MathD2: Towards Disambiguation of Mathematical Terms

Shufan Jiang*[1, 2], Mary Ann Tan*[1, 2], and Harald Sack[1, 2]

[1]FIZ Karlsruhe – Leibniz Institute for Information Infrastructure,
Eggenstein-Leopoldshafen, Germany
[2]Karlsruhe Institute of Technology, Karlsruhe, Germany
{shufan.jiang, ann.tan, harald.sack}fiz-karlsruhe.de

Abstract

In mathematical literature, terms can have multiple meanings based on context. Manual term disambiguation across scholarly articles demands massive efforts from mathematicians. This paper addresses the challenge of automatically determining whether two or more definitions of a mathematical term are semantically different. Specifically, the difficulties of understanding how contextualized textual representation can help solve the problem are investigated. A new dataset MathD2 for mathematical term disambiguation is constructed with ProofWiki's disambiguation pages. Then three approaches based on contextualized textual representation are studied: (1) supervised classification based on the embeddings of concatenated definitions and titles; (2) zero-shot prediction based on semantic textual similarity (STS) between definition and title and (3) zero-shot LLM prompting. The first two approaches achieve accuracy greater than 0.9 on the ground truth dataset, demonstrating the effectiveness of our methods for automatic disambiguation of mathematical definitions. Our dataset and source code are available here: https://github.com/sufianj/MathTermDisambiguation.

1 Introduction

Mathematical scholarly articles contain highly structured statements, such as definitions, axioms, theorems, and proofs. Despite adhering to strict conventions and consistent usage of terminologies, these articles cannot be easily searched or explored through traditional keyword searches.

Mathematical definitions are rich sources of information. The terms defined therein known as *definienda* (singular: *definiendum*) can be automatically extracted. Extracted terms can be used to populate a knowledge base (KB), thereby facilitating knowledge discovery. In addition, these terms are utilized to index relevant mathematical statements and articles for efficient lookup.

To this end, several initiatives have emerged: Argot (Berlioz, 2021) is a collection of term-definition pairs automatically extracted from mathematical papers, allowing users to retrieve all definitions of a given term, while MathMex (Durgin et al., 2024) is a recent search engine for mathematical definitions based on the semantic similarity between a user's query and the definition. Both projects show promising usage of different word embeddings.

Existing research in this area focuses on automatically extracting mathematical definitions from scholarly articles (Berlioz, 2023; Nakagawa et al., 2004; Sun and Zhuge, 2023; Vanetik et al., 2020) and disambiguating definienda (Berlioz, 2021; Jiang and Senellart, 2023). Definienda disambiguation involves identifying and connecting terms to their corresponding mathematical definitions in a reference KB. It is particularly challenging when identical terms for the same concept are defined in various ways (e.g., "path") or when polysemous terms (e.g., "block") refer to distinct concepts in various subtopics (see Table 1). Argot cannot disambiguate polysemous terms, while MathMex cannot guarantee that the retrieved definitions accurately define the queried term.

For this study, ProofWiki[1] serves as the reference list. It is a crowd-sourced online collection of categorized mathematical proofs, including 500 disambiguation pages [2]. Similar to Wikipedia, these disambiguation pages list identical terms, each linking to a dedicated article. The heading of each article is composed of a unique title, appended by the category where the term can be found (e.g. algebra or

*These authors contributed equally to this work.

[1]https://ProofWiki.org/wiki/Main_Page

[2]ProofWiki Disambiguation Pages, https://proofwiki.org/wiki/Category:Definition_Disambiguation_Pages

Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025), pages 17–30

Definiendum	Definition in Source Article
block	A *block* in H is a maximal set of tightly-connected hyperedges. (Ergemlidze et al., 2019)
block	A *block* of indices is a set of numbers S where every term $SG_{a,b}(s)$ depends on the same value via division, for all $s \in S$. (Kupin, 2011)
path	If the vertices $v_0, v_1, \ldots, v_k$ of a walk W are distinct then W is called a *Path*. A path with n vertices will be denoted by P_n. P_n has length $n-1$. (Kalayathankal et al., 2015)
path	Let $G = (V, E)$ be a graph. A *path* in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. This is denoted by $P = (u = v_0, v_1 \ldots, v_k = v)$, where $(v_i, v_{i+1}) \in E$ for $0 \leq i \leq k-1$. (Perera and Mizoguchi, 2012)

Table 1: Definitions extracted from different scholarly articles (Jiang and Senellart, 2023). The definition of "path" has different formulations. The notion of "block" has different meanings.

geometry).

This work addresses the following research questions:

RQ1. How well can contextualized word embeddings help the disambiguation of mathematical terms?

RQ2. Which architectures and pre-training strategies are best suited for this task?

RQ3. How well do models trained in the preceding learning paradigm of pre-train + fine-tune compare with state-of-the-art (SOTA) *Instruction-Tuned* Large Language Models (LLMs)?

The main contributions of this work are:

- `MathD2` - a new dataset for **Math**ematical **D**efiniendum **D**isambiguation.
- Exploration of **three different approaches** demonstrating how the disambiguation task can benefit from contextualized semantic representations.
- **Experiment-supported evidence** highlighting the efficiency of sentence embeddings for the addressed disambiguation task.

2 Related Work

The challenges posed by this task are:

(a) the lack of labeled datasets for equivalent mathematical definitions – there is only one example for each definiendum and definition;

(b) the limited number of disambiguation pages;

(c) the unstructured nature of definitions that combine mathematical notations, formulas, and general discourse.

To address (a), entity linking and sentence similarity approaches for mathematical terms are reviewed. To tackle (b) and (c), transformer models (Vaswani et al., 2023) are employed for their capabilities to produce rich, contextualized representations.

