# Contextualized Formula Search Using Abstract Meaning Representation

Behrooz Mansouri
Rochester Institute of Technology
Rochester, NY, USA
bm3302@rit.edu

Douglas W. Oard
University of Maryland
College Park, MD, USA
oard@umd.edu

Richard Zanibbi
Rochester Institute of Technology
Rochester, NY, USA
rxzvcs@rit.edu

## ABSTRACT

In math formula search, relevance is determined not only by the similarity of formulas in isolation, but also by their surrounding context. We introduce MathAMR, a new unified representation for sentences containing math. MathAMR generalizes Abstract Meaning Representation (AMR) graphs to include math formula operations and arguments. We then use Sentence-BERT to embed linearized MathAMR graphs for use in formula retrieval. In our first experiment, we compare MathAMR against raw text using the same formula representation (Operator Trees), and find that MathAMR produces more effective rankings. We then apply our MathAMR embeddings to reranking runs from the ARQMath-2 formula retrieval task, where in most cases effectiveness measures are improved. The strongest reranked run matches the best P′@10 for an original run, and exceeds the original runs in nDCG′@10.

## KEYWORDS

Abstract Meaning Representation, Math IR, Formula Search

**ACM Reference Format:**
Behrooz Mansouri, Douglas W. Oard, and Richard Zanibbi. 2022. Contextualized Formula Search Using Abstract Meaning Representation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557567

## 1 INTRODUCTION

In *contextualized* math formula search, a formula selected from a document is used as a query, and formulas in documents are returned. For example, a formula might be selected from a question post on a Community Question Answering (CQA) site, and then issued as a query to find posts providing information about the formula and/or the formula's associated question. The relevance of retrieved formulas is defined by both formula similarity *and* the interpretation of formulas where they appear.

For retrieval, using context can be helpful in two ways. First, context features can improve recall through characterizing a formula's meaning, or a setting where it might be employed. Second, context can improve precision through including constraints, such as on the range, domain, or types of variables [22]. Given this, we expect that formula search systems exploiting both a formula and its context yield better results than systems that ignore formula context. In this paper, we show results consistent with that hypothesis.

As we explain in Section 2, mathematical notation is hierarchical, and represented naturally by trees. In this paper we represent formulas using Operator Trees (OPTs), which are widely used for computation and isolated formula search [7, 13, 36]. Text by contrast is expressed linearly, but the latent meaning of text is complex, and graphs have been devised to encode aspects of that meaning. We consider one semantic encoding for text, the Abstract Meaning Representation (AMR) graph.

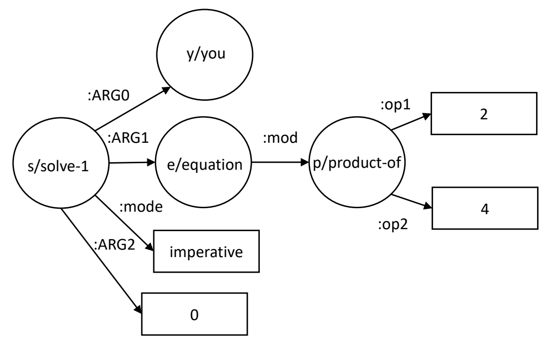



**Figure 1: AMR with incorrect formula representation for "Solve the equation $x^2 - 4 = 0$." A method for avoiding such errors is shown in Figure 3.**

Unfortunately, current AMR parsers do not interpret formulas correctly, as illustrated for a BART-based parser[1] in Figure 1. To address this, we unify AMR and OPT representations to capture the meaning of a formula in its context, which we call *MathAMR*.[2] Using the ARQMath Formula Retrieval task (Task 2 [22]), we show that reranking using MathAMR representations improves effectiveness for both state-of-the-art isolated formula search techniques, and retrieval models that fuse text and formula retrieval results.

## 2 RELATED WORK

Abstract Meaning Representation (AMR) is a semantic encoding for text introduced in 1998 by Langlade and Knight in the Nitrogen system [15] to map meanings onto word lattices. Banarescu et al. [2] later used PropBank [4] notation, and defined AMR annotations as rooted Directed Acyclic Graphs (DAGs) with nodes representing core concepts as either words (typically, adjectives or stemmed nouns and adverbs), or frames extracted from Propbank. Labeled directed edges represent semantic relationships (see Figure 1).

