Extracting Mathematical Concepts with Large Language Models

Valeria de Paiva*1*, Qiyue Gao*2*, Pavel Kovalev*3* and Lawrence S. Moss*4,†*

Abstract

We extract mathematical concepts from mathematical text using generative large language models (LLMs) like ChatGPT, contributing to the field of automatic term extraction (ATE) and mathematical text processing, and also to the study of LLMs themselves. Our work builds on that of others, in that we aim for automatic extraction of terms (keywords) in one mathematical field, category theory, using as a corpus the 755 abstracts from a snapshot of the online journal *Theory and Applications of Categories*, circa 2020. Where our study diverges from previous work is in (1) providing a more thorough analysis of what makes mathematical term extraction a difficult problem to begin with; (2) paying close attention to inter-annotator disagreements; (3) providing a set of guidelines which both human and machine annotators could use to standardize the extraction process; (4) introducing a new annotation tool to help humans with ATE, applicable to any mathematical field and even beyond mathematics; (5) using prompts to ChatGPT as part of the extraction process, and proposing best practices for such prompts; and (6) raising the question of whether ChatGPT could be used as an annotator on the same level as human experts. Our overall findings are that the matter of mathematical ATE is an interesting field which can benefit from participation by LLMs, but LLMs themselves cannot at this time surpass human performance on it.

Keywords
mathematical terminology, knowledge graph, annotation, large language model

1. Introduction

This paper addresses an issue which must be confronted in large-scale user interaction with mathematical text, the isolation of mathematical concepts in text itself. We ask whether large language models (LLMs) can assist in the automatic extraction of terms from text. For example, could one input a textbook or research monograph and ask an LLM to construct a reasonable index for it? Obviously, if this were possible, then it would save authors time and energy. And if an LLM is of limited use, would it still be sensible to ask? What would be the best way to prompt the LLM?

1.1. Mathematical text and mathematical concepts

Mathematical text is different from news or encyclopedic (Wikipedia-style) text. For a start, it has equations and diagrams and special fonts, usually laid out in LaTeX. It usually requires special processing. Mathematical text employs highly specialized vocabulary; these are the topic of this paper. It also adheres to special conventions that can put off beginners, but these make concepts and statements as clear and non-ambiguous as the author can possibly make them. Those conventions also come with their own terminology (see Section 2.1), but in this paper we make a distinction between terms in the mathematical vocabulary and terms in the mathematical practice. We are interested in automatically finding the former rather than the latter.

The emergence of large language models (LLMs, e.g. GPT-3 [1]) has permanently altered the work of natural language processing (NLP) researchers. It goes without saying that the general public has

---

✉ valeria@topos.institute (V. de Paiva); gaoq@rose-hulman.edu (Q. Gao); pakova@indiana.edu (P. Kovalev); lmoss@indiana.edu (L. S. Moss)

CEUR Workshop Proceedings (CEUR-WS.org)

become aware of LLMs, and over 100 million of people have used ChatGPT [2]. This is because LLMs have the capacity of learning downstream tasks given a few question-answer examples (*in-context examples*) or even no example whatsoever in the input. This mysterious and charming capacity motivates us to discover what role LLMs could play in our task of *extracting mathematical concepts from text.* That is, we are not interested in using LLMs to prove theorems or to search the world's mathematical corpus for results. We set our sights on something which at first glance looks "smaller" but which we have made a quick advance on. We build a *mathematical concept extraction pipeline* based on using ChatGPT. We ask whether if it serves as a good annotator, comparable to human experts, or whether instead it is a decent helper that only produces preliminary results, in need of additional processing from human experts.

Contributions   In this work we use ChatGPT [2] to extract mathematical concepts[1] from academic papers in Category Theory, a reasonably recent (mid-last-century) branch of pure mathematics. ChatGPT is the only large language model which we use in this paper. While the results of our work might well depend on the particular LLM used, as ChatGPT being one of the best performing models in the field, we expect the main conclusions of our work would hold for other LLMs. Using NLP tools for dealing with a specific domain such as mathematics is not very easy, as there is basically very little data annotated for mathematics. We make two contributions. First, we analyze mathematical terms (Section 2) and then propose guidelines for human annotators (3). Second, we prompted ChatGPT for such terms (5) and conducted three experiments (6) using a corpus (4). We report on these experiments and draw conclusions. (7)

