Chapter 14

Can a Hammer Categorize Highly Technical Articles?

Samuel Hansen
University of Michigan

When everything looks like a nail...

I was sure I had the most brilliant research project idea for my course in Digital Scholarship techniques. I would use the Mathematical Subject Classification (MSC) values assigned to the publications in MathSciNet[1] to create a temporal citation network which would allow me to visualize how new mathematical subfields were created and perhaps even predict them while they were still in their infancy. I thought it would be an easy enough project. I already knew how to analyze network data and the data I needed already existed, I just had to get my hands on it. I even sold a couple of my fellow coursemates on the idea and they agreed to work with me. Of course nothing is as easy as that, and numerous requests for data went without response. Even after I reached out to personal contacts at MathSciNet, we came to understand we would not be getting the MSC data the entire project relied upon. Not that we were going to let a little setback like not having the necessary data stop us.

After all, this was early 2018 and there had already been years of stories about how artificial intelligence, machine learning in particular, was going to revolutionize every aspect of our world (Kelly 2014; Clark 2015; Parloff 2016; Sangwani 2017; Tank 2017). All the coverage made it seem like AI was not only a tool with as many applications as a hammer, but that it also magically turned all problems into nails. While none of us were AI experts, we knew that machine learning was supposed to be good at classification and categorization. The promise seemed to be that if you had stacks of data, a machine learning algorithm could dive in, find the needles, and arrange them into neatly divided piles of similar sharpness and length. Not only that, but there were prebuilt tools that made it so almost anyone could do it. For a group of people whose project was on

[1] See https://mathscinet.ams.org/

life support because we could not get the categorization data we needed, machine learning began to look like our only potential savior. So, machine learning is what we used.

I will not go too deep into the actual process, but I will give a brief outline of the techniques we employed. Machine-learning-based categorization needs data to classify, which in our case were mathematics publications. While this can be done with titles and abstracts we wanted to provide the machine with as much data as we could, so we decided to work with full-text articles. Since we were at the University of Wisconsin at the time, we were able to connect with the team behind GeoDeepDive[2] who have agreements with many publishers to provide the full text of articles for text and data mining research ("GeoDeepDive: Project Overview" n.d.). GeoDeepDive provided us with the full text of 22,397 mathematics articles which we used as our corpus. In order to classify these articles, which were already pre-processed by GeoDeepDive with CoreNLP,[3] we first used the Python package Gensim[4] to process the articles into a Python-friendly format and to remove stopwords. Then we randomly sampled ⅓ of the corpus to create a topic model using the MALLET[5] topic modeling tool. Finally, we applied the model to the remaining articles in our corpus. We then coded the words within the generated topics to subfields within mathematics and used those codes to assign articles a subfield category. In order to make sure our results were not just a one-off, we repeated this process multiple times and checked for variance in the results. There was none, the results were uniformly poor.

That might not be entirely fair. There were interesting aspects to the results of the topic modeling, but when it came to categorization they were useless. Of the subfield codes assigned to articles, only two were ever the dominant result for any given article: Graph Theory and Undefined, which does not really tell the whole story as Undefined was the run-away winner in the article classification race with more than 70% of articles classified as Undefined in each run, including one for which it hit 95%. The topics generated by MALLET were often plagued by gibberish caused by equations in the mathematics articles and there was at least one topic in each run that was filled with the names of months and locations. Add how the technical language of mathematics is filled with words that have non-technical definitions (for example, map or space), or words which have their own subfield-specific meanings (such as homomorphism or degree), both of which frustrate attempts to code a subfield. These issues help make it clear why so many articles ended up as "Undefined." Even for the one subfield which had a unique enough vocabulary for our topic model to partially be able to identify, Graph Theory, the results were marginally positive at best. We were able to obtain Mathematical Subject Classification (MSC) values for around 10% of our corpus. When we compared the articles we categorized as Graph Theory to the articles which had been assigned the MSC value for Graph Theory (05Cxx), we found we had a textbook recall-versus-precision problem. We could either correctly categorize nearly all of the Graph Theory articles with a very high rate of false positives (high recall and low precision) or we could almost never incorrectly categorize an article as Graph Theory, but miss over 30% that we should have categorized as Graph Theory (high precision and low recall).