We consider three categories of AMR parser:

(1) **Graph-based AMR parser.** Construct AMR graphs by searching for Maximum Spanning Connected Subgraphs (MSCGs) from an edge-labeled directed graph of all possible relations. The first AMR parser was of this type (JAMR [8], in 2014).
(2) **Dependency-based AMR parser.** Parsers such as CAMR [31] generate a dependency parse for a sentence, and then transform it into an AMR graph using transition rules.

[1] `xmf-bart` model from the `amrlib` python library
[2] MathAMR source code: https://github.com/BehroozMansouri/MathAMR

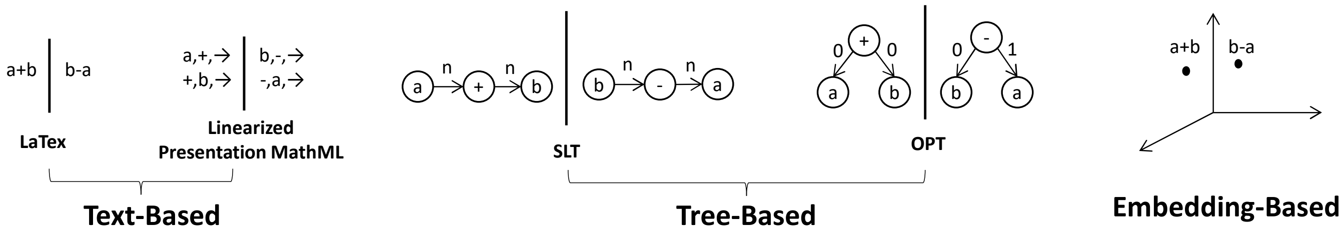



**Figure 2: Text, tree, and embedding-based formula search approaches using $a + b$ (left side) and $b - a$ (right side) as examples.**

(3) **Neural AMR parser.** Viewing the problem as a machine translation task, these models directly convert raw text into linearized AMR representations. For example, the SPRING parser [3] translates text to a depth-first linearization of AMR graphs. SPRING leverages a BART transformer model [16] by modifying its tokenizer to handle AMR tokens.

AMRs are commonly used for tasks such as summarization [17, 18], question answering [12, 32], and information extraction [10, 35]. For example, Liu et al. [18] summarized text by generating AMRs for individual sentences in a document, and then merging the AMRs by collapsing named and date entities. Next, a summary sub-graph was generated using integer linear programming, after which summary text was generated from that sub-graph using the JAMR graph-based AMR parser [8].

**Formula Representation.** Two tree representations are commonly used for formula search: Symbol Layout Tree (SLT) and Operator Tree (OPT). In SLT, nodes are formula elements (e.g., symbols, roots, fraction lines) and edge labels indicate spatial relationships. In OPT, edge labels show operator argument (operand) order. For an ordered operation such as subtraction (e.g., 'b - a'), the first operand is the initial value, and the second is the quantity to subtract. For commutative operators such as '+', edge labels are identical. In the middle of Figure 2, the SLT and OPT for $a + b$ and $a - b$ are shown. In SLT, because the formula elements are located horizontally adjacent to one another, the edge label 'n' ('next') is used. For OPT, '+' is commutative, so both edge labels are '0'.

**Formula Search.** Depending upon a user's goals, relevance for retrieved formulas may be defined differently. For example, a user might search for a formula to see what other applications it has, in which case exact matches are highly relevant. Exact matches were also considered highly relevant for the formula browsing task in NTCIR-12 [33], where relevance was decided by comparing formulas in isolation. As a different example, a user may issue a formula query to find formulas for accomplishing a specific task. For instance, in ARQMath-2 [22] formula queries were taken from math question posts in Math Stack Exchange, and the relevance of returned formulas was decided by whether material associated with a formula is likely to help answer the original question.

There have been three main approaches to formula search. Figure 2 illustrates how two formulas are compared in each approach.