1.2. Related work

We have not been able to find much related work, especially, related data for the task we set ourselves. The only official NLP competition data we found, discussed in [3, 4], was the corpus SciERC[2]. This dataset includes annotations for scientific entities, their relations, and coreference clusters for 500 scientific abstracts. (These abstracts are taken from 12 AI conference/workshop proceedings in four AI communities, from the Semantic Scholar Corpus.) SciERC extends previous datasets in scientific articles related to the SemEval 2017 Task 10 and SemEval 2018 Task 7 meetings by extending entity types, relation types, relation coverage, and adding cross-sentence relations using coreference links. However, as shown by [5, 6], there seems to be a considerable difference between "scientific entities" and "mathematical entities" as the results reported by the software DyGIE++ of Wadden et al are surprisingly low in the mathematical text. Details of this corpus of annotated scientific entities can be found at [7].

Most of the work on extracting information from mathematical texts has concentrated on extracting meaning from formulas [8], which we do not consider. We want to see how much information we can obtain from mathematical vernacular alone, without symbols, formulas, or diagrams.

Another line of work that might be relevant is the work on "flexiformalization". While we share the motivations and goals of [9], especially in believing that our extracted concepts should be mapped to an ontology such as WikiData, we do not strive for a formalization of the data model obtained from our concepts. We are trying to work with existing mathematical text (category theory in this paper) using tools from contemporary NLP. An important first step for us would be to construct mathematical datasets to be able to test our procedures, and this is what we do: we construct several small datasets of mathematical text annotated for mathematical entities.

---

[1]In this paper we use the terms "term", "concept", and "keyword" interchangeably. We know that there are situations where it is sensible to distinguish these terms, but in the present paper we have no reason to do so.

[2]http://nlp.cs.washington.edu/sciIE/

2. Background: difficulties with the isolation of mathematical concepts in texts

The automatic extraction of mathematical terms was not as easy as we thought it would be. Certainly for machines it turns out to be problematic. For that matter, we have also found that human annotation of terms in mathematical texts comes with its own problems. Our annotation has brought to the surface quite a number of issues concerning mathematical writing, and so we discuss these in this section.

2.1. Words coming from mathematical writing that have little or no mathematical content

One persistent problem is that *mathematical text contains quite a few words which are special to the genre of mathematical writing but which do not themselves bear mathematical content.* Examples include *assumption*, *characterization*, *conjecture*, *consequence*, *contradiction*, *counter-example*, *paper*, and *therefore*. In some cases, the words are more generally part of academic writing beyond mathematics, and they have different meanings in "real life" than in the academic world: *paper* and *proposition*, for example. We work with the assumption that such words would not be part of the index of mathematical terms in any book. Yet if one asks a LLM for a list of words that are especially found in mathematical texts, the words above are likely to be found.

2.2. Difficulties with alternate forms of words

The ultimate goal of this work is the automatic extraction of a list of mathematical terms from text. For this, it would be advantageous to avoid repeated entries, or even to have closely related entries.

1. *Should nouns be listed in the singular or the plural, even if their appearance in the source text is different?* The difficulty here is not special to mathematics: it affects terminological indices in every subject whatsoever.
2. *Should adjectives be listed alone, or only with the nouns they modify?* It would be simpler to drop the head noun, but this will not work. Frequently adjectives have different meaning depending on the nouns they modify: *regular polygon*, *regular expression*, etc. Even in our target subject of category theory, *regular monomorphism* and *regular epimorphism* have different (but related) meanings. This point also affects all disciplines.
3. *If an adjective is inflectionally derived from a verb, should the automatic system prefer one form over another?* For example, in *interpolated function*, should an index show *interpolate*, even though it might be missing from the text? Or should it show *interpolated*, or *interpolated function*, or even all of these?
4. *In cases where there are constructions involving two or more adjectives handling a single noun, what should the system do?* If a text says *differential graded category*, should we also add to an index sub-expressions like *differential category* and *graded category*? In many cases, the decision on this requires expert knowledge (or perhaps access to much more data). This situation is fairly common. How should it be handled?
5. *How should an automatic system handle expressions with prepositions?* For example *sheaf of germs of analytic functions*? Should it include this expression, and also *germs of analytic functions* and also *analytic functions*? Mathematical text is full of such expressions: *area between the tangents to two circles*, for example.
6. *How should the system handle proper names, or upper-case symbols in the source data?* In mathematical text, one does find proper names. They might occur in adjective-like positions: as in *Shanin's method*, *Lagrange interpolation*, *Birkhoff's Variety Theorem*. In some cases, the