Needless to say, we were not able to create the temporal subfield network I had imagined. While we could reasonably claim that we learned very interesting things about the language of mathematics and its subfields, we could not claim we even came close to automatically categorizing mathematics articles. When we had to report back on our work at the end of the course,

[2] See `https://geodeepdive.org/`
[3] See `https://stanfordnlp.github.io/CoreNLP/`
[4] See `https://radimrehurek.com/gensim/`
[5] See `http://mallet.cs.umass.edu/topics.php`

our main result was that basic, off-the-shelf topic modelling does not work well when it comes to highly technical articles from subjects like mathematics. It was also a welcome lesson in not believing the hype of machine learning, even when a problem looks exactly like the kind machine learning was supposed to excel at solving. While we had a hammer and our problem looked like a nail, it seemed that the former was a ball peen and the latter a railroad tie. In the end, even in the land of hammers and nails, the tool has to match the task. Though we failed to accomplish automated categorization of mathematics, we were dilettantes in the world of machine learning. I believe our project is a good example of how machine learning is still a long way from being the magic tool as some, though not all (Rahimi and Recht 2017), have portrayed it. Let us look at what happens when smarter and more capable minds tackle the problem of classifying mathematics and other highly technical subjects using advanced machine learning techniques.

Finding the Right Hammer

To illustrate the quest to find the right hammer I am going to focus on three different projects that tackled the automated categorization of highly technical content, two of which also attempted to categorize mathematical content and one that looked to categorize scholarly works in general. These three projects provide examples of many of the approaches and practices employed by experts in automated classification and demonstrate the two main paths that these types of projects follow to accomplish their goals. Since we have been discussing mathematics, let us start with those two projects.

Both projects began because the participants were struggling to categorize mathematics publications so they would be properly indexed and searchable in digital mathematics databases: the Czech Digital Mathematics Library (DML-CZ)[6] and NUMDAM[7] in the case of Radim Řehůřek and Petr Sojka (Řehůřek and Sojka 2008), and Zentralblatt MATH (zbMath)[8] in the case of Simon Barthel, Sascha Tönnies, and Wolf-Tilo Balke (Barthel, Tönnies, and Balke 2013). All of these databases rely on the aforementioned MSC[9] to aid in indexing and retrieval, and so their goal was to automate the assignment of MSC values to lower the time and labor cost of requiring humans to do this task. The main differences between their tasks related to the number of documents they were working with (thousands for Řehůřek and Sojka and millions for Barthel, Tönnies, and Balke), the amount of the works available (full text for Řehůřek and Sojka, and titles, authors, and abstracts for Barthel, Tönnies, and Balke), and the quality of the data (mostly OCR scans for Řehůřek and Sojka and mostly TeX for Barthel, Tönnies, and Balke). Even with these differences, both projects took a similar approach, and it is the first of the two main pathways toward classification I spoke of earlier: using a predetermined taxonomy and a set of pre-categorized data to build a machine learning categorizer.

In the end, while both projects determined that the use of Support Vector Machines (Gandhi 2018)[10] provided the best categorization results, their implementations were different. The Ře-

[6] See `https://dml.cz/`.

[7] See `http://www.numdam.org/`.

[8] See `https://zbmath.org/`.

[9] Mathematical Subject Classification (MSC) values in MathSciNet and zbMath are a particularly interesting categorization set to work with as they are assigned and reviewed by a subject area expert editor and an active researcher in the same, or closely related, subfield as the article' s content before they are published. This multi-step process of review yields a built-in accuracy check for the categorization.

[10] Support Vector Machines (SVMs) are machine learning models which are trained using a pre-classified corpus to split a vector space into a set of differentiated areas (or categories) and then attempt to classify new items by where in the

