# Large-Scale Experiments for Mathematical Document Classification

Simon Barthel[1], Sascha Tönnies[2], and Wolf-Tilo Balke[1,2]

[1] IFIS TU Braunschweig, Mühlenpfordstraße 23, 38106 Braunschweig, Germany
[2] L3S Research Center, Appelstraße 9a, 30167 Hannover, Germany
{barthel,balke}@ifis.cs.tu-bs.de, toennies@l3s.de

**Abstract.** The ever increasing amount of digitally available information is curse and blessing at the same time. On the one hand, users have increasingly large amounts of information at their fingertips. On the other hand, the assessment and refinement of web search results becomes more and more tiresome and difficult for non-experts in a domain. Therefore, established digital libraries offer specialized collections with a certain degree of quality. This quality can largely be attributed to the great effort invested into semantic enrichment of the provided documents e.g. by annotating their documents with respect to a domain-specific taxonomy. This process is still done manually in many domains, e.g. chemistry (CAS), medicine (MeSH), or mathematics (MSC). But due to the growing amount of data, this manual task gets more and more time consuming and expensive. The only solution for this problem seems to employ automated classification algorithms, but from evaluations done in previous research, conclusions to a real world scenario are difficult to make. We therefore conducted a large scale feasibility study on a real world data set from one of the biggest mathematical digital libraries, i.e. Zentralblatt MATH, with special focus on its practical applicability.

**Keywords:** Text Classification, Mathematical Documents, Experiments.

## 1 Introduction

Digital libraries offer specialized document collections for many scientific domains combined with user interfaces and retrieval functionalities that are customized to the respective domain. The retrieval facilities of a digital library generally rely on two different types of metadata: classic bibliographic metadata (such as author, title, year of publication, and publisher) and semantic metadata (describing the content of a document). Today, especially semantic metadata is essential for the development of innovative methods for document retrieval, for instance explorative search, contextualization, personalization, and the creation of synergies between various digital libraries [1], [2].

There is a variety of possible semantic metadata annotations ranging from free tags over author keywords to terms from domain-specific taxonomies. In contrast to free tags (e.g., extracted from the Social Web) or keywords provided by authors, taxonomical metadata offers exceptional quality: it features a controlled vocabulary that is

S.R. Urs, J.-C. Na, and G. Buchanan (Eds.): ICADL 2013, LNCS 8279, pp. 83–92, 2013.

well understood by users in a domain, and is maintained and regularly updated by domain experts. While this quality is essential for libraries as controlled quality information providers, it comes at a price: not only is the maintenance of a taxonomy itself expensive, but the annotations of individual documents in a collection are too, since they usually are performed manually by domain experts.

Consider for instance the field of mathematics. There are two important digital libraries in this domain, Mathematical Reviews[1] in North America and Zentralblatt MATH[2] in Europe. Both provide abstracts and reviews covering the entire field of mathematics, e.g., for Zentralblatt MATH about 2,300 journals and serials world-wide, as well as books and conference proceedings. To offer the aforementioned assets for users (in contrast to general purpose web search engines), all provided documents are indeed manually annotated according to the Mathematics Subject Classification (MSC) taxonomy maintained by both organizations.

But is this effort in manual indexing sustainable? Currently, an exponential growth in the number of publications can be observed across all fields of science. Given the limited financial resources of libraries, this problem obviously cannot be handled by employing more domain experts for indexing tasks. Thus, the only solution is to provide more efficient (semi-)automatic indexing methods effectively reducing the manual indexing work while maintaining the resulting metadata quality. Fortunately, for the task of indexing the field of machine learning offers a multitude of automated text categorization methods which have already been applied to many text corpora with great success, see e.g. [3-7]. Indeed, it seems intuitive that text classification is key to being able to cope with the existing information flood. For instance, using MSC the approach in [8] achieved very good results with $F_1$ measures of 0.89. But looking closer at the experimental settings the experiments were performed in, they are hardly applicable to the workflow of a digital library because hard constraints to the incoming data were performed and full-texts were available. Moreover, it is also unclear what an $F_1$ measure of 0.89 really means in terms of quality for the applicability.