- Try to treat math concepts as black-boxes, as much as possible. We do not expect users to know their full meaning. We intend to annotate any term that we think a mathematician might not know and might want to check in some glossary or in some index.
- Use the singular, instead of the plural, for concepts. Thus, annotators are expected to extract from the sentence, *We show that both approaches give equivalent bicategories* the concept *equivalent bicategory*.
- A generic decision was that terms like *theorem, corollary, conjecture, theory* are very used in mathematics, but are not the kind of mathematical concept we want to extract from texts. These are almost meta-concepts.
- Even if names of important mathematicians sometimes appear in book indices, the mathematicians themselves are not mathematical concepts. Thus while *Grothendieck's construction* is a concept, *Grothendieck* by itself is not a mathematical concept.
- If one has a long span that is a concept, e.g. *enriched accessible categories*, we should also list the sensible sub-spans like *accessible category*.
- If a sentence contains a mathematical adjective together with a mathematical noun (e.g. *nilpotent algebra*), then we include the adjective together with the noun, but not the standalone adjective. (And we also include the noun itself since we want all smaller spans.) If a sentence contains a mathematical adjective without a mathematical noun (e.g. *nilpotent case*), then we only include the standalone adjective. If a sentence contains a mathematical adjective without any noun, we include the adjective.
- Try as much as possible to avoid prepositions occurring inside of concepts. In an example like *sheaf of germs of analytic functions* we want *sheaf, germ*, and *analytic function*, but we don't want *germs of analytic function* and we don't want the original (long) span. That is, these complex concepts are something that a reader would naturally understand but not find in an index.

Figure 1: Annotation guidelines.

name is part of the concept; the concept would not be understandable without the name. For example: a mathematical text which dropped *Cauchy* from *Cauchy sequence* would likely have an error, and an index that omitted *Cauchy sequence* in favor of *sequence* would not be helpful. The same holds for *Gauss' Lemma* and *Turing machine*. Certainly *Newton's Method* is a standard concept that books on calculus would want to index. In other cases, the proper name is much less standard. Our data included *Shanin's method*, and we ourselves do not know whether this is a standard usage or a nonce.

3. Human annotation and guidelines

Looking simply at one or two examples this task of extracting concepts seemed easy enough. For a sentence like *A notion of central importance in categorical topology is that of topological functor*, it was clear that we needed to extract the concepts of *categorical topology* and *topological functor*. But it was also clear that one should know the concept of a *functor* and maybe also the one of *topology*. Thus, some sub-concepts of extracted concepts should also be concepts. But which ones?

We realized that we needed some guidelines for the extraction of concepts and decided to annotate a pilot set of a hundred sentences to determine the necessary guidelines to guarantee a good inter-annotator agreement between the human annotators. The set of these hundred sentences can be found at https://github.com/vcvpaiva/NLIMath/blob/main/PilotTest100.txt.

We formulate some generic guidelines for our annotations and list them in Figure 1.

However, while annotating carefully the initial sentences we noticed that these generic guidelines were not enough. Some further guidelines for annotation were created from this pilot annotation exercise.

1. Certain words and expressions can be either math concepts or can be used in their English sense, for example *method* in *Shanin's method*. If the word is used as a mathematical concept, we want to use the whole expression. So for example in *abelian group*, the word *group* is used

in the mathematical sense, so the concept is the whole expression. But if the word is used as a common English noun we do not want it as part of the concept. This is one of the reasons for disagreements between annotators, is *method* above used as common English word or as mathematical concept? Similarly, for *2-categorical refinement*, is this refinement a mathematical concept?

2. We believe we should do 'coreference resolution' for our terms. So for the sentence *We introduce a new class of categories generalizing locally presentable ones* we want to extract the term *locally presentable category*.

3. For a long mathematical expression like *skew monoidal categories*, suppose that the annotator is not sure of its meaning. Suppose also that some sub-strings do make sense (monoidal category), while others might not e.g. *skew category*. Unless we know the math involved, we cannot tell all the sub-concepts of a long expression. So this is a known reason for disagreement between annotators. Even though the annotators are instructed to set aside their mathematical training, this is frequently too much to ask.

4. The hardest piece of guideline is what to do about adjectives like *analytic*, *algebraic* or *categorical* that are not concepts themselves, but that might indicate, the existence of a concept, in say *2-categorical refinement* above.

Then we asked ChatGPT to find concepts following the theme of our guidelines above and compared results obtained by ChatGPT with the ones from our human annotators.