hůřek and Sojka SVMs were trained with terms weighted using augmented term frequency[11] and dynamic decision threshold[12] selection using *s-cut*[13] (Řehůřek and Sojka 2008, 549) and Barthel, Tönnies, and Balke's with term weighting using term frequency–inverse document frequency[14] and Euclidean normalization[15] (Barthel, Tönnies, and Balke 2013, 88), but the main difference was how they handled formulae. In particular the Barthel, Tönnies, and Balke group split their corpus into words and formulae and mapped them to separate vectors which were then merged together for a combined vector used for categorization. Řehůřek and Sojka did not differentiate between words and formulae in their corpus, and they did note that their OCR scans' poor handling of formulae could have hindered their results (Řehůřek and Sojka 2008, 555). In the end, not having the ability to handle formulae separately did not seem to matter as Řehůřek and Sojka claimed microaveraged F_1 scores of 89.03% (Řehůřek and Sojka 2008, 549) when classifying the top level MSC category with their best performing SVM. When this is compared to the microaveraged F_1 of 67.3% obtained by Barthel, Tönnies, and Balke (Barthel, Tönnies, and Balke 2013, 88), it would seem that either Řehůřek's and Sojka's implementation of SVMs or their access to full-text led to a clear advantage. This advantage becomes less clear when one takes into account that Řehůřek and Sojka were only working with top level MSCs where they had at least 30 (60 in the case of their best result) articles, and their limited corpus meant that many top-level MSC categories would not have been included. Looking at the work done by Barthel, Tönnies, and Balke makes it clear that these less common MSC categories such as K-Theory or Potential Theory, for which Barthel, Tönnies, and Balke achieved microaveraged F_1 measures of 18.2% and 24% respectively, have a large impact on the overall effectiveness of the automated categorization. Remember, this is only for the top level of MSC codes, and the work of Barthel, Tönnies, and Balke suggests it would get worse when trying to apply the second and third level for full MSC categorization to these less-common categories. This leads me to believe that in the case of categorizing highly technical mathematical works to an existing taxonomy, people have come close to identifying the overall size of the machine learning hammer, but are still a long way away from finding the right match for the categorization nail.

Now let us shift from mathematics-specific categorization to subject categorization in general and look at the work Microsoft has done assigning Fields of Study (FoS) in the Microsoft Academic Graph (MAG) which is used to create their Microsoft Academic article search product.[16] While the MAG FoS project is also attempting to categorize articles for proper indexing and search, it represents the second path which is taken by automated categorization projects: using machine learning techniques to both create the taxonomy and to classify.

Microsoft took a unique approach in the development of their taxonomy. Instead of rely-

---

vector space the trained model places them. For a more in-depth, technical explanation, see: `https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47`.

[11] Augmented term frequency refers to the number of times a term occurs in the document divided by the number of times the most frequent occurring term appears in the document.

[12] The decision threshold is the cut-off for how close to a category the SVM must determine an item to be in order for it to be assigned that category. Řehůřek and Sojka' s work varied this threshold dynamically.

[13] Score-based local optimization, or s-cut, allows a machine-learning model to set different thresholds for each category with an emphasis on local, or category, instead of global performance.

[14] Term frequency–inverse document frequency provides a weight for terms depending on how frequently it occurs across the corpus. A term which occurs rarely across the corpus but with a high frequency within a single document will have a higher weight when classifying the document in question.

[15] A Euclidean norm provides the distance from the origin to a point in an n-dimensional space. It is calculated by taking the square root of the sum of the squares of all coordinate values.

[16] See `https://academic.microsoft.com/`.

ing on the corpus of articles in the MAG to develop it, they relied primarily on Wikipedia for its creation. They generated an initial seed by referencing the Science Metrix classification scheme[17] and a couple thousand FoS Wikipedia articles they identified internally. They then used an iterative process to identify more FoS in Wikipedia based on whether they were linked to Wikipedia articles that were already identified as FoS and whether the new articles represented valid entity types—e.g. an entity type of protein would be added and an entity type of person would be excluded (Shen, Ma, and Wang 2018, 3). This work allowed Microsoft to develop a list of more than 200,000 Fields of Study for use as categories in the MAG.

Microsoft then used machine learning techniques to apply these FoS to their corpus of over 140 million academic articles. The specific techniques are not as clear as they were with the previous examples, likely due to Microsoft protecting their specific methods from competitors, but the article published to the arXiv by their researchers (Shen, Ma, and Wang 2018) and the write up on the MAG website does make it clear they used vector based convolutional neural networks which relied on Skip-gram (Mikolov et al. 2013) embeddings and bag-of-words/entities features to create their vectors ("Microsoft Academic Increases Power of Semantic Search by Adding More Fields of Study—Microsoft Research" 2018). One really interesting part of the machine learning method used by Microsoft was that it did not rely only on information from the article being categorized. It also utilized the citations to and references from information about the article in the MAG, and used the FoS the citations and references were assigned in order to influence the FoS of the original article.