To provide additional perspective on these evaluations, we conducted a large-scale study on the feasibility of using different classification techniques for automatic indexing in practice. Motivated by the promising results in [8] we focused on mathematical documents according to the MSC taxonomy.

- Our document corpus taken from Zentralblatt MATH includes more than three million entries manually annotated with MSC classes and was chosen to ensure the applicability of our results to a real world scenario.
- We employed state-of-the-art text classification algorithms like support vector machines (SVM), Naïve Bayes classifiers, and C4.5 decision trees that were even specifically adapted to the domain, e.g. taking mathematical formulae into account for boosting classification performance.
- For the evaluation we do not only look at traditional $F_1$ measures or microaveraged break-even points, but also look behind these measures and evaluate what these numbers actually mean for practical application.

[1] http://www.ams.org/mr-database
[2] http://www.zentralblatt-math.org/zbmath/

Our contribution thus is threefold. First, we conduct a large-scale evaluation of text classifiers in a realistic taxonomy-based setting. We then provide an in-depth analysis of classification problems, and draw conclusions for today's digital libraries.

The rest of the paper is organized as follows: In section 2 we review related work presenting results for automated indexing based on bibliographic metadata. Section 3 introduces our experiments on the practical Zentralblatt MATH corpus. Section 4 addresses the applicability of automatic classifiers in a real-world scenario and explains the results in depth by additional experiments. Finally, section 5 closes with our conclusions for the practical application of text classification in digital libraries.

## 2 Related Work

Semi-automatic techniques for annotation of semantic metadata are covered by the field of tag recommendation. Tag recommendation are mainly based on two main approaches: co-occurences between tags [9], [10] and content-based tag recommendation [11]. A collaborative method for tag recommendation is presented in [10]. The authors used tag co-occurrences and tag aggregation methods to recommend Flickr tags. With this approach users can provide one or two initial tags, whereupon they receive recommendations for additional tags. On the other hand, content-based approaches as presented in [11] usually map the items to be tagged into a vector space, where either typical difference metrics between item vector and tag reference vectors are applied or a tag is recommended with respect to the output of a classifier which has been trained for that tag.

In [12] the author conducted experiments using bibliographic metadata of the Library of Congress and ranked Library of Congress class numbers for a given document. This was done by building reference vectors for each considered class and by ranking the classes for a new document according to the product of the document vector with each class reference vector. Reasonable results could be achieved by using the subject headings of a document. Here, the average rank for a relevant class was 1.36. However, as subject headings are normally already linked with a recommendation of a Library of Congress class number, this result is not particularly surprising. Furthermore, Library of Congress subject headings also belong to semantic metadata and have to be annotated manually. Without subject headings used in the ranking progress, the average rank of relevant classes raised up to 50.53. In [13] the authors performed a classification task for the ACM Computing Classification System. The eleven ACM categories could be predicted with a microaveraged $F_1$ measure of 60.81.

For the MSC the authors of [8] achieved a very good $F_1$ measure of 0.89. However, their experiments were applied in a setting which differs greatly from real world scenarios such as digital libraries. For instance, full texts were used, which are not generally available in the workflow of a digital library. Additionally, the whole corpus was filtered to only those documents with no secondary classes annotated. As the majority of documents have one main class and several secondary classes annotated, this constraint introduces a strong bias into the evaluation. Consequently, only 20 of the 63 top-level classes could be evaluated on the remaining corpus of 4,127 documents.

# 3 Experiments

In section 2 we presented related work with varying results for different text categorization tasks. From those results, conclusions for practical use can hardly be drawn. We therefore conducted a text classification task based on the data that is actually available in the workflow of a digital library and analyzed the results with respect to the practical usage. For our experiments we used a document corpus containing 2,051,392 documents covering the years from 1931 to 2013 delivered by the Zentralblatt MATH. The corpus contains titles, abstracts, authors, journals, and an unordered list of author keywords. To maintain applicability for a real world setting, we only applied a realistic and affordable data cleaning method. This included the application of a language guesser confining the corpus to English texts and the elimination of documents with missing abstracts or abstracts that only consist of one sentence. Statistical information about the corpus, including the distribution of documents over time, the distribution of categories over documents and the changes of the distribution of categories over time can be found at figshare[3].

