiens`.

During the development of the evaluation framework, we discovered several weaknesses of the MLP pipeline. In step 3 (find candidates for identifier - definiens pairs, cf. Sect. 2.1), we discovered that certain operators, such as special cases of 'd' in integrals and the '∞' symbol were misclassified as identifiers. While addressing this issue, we also created unit tests with the data from the gold standard to prevent future regressions in the identifier extraction. In addition, we discovered that many false positives included the identifier 'a'. Thus, we improved the identifier detection for simple Latin charters using style information from the Wikitext markup.

3.2 Evaluating Existing Approaches

Using the evaluation framework, we could accurately judge the impact of changes in the MLP pipeline and develop a new approach for the identifier-definiens scoring. As a first experiment, we evaluated the statistical feature analysis (cf. Sect. 2.1) and the pattern matching approach (cf. Sect. 2.2) individually, with and without the improvements to the preprocessing steps of the MLP pipeline. Additionally, we evaluated the union of the identifier definiens tuples returned by both approaches.

3.3 New Machine Learning Approach (ML)

Following the idea of Kristianto et al. [7], we employed a support vector machine to combine the strengths of the statistical feature analysis of Schubotz et al. [13] (cf. Sect. 2.1) and the pattern matching approach of Pagel et al. [9] (cf. Sect. 2.2) as well as to implicitly tune the parameters of the approaches. The SVM accepts as input nominal features, e.g., whether an identifier appears before its definiens, and ordinal features, e.g., the relative term frequency of identifiers. Using a filter that converts strings to word vectors, we can also use the SVM to train on parts of the original sentences and POS-Tags. After the feature vector generation phase, we obtain 7902 feature vectors of which 244 are actual matches of the gold standard and 7658 are true negatives. We use a combination of oversampling of the minority class and undersampling the majority class approach to balance the data for training. We chose an radial basis function (RDF) kernel, due to the non-linear characteristics of some of the features. We found the best hyperparameters in cost $= 1$ and $\gamma \approx 0.0186$. We trained four different classifiers examine

the performance of different feature classes: A classifier ML_ST_PM using only simple features (1–16 and 27–29 in Table 1), ML_no_DG without the features using the expensive dependency graph generation (1–21 and 27–29 in Table 1), ML_no_PM without the hand-crafted patterns (11–29 in Table 1) and ML_full with all features. The features are a combination of features used for the three approaches described in Sect. 2. Some features used by Kristianto et al. were too specific to the task of classifying formulae descriptions instead of identifier-definiens pairs and thus were ignored for our approach. The remaining features were adjusted to be compatible with the MLP pipeline (cf. Sect. 2.1).

For training, we used all extracted identifier-definiens candidates and annotated them with the information from the gold standard. We employed a 10-fold cross validation using the entire gold standard. We divided the test and the training sets on document level, i.e. we trained the model using the data from 90 documents and evaluated the model using the data from 10 other documents.

3.4 Evaluating the Influence of Namespaces

So far, we described approaches that operate on the level of individual documents to extract identifier definitions, i.e. approaches that implement the steps 1–4 of the MLP pipeline (cf. Sect. 2.1). We also executed and evaluated the computationally expensive step 5 of the MLP pipeline (namespace discovery), which requires to process the entire test collection. To enable an unbiased comparison, we build namespaces using each of the methods individually. In other words, we use each extraction approach to collect the identifier-definiens pairs from all documents. We then use the identifier-definiens pairs as features to cluster documents and label the obtained clusters with suitable categories from well-known topic categorizations, such as the Mathematics Subject Classification provided by the American Mathematical Society. For details please refer to [13].

4 Results

We re-evaluated the pattern matching approach (PM) of Pagel et al. (cf. Sect. 2.2) [9] and the statistical feature analysis (ST) of Schubotz et al. (cf. Sect. 2.1) [13] to obtain the '_before' results shown in Table 2. The '_before' suffix indicates the use of the MLP pipeline as presented in [13], i.e. before making the improvements described in Sect. 3.1.