(1) **Text-based models.** LaTeX or linearized MathML representations used with traditional text retrieval models such as TF-IDF (e.g., MIaS [30]). Text-based approaches lose the hierarchical structure of formulas, and may fail to characterize formula structure well.
(2) **Tree-based models.** SLT and/or OPT representations are compared using subtrees and/or paths. For example, Approach0 [36] retrieves formulas using paths from operator trees generated by parsing LaTeX with a relatively small expression grammar. Candidates are scored using up to three best-matching sub-trees.
(3) **Embedding-based models.** Formulas are represented by $d$-dimensional vectors, and vector similarity measures (e.g., cosine) are used for ranking. Embedding models such as Tangent-CFT [20] ignore surrounding text, while in MathBERT [26] each formula is represented by its tokens, associated text, and OPT representation.

In addition to MathBERT, there are other approaches that also take advantage of text near a formula. Ng et al. [25] combined retrieval results from Tangent-L [9] and BM25+. In the work of Krstovski et al. [14], equation embeddings generated unified representations by linearizing formulas as tuples, and then treating them as tokens in the text. These embeddings used context windows around formulas and applied a word embedding model [24] to build vector representations for formulas.

In this work, we introduce a new unified representation of math and text, MathAMR, and apply it to contextualized formula search.

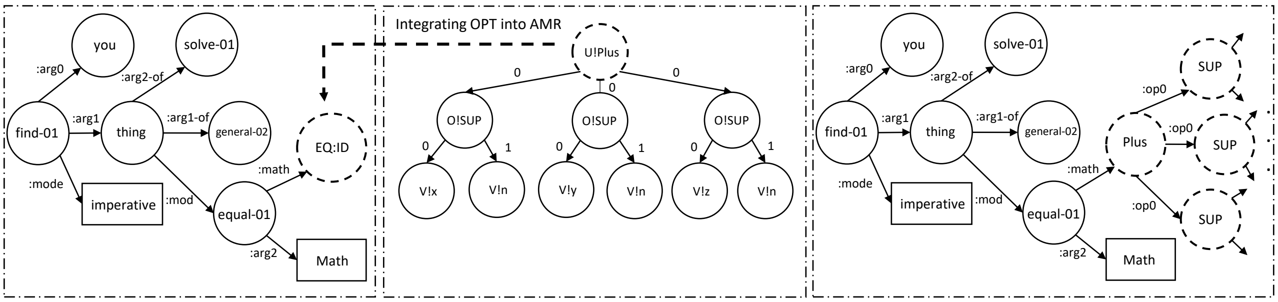
## 3 MATHAMR

The application of operations represented by an operator tree is very similar to how text semantics are represented by Abstract Meaning Representation (AMR) graphs, which can roughly be understood as representing "who is doing what to whom."[3] AMR graphs and operator trees also share similar edge labelings: in AMR 'opx' edge labels indicate node ordering, where 'x' is an integer enumerator (e.g., `op0`, `op1`).

Unfortunately, as we saw in Figure 1, current AMR parsers are not able to interpret formulas correctly. For some domains (e.g., biomedical), there are specialized pre-trained AMR parsers [23], but not for mathematical text. To address this, we introduce MathAMR, in which math formulas are represented using operator trees. Figure 3 uses an example topic to illustrate the steps used to generate MathAMR, which we describe below.

(1) **Select the context window.** As AMR was designed for sentences, we use Spacy[4] to split paragraphs into sentences and choose the sentence the formula appears in. It is common to see sentence punctuation inside formula regions, so before

[3]https://github.com/amrisi/amr-guidelines/blob/master/amr.md#part-i-introduction
[4]https://spacy.io/




(a) Abstract Meaning Representation (b) Operator Tree (c) Math Abstract Meaning Representation

**Figure 3: Generating MathAMR for "Find $x^n + y^n + z^n$ general solution" (ARQMath-2 topic B.289). (a) AMR tree with formula replaced by formula id (b) OPT formula representation. (c) OPT root replaces formula id. Part of the OPT is not shown in (c).**

using Spacy we move any punctuation (. , ! ?) from the end of formula regions to after the final formula delimiter. For example, in LaTeX $a+b=c.$ becomes $a+b=c$.