4. Corpus preparation

We use an already prepared corpus, consisting of 755 abstracts from the open source journal *Theory and Application of Categories*[3], processed around 2020. This small corpus (around 3.2K sentences) is available from GitHub[4]. The abstracts were processed with the BERT version of spaCy[5]. The corpus was already used before for extracting concepts using NLP tools [5]. Here we are exploring what generative models, especially ChatGPT, can help with.

Since we are dealing with abstracts, we have less LaTeX markup than if we were using full papers. We also have no diagrams, and spaCy provides us with lists of nouns, proper nouns, compounds, and adjective-noun terms, following the conventions of Universal Dependencies [10]. Long sentences are likely to complicate the parsing and very short ones can be parsing or sentencization errors, so we use an extracted collection of sentences with moderate size and free from LaTeX mark-up. These can be found at https://bit.ly/tac-examples. This file also has examples of humanly-extracted concepts for each sentence. Using this collection of sentences, a set of humanly-extracted mathematical concepts was created for a random set of 436 sentences from the TAC abstracts. We used three annotators from our team. They could check the context of the sentences, if desired. Then we wanted to measure the extent of agreement between these three annotators. This turned out to be more complicated than expected.

4.1. New annotation tool

To discover true disagreements and reduce the noise produced by minor mistakes, we have repurposed a tool from previous work of Chen, Gao and Moss [11] to help with the human annotation. Figure 4 in Appendix A gives an overview of this annotation platform. After uploading the dataset in the 'upload' tab, users can select the dataset in a drop-down menu and specify a starting index to initiate the annotation process. One can then select consecutive words from the provided sentence, and they

---

[3] http://www.tac.mta.ca/tac/

[4] https://github.com/ToposInstitute/tac-corpus

[5] https://spacy.io/

will be entered into the textbox below, where further edits could be done such as changing from plural to singular forms. Each annotated concept is listed under the sentence and can be removed, if necessary. These annotations are stored in our database upon submission. To retrieve the annotations for a specific dataset, users can use the 'download' button at the bottom of the platform.

5. Prompting ChatGPT

Given the following Context, extract the words that denote Math concepts.

Here are some examples:
{*in-context   example*}

Now please solve the following problem.

Context: {*math_sentence*}
Concepts:

Table 1
The initial prompt template, where *math_sentence* denotes the sentence from which we wish to extract math concepts.

Context: 'Let PreOrd(C) be the category of internal preorders in an exact category C.'
Concepts: ['internal preorder', 'exact category']

Table 2
An example of the in-context demonstration.

Table 1 shows the initial prompt given to ChatGPT. The prompt starts with a concise description of the task followed by several in-context examples. An example of in-context demonstration is shown in Table 2. After these demonstrations, the real problem is given and the model is asked to extract concepts from this context. We first conducted a pilot study using this prompt on the random set of 300 sentences from the TAC abstracts. The evaluation from our human experts shows that the concepts generated from each sentence share common mistakes, including keeping the plural form as it appears in the context, generating a person's name and identifying everyday non-mathematical used words as concepts. To help the model avoid such mistakes, we added specialized instructions in the prompt, as shown in Table 3. These instructions alleviate the above issues but do not solve them, since the model failed to generalize to words not mentioned in these instructions. We find common words like "example" have been extracted, and plural words like "measures" not been transformed to the singular. We think that detailed explanations of why some words are not mathematical concepts might help the model to understand our criteria and the categorization of mathematical concepts. This motivates us to construct detailed in-context examples(e.g. Table 4) that constitutes our final prompt, where ordinary in-context examples are replaced by a few detailed ones.

Given the following Context, extract the words that denote Math concepts.
Be sure to make the concept words singular. For example when we see 'functors' in a sentence, we would extract 'functor' rather than 'functors' or when we see 'categories' we would like to extract 'category' instead of 'categories'!

Here are some examples:
{*in-context example*}

Also note that we are looking for concepts like modulation, enriched orthogonality, holonomy, localization, variety, but not words shown in daily English sentences like 'future work', 'conclusion', 'this property'!

We don't want a person's name to be extracted as a math concept although we understand that a person's name could be part of the phrase that denotes a math concept.

Now please solve the following problem.

Context: {*math_sentence*}
Concepts:

Table 3
The updated prompt template with specialized instructions shown in different colors.