As a ground truth, the documents are annotated manually with MSC classes by the technical editors of the Zentralblatt MATH. The MSC is a taxonomy used by many mathematic journals for semantic enrichment of their data. It is maintained and regularly updated by the two most important digital libraries in the area of mathematics, the Zentralblatt MATH and Mathematical Reviews. The last version of the MSC taxonomy (MSC2010) has three levels, containing 63 classes on the first level, 530 on the second and on 5202 on the third. An MSC class (e.g. 05D10 for Ramsey theory) is organized as follows: The first two digits determine the top level category (05 for combinatorics), followed by a character indicating the second level (D for extremal combinatorics) and two digits for the third level (10 for Ramsey theory). Each document has exactly one main class assigned and may have an arbitrary number of secondary categories assigned.

## 3.1 Text Classification

For document classification we evaluated Support Vector Machines, C4.5 decision trees as well as Naïve Bayes classifiers. For the document indexing we used a tokenizer that was adapted to our corpus and can distinguish formulae, references and plain text within the abstracts. We also analyzed the benefit of several standard text preparation, term reweighing and term selection methods like stemming, TF-IDF, latent semantic indexing, Euclidian normalization, and local term selection according to a feature scoring metric. Detailed explanations of these techniques are not in the scope of this paper, we refer to [14] for further details.

As proposed in [15], we focused on the three levels of the MSC individually, applying the same algorithms to each level in a hierarchical fashion with only minor adaptions.

For the training and evaluation of the second MSC level the corpus was projected to those documents that have the respective top class annotated. The same applies for level three. The classification error therefore sums up when a complete classification for all three levels has to be performed.

[3] http://dx.doi.org/10.6084/m9.figshare.796397

### 3.2 Formula Classification

In our mathematical corpus, formulae are the most important domain-specific feature. Since the naïve approach of treating formulae as simple text tokens had negligible impact on classification, we used a more sophisticated way to utilize formulae for classification.

In this experiment we used the formula search index described in [16]. In contrast to previous research we did not use this index to perform a search [17] [18], but to map formulae into a vector space. When a search query is performed on the index, multiple index nodes are visited. An index node can represent a complete formula, a sub formula, a terminal symbol or an abstract formula with no terminal symbols. Therefore, the nodes visited during the query evaluation yield a good semantic representation of a formula. When considering each index node as a dimension in a vector space, a formula can be mapped into that vector space by setting the coordinate of each index node to 1 if it was visited during the search query and to 0 otherwise.

One problem with this method is, that many formulae contained in abstracts are trivial (like e.g. $\lambda$) or not very complex and are therefore not adequate for formula classification. Therefore, the formula vector is merged with the vector obtained from a traditional bag-of-words approach on the plain text. This avoids the problem of finding an appropriate "complexity threshold" for formulae to consider them in formula classification and also increases the overall classification quality.

As many abstracts do not even contain a single formula, to verify the effectiveness of this approach we created a smaller collection of documents featuring formulae in their abstracts. The performance gain shown in Table 1 can therefore not be applied directly to the global performance but only serves as an argument for the general plausibility of this approach.

**Table 1.** Table of three best performing categories for formula classification in terms of $F_1$ measures.

| Top-level Category | only text | only formulae | combination |
|---|---|---|---|
| 34 (Ordinary differential equations) | 0.623 | 0.613 | 0.667 |
| 35 (Partial differential equations) | 0.674 | 0.609 | **0.734** |
| 11 (Number theory) | 0.664 | 0.531 | 0.667 |

We can see that even the relatively simple formulae mentioned in abstracts can be used to perform a classification based on formulae exclusively, or in a combined manner to improve the classification quality beyond that of pure text classification.