(2) **Replace formulas by identifier tokens.** To avoid AMR parsing problems, we replace each formula in the input sentence by a single token. For the example shown in Figure 3, the LaTeX formula $x^n + y^n + z^n$ is replaced by `EQ:ID`, where ID is an integer identifier (for this example, EQ:766).
(3) **Generate AMR graph.** We use an embedding-based AMR parser to generate the sentence's AMR graph, the AMRLib[5] "model_parse_xfm_bart_large" model [16]. We introduce a new edge label 'math', used to connect a formula's placeholder node to its parent (see Figure 3 (a)).
(4) **Replace formula token by formula OPT.** AMR formula id nodes are replaced by the corresponding root nodes of formula OPTs, creating the *MathAMR* graph. This is shown in Figure 3 (c) for the OPT shown in Figure 3 (b). To follow AMR conventions, OPT edge labels are renamed from numbers to 'opX', where 'X' is the integer from the original OPT.

This is our first attempt at a unifying text and formulas in AMR graphs, so we have kept our model simple. For example, we use only OPT formula representations, even though prior research has shown that also using SLT representations can be helpful [20, 21].

# 4 MATHAMR SENTENCE-BERT EMBEDDINGS

For retrieval, we embed linearized MathAMR graphs. MathAMR graphs are linearized using a depth-first traversal, ignoring edge labels for simplicity. For Figure 3(c) using the full OPT from Figure 3(b), the linearized MathAMR string is:

```
find-01 you thing general-02 solve-01 equal-01
math plus SUP z n SUP y n SUP x n imperative
```

MathAMR strings are embedded using Sentence-BERT [27], and retrieval performed using Sentence-BERT's cosine similarity implementation. To train Sentence-BERT models, we used the pre-trained `all-distilroberta-v1` model, with ARQMath-1 [34] topics and the training topics from ARQMath-2 [22] used for fine-tuning.

Formula search results in ARQMath are scored as high, medium, low, or not-relevant. For training, we assigned a relevance score of 1 for high and medium, 0.5 for low, and 0 for non-relevant. Training data contained triplets of the form: `(query formula, candidate formula, relevance score)`.

For training, we employ a model and multi-task learning framework in Sentence-BERT [6] used previously to detect duplicate Quora questions by first minimizing the distance between positive pairs and maximizing the distance between negative pairs using a contrastive loss function [5]. Then, the multiple negatives ranking loss function [11] is used, which considers only positive pairs, minimizing the distance between positive pairs out of a large set of possible candidates, making it well-suited to ranking tasks.

During training, in each epoch we compute the Spearman correlation between the embedding cosine similarity and the label score on the validation set. After training the model with a fixed number of epochs, the model with the lowest validation loss is selected.

# 5 EXPERIMENTS

**Test Collection and Training Data.** For training we use all 74 assessed ARQMath-1 topics, along with the 12 training topics from ARQMath-2. This provides a total of 21,411 training triples of the form described in the previous section.

To create a validation set, we separated the triples for each relevance rating (high, medium, low, non-relevant) into sets, and then divided each triple set randomly, using a 90%/10% split. The 10% splits were then combined and used as our validation set during training. This validation set contained 2,160 triples, with 535 having *high* relevance, 142 *medium* relevance, 141 *low* relevance, and 1,342 *non-relevant.* The remaining 90% of triples from each relevance level comprised our training set, with relevance degree distribution: { high: 531, medium: 134, low: 134, non-relevant: 1340 }.

Our results are reported on the 58 ARQMath-2 test topics that were not used for any training purposes.

**Sentence-BERT parameters.** Sentence-BERT was trained for 50 epochs, choosing the model with an epoch obtaining the minimum training loss on a validation set. For candidate formulas, the average token length in linearized MathAMR strings was 53.2 ($\sigma$ = 52.8) and for RawText with OPT was 96.3 ($\sigma$ = 141.6). As a result, we used batch size 16 and maximum sequence length 128 for all representations other than RawText with OPT, for which we used 256 tokens.

[5]https://github.com/bjascob/amrlib

[6]https://www.sbert.net; `distilbert-base-nli-stsb-quora-ranking` model.