While PM has not been evaluated using this gold standard before, we already evaluated ST using the same gold standard in the past [13]. However, in the previous evaluation of ST, we manually judged the relevance of the extracted identifier-definiens pairs and used a different policy to judge true positives than employed by the automated evaluation procedure in our framework. As described in Sect. 3.1, our framework ignores cases in which more than one correct definition is retrieved for calculating the performance metrics. Opposed to that, we counted each correctly extracted definition as a true positive in our previous

Table 2. Performance comparison of the pattern matching (PM), statistical feature analysis (ST), and machine learning (ML) methods. See Sect. 4 for details.

Baseline	tp	fp	Prec%	Rec%	F_1%
ST_before	69	351	16.43	22.26	18.90
ST_after	70	322	17.85	22.58	19.94
PM_before	56	290	16.18	18.06	17.07
PM_after	56	199	22.00	18.06	19.80
Without namespaces					
PM_after ∪ ST_after	77	551	12.26	24.84	16.42
ML_ST_PM	60	181	24.90	19.35	21.78
ML_no_DG	79	171	31.60	25.48	28.21
ML_no_PM	86	111	43.65	27.74	33.92
ML_full	87	92	48.60	28.06	35.58
With namespaces					
ST_after + NS	75	340	18.07	24.19	20.69
ML_full + NS	93	118	43.66	30.00	35.56

work [13]. In our previous manual evaluation, we also counted several synonymous identifier-definiens relations as true positives. For example, we manually matched *'eigenfrequencies'* to the gold standard entry *'natural frequency'* and the wikidata item Q946764, which does not (yet) have an alias for *'eigenfrequencies'*. Realizing that both terms are synonyms is trivial for human assessors, but beyond the capabilities of our current automated evaluation framework. Since these limitations of the framework apply to all evaluated methods, the relative performance scores of the methods should be unaffected.

Due to the different evaluation policies, the measured performance of ST decreased from $p \approx .21$, $r \approx .28$, $F_1 \approx .24$ in our previous manual evaluation [13] to $p \approx .16$, $r \approx .22$, $F_1 \approx .19$ in the current automated evaluation. The absolute number of true positives (tp) declined by 18 from 88 to 70. Identifiers for which more than one definiens was found account for 9 fewer tp and the inability of the evaluation framework to resolve synonymous identifier-definiens pairs accounts for 8 fewer tp.

Our improvements to the MLP pipeline slightly increased the precision achieved by the pattern matching approach (PM) and the statistical feature analysis (ST). Refer to the results with the suffix '_after' in Table 2.

The PM and the ST approach extracted 48 identical and 29 different definientia, which means that combining both results will achieve a higher recall. The simple union (see Table 2) yields a higher recall, but the precision drops disproportionately. To create a combined classifier of both approaches that achieves better precision, we must rank nominal features, e.g., pattern matches, together with ordinal features, e.g., term frequency.

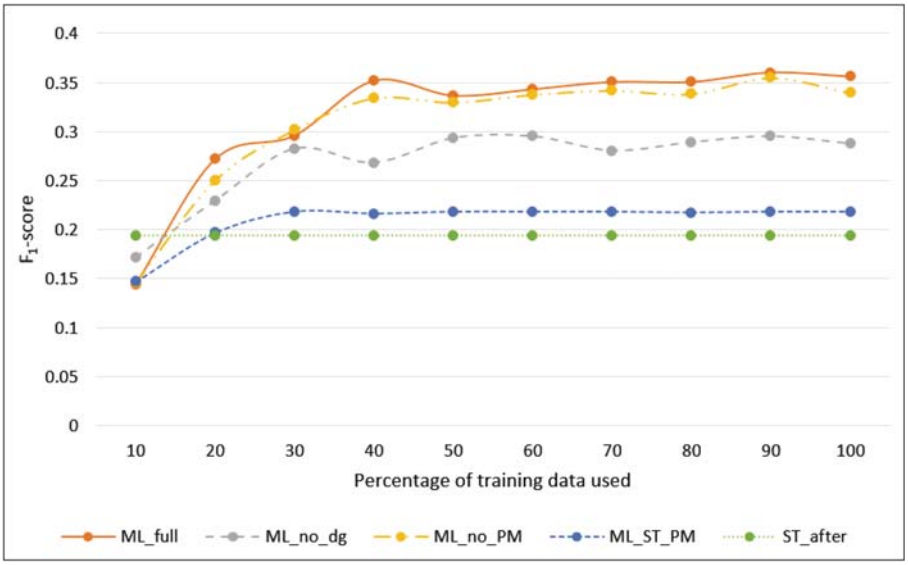

Fig. 2. Extraction performance (F_1-score) for different sizes of the training dataset.