Context: 'If the category is additive, we define a sheaf of categories of analytic functions.'
Concepts: ['additive category', 'sheaf', 'analytic function']

Reason: the concept 'additive category' is generalized from the sentence because of the original phrase 'category is additive';
the concept 'sheaf' is a known math concept shown in the sentence;
the concept 'analytic function' is extracted from the original phrase 'analytic functions' by removing the plural form;
here we don't want the single 'category' and 'additive' extracted as math concepts since they are usual words
nor do we want the phrase 'category of analytic functions' since the more concise phrase 'analytic function'
should be extracted instead as a math concept.

Table 4
An example of the **detailed** in-context demonstration.

6. Experiments and evaluation

We have carried out three experiments with human and machine annotators.

1. An in-depth pilot with 100 sentences from the TAC corpus. We aimed for sentences with little or no special symbols and which were of moderate length.
2. A longer study with sentences from the same corpus. We chose 436 sentences from TAC abstracts (see bit.ly/3DKEitc). This file also has examples of humanly-extracted concepts for each sentence. For this longer set we use an industry strategy: We compare one human annotator with ChatGPT for all terms extracted. If they disagree, a second human adjudicator is invoked and their decision is final.
3. A full-scale attempt to use ChatGPT to annotate 55K sentences from the nLab website. We discuss this in Section 6.4.

6.1. First experiment: pilot with 100 sentences from the TAC corpus

As mentioned, our first experiment was based on 100 sentences from the TAC corpus. Here are examples of sentences from this study:

> We check these extra assumptions in several categories with pretopologies.
> Functors between groupoids may be localised at equivalences in two ways.
> We show that both approaches give equivalent bicategories.
> In this paper, we use the language of operads to study open dynamical systems.
> The syntactic architecture of such interconnections is encoded using the visual language of wiring diagrams.

Three members of our team annotated 100 of these sentences, and at the same time ChatGPT was prompted. The annotators were given a spreadsheet with the sentences in a single column, and next to this column were empty columns in which they were asked to list all of the terms that should appear in a mathematical knowledge graph or an index of terms in category theory. At first many 'minor mistakes' were made by the annotators. They introduced typos, empty spaces and forgot to reduce concepts to singular terms, for instance. There were also many cases of annotators forgetting to read part of a sentence or treating some mathematical terms as common English nouns.

It was for this reason that we formulated the guidelines mentioned in Section 3. This, too, was not a straightforward matter. There were differences at every step. Guidelines coordinate the annotation practice of the team members, but occasionally they sacrifice completeness in the process.

In order to make the annotation process easier and more reliable, we used the tool mentioned in Section 4.1.

Here are our results on annotation of 100 sentences from the corpus, after filtering. Out of 327 concepts extracted by the 4 annotators (three humans and ChatGPT) from the 100 sentences in the pilot experiment, we have 120 concepts that all four annotators agree on. This gives (only) 37% full agreement. This is pre-filtering. After filtering and some light changes (removing plurals and some common nouns like "decade", etc.) we get about 40% full agreement. This still seems low as a measure of full agreement.

Annotator	Number of Concepts
Annotator 1	206
Annotator 2	199
Annotator 3	194
ChatGPT (pre-filtering)	235
The Union of All (pre-filtering)	327
ChatGPT (post-filtering)	226
The Union of All (post-filtering)	310

There are 40 concepts that the three human annotators agree are concepts, but ChatGPT does not extract (about 12% of the total). Some examples include *internal presheaf, gerbe* that only appear in the plural. Many are terms that human annotators considered important subspans of terms ChatGPT considers mathematical concepts. Examples include *triple category, probability distribution, topology*, respectively, subspans from terms *strict triple category, joint probability distributions, categorical topology*. See Figure 2 for further examples.

It is not so surprising that ChatGPT did not extract all the concepts we want. What seems really surprising is the lack of agreement between the human annotators. Especially because the lack of agreement does not look conceptual, but simply noise in the extraction process. But how much is conceptual, how much is noise or minor mistakes?