**Table 1: ARQMath-2 Formula Search Results for Single Sentence Embeddings (Sentence-BERT, Assessed formulas).** ***w. OPTText*****: formulas included as linearized OPTs.**

| Context Representation | Evaluation Measures nDCG′@5 | nDCG′@10 | P′@5 | P′@10 |
|---|---|---|---|---|
| AMRText w. OPTText | **0.58** | **0.56** | **0.51** | **0.47** |
| AMRText w. LaTeX | 0.38 | 0.39 | 0.35 | 0.34 |
| RawText w. OPTText | 0.54 | 0.51 | 0.48 | 0.43 |
| RawText w. LaTeX | 0.43 | 0.42 | 0.40 | 0.37 |
| OPTText | 0.35 | 0.32 | 0.30 | 0.24 |

**Table 2: Results for Original ARQMath-2 runs, runs Reranked by MathAMR, and RRF of Original & Rerank. *Approach0 is a manual run. †MathDowers used text & formulas.**

| Model | nDCG′@10 Original | Rerank | RRF | P′@10 Original | Rerank | RRF |
|---|---|---|---|---|---|---|
| *Approach0 | 0.59 | 0.57 | **0.63** | 0.49 | 0.48 | **0.54** |
| DPRL | 0.60 | 0.59 | **0.62** | **0.54** | 0.51 | 0.52 |
| †MathDowsers | 0.54 | 0.55 | **0.58** | 0.45 | 0.44 | **0.48** |
| XY-Phoc | 0.51 | 0.54 | **0.55** | 0.43 | **0.45** | **0.45** |
| NLP_NIST | 0.26 | 0.33 | **0.46** | 0.20 | 0.26 | **0.40** |
| TU-DBS | 0.25 | 0.28 | **0.41** | 0.22 | 0.23 | **0.35** |

**Evaluation Measures.** We used nDCG′ [29] and P′, at cutoffs of 5 and 10. These measures are nDCG@k and P@k for k={5,10} after removing unjudged results from the ranked lists. Following the ARQMath evaluation protocol, to calculate P′@k, we treat only high and medium ratings as relevant, and all measures are calculated after deduplication. Retrieved formula instances are deduplicated using ARQMath identifiers for visually distinct groupings.

**Context Representation Results (Table 1).** To study the effectiveness of the MathAMR representation, we consider four other representations similar to MathAMR. For simplicity, we compared these models using only assessed formula hits; this may produce a bias, as all relevant formulas are in the ranked set. Our baseline uses only linearized OPTs (*'OPTText'*), ignoring surrounding text. *'AMRText w. OPTText'* corresponds to Figure 3 (c). For the model *'AMRText w. LaTeX'*, we produce AMRs as shown in Figure 3 (a), and replace the formula placeholder node with the original formula LaTeX. The final two representations use raw text with the original LaTeX or linearized OPTs. Note that the tokenizer for RawText can correctly handle formula operators (e.g., '$a + b = c$' is tokenized as: $[a, +, b, =, c]$). A single Sentence-BERT setting is used for all conditions other than *'RawText w. OPTText'* (see above).

**ARQMath-2 Reranking Results (Table 2)**. To evaluate MathAMR's utility for contextual formula search, we rerank runs with the highest effectiveness from each participating team in ARQMath-2: Approach0 [37], MathDowsers [25], TU-DBS [28], DPRL [19], XY-Phoc [1], and NLP_NITS [6]. Approach0 was a "manual" run that included human intervention, and only MathDowers used both text and formulas. Table 2 shows the effectiveness using cosine similarity of MathAMR embeddings (AMR w. OPTText) and then combining original scores with MathAMR embedding similarities by score-weighted Reciprocal Rank Fusion (RRF) [19].

**Discussion.** Table 1 compares effectiveness measures for MathAMR representations and other context representations. MathAMR (i.e., AMRText with OPTText) does best in all measures averaged over the 58 ARQMath-2 test topics. The closest competitor is the RawText with OPTText condition. With the exception of the RawText with OPTText condition, MathAMR results were significantly better than other representations by all four measures ($p < 0.05$, two-tailed paired $t$-test with Bonferroni correction).