Contextualized representations produced by BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019) encode the meaning of a word according to its context. This means that polysemous words have several, more accurate representations depending on their location in the sentence. BERT is pre-trained on two key tasks: Masked Language Modeling (MLM), where random tokens in a sentence are masked and predicted based on context, and Next Sentence Prediction (NSP), which trains BERT to determine whether a sentence follows another in a discourse. Pre-training with MLM is widely applied for domain adaptation, especially when there is a dearth of data for fine-tuning (Mishra et al., 2021; Jiang et al., 2022). In addition, fine-tuning BERT for specific downstream tasks and domains is straightforward. For instance, by combining BERT's output with a classification layer, it has been adapted for mathematical notation prediction (Jo et al., 2021), definiendum extraction (Jiang and Senellart, 2023) and mathematical statement extraction (Mishra et al., 2024). The Natural Language Inferernce (NLI) datasets (Bowman et al., 2015; Williams et al., 2018) used by BERT's NSP pre-training are related to the task at hand. A piece of supporting

evidence is AcroBERT (Chen et al., 2023), an entity linker that reuses BERT for NSP's pre-trained weights and is fine-tuned to link acronyms to their long forms. AcroBERT outperforms BERT and other domain-adapted BERT-based models.

However, the nature of the BERT's pre-training tasks makes it unsuitable for measuring semantic similarity. Sentence BERT (SBERT) [3] (Reimers and Gurevych, 2019) modifies the architecture of BERT to produce meaningful sentence embeddings that can be compared using cosine similarity. Out-of-the-box SBERT achieves superior performance across varied classification tasks involving mathematical texts (Steinfeldt and Mihaljević, 2024). In one such task, the proponents measure the similarity of SBERT embeddings between an input text and the combination of titles and abstracts of mathematical publications in arXiv [4] and zbMATH [5] to predict the classification code of the respective repositories. In the same vein, this study aims to evaluate the effectiveness of semantic textual similarity in linking definitions to titles. Since BERT for NSP and SBERT require different domain adaptation strategies (Reimers and Gurevych, 2019; Steinfeldt and Mihaljević, 2024), this work first identifies the architecture that performs better for the task.

Since the release of powerful LLMs, these models have been applied to various Information Extraction (IE) tasks, including entity linking. Particularly for long-tail entities, LLMAEL (Xin et al., 2024) instructed LLMs to augment the context by expanding entity mentions. The augmented context then serves as additional input to the entity disambiguation component of an IE pipeline (Xin et al., 2024). Meanwhile, (Vollmers et al., 2025) attempted to use LLMs in several IE pipeline components: first by prompting the LLM to identify entity mentions (NER), followed by context expansion using prompts reminiscent of LLMAEL's. Additional experiments conducted in this paper aim to find out the comparability of the proposed textual similarity models and LLMs in identifying another example of long-tail entities embodied by mathematical terms.

[3] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[4] https://arxiv.org/

[5] https://zbmath.org/

3 Methodology

Mathermatical term disambiguation is formalized as an entity linking task, where the entities refer to the unique article titles in ProofWiki. That is, given (1) a definition and an ambiguous definiendum and (2) a dictionary that maps the ambiguous definiendum to entities, the goal is to find the title that best matches the definition. The proposed method is described in two steps. First, the ground truth dataset is constructed. Second, three applicable approaches are considered.

3.1 Construction of the `MathD2` Dataset

A dump of the whole ProofWiki was extracted on the 5th of February, 2025, using WikiTeam (WikiTeam). This dump is then parsed with the disambiguation pages serving as a jump-off point for constructing the dictionary and the corpus used for training the proposed models.

The dictionary is composed of a list of terms and their corresponding candidate titles. Each term has a disambiguation page. This page contains links to associated articles, where each article is assigned a unique title.

The list of candidate titles for the dictionary is extracted from the hierarchical list of articles on each disambiguation page. It is important to note that not all articles appearing on a term's disambiguation page are automatically added as candidates for that term.

In addition, the hierarchy of topics is also taken into account when building the dictionary. More specific topics, or those belonging to the lower levels in the hierarchy, take precedence over higher level topics, when the former are also included in the latter's definition. The disambiguation page of "Equivalence" [6] illustrates this example: "Logical Equivalence" is not included in the candidate list of the term "Equivalence", since it is included already in the definitions of both "Semantic Equivalence" and "Provable Equivalence."

Aside from the hierarchy, the surface forms of the topics listed on the disambiguation page are also taken into account. Topics that do not include the term in question are not added as candidates (See "Set Theory" from the disambiguation page of the term, "Loop" [7].

[6] https://proofwiki.org/wiki/Definition:Equivalence

[7] https://proofwiki.org/wiki/Definition:Loop

Finally, terms mapped to less than two titles are removed. Table 2 shows (definition, title) pairs extracted from the disambiguation page of "Bilinear Form [8]."

The training corpus is extracted from the articles of the candidate titles. Only the *Definition* sections are utilized. They undergo post-processing which involves parsing of redirects and converting LaTeX content into plain text.

The `MathD2` dataset contains 365 ambiguous terms, mapping to 1984 *definition-title* pairs. For the fine-tuning in Section 3.2, the dataset is split for 5-fold cross validation as follows:

- 20% ambiguous terms and the corresponding *definition-title* pairs make a test set $test_{term}$. These terms are not seen in the training set, thereby testing model's ability to generalize on unseen ambiguous terms.
- of the 80% remaining ambiguous terms, the split between the training set and the second test set ($test_{title}$) is dependent on the number of (definition, title) pairs for each term. If a term has less than 8 (definition, title) pairs, all the pairs are are assigned to the training set. When the term has more than 8 definitions, the first 8 of those (definition, title) pairs are assigned to the training set, while the rest are assigned to $test_{title}$. Terms having not more than 8 definitions are automatically assigned to the training set. The purpose of $test_{title}$ is to evaluate the model's generalizability on new candidate titles to seen ambiguous terms.