In addition, ChatGPT extracted concepts that are unwanted: again, see Figure 2. The one member of our team who did not annotate then *filtered* the terms extracted by ChatGPT. They looked at the

Here are some of the concepts which all three human annotators found but ChatGPT missed:

Grothendieck's six operations	group homotopy	quotient triangulated category
Lie algebra	left proper model structure	representation
arithmetic variety	morphism axiom	smooth stack
cartesian closed category	pointed regular	sup-lattice
closed category	protomodular category	topology
graph rewriting	probability distribution	triple category

Here are some of the concepts found by ChatGPT which none of the three human annotators found:

Grothendieck	categorical property	language	property
acyclic models method	category theory	localize	prove
algebraic context	characterization	locally presentable	result
analysis	closed monoidal	mathematically natural	six operation
application	consequence	motivation	structure
approach	construction	non-abelian	symmetric monoidal
axiom	corollary	open question	tame
balanced category	definition	perspective	theory
calculate	example	previous work	trivial
categorical	issue	proof	uniqueness statement

Figure 2: Results discussed in Section 6.1.

list of concepts that appeared in the ChatGPT list but not in the human list and deleted items that they considered to not be mathematical concepts; a few items were also added.

6.2. Evaluating the pilot

The evaluation strategy uses a Jupyter notebook to transform the chosen terms into zeros and ones to calculate agreement. We measure agreement using the *Jaccard similarity index*. For two sets A and B, the definition is

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

We are given lists of terms, one for each annotator (three humans and one non-human). We produced one "master-list" that contains all concepts listed by at least one annotator (with no repetitions). We then created an empty dictionary with the purpose of storing the counts of each unique concept for each annotator. By iterating over each of the four lists and utilizing the master-list, we populated the dictionary with the desired value counts for each unique concept and converted it into a dataframe where the columns correspond to the annotators and the rows correspond to the unique concepts from the master-list. Since each concept from the master-list is present at most once in each of the four individual lists of concepts, the possible value counts are either 0 or 1. The entry at the intersection of row i and column j is 1 if and only if annotator j listed concept i.

Figure 3 presents a set of comparisons of the Jaccard similarity indices. The main message is that by agreeing to guidelines, the human annotators were able to achieve pairwise similarity scores in the range $0.75 - 0.8$. They were not able to go higher. On the other hand, ChatGPT's Jaccard index relative to humans is about 0.5 even with filtering.

6.3. Second experiment: 436 sentences from the TAC corpus

As previously mentioned, in experiment 2, one human annotator and ChatGPT independently extracted concepts from 436 TAC sentences, and then a second human *adjudicated* conflicts between the

Annotators Being Compared	Jaccard Score
annotator 1 and annotator 2	0.753
annotator 1 and annotator 3	0.794
annotator 2 and annotator 3	0.746
ChatGPT and annotator 1	0.485
ChatGPT and annotator 2	0.518
ChatGPT and annotator 3	0.505
ChatGPT and union of the humans	0.45
ChatGPT (after filtering) and union of the humans	0.5

Figure 3: Comparison of Jaccard similarities in experiment 1.

human and ChatGPT. This same person then filtered ChatGPT's output, as was done in experiment 1. Out of 250 items, 147 (or about 59%) were deleted. It is worth noting that they were not very strict. For example, the adjudicator did not remove standalone adjectives (e.g. *abelian* or *accessible*) even if they already appeared as part of larger math concepts in ChatGPT's list. This is more relaxed than our guidelines for humans, which say that standalone adjectives may only be included in certain cases. But it is not reasonable to expect that ChatGPT would be able to follow such a complicated guideline without being provided a variety of examples illustrating when it does and does not apply.

The adjudication-cum-filtering process increased the Jaccard similarity index between the human annotator from 0.531 to 0.631. The adjudicator/filterer did not alter the other difference list – the human list minus ChatGPT list – mainly because humans are unlikely to include items that are obviously not mathematical concepts. There might be disagreement between the human annotator and adjudicator as to whether an item like *Shanin's method* is indeed a concept, but settling such cases was not in the agenda of experiment 2. Instead, we wanted to only exclude items that are obviously not mathematical concepts. For example, *previous work* is a concept found by ChatGPT. But this is an error: no human being would think or say that this is a concept of mathematics.

Here are the results of this experiment.

Measurement	Value
Length of Human	673
Length of ChatGPT	740
Length of ChatGPT minus Human	250
Length of Human minus ChatGPT	183
Length of Adjudicated and Filtered ChatGPT minus Human	103
Length of ChatGPT after Adjudication and Filtering	593

Measurement	Jaccard Score
Between Human and Un-filtered ChatGPT	0.531
Between Human and Filtered ChatGPT	0.631

Experiment 2 had two main problems: The results depend too much on the humans used for the experiment; we would prefer to have more humans involved, to decrease the subjective aspects. Second, the more specialized the mathematics the harder it is for mathematicians to decide what is really a concept that it is worth keeping, versus what is an auxiliary definition that will not be adopted by others. From this perspective it would make more sense to work with text from more established mathematics, textbooks, where concepts have been crystallized by use, as opposed to journal articles, where the concepts are being forged. This leads us to our third experiment, where

the text comes from a standard Category Theory resource, the wiki nLab.