### 3.3 Results

In this section we present the results achieved by the machine learning algorithms mentioned above. For the training of the classifiers ultimately employed, we used titles concatenated with abstracts with no stemming applied. Formulae were ignored for bag-of-words indexing and were instead processed by the formula indexing

method introduced in section 3.2. We then applied TF-IDF reweighting and Euclidean normalization on the document vectors and used these vectors to train Support Vector Machines. In the training process we used only those documents as positive examples which were tagged with the respective MSC class as main class. To prevent overfitting, the amount of positive and negative training examples were balanced to an equal number, where the negative examples were drawn equally distributed over all negative categories. By means of a tuning set, the threshold of the resulting classifiers were afterwards adapted to return an optimized $F_1$ measure.

The performance of the resulting classification system in terms of microaveraged $F_1$ measures are summarized in Table 2.

**Table 2.** Performance of the classification system in terms of microaveraged $F_1$ measures

| | Top Level | Second Level | Third Level |
|---|---|---|---|
| Top 10% | 0.815 | 0.898 | 0.919 |
| All | 0.673 | 0.665 | 0.538 |

## 4 Classification Performance in a Real World Workflow of a Digital Library

Averaged results as shown at the end of the last chapter are commonly used to show the significance of the result of an experiment. They are often combined with microaveraged $F_1$ measures of the top *k* best performing categories or examples for exceedingly favorable results. While for the evaluation of novel approaches for machine learning this manner of representation might be valid, for practical use cases it tends to be misleading. Let us therefore focus on the worst performing classifiers shown in Table 3.

**Table 3.** Performance of 5 worst categories in terms of the $F_1$ measure

| Top-level Category | $F_1$ measure |
|---|---|
| 19 ($K$-theory) | 0.182 |
| 12 (Field theory and polynomials) | 0.230 |
| 31 (Potential theory) | 0.240 |
| 08 (General algebraic systems) | 0.241 |
| 43 (Abstract harmonic analysis) | 0.242 |

Of course, in digital libraries, the annotations for each document have to be correct and complete to cater for effective subsequent searches. Since there are classifiers included in the classification system with performances as shown in Table 3, a fully automatic indexing is out of the question. Still, there are two possible ways to use the classification system.

One approach is to **tweak the precision** of each classifier to an expected precision of e.g. 0.95 by shifting the threshold for positive classifications appropriately. Consequently,

classifiers as shown in Table 3 would then return poor recall values, meaning that these categories will hardly ever be annotated. Knowing that only 2.79% of all documents are annotated with more than three top level categories, we may consider a document to be annotated correctly if at least three high precision classifiers were triggered. After implementing and generalizing this approach for a hierarchical setting, we found that only about 1% of all documents could be automatically annotated with high precision MSC tags.

In the other approach the classifiers' thresholds are adjusted to **provide a good recall** of 0.95. These classifiers can then be used as tag recommendation service within the indexing workflow. In this setting, weak classifiers will be triggered far too often, invalidating the recommendation lists. A technical editor thus would have to work with recommendation lists containing a lot of irrelevant categories.

Both scenarios do not really contribute to the reduction of manual work for technical editors. In both cases the reason is not connected to the average performance of the classification system or the performance of the best classifiers but solely depends on the quality of the worst performing classifiers. For practical use, we therefore claim that the main focus must lie on the worst performing classifiers.

## 5 Confusion of Different Categories

To examine the extent of the problem we conducted further experiments with our MSC classifiers. In particular, we analyzed the confusion between all top level categories to determine whether a high degree of confusion between various categories might be the source of the problem. We defined a confusion matrix as

$$conf(c_1, c_2) = \frac{|\{\, d \mid Cons(d, c_1, c_2) \wedge Miss(d, c_1, c_2)\}|}{|\{d \mid Cons(d, c_1, c_2)\}|}$$

where $Cons$ is used to specify the documents considered for the calculation of the confusion between $c_1$ and $c_2$. In our case all Documents are considered that have $c_1$ or $c_2$ annotated but not both $c_1$ and $c_2$. And $Miss$ indicates a miss-classification of $d$ with respect to $c_1$ and $c_2$, meaning that $c_1$ or $c_2$ are missing or $c_1$ or $c_2$ have been mistakenly annotated. The resulting confusion matrix can be seen in Fig. 1.