The key difference between the two tests is that $test_{term}$ only contains unseen terms and the corresponding unseen candidate titles, while $test_{title}$ includes only seen terms and candidate titles not seen in the training set. Since there are more candidate titles per term on average in textittest$_{\text{title}}$, these terms are more ambiguous, making the test more difficult than $test_{term}$. This is reflected in the results shown on Table 4. In addition, inference on $test_{title}$ takes more time due to additional pairwise comparisons.

In the fine-tuning of Section 3.2, for each ambiguous term, two definitions and their titles are randomly selected to make positive pairs, and the titles of two other random definitions to make negative pairs. Table 3 shows the MathD2 dataset statistics. All approaches are evaluated on the 5 folds of 2 test sets.

[8] https://proofwiki.org/wiki/Definition:Bilinear_Form

3.2 Classification Based on One Concatenated Embedding

Following the finetuning setup of AcroBERT (Chen et al., 2023), BERT for NSP is adapted to build a supervised sentence pair classifier to link definitions to their page titles in ProofWiki. Every pair of (definition, candidate title with the matching ambiguous term in ProofWiki) is concatenated as an input sequence. The sequence begins with a [CLS] token, followed by a candidate title, a [SEP] token, and then the definition, ending with [SEP]. The input sequence passes through BERT's transformer layers. These layers produce contextual embeddings for each token in the sequence. Then, the embedding of [CLS] is fed into a softmax classification layer, which outputs a score to judge how coherent the concatenated sequence is. The pair with the highest score is selected as the final predicted output. First the out-of-box BERT for NSP serves as the baseline to see how well the pre-retained natural language inference model can describe the entailment between the titles and definitions. Then the pretrained BERT for NSP is finetuned with the training set using a triplet loss function

$$\mathcal{L} = \max\left\{0, \lambda - d_{\text{neg}} + d_{\text{pos}}\right\}$$

which aims to assign higher scores to the titles that match the input definition while reducing the scores of irrelevant candidates. $\lambda = 0.2$ is the margin value, and d_{pos} and d_{neg} are the distances for positive pairs (good matches of definition and title) and negative pairs (definition and irrelevant candidate title), respectively. This approach is implemented with PyTorch (Paszke et al., 2019) and transfomers (Wolf et al., 2020). The batch size is chosen among [8, 16, 32]. The learning rate is chosen among [1e-5, 2e-6] for Adam optimizer. The learning rate exponentially decays at a rate of 0.95 every 1000 steps. The model is trained with the training dataset for 200 epochs. After each epoch, a checkpoint (copy of the current model weights) is saved. Each checkpoint is then evaluated with the test dataset so that test data do not impact the model weights. The best evaluation scores are recorded in Appendix B.

Definition	Title
Let R be a ring. Let R_R denote the R-module R. Let M_R be an R-module. A bilinear form on M_R is a bilinear mapping $B : M_R \times M_R \to R_R$.	Definition: Bilinear Form (Linear Algebra)
A bilinear form is a linear form of order 2.	Definition: Bilinear Form (Polynomial Theory)

Table 2: Data extracted from a ProofWiki disambiguation page.

Fold	1	2	3	4	5
Train					
Term	292	292	292	292	292
Pairs	1153	1181	1181	1160	1169
Test$_{\text{term}}$					
Term	73	73	73	73	73
Pairs	412	362	342	445	423
Test$_{\text{title}}$					
Term	48	49	49	42	48
Pairs	419	441	461	379	392

Table 3: Cross-validation splits statistics. Terms in Test$_{\text{title}}$ sets are also in Train sets.

3.3 Zero-shot Prediction Based on Semantic Textual Similarity

A shortcoming of the previous solution is that the NSP inference has to be run for every (definition, title) pair mapped to an ambiguous term. Motivated to make a computationally more efficient solution, the sentence embeddings of the definitions and titles are explored. In this setup, the sentence embedding of the titles and the definitions only need to be calculated once. For the definition and each candidate title with the matching ambiguous term, the title with the highest cosine similarity to the embedding of the definition is selected as the final predicted output. To explore the potential benefits of different pretraining corpus and related tasks, the following models are studied:

- Out-of-box SBERT (`SBERT-all-mpnet-base-v2`) (Reimers and Gurevych, 2019).
- Mean pooled out-of-box BERT, to compare with the pretraining of SBERT.
- Mean pooled CC-BERT (Mishra et al., 2021), a from-scratch model pretrained with MLM on mathematical papers. This experiment studies the impact of domain-specific MLM pretraining and domain-specific tokenization, comparing to mean pooled out-of-box BERT.
- The best-performing sentence transformers for Semantic Textual Similarity(STS) tasks for short mathematical text as reported in (Steinfeldt and Mihaljević, 2024), including `Bert-MLM_arXiv-MP-class_zbMath` (Steinfeldt and Mihaljević, 2024) (noted as Adapted SBERT in Table 4), `SBERT-all-MiniLM-L6-v2` (Wang et al., 2020), and `SBERT-all-MiniLM-L12-v2` (Wang et al., 2020).

Following SBERT's default setting (Reimers and Gurevych, 2019), the mean pooling strategy is used to calculate the sentence embeddings with out-of-box BERT and CC-BERT.

3.4 LLM Prompting

Recently, Large Language Models (LLMs) have been incorporated to improve entity disambiguation tasks (Xin et al., 2024). Experiments conducted with LLMs are framed as a Zero-Shot Open Generative Question and Answer, where the LLM is instructed to identify the correct article title given a mathematical term and its ProofWiki definition as context.