6.4. For the near future: terms extracted from 55K sentences in a standard resource

A currently standard source for category theory at all levels is the nLab website https://ncatlab.org/. We have run ChatGPT with our prompts on a subset of 55K sentences from this website; see https://raw.githubusercontent.com/ToposInstitute/nlab-corpus/main/nlab_examples.csv. Of course we do not have human judgments to serve as a gold standard for these 55K sentences. We do have ChatGPT's concepts and we propose to use them as the automatically extracted concepts (filtered to some degree), given the relatively good results obtained in the two previous experiments.

Our aim is to put this information on a website as part of a large-scale effort to allow mathematicians to interact with our work. Users will be able to approve or disapprove of ChatGPT's selection of concepts. We plan to use a modification of the same tool which we mentioned in Section 4.1. This "living experiment" is somewhat similar to the first half of the experiment proposed in the blogpost "Introducing the MathFoldr Project"[6]. While on the earlier experiment we used NLP tools (then spaCy and Universal Dependencies) to produce the concepts that we wanted to map to WikiData, now we can use ChatGPT for the same purpose and hopefully the results are better.

Our task all along has been to ask whether a given word is or is not a "mathematical concept". We do this by asking: does the word belong in a knowledge graph of the mathematical subject under discussion, and does it belong in an index? Our work has shown that this question is subtle and that there are many borderline cases. We plan to ask the community of category theorists who use platforms like Zulip for assistance.

This will have three results: first, it is likely to lead to additional sharpening of our understanding of mathematical terminology in the first place. (We are not aware of any individuals or groups looking at this topic from a data-driven perspective.) Also, the involvement of the community will lead to a better appreciation of our topic, and of Mathematical User Interaction more broadly. Thirdly, once in possession of a substantial number of concepts in several areas of mathematics we can organize this knowledge in structured ways, building a comprehensive knowledge graph. More importantly, the organized concepts should allow us to help with the formalization of mathematical results. There is a large movement towards formalizing mathematics using proof assistants nowadays. We believe that this kind of formalization will be more successful if we build it from the actual lingua franca of mathematicians, which is mathematical English.

7. Conclusions

ChatGPT can help with the extraction of mathematical terms from mathematical texts, but at this time LLMs cannot replace humans. With judicious prompting, ChatGPT can find terms, but it misses some important ones that a mathematically-literate human would want, such as terms which are not fully explicit in a text but which a human would construct on the fly. For example, in a sentence like *We conjecture that this holds for all rings, including non-Artinian ones*, a reader would know that *Artinian rings* belongs in a knowledge graph, even though it is not explicit. An LLM might not extract that term. At the same time it is likely to extract terms that would not normally be considered "mathematical terms," such as *conjecture* or even *ones*.

The fact that ChatGPT may be called almost effortlessly and gives acceptable preliminary results suggests that it will be used in the future. Our work gives numbers that serve as baselines to future work in the area, both in terms of inner-annotator agreement by humans (measured by Jaccard index) and also as in the quality of the LLMs findings.

We ourselves learned a lot about mathematical text by doing the annotations in a serious way. Three people from our team have extensive experience with both mathematics (even category theory)

---

[6]https://topos.site/blog/2021/07/introducing-the-mathfoldr-project/2020

and computational linguistics. Two of us have spent years close-reading and correcting a corpus of natural language inference. Yet we found it surprising that it was difficult to agree on borderline cases, such as multi-word expressions containing adjectives or prepositions. Returning to our example above, is *non-Artinian* a concept one would want in a knowledge graph or index? We think not. But again, it has proved too difficult to characterize the set of terms that one would want. We believe that our guidelines for annotators are a good first step in such a characterization.
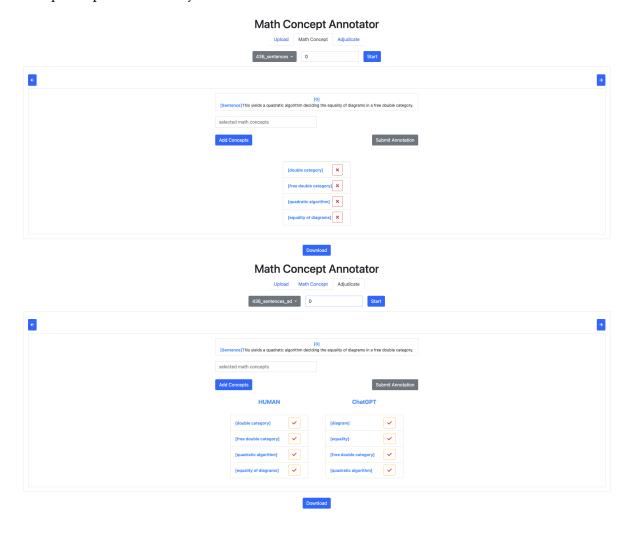
References