Our new machine learning method (ML) extracted 87 definientia correctly with 92 false positives, resulting in $p = .4860$, $r = .2806$, $F_1 = .3558$. In addition, we trained the following classifiers using subsets of the features: ML_ST_PM indicates the combination of only statistical features (27–29 in Table 1) with the pattern matcher (1–10 in Table 1). This approach yielded $F_1 = .2178$, which is comparable to the performance of the approaches in the previous manual investigation. Adding the string features (11–21 in Table 1), but leaving out the computationally intensive dependency graph features increased the performance to $F_1 = .2821$ (see ML_no_DG in Table 2). Training with all features except for the patterns 1–10 in Table 1 yielded a good performance of $F_1 = .3276$ (see ML_no_PM in Table 2). When comparing this result to the full classifier (ML_full), which achieves $F_1 = .3558$, shows that the improvement achieved by including the pattern-based features is small. In other words, creating a well-performing classifier without the language-dependent pattern features is possible.

Figure 2 plots the F_1-scores for different sizes of the training dataset. The different models seem to converge to a fixed threshold with a decreasing gradient. This indicates that the classifier can be trained well despite the limited number of positive instances in the training data and the complexity of the features. Additionally, all ML classifiers outperform the best individual extraction method (ST or PM) if more than 20% of the training data is used.

Investigating the effects of considering namespaces for the identifier definition extraction, we could confirm the finding in [13] that namespaces improve both precision and recall for the statistical approach. However, the gain for the

machine learning classifier is minimal, since the increase in recall is traded for precision. We could create 216 namespaces with a purity of more than 0.8 while retaining the total number of definientia. In our previous evaluation, we could form 169 namespaces with an average purity of 0.8 [13]. Purity is a cluster quality metric computed using the Wikipedia category information (see [13] for details). The results indicate that the new machine learning approach yields fewer false positives in forming namespaces.

Performing the classification task, which considered 5‘400‘702 identifier-definiens pairs, required approx. 3.5 h on a compute server with 80 2.6 Ghz Xeon cores. This runtime included the time-consuming calculation of the dependency graphs for the sentences that contain identifiers.

5 Conclusion and Outlook

This paper provides an openly available, automated framework to evaluate the extraction of identifier definitions from mathematical datasets and presents a new statistical machine learning approach to perform this task. The framework extends and improves the pipeline for mathematical language processing using Wikitext input that we presented in [13]. The evaluation framework uses parts of the dataset of the NTCIR-11 Math Retrieval Wikipedia task [12] and a manually created gold standard that contains definiens for all the 310 mathematical identifiers in the dataset.

Using the newly developed evaluation framework, we compared existing approaches for identifier definition extraction. The previously best-performing approach achieved a precision $p \approx .18$, recall $r \approx .23$, and $F_1 \approx .20$. Our newly developed machine learning approach significantly increased the extraction performance to $p \approx .49,\ r \approx .28,\ F_1 \approx .36$.

Despite the improvements of the preprocessing pipeline, we could see that the statistical feature analysis (ST) clearly outperforms the pattern matching (PM) approach. In addition, our machine learning (ML) approach significantly reduces the number of false positives, even without relying on language-dependent patterns. At its core, the developed machine learning approach relies heavily on features developed for the statistical feature analysis, which the new approach combines with a better method for tuning the extraction parameters.

Even the newly developed machine learning approach achieves a relatively low performance when compared to approaches for other information extraction tasks. The results indicate that a large potential for future improvements of identifier extraction approaches remains. While our newly proposed method significantly reduced the number false positives, new strategies are needed to further improve the number of true positives.

For future research, we advise against using our gold standard dataset for training purposes, since the gold standard contains identifier definitions that cannot be identified by any of the features we examined in this paper. This limitation lies rooted in the creation history of the gold standard, which involved tedious logical inference and the consultation of tertiary sources by the domain experts

who created the gold standard. The experts deduced information, incorporated world knowledge and exhibited a higher fault tolerance than can be expected from automated systems. For instance, extracting the identifiers η, Q_1, Q_2 from the malformed input formula $\eta = \frac{workdone}{heatabsorbed} = \frac{Q1-Q2}{Q2}$, as the domains experts did when creating the gold standard, is likely a suboptimal training for future extraction approaches.