In order to get the best results from the LLM, the prompt is constructed following best practices:

1. **Task Description.** "`Your task is to find the correct article title given a mathematical term and definition as context.`"
2. **Hallucination Prevention.** "`Reply with "I don't know" when uncertain.`"
3. **Expectation Setting.** "`Only select one answer from the provided list. Do not provide justifications.`"
4. **Multiple Choice.** "`Identify the correct title from this list:[...]`"

The LLMs used for testing are open-source and are

categorized as *Instruction-tuned* models (Zhang et al., 2024). These LLMs undergo a supervised fine-tuning step with a dataset consisting of human instructions paired with their desired generated outputs. The list of titles provided in the prompt are extracted from the dictionary mentioned in Section 3.1. Answers are only considered correct when the article title in the ground truth exists in full in the LLM's answer as in Example 1.

```
<s> [INST] Your task is to
find ...
Identify the correct
definition title from this
list: ...
[/INST] Indexing Set /
Term"</s>
```

Example 1: An Example of a Precise Response from `Mistral-7B-Instruct-v0.2`.

As can be seen in Example 2, there are instances when the LLM insists on providing lengthy justifications to its answer. Even when the text in the ground truth exists in the justification, this kind of answer will still be considered as incorrect.

```
<s> [INST] Your task is to
find ...
...following mathematical
definition as context:
Let G be a group...
[/INST] I don't know. The
term "complex" in the given
context refers to a subset
of a group...
```

Example 2: An Example of a response from `Mistral-7B-Instruct-v0.2` not following instructions.

The different LLMs used for the experiments are prompted with identical instructions. Inference call arguments, such as `max_tokens` or `temperature`, are adapted from the Hugging Face model card specifications pages.

4 Results and Discussion

4.1 Overall Performance

The evaluation measure used for comparison is *Accuracy* or micro F1-score (Equation 1) (Shen et al., 2015). Macro F1-score is not considered due to the characteristic of the test set, where there is only a single sample for each definition-title pair.

$$F1_{micro} = Acc = \frac{\text{\# correctly identified title}}{\text{\# of titles}} \quad (1)$$

Table 4 shows the experimental results of all three methods. Overall, the best-performing models are finetuned BERT for NSP, and generic SBERT-like models for STS. The differences between these models are not statistically significant (see Appendix B.1). Notably, the out-of-the-box SBERT demonstrated excellent performance with much less inference time.

Regarding the NSP approach, finetuned BERT on the MathD2 dataset significantly outperforms out-of-box BERT, validating AcroBERT's set-up, the informativeness of MathD2 data for finetuning, and the helpfulness of BERT's pretrained weights.

Regarding the STS approach, the performance of SBERT models is aligned with the results of (Reimers and Gurevych, 2019) and (Steinfeldt and Mihaljević, 2024). The experiments with the mean pooled out-of-box BERT and CC-BERT show that MLM domain-adaptation over mathematical papers slightly improves this task but is far less efficient than adapted SBERT, which has been pre-trained with fewer data but on a better task.

Given that both BERT for NSP and SBERT are pretrained on NLI tasks (Devlin et al., 2019; Reimers and Gurevych, 2019), it may be deduced that: i) Compared to using the [CLS] representation of concatenated sequences, using separated sentence embeddings captures more information for our task. ii) SBERT's pretraining on (title, abstract) pairs from S2ORC dataset (Lo et al., 2020) helps to better understand the entailment between titles and body texts. However, Bert-MLM_arXiv-MP-class_zbMath, the domain-adapted SBERT model [9] that the authors of (Steinfeldt and Mihaljević, 2024) fine-tuned with multiple tasks using titles and abstracts of mathematical papers does not yield better results. This might be due to the model being solely trained on titles and abstracts, diminishing the model's representational capacity for both formulas and general text.

In comparison, the results of the zero-shot experiments with LLMs are worse than those of the other

[9] https://huggingface.co/math-similarity/Bert-MLM_arXiv-MP-class_arXiv

Model	Approach	Test$_{term}$	Test$_{title}$
BERT (Devlin et al., 2019)	NSP	84.6	84.0
BERT(finetuned)	NSP	**92.3**	**91.6**
BERT(mean pooled)	STS	39.9	27.2
CC-BERT (Mishra et al., 2021)	STS	44.0	32.7
SBERT-all-mpnet-base-v2 (Reimers and Gurevych, 2019)	STS	**92.8**	**91.9**
SBERT-all-MiniLM-L6-v2	STS	91.6	**91.4**
SBERT-all-MiniLM-L12-v2	STS	**92.6**	**91.4**
Adapted SBERT (Steinfeldt and Mihaljević, 2024)	STS	59.8	48.4
Mistral-7B-Instruct-v0.2 (Jiang et al., 2023)	LLM-Instruct	45.9	50.4
Mistral-7B-Instruct-v0.3	LLM-Instruct	60.0	52.1
Meta-Llama-3-8B-Instruct	LLM-Instruct	75.0	71.9

Table 4: Averaged accuracy scores of five tests. Values are reported as $\rho \cdot 100$. Best scores are in bold. Detailed results and pairwise t-statistics can be found in Appendix B.

approaches. When running the experiments on older GPUs, some samples caused out-of-memory runtime errors due to the lengthy ProofWiki definition sections. For example, the definition section of *Matrix Product* [10] have matrices within it which could have caused the error. One solution is to limit the maximum token size during inference to 255. However, this curtails contexts that may help the model disambiguate highly ambiguous terms.

4.2 Errors Generated by LLMs

In order to understand the types of errors encountered by LLMs, all responses from the $\texttt{test}_{\texttt{term}}$ split that are considered incorrect are manually scrutinized. These amounted to almost a quarter of $\texttt{test}_{\texttt{term}}$.