[1] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, 2020. arXiv:2005.14165.

[2] OpenAI, Introducing chatgpt, https://openai.com/blog/chatgpt, 2022.

[3] Y. Luan, L. He, M. Ostendorf, H. Hajishirzi, Multi-task identification of entities, relations, and coreferencefor scientific knowledge graph construction, in: Proc. Conf. Empirical Methods Natural Language Process. (EMNLP), 2018.

[4] D. Wadden, U. Wennberg, Y. Luan, H. Hajishirzi, Entity, relation, and event extraction with contextualized span representations, CoRR abs/1909.03546 (2019). URL: http://arxiv.org/abs/1909.03546. arXiv:1909.03546.

[5] J. Collard, V. de Paiva, B. Fong, E. Subrahmanian, Extracting mathematical concepts from text, in: Proceedings of the Eighth Workshop on Noisy User-generated Text (W-NUT 2022), Association for Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 15–23. URL: https://aclanthology.org/2022.wnut-1.2.

[6] J. Collard, V. de Paiva, E. Subrahmanian, Parmesan: mathematical concept extraction for education, 2023. arXiv:2307.06699.

[7] I. Augenstein, M. Das, S. Riedel, L. Vikraman, A. McCallum, SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 546–555. URL: https://aclanthology.org/S17-2091. doi:10.18653/v1/S17-2091.

[8] T. Asakura, A. Greiner-Petter, A. Aizawa, Y. Miyao, Towards grounding of formulae, in: Proceedings of the First Workshop on Scholarly Document Processing, Association for Computational Linguistics, Online, 2020, pp. 138–147. URL: https://aclanthology.org/2020.sdp-1.16. doi:10.18653/v1/2020.sdp-1.16.

[9] M. Kohlhase, A data model and encoding for a semantic, multilingual terminology of mathematics, in: S. M. Watt, J. H. Davenport, A. P. Sexton, P. Sojka, J. Urban (Eds.), Intelligent Computer Mathematics - International Conference, CICM 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings, volume 8543 of *Lecture Notes in Computer Science*, Springer, 2014, pp. 169–183. URL: https://doi.org/10.1007/978-3-319-08434-3_13. doi:10.1007/978-3-319-08434-3_13.

[10] J. Nivre, M.-C. de Marneffe, F. Ginter, J. Hajič, C. D. Manning, S. Pyysalo, S. Schuster, F. Tyers, D. Zeman, Universal Dependencies v2: An evergrowing multilingual treebank collection, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 4034–4043. URL: https://aclanthology.org/2020.lrec-1.497.

[11] Z. Chen, Q. Gao, L. S. Moss, NeuralLog: Natural language inference with joint neural and logical reasoning, in: Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics, Association for Computational Linguistics, Online, 2021, pp. 78–88. URL: https://aclanthology.org/2021.starsem-1.7. doi:10.18653/v1/2021.starsem-1.7.

A. Annotation platform

To help human annotators create eventual golden standards for the extraction of mathematical concepts, we repurposed a tool from the Natural Language Inference (NLI) field. The tool shows annotators a sentence (possibly) containing mathematical concepts, and annotators will cut-and-paste spans of text that they believe are mathematical concepts. If the annotator believes there are no mathematical concepts, they can simply skip to the next sentence. Annotators can also modify the span: they can change plurals to singluras, or add or delete words, etc.

Development of this tool is ongoing. But it can be used as-is with other experiments. It can compare the performance of human annotators or between humans and LLMs. We also have the possibility of "adjudicating" disagreements between annotations. In the near future we hope to expand the tool to add crowd-source interface, in order to allow mathematicians to "correct" machine annotations and to express opinions on tricky cases.

Figure 4: Overview of the annotation platform when an input sentence is selected.

The tool is availalble at https://gaoq111.github.io/math_concept_annotation/.