The identification of potential FoS and their assignment to articles was only a part of Microsoft's purpose. In order to fully index the MAG and make it searchable they also wished to determine the relationships between the FoS; in other words they wanted to build a hierarchical taxonomy. To achieve this they used the article categorizations and defined a Field of Study A as the parent of B if the articles categorized as B were close to a subset of the articles categorized as A (a more formal definition can be found in (Shen, Ma, and Wang 2018, 4). This work, which created a six-level hierarchy, was mostly automated, but Microsoft did inspect and manually adjust the relationships between FoS on the highest two levels.

To evaluate the quality of their FoS taxonomy and categorization work, Microsoft randomly sampled data at each of the three steps of the project and used human judges to assess their accuracy. The accuracy assessments of the three steps were not as complete as they would be with the mathematics categorization, as that approach would evaluate terms across the whole of their data sets, but the projects are of very different scales so different methods are appropriate. In the end Microsoft estimates the accuracy of the FoS at 94.75%, the article categorization at 81.2%, and the hierarchy at 78% (Shen, Ma, and Wang 2018, 5). Since MSC was created by humans there is no meaningful way to compare the FoS accuracy measurements, but the categorization accuracy falls somewhere between that of the two mathematics projects. This is a very impressive result, especially when the aforementioned scale is taken into account. Instead of trying to replace the work of humans categorizing mathematics articles indexed in a database, which for 2018 was 120,324 items in MathSciNet[18] and 97,819 in zbMath,[19] the FoS project is trying to replace the human categorization of all items indexed in MAG, which was 10,616,601 in 2018.[20]

[17] See `http://science-metrix.com/?q=en/classification`.
[18] See `https://mathscinet.ams.org/mathscinet/search/publications.html?dr=pubyear&yrop=eq&arg3=2018`.
[19] See `https://zbmath.org/?q=py%3A2018`.
[20] See `https://academic.microsoft.com/publications/33923547`.

Both zbMath and MathSciNet were capable of providing the human labor to do the work of assigning MSC values to the mathematics articles they indexed in 2018.[21] Therefore using an automated categorization, which at best could only get the top level right with 90% accuracy, was not the right approach. On the other hand, it seems clear that no one could feasibly provide the human labor to categorize all articles indexed by MAG in 2018 so an 80% accurate categorization is a significant accomplishment. To go back to the nail and hammer analogy, Microsoft may have used a sledgehammer but they were hammering a rather giant nail.

Are You Sure it's a Nail?

I started this chapter talking about how we have all been told that AI and machine learning were going to revolutionize everything in the world. That they were the hammers and all the world's problems were nails. I found that this was not the case when we tried to employ it, in an admittedly rather naive fashion, to automatically categorize mathematical articles. From the other examples I included, it is also clear computational experts find the automatic categorization of highly technical content a hard problem to tackle, one where success is very much dependent on what it is being measured against. In the case of classifying mathematics, machine learning can do a decent job but not enough to compete with humans. In the case of classifying everything, scale gives machines an edge, as long as you have the computational power and knowledge wielded by a company like Microsoft.

This collection is about the intersection of AI, machine learning, deep learning, and libraries. While there are definitely problems in libraries where these techniques will be the answer, I think it is important to pause and consider if artificial intelligence techniques are the best approach before trying to use them. Libraries, even those like the one I work in, which are lucky enough to boast of incredibly talented IT departments, do not tend to have access to a large amount of unused computational power or numerous experts in bleeding-edge AI. They are also rather notoriously limited budget-wise and would likely have to decide between existing budget items and developing an in-house machine learning program. Those realities combined with the legitimate questions which can be raised about the efficacy of machine learning and AI with respect to the types of problems a library may encounter, such as categorizing the contents of highly technical articles, make me worry. While there will be many cases where using AI makes sense, I want to be sure libraries are asking themselves a lot of questions before starting to use it. Questions like: is this problem large enough in scale to substitute machines for human labor given that machines will likely be less accurate? Or: will using machines to solve this problem cost us more in equipment and highly technical staff than our current solution, and has that factored in the people and services a library may need to cut to afford them? Or: does the data we have to train a machine contain bias and therefore will produce a biased model which will only serve to perpetuate existing inequities and systemic oppression? Not to mention: is this really a problem or are we just looking for a way to employ machine learning to say that we did? In the cases where the answers to these questions are yes, it will make sense for libraries to employ machine learning. I just want libraries to look really carefully at how they approach problems and solutions, to make sure that