Stratifying our analysis using high/medium/low complexity topic labels distributed with the test collection, we see that both the RawText with OPTText and AMRText with OPTText conditions have identical P′@10 for low complexity topics, and that MathAMR's superior results comes entirely from a better average P′@10 on medium and high complexity topics. For several topics, MathAMR achieves better results due to its selective focus on text in the sentence containing the formula, rather than the whole input text. For example, for the formula query in the sentence, "How to show $(a_1 a_2 \dots a_n)^{\frac{1}{n}} \leq \frac{\sum_{i=1}^{n} a_i}{n}$", P′@10 increases from 0.1 for RawText with OPTText to 0.9 for MathAMR, as many candidates occur in sentences with uninformative words that AMR ignores, and that RawText includes in Sentence-BERT input.

Table 2 shows P′@10 and nDCG′@10 for the best ARQMath-2 run submitted by each team to ARQMath-2021 [22]. Using linearized MathAMR embeddings to rerank candidates using the cosine similarity sometimes decreases P′@10, partly because there can be several formulas in one sentence, all of which share a single MathAMR similarity score. For this reason, a score-weighted Reciprocal Rank Fusion of the original and MathAMR-reranked results performs better than reranking alone. nDCG′@10 is significantly improved over the original rankings for all systems other than DPRL ($p<0.05$, $t$-test with Bonferroni correction). The one case where RRF did not help (DPRL) likely results from its use of tree-edit distance for final matching, which makes top results often very similar to the query. However, if we consider all retrieved instances, nDCG′@1000 increases from 0.57 to 0.76 when using RRF to combine DPRL and MathAMR; so even here there is a benefit, it is just seen lower in the ranking.

## 6 CONCLUSION

MathAMR is a new semantic representation for sentences that integrates Abstract Meaning Representation graphs with Operator Tree formula representations. We have made a first study of formula search using linearized MathAMR graphs for single sentences that are embedded using Sentence-BERT. Compared to raw text embeddings, MathAMR achieved better results, and using MathAMR embeddings to rerank results from other formula search techniques yielded improvements. In future work, we plan to explore using MathAMR without employing linearization, and to study the utility of MathAMR for ARQMath's Answer Retrieval task (Task 1). Also, the current MathAMR representation can be improved: for example, in future work we may consider using/adding other math formula representations such as symbol layout trees (SLTs).

## ACKNOWLEDGMENTS

We thank Heng Ji and Kevin Knight for helpful discussions about AMR and multimodal representations. This material is based upon work supported by the Alfred P. Sloan Foundation under Grant No. G-2017-9827 and the National Science Foundation (USA) under Grant No. IIS-1717997.

# REFERENCES

[1] Robin Avenoso, Behrooz Mansouri, and Richard Zanibbi. 2021. In *XY-PHOC Symbol Location Embeddings for Math Formula Retrieval and Autocompletion.*

[2] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability With Discourse.*

[3] Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One SPRING to Rule Them Both: Symmetric AMR Semantic Parsing and Generation Without a Complex Pipeline. In *Proceedings of AAAI.*

[4] Claire Bonial, Julia Bonn, Kathryn Conger, Jena D Hwang, and Martha Palmer. 2014. Propbank: Semantics of New Predicate Types. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14).*

[5] Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).* IEEE.

[6] Pankaj Dadure, Partha Pakray, and Sivaji Bandyopadhyay. 2021. BERT-Based Embedding Model for Formula Retrieval. In *Working Notes of CLEF.*

[7] Kenny Davila and Richard Zanibbi. 2017. Layout and Semantics: Combining Representations for Mathematical Formula Search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[8] Jeffrey Flanigan, Sam Thomson, Jaime G Carbonell, Chris Dyer, and Noah A Smith. 2014. A Discriminative Graph-Based Parser for the Abstract Meaning Representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.*

[9] Dallas Fraser, Andrew Kane, and Frank Wm Tompa. 2018. Choosing Math Features for BM25 Ranking with Tangent-L. In *Proceedings of the ACM Symposium on Document Engineering 2018.*

[10] Sahil Garg, Aram Galstyan, Ulf Hermjakob, and Daniel Marcu. 2016. Extracting Biomolecular Interactions using Semantic Parsing of Biomedical Text. In *Thirtieth AAAI Conference on Artificial Intelligence.*

[11] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. *arXiv preprint arXiv:1705.00652* (2017).