Future research should focus on increasing recall, because current methods exclusively find exact definitions for approx. 1/3 of all identifiers. New approaches may further improve the task at hand, e.g., by using logical deduction [11]. Likewise, the use of multilingual features of Wikipedia, e.g., by applying approaches like multilingual semantic role labeling [1], can prove beneficial. In the medium term, the semantic granularity of the corresponding Wikidata concepts should be considered [3]. Lastly, the proposed approach could also be applied in other domains. One use case would be to identify biased media coverage by analyzing the relations between words in image captions and texts of news articles (cf. [4,5]). Another idea would be to adapt the approach for resolution of abbreviations and synonyms [6].

In conclusion, by evaluating and combining existing approaches we achieved a significant performance improvement in extracting mathematical identifier definitions. We also identified several promising directions for future research to further improve the extraction performance.

References

1. Akbik, A., Guan, X., Li, Y.: Multilingual aliasing for auto-generating proposition banks. In: Calzolari, N., Matsumoto, Y., Prasad, R. (eds.) 6th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers (COLING 2016), December 11–16, 2016, Osaka, Japan, pp. 3466–3474. ACL (2016)
2. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines. In: ACM Transactions on Intelligent Systems and Technology (TIST 2011) vol. 2, no. 3, p. 27 (2011)
3. Corneli, J., Schubotz, M.: math.wikipedia.org: a vision for a collaborative semi-formal, language independent math(s) encyclopedia. In: Conference on Artificial Intelligence and Theorem Proving (AITP 2017) (2017)
4. Hamborg, F., Meuschke, N., Gipp, B.: Matrix-based news aggregation: exploring different news perspectives. In: Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL) (2017)
5. Hamborg, F., et al.: Identification and analysis of media bias in news articles. In: Gaede, M., Trkulja, V., Petra, V. (eds.) Proceedings of the 15th International Symposium of Information Science, Berlin, pp. 224–236, March 2017
6. Henriksson, A., et al.: Synonym extraction and abbreviation expansion with ensembles of semantic spaces. J. Biomed. Semant. **5**(1), 6 (2014)
7. Kristianto, G.Y., Topic, G., Aizawa, A.: Extracting textual descriptions of mathematical expressions in scientific papers. In: D-Lib Magazine (D-Lib 2014), vol. 20, no. 11, p. 9 (2014)

8. Manning, C.D., et al.: The stanford CoreNLP natural language processing toolkit. In: Association for Computational Linguistics (ACL) System Demonstrations (ACL 2014), pp. 55–60 (2014)
9. Pagel, R., Schubotz, M.: Mathematical language processing project. In: England, M., et al. (eds.) Joint Proceedings of the MathUI, OpenMath and ThEdu Workshops and Work in Progress Track at CICM Co-located with Conferences on Intelligent Computer Mathematics (CICM 2014), Coimbra, Portugal, July 7–11, 2014, vol. 1186. CEUR Workshop Proceedings. CEUR-WS.org (2014)
10. Schubotz, M.: Augmenting Mathematical Formulae for More Effective Querying & Efficient Presentation. Epubli Verlag, Berlin (2017). ISBN: 9783745062083
11. Schubotz, M., Veenhuis, D., Cohl, H.S.: Getting the units right. In: Kohlhase, A., et al. (ed.) Joint Proceedings of the FM4M, MathUI, and ThEdu Workshops, Doctoral Program, and Work in Progress at the Conference on Intelligent Computer Mathematics 2016 Co-located with the 9th Conference on Intelligent Computer Mathematics (CICM 2016), Bialystok, Poland, July 25–29, 2016, Vol. 1785. CEUR Workshop Proceedings. CEUR-WS.org (2016)
12. Schubotz, M., et al.: Challenges of mathematical information retrieval in the NTCIR-11 math Wikipedia task. In: Baeza-Yates, R.A., et al. (eds.) Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015), pp. 951–954. ACM, Santiago (2015). ISBN: 978-1-4503-3621-5
13. Schubotz, M., et al.: Semantification of identifiers in mathematics for better math information retrieval. In: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016), pp. 135–144. ACM, Pisa (2016). ISBN: 978-1-4503-4069-4