Appendix A provides examples of each category of errors. Erroneous LLM responses are of the following types:

1. **No Prediction (NP)**. This is when the LLM responds with *"I don't know."*

2. **Not Following Instructions (NFI).** These are scenarios when the LLM chose answers not included in the list of choices or when the answer is in the justification.

3. **Learning Bias (LB).** This is when the LLM 's answer is closest to the ground truth (e.g. "Degrees of Arc" instead of "Degree of Arc."). NFIs and LBs are often hard distinguish. As a rule of thumb, an error is considered an NFI, when the LLMs try to change the categorical structure of the titles into prose (e.g. "Right Distributive Operation" instead of "Distributive Operation/Right", as provided in the list of choices).

4. **Wrong Prediction (WP).** These errors are easy to distinguish. In most cases, the incorrect answers are included in the list of candidates.

[10] https://proofwiki.org/wiki/Definition:Matrix_Product

Error Type	Mistral v2	Mistral v3	Llama v3.1
NP	20.6	0.0	2.2
NFI	169.8	126.0	73.4
LB	3.2	1.0	0.4
WP	21.0	33.4	23.2

Table 5: Average Number of Errors per Type Produced by LLMs on 5 $\texttt{test}_{\texttt{term}}$ sets. NP = No Prediction, NFI = Not Following Instructions , LB = Learning Bias, WP = Wrong Prediction. Detailed error distribution is given in Appendix B.1.

Existing literature points to the tendencies of LLMs to hallucinate (Huang et al., 2025). Among the aforementioned error types, NFIs and LBs errors exhibit this behavior. Instead of admitting uncertainty or the lack of knowledge, these errors show that the model regresses to making up answers. Our experimental results also show that when the number of candidates increases, Mistral models are more likely to produce NFI errors (see Figure B.1 and Figure B.2 in Appendix B.1), and the correct rate decreases correspondingly.

Table 5 shows that older models, such as `Mistral-7B-Instruct-v0.2`, are likely not to know the answer with the highest number NPs and

not follow explicit instructions (NFIs). Compared to its predecessor, `Mistral-7B-Instruct-v0.3` did not abstain from making predictions (0% NPs) but produced more wrong predictions. Not surprisingly, it is more likely to follow instructions than its predecessor. While the best performing model is `Meta-Llama-3-8B-Instruct` with considerable fewer errors across the board.

4.3 Limitations

An interesting finding is that all three approaches make some common mistakes, indicating the limits of using only semantic representations. The most common error is when the definition statement includes nested definitions. Another typical error is that the predicted result is in the correct category but not the definiendum, mainly when the definition contains morphemes in the predicted title or when the definition does not contain some morphemes in the expected title. For example, the definition of "Consequence Function" starts with "Let $\mathbf{G}$ be a game..." [11] , and the predicted title is "Definition:Consequence(Game Theory)' [12] . Thus, enhancing sentence embedding's comprehension of semantic and syntactic knowledge of mathematical definitions is still worth investigating. Other common mistakes reveal the noises in the dataset due to errors in Proofwiki [13], or automatic scrapping and LaTeXconversion of irregular ProofWiki pages.

Practical Considerations: One reason for comparing traditional transformer-based model paradigm of Pre-train+Fine-Tune and Large Language Models is the consideration of computing resource constraints. SOTA LLMs, such as `Meta-Llama-3-8B-Instruct`, require Cuda libraries with version 12.0 (Nvidia, 2024).

Experiments involving BERT/SBERT-based models are conducted on NVIDIA Tesla V100S-PCIE 32GB having compute capability of 5.0 with 14.5 TFLOPS [14]. On the other hand, experiments with LLMs used NVIDIA A100 80GB PCIe with with 19 TFLOPS, belonging to a line of Graphics Processing Units (GPUs) with compute capability of 7.0.

Compute capability dictates how much computing resources are required to run experiments. Newer LLMs require higher versions of Cuda. Cuda libraries require a specific version of NVIDIA drivers, and consequently, the array of GPUs capable of running the driver version.

5 Conclusion and Future Works

This work introduces `MathD2`, a new dataset for mathematical term disambiguation extracted from ProofWiki. Two entity-linking approaches have been implemented and shown to yield advantages in the usage of contextualized embeddings to differentiate mathematical definitions. The experimental results proved the efficiency and effectiveness of using out-of-the-box SBERT.

Additional experiments with SOTA LLMs also show that the proposed models performed better and have fewer computing resource constraints. Moreover, error analysis shows the inherent tendency of LLMs to hallucinate.

Further work is planned on applying the proposed approaches to scholarly papers. Regarding the closed scores of the best models, evaluation with more data and significance tests are planned. In addition, the current approach is to be extended to include document-level representation and citation information to differentiate definitions in scholarly papers. This work also indicates the need for further study on building sentence transformers that benefit from domain-specific MLM and task-related pre-training.

6 Acknowledgements

We would like to express our gratitude to Frank Pöhlmann who dedicated his time and editing experience to provide valuable feedback and constructive criticism on this paper.

[11] https://proofwiki.org/wiki/Definition:Consequence_Function

[12] https://proofwiki.org/wiki/Definition:Consequence_(Game_Theory)

[13] For example, the definiendum in https://proofwiki.org/wiki/Definition:Ideal_of_Algebra/Right_Ideal should be *right ideal*, but is wrongly written as *left ideal*.

[14] TeraFLOPS specifies the number of floating point operations per second that the hardware can accomplish.

References

Luis Berlioz. 2021. ArGoT: A Glossary of Terms extracted from the arXiv. *Electronic Proceedings in Theoretical Computer Science*, 342:14–21.

Luis Berlioz. 2023. *Hierarchical Representations from Large Mathematical Corpora*. Ph.D. thesis, University of Pittsburgh.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Lihu Chen, Gael Varoquaux, and Fabian M. Suchanek. 2023. GLADIS: A general and large acronym disambiguation benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2073–2088, Dubrovnik, Croatia.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shea Durgin, James Gore, and Behrooz Mansouri. 2024. Mathmex: Search engine for math definitions. In *European Conference on Information Retrieval*, pages 194–199. Springer.