[21] When an article is indexed by MathSciNet it receives initial MSC values from a subject area editor who then passes the article along to an external expert reviewer who suggests new MSC values, completes partial values, and provides potential corrections to the MSC values assigned by the editors ("Mathematical Reviews Guide For Reviewers" 2020) and then the subject area editors will make the final determination in order to make sure internal styles are followed. zbMath follows a similar procedure.

their problem is, in fact, a nail, and then to look even closer and make sure it is the type of nail a machine-learning hammer can hit.

References

Barthel, Simon, Sascha Tönnies, and Wolf-Tilo Balke. 2013. "Large-Scale Experiments for Mathematical Document Classification." In *Digital Libraries: Social Media and Community Networks,* edited by Shalini R. Urs, Jin-Cheon Na, and George Buchanan, 83–92. Cham: Springer International Publishing.

Clark, Jack. 2015. "Why 2015 Was a Breakthrough Year in Artificial Intelligence." *Bloomberg*, December 8, 2015. https://www.bloomberg.com/news/articles/2015-12-08/why-2015-was-a-breakthrough-year-in-artificial-intelligence.

Gandhi, Rohith. 2018. "Support Vector Machine—Introduction to Machine Learning Algorithms." Medium. July 5, 2018. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

"GeoDeepDive: Project Overview.' n.d. Accessed May 7, 2018. https://geodeepdive.org/about.html.

Kelly, Kevin. 2014. "The Three Breakthroughs That Have Finally Unleashed AI on the World." *Wired*, October 27, 2014. https://www.wired.com/2014/10/future-of-artificial-intelligence/.

"Mathematical Reviews Guide For Reviewers." 2015. *American Mathematical Society.* February 2015. https://mathscinet.ams.org/mresubs/guide-reviewers.html.

"Microsoft Academic Increases Power of Semantic Search by Adding More Fields of Study." 2018. *Microsoft Academic* (blog). February 15, 2018. https://www.microsoft.com/en-us/research/project/academic/articles/microsoft-academic-increases-power-semantic-search-adding-fields-study/.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." In *Advances in Neural Information Processing Systems 26*, edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, 3111–3119. Curran Associates, Inc. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Parloff, Roger. 2016. "From 2016: Why Deep Learning Is Suddenly Changing Your Life." *Fortune*. September 28, 2016. https://fortune.com/longform/ai-artificial-intelligence-deep-machine-learning/.

Rahimi, Ali, and Benjamin Recht. 2017. "Back When We Were Kids." Presentation at the NIPS 2017 Conference. https://www.youtube.com/watch?v=Qi1Yry33TQE.

Řehůřek, Radim, and Petr Sojka. 2008. "Automated Classification and Categorization of Mathematical Knowledge." In *Intelligent Computer Mathematics*, edited by Serge Autexier, John Campbell, Julio Rubio, Volker Sorge, Masakazu Suzuki, and Freek Wiedijk, 543–57. Berlin: Springer Verlag.

Sangwani, Gaurav. 2017. "2017 Is the Year of Machine Learning. Here's Why." *Business Insider*, January 13, 2017. https://www.businessinsider.in/2017-is-the-year-of-machine-learning-heres-why/articleshow/56514535.cms.

Shen, Zhihong, Hao Ma, and Kuansan Wang. 2018. "A Web-Scale System for Scientific Knowledge Exploration." Paper presented at the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, July 2018. http://arxiv.org/abs/1805.12216.

Tank, Aytekin. 2017. "This Is the Year of the Machine Learning Revolution." *Entrepreneur*, January 12, 2017. https://www.entrepreneur.com/article/287324.