We can see that there are quite a lot of categories with a confusion of 50% and higher. An explanation for this fact is that the information based on bag-of-words and formulae contained in title and abstract are not sufficient to separate the categories.

As a last experiment we analyzed the intra-class text similarity with Apache Lucene, one of the most popular full text indexing libraries. The Lucene full text index is used by many digital libraries as retrieval system and is therefore an important baseline to compare with. In the experiment we built a Lucene index with the given corpus and used every document for a "More like this" query. If textual content were a discriminating feature, documents in the result list should at least show the same main category as the query document. The result of this experiment can be seen in Fig. 2. Considering the top $k$ documents in the result list, the graph shows the percentage of documents which are annotated with the query documents main category.

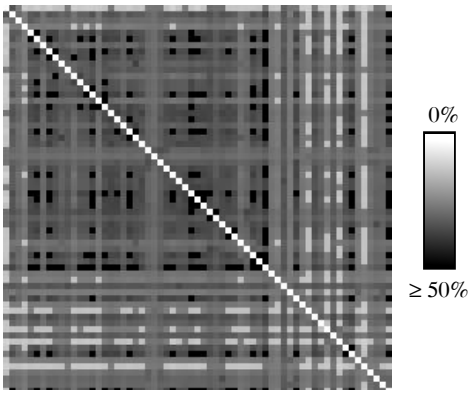


**Fig. 1.** Confusion between the top level classes of the MSC

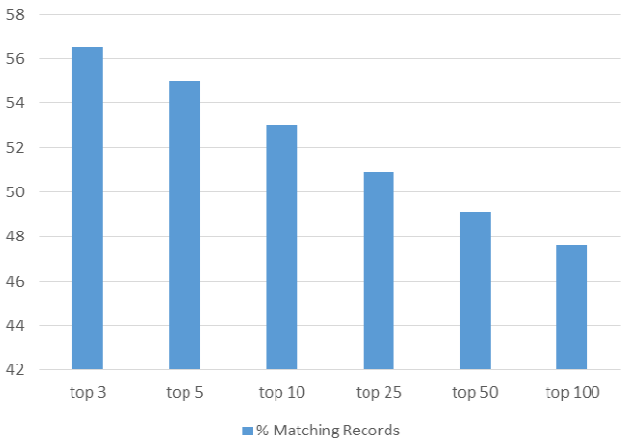


**Fig. 2.** Average matching of categories with Lucene "More like this" Queries

Fig. 2 shows two interesting points. On one hand, it shows that even for the top ranked documents according to the Lucene text similarity there is no confidence that the returned documents belong to the same category. On the other hand, the result show that even for rank 100 the confidence of 48% for receiving some relevant document is not significantly lower than on the ranks 1-3.

## 6 Conclusion

In this paper we studied the level of quality that state-of-the-art text categorization techniques can achieve for automated annotation of semantic metadata. With respect to microaveraged $F_1$ measures we can say that our results are comparable to current work in mathematical text classification. But we found out that for practical

applications microaveraged $F_1$ measure may be misleading. As we have argued, for digital libraries the performance of a classification system is mainly dependent on the performance of the worst classifiers. This fact became obvious when we applied our MSC classifiers either for fully automatic indexing by tweaking the precision or for tag recommendation by boosting the recall. Both approaches resulted in a minimal reduction of manual work for technical editors.

Assuming that for every relevant classification task in digital libraries there are always bad performing classifiers involved, we can conclude that automated text classification alone cannot be used to reduce the manual work for indexing tasks in digital libraries. In our scenario this fact was true even though a notable amount of over two million documents were available for training and only the first level of the MSC were considered. In future work we plan to find out why this is the case and what can be done to solve the problem. First, we want to evaluate the inter-rater reliability for our classification task to see if humans can achieve a significantly higher categorization performance than automated classifiers, especially for those classes with high confusion. If this is the case, the question remains why humans are able to classify documents accurately while machines can't. Otherwise, if even human ratings are not consistent, it is not surprising that machines cannot perform significantly better. In this case it is also questionable if a strong annotation of taxonomy terms is sensible and if there is no need for alternative ways to enrich documents for the users of digital libraries.