Beka Ergemlidze, Ervin Győri, and Abhishek Methuku. 2019. 3-uniform hypergraphs without a cycle of length five. *arXiv preprint arXiv:1902.06257*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Shufan Jiang, Rafael Angarita, Stéphane Cormier, Julien Orensanz, and Francis Rousseaux. 2022. Choubert: Pretraining french language model for crowdsensing with tweets in phytosanitary context. In *International Conference on Research Challenges in Information Science*, pages 653–661. Springer.

Shufan Jiang and Pierre Senellart. 2023. Extracting definienda in mathematical scholarly articles with transformers. *IJCNLP-AACL 2023*, page 31.

Hwiyeol Jo, Dongyeop Kang, Andrew Head, and Marti A Hearst. 2021. Modeling mathematical notation semantics in academic papers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3102–3115.

Sunny Joseph Kalayathankal et al. 2015. Operations on covering numbers of certain graph classes. *arXiv preprint arXiv:1506.03251*.

Elizabeth Kupin. 2011. Subtraction division games. *arXiv preprint arXiv:1201.0171*.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Shrey Mishra, Yacine Brihmouche, Theo Delemazure, Antoine Gauquier, and Pierre Senellart. 2024. First steps in building a knowledge base of mathematical results. In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, pages 165–174.

Shrey Mishra, Lucas Pluvinage, and Pierre Senellart. 2021. Towards extraction of theorems and proofs in scholarly articles. In *DocEng '21: ACM Symposium on Document Engineering 2021, Limerick, Ireland, August 24-27, 2021*, pages 25:1–25:4. ACM.

Koji Nakagawa, Akihiro Nomura, and Masakazu Suzuki. 2004. Extraction of Logical Structure from Articles in Mathematics. In Andrea Asperti, Grzegorz Bancerek, and Andrzej Trybulec, editors, *Mathematical Knowledge Management*, volume 3119, pages 276–289. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Lecture Notes in Computer Science.

Nvidia. 2024. Cuda12 support for v100 gpu.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

KKKR Perera and Yoshihiro Mizoguchi. 2012. Bipartition of graphs based on the normalized cut and spectral methods. *arXiv preprint arXiv:1210.7253*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460.

Christian Steinfeldt and Helena Mihaljević. 2024. Evaluation and domain adaptation of similarity models for short mathematical texts. In *International Conference on Intelligent Computer Mathematics*, pages 241–260. Springer.

Yutian Sun and Hai Zhuge. 2023. Discovering patterns

of definitions and methods from scientific documents. *arXiv preprint arXiv:2307.01216*.

Natalia Vanetik, Marina Litvak, Sergey Shevchuk, and Lior Reznik. 2020. Automated discovery of mathematical definitions in text. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2086–2094.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Daniel Vollmers, Hamada Zahera, Diego Moussallem, and Axel-Cyrille Ngonga Ngomo. 2025. Contextual augmentation for entity linking using large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8535–8545, Abu Dhabi, UAE. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

WikiTeam. Wikiteam. Original-date: 2014-06-25T10:18:03Z.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Amy Xin, Yunjia Qi, Zijun Yao, Fangwei Zhu, Kaisheng Zeng, Xu Bin, Lei Hou, and Juanzi Li. 2024. Llmael: Large language models are good context augmenters for entity linking. *Preprint*, arXiv:2407.04020.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Appendices

A Error Analysis of LLMs' Response

Below are examples of actual LLM answers according to the type of error specified in Section 4.

1. No Prediction (NP).
   $\texttt{Test}_{\texttt{term-idx}}$: 269
   Ground Truth: Composition of Ratio
   Answer: I don't know

2. Not Following Instructions (NFI).
   $\texttt{Test}_{\texttt{term-idx}}$: 51
   Context: Identify the correct definition title from this list: ['Image (Relation Theory)/Mapping/Mapping', 'Image (Relation Theory)/Relation/Relation', 'Direct Image Mapping/Mapping', 'Direct Image Mapping/Relation', 'Direct Image of Sheaf']
   Ground Truth: Direct Image of Sheaf
   Answer: Direct Image Mapping/Sheaf

3. Learning Bias (LB).
   $\texttt{Test}_{\texttt{term-idx}}$: 338
   Ground Truth: Cut-Vertex
   Answer: Vertex Cut

4. Wrong Prediction (WP).
   $\texttt{Test}_{\texttt{term-idx}}$: 2
   Context: Complex analysis is a branch of mathematics that studies complex functions.
   Ground Truth: Analysis/Complex
   Answer: Complex Function

B Detailed Results

Table 6 and Table 7 show the 5-fold cross-validation accuracy scores.

B.1 Comparing Performance of Models

Table 8 and Table 9 compare models with close scores in Table 6 and Table 7. Paired Student's t-test is used to determine if one model is significantly better than another. Given $n = 5$ folds, let d_i represent the difference in accuracy between Model A and Model B for the i-th fold:

$$d_i = \text{Accuracy}_A^{(i)} - \text{Accuracy}_B^{(i)}, \quad i = 1, 2, \ldots, 5$$

Mean difference

$$\bar{d} = \frac{1}{n} \sum_{i=1}^{n} d_i$$

Sample standard deviation

$$s_d = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (d_i - \bar{d})^2}$$

Standard Error

$$\text{SE} = \frac{s_d}{\sqrt{n}}$$

t-statistic

$$t = \frac{\bar{d}}{\text{SE}} = \frac{\bar{d}}{s_d/\sqrt{n}}$$

Degrees of Freedom DF $= n - 1 = 4$

Two-Tailed p-value

$$p\text{-value} = 2 \cdot P(T \geq |t|) \quad \text{where } T \sim t_{\text{DF}=4}$$

We consider the difference between the performance of two ML models to be statistically significant if p-value is smaller than 0.05.

Model	**Approach**	**Test$_{term}$ 1**	**Test$_{term}$ 2**	**Test$_{term}$ 3**	**Test$_{term}$ 4**	**Test$_{term}$ 5**
BERT	NSP	0.799	0.873	0.868	0.845	0.844
BERT (finetuned)	NSP	0.903	**0.972**	**0.927**	0.910	0.903
BERT (mean pooled)	STS	0.381	0.434	0.453	0.369	0.359
CC-BERT (mean pooled)	STS	0.427	0.448	0.462	0.434	0.430
SBERT-all-mpnet-base-v2	STS	**0.923**	0.936	0.918	0.928	**0.934**
SBERT-all-MiniLM-L6-v2	STS	0.871	0.923	**0.927**	0.935	0.922
SBERT-all-MiniLM-L12-v2	STS	0.893	0.939	0.921	**0.942**	**0.934**
Adapted SBERT	STS	0.568	0.655	0.611	0.580	0.577
Mistral-7B-Instruct-v0.2	LLM	0.483	0.506	0.421	0.456	0.430
Mistral-7B-Instruct-v0.3	LLM	0.619	0.635	0.667	0.526	0.556
Meta-Llama-3-8B-Instruct	LLM	0.731	0.815	0.719	0.780	0.707

Table 6: Accuracy scores on new terms. The best scores are in bold.

Model	**Approach**	**Test$_{title}$ 1**	**Test$_{title}$ 2**	**Test$_{title}$ 3**	**Test$_{title}$ 4**	**Test$_{title}$ 5**
BERT	NSP	0.847	0.823	0.833	0.850	0.847
BERT (finetuned)	NSP	0.926	**0.927**	0.911	0.900	**0.918**
BERT (mean pooled)	STS	0.258	0.274	0.260	0.290	0.278
CC-BERT (mean pooled)	STS	0.329	0.336	0.315	0.319	0.337
SBERT-all-mpnet-base-v2	STS	0.896	0.923	**0.928**	**0.934**	0.916
SBERT-all-MiniLM-L6-v2	STS	0.924	0.909	0.911	0.910	0.916
SBERT-all-MiniLM-L12-v2	STS	**0.928**	0.902	0.915	0.913	0.911
Adapted SBERT	STS	0.494	0.485	0.479	0.472	0.487
Mistral-7B-Instruct-v0.2	LLM	0.492	0.499	0.495	0.533	0.503
Mistral-7B-Instruct-v0.3	LLM	0.506	0.522	0.505	0.549	0.523
Meta-Llama-3-8B-Instruct	LLM	0.747	0.703	0.709	0.683	0.753

Table 7: Accuracy scores on new titles. The best scores are in bold.

Model 1	**Model 2**	**t-statistic t**	**p-value p**	**Significant**
SBERT-all-mpnet-base-v2	BERT (finetuned)	0.405	0.706	no
BERT (finetuned)	SBERT-all-MiniLM-L12-v2	-0.222	0.835	no
SBERT-all-MiniLM-L12-v2	SBERT-all-MiniLM-L6-v2	2.160	0.097	no
BERT (finetuned)	BERT	7.637	0.002	yes
BERT (mean pooled)	CC-BERT (mean pooled)	-3.197	0.033	yes
Mistral-7B-Instruct-v0.3	Mistral-7B-Instruct-v0.2	-4.928	0.008	yes

Table 8: Comparing models on new terms. Statistical significance: $p < 0.05$

.

Model 1	Model 2	t-statistic t	p-value p	Significant
SBERT-all-mpnet-base-v2	BERT (finetuned)	0.272	0.819	no
BERT (finetuned)	SBERT-all-MiniLM-L12-v2	0.377	0.753	no
SBERT-all-MiniLM-L12-v2	SBERT-all-MiniLM-L6-v2	-0.013	0.992	no
BERT (finetuned)	BERT	8.921	0.001	yes
BERT (mean pooled)	CC-BERT (mean pooled)	-7.759	0.002	yes
Mistral-7B-Instruct-v0.3	Mistral-7B-Instruct-v0.2	-7.948	0.002	yes

Table 9: Comparing models on new titles. Statistical significance: $p < 0.05$

.