We also want to extend our experiments to full text documents, which might – regarding formula classification – to some degree increase classification quality. In this case this means that digital libraries will need access to full text even if this full texts are not delivered as plain text but e.g. in the form of a feature vector. Otherwise, digital libraries can only restrict their scope and process less sources, lower the demands on delivered quality, or rely on different types of semantic metadata like free tags, author networks, or citation networks. Future work will show if using this kind of semantic metadata an adequate quality in retrieval and customization can be achieved.

**Acknowledgements.** Special thanks are extended to the German National Science Foundation (DFG) for supporting this study. We also wish to acknowledge the help provided by Corneliu-Claudiu Prodescu and Prof. Dr. Michael Kohlhase by applying the MathWebSearch index on the Zentralblatt MATH corpus. Moreover, we would like to express our appreciation to the Zentralblatt MATH for the provision of the corpus used in this study.

## References

1. Chirita, P.A., Nejdl, W., Paiu, R., Kohlschütter, C.: Using ODP metadata to personalize search. In: SIGIR 2005, Salvador, Brazil (2005)
2. Mirizzi, R., Ragone, A., Di Noia, T., Di Sciascio, E.: Semantic wonder cloud: exploratory search in DBpedia. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 138–149. Springer, Heidelberg (2010)

3. Homoceanu, S., Dechand, S., Balke, W.-T.: Review Driven Customer Segmentation for Improved E-Shopping Experience. ACM Web Science (2011)
4. Shen, D., Ruvini, J.-D., Sarwar, B.: Large-scale item categorization for e-commerce. In: CIKM 2012, Maui, Hawaii, USA (2012)
5. Cheng, W., Kasneci, G., Graepel, T., Stern, D., Herbrich, R.: Automated feature generation from structured knowledge. In: CIKM 2011, Glasgow, Scotland, UK (2011)
6. Dumais, S., Platt, J., Heckerman, D., Sahami, M.: Inductive learning algorithms and representations for text categorization. In: CIKM 1998, Bethesda, Maryland, USA (1998)
7. Cohen, W.W., Singer, Y.: Context-sensitive learning methods for text categorization. ACM Trans. Inf. Syst., pp. 141–173 (April 1999)
8. Řehůřek, R., Sojka, P.: Automated Classification and Categorization of Mathematical Knowledge. In: CICM 2008, pp. 543–557 (2008)
9. Song, Y., Zhuang, Z., Li, H., Zhao, Q., Li, J., Lee, W.-C., Giles, C.L.: Real-time automatic tag recommendation. In: SIGIR 2008 (2008)
10. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: WWW 2008, Beijing, China (2008)
11. Byde, A., Wan, H., Cayzer, S.: Personalized Tag Recommendations via Tagging and Content-based Similarity Metrics. In: ICWSM 2007 (2007)
12. Larson, R.R.: Experiments in automatic library of congress classification. In: JASIS 1992, pp. 130–148 (1992)
13. Zhang, B., Gonçalves, M.A., Fan, W., Chen, Y., Fox, E.A., Calado, P., Cristo, M.: Combining structural and citation-based evidence for text classification. In: ICKM 2004 (2004)
14. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys, 1–47 (2002)
15. Sun, A., Lim, E.-P.: Hierarchical text classification and evaluation. In: ICDM 2001 (2001)
16. Prodescu, C.C., Kohlhase, M.: Mathwebsearch 0.5-open formula search engine. In: Wissens-und Erfahrungsmanagement Conference Proceedings (2011)
17. Kohlhase, M., Matican, B.A., Prodescu, C.-C.: MathWebSearch 0.5: scaling an open formula search engine. In: CICM 2012, pp. 342–357 (2012)
18. Iancu, M., Kohlhase, M., Rabe, F., Urban, J.: The Mizar Mathematical Library in OMDoc: Translation and Applications. Journal of Automated Reasoning, 191–202 (2013)