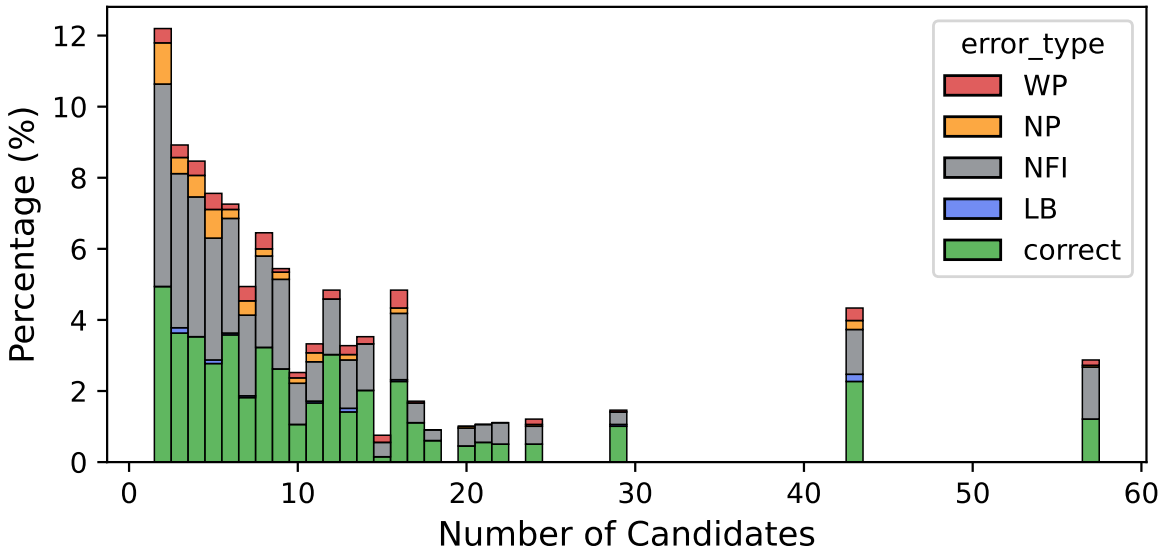

Figure B.1: Error Type Distribution by Candidate Number - mistralv2. NP = No Prediction, NFI = Not Following Instructions , LB = Learning Bias, WP = Wrong Prediction. The proportion of grey in a bar grows when the number of candidates increases, suggesting that NFI is more likely to happen when given more options.

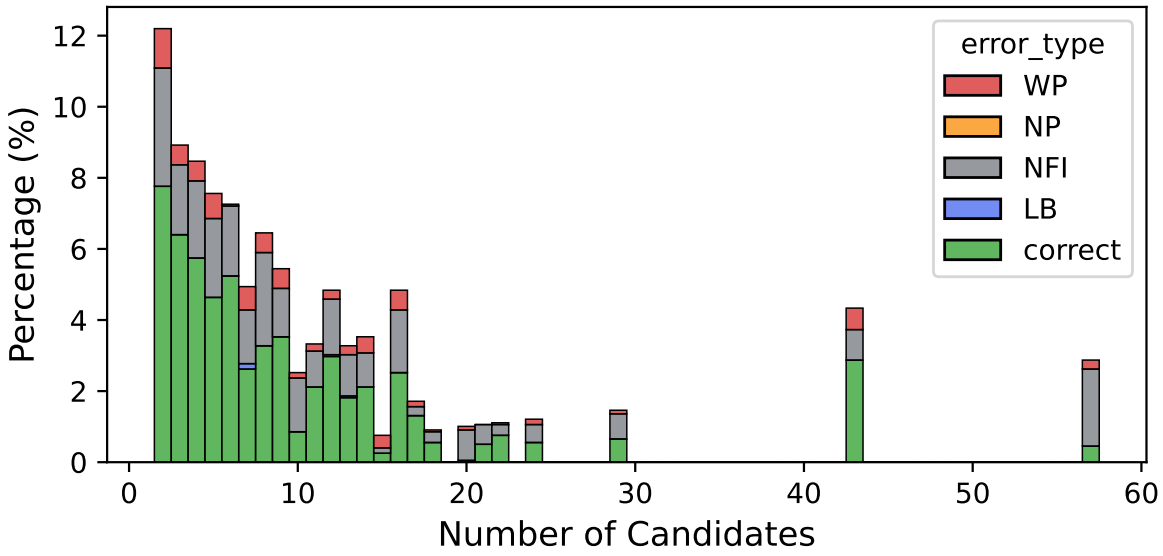

Figure B.2: Error Type Distribution by Candidate Number - mistralv3. NP = No Prediction, NFI = Not Following Instructions , LB = Learning Bias, WP = Wrong Prediction. The proportion of grey in a bar grows when the number of candiadtes increases, suggesting that NFI is more likely to happen when given more options.

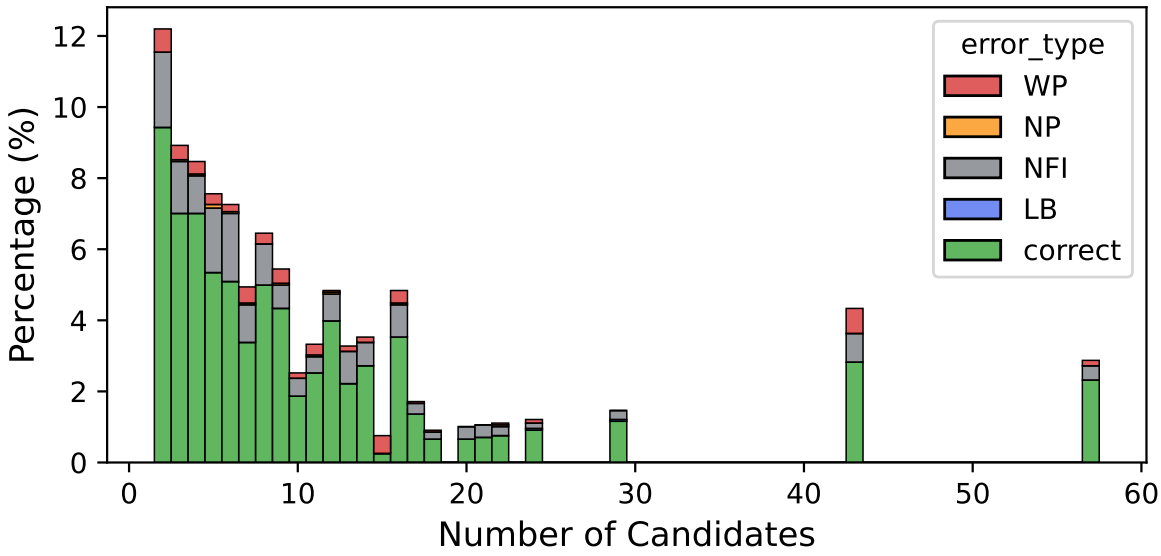

Figure B.3: Error Type Distribution by Candidate Number - llama3. NP = No Prediction, NFI = Not Following Instructions , LB = Learning Bias, WP = Wrong Prediction.