[12] Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramón Fernandez Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue-Nkoutche, et al. 2021. Leveraging Abstract Meaning Representation for Knowledge Base Question Answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021.*

[13] Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. 2016. MCAT Math Retrieval System for NTCIR-12 MathIR Task. In *NTCIR.*

[14] Kriste Krstovski and David M Blei. 2018. Equation Embeddings. *arXiv preprint arXiv:1803.09123* (2018).

[15] Irene Langkilde and Kevin Knight. 1998. Generation That Exploits Corpus-Based Statistical Knowledge. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics.*

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.*

[17] Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for Multi-Document Summarization. In *Proceedings of the 27th International Conference on Computational Linguistics.*

[18] Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman Sadeh, and Noah A Smith. 2015. Toward Abstractive Summarization Using Semantic Representations. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[19] Behrooz Mansouri, Douglas W Oard, and Richard Zanibbi. 2021. DPRL Systems in the CLEF 2021 ARQMath Lab: Sentence-BERT for Answer Retrieval, Learning-to-Rank for Formula Retrieval. (2021).

[20] Behrooz Mansouri, Shaurya Rohatgi, Douglas W Oard, Jian Wu, C Lee Giles, and Richard Zanibbi. 2019. Tangent-CFT: An Embedding Model for Mathematical Formulas. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval.*

[21] Behrooz Mansouri, Richard Zanibbi, and Douglas W Oard. 2021. Learning to Rank for Mathematical Formula Retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[22] Behrooz Mansouri, Richard Zanibbi, Douglas W Oard, and Anurag Agarwal. 2021. Overview of ARQMath-2 (2021): Second CLEF Lab on Answer Retrieval for Questions on Math. In *International Conference of the Cross-Language Evaluation Forum for European Languages.* Springer.

[23] Jonathan May and Jay Priyadarshi. 2017. SemEval-2017 Task 9: Abstract Meaning Representation Parsing and Generation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017).*

[24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. *Advances in Neural Information Processing Systems* (2013).

[25] Yin Ki Ng, Dallas Fraser, Besat Kassaie, and Frank Tompa. 2021. Dowsing for Answers to Math Questions: Ongoing Viability of Traditional MathIR. In *Working Notes of CLEF.*

[26] Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. 2021. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. *arXiv preprint arXiv:2105.00377* (2021).

[27] 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).*

[28] Anja Reusch, Maik Thiele, and Wolfgang Lehner. 2021. TU_DBS in the ARQMath Lab 2021. In *Working Notes of CLEF.*

[29] Tetsuya Sakai and Noriko Kando. 2008. On Information Retrieval Metrics Designed for Evaluation with Incomplete Relevance Assessments. *Information Retrieval* (2008).

[30] Petr Sojka and Martin Líška. 2011. Indexing and Searching Mathematics in Digital Libraries. In *International Conference on Intelligent Computer Mathematics.* Springer.

[31] Chuan Wang, Nianwen Xue, and Sameer Pradhan. 2015. A Transition-Based Algorithm for AMR Parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[32] Weiwen Xu, Huihui Zhang, Deng Cai, and Wai Lam. 2021. Dynamic Semantic Graph Construction and Reasoning for Explainable Multi-hop Science Question Answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021.*

[33] Richard Zanibbi, Akiko Aizawa, Michael Kohlhase, Iadh Ounis, Goran Topic, and Kenny Davila. 2016. NTCIR-12 MathIR Task Overview. In *Proceedings of the 16th NTCIR.*

[34] Richard Zanibbi, Douglas W Oard, Anurag Agarwal, and Behrooz Mansouri. 2020. Overview of ARQMath 2020: CLEF Lab on Answer Retrieval for Questions on Math. In *International Conference of the Cross-Language Evaluation Forum for European Languages.* Springer.

[35] Zixuan Zhang and Heng Ji. 2021. Abstract Meaning Representation Guided Graph Encoding and Decoding for Joint Information Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

[36] Wei Zhong, Shaurya Rohatgi, Jian Wu, C Lee Giles, and Richard Zanibbi. 2020. Accelerating Substructure Similarity Search for Formula Retrieval. In *European Conference on Information Retrieval.* Springer.

[37] Wei Zhong, Xinyu Zhang, Ji Xin, Jimmy Lin, and Richard Zanibbi. 2021. Approach Zero and Anserini at the CLEF-2021 ARQMath Track: Applying Substructure Search and BM25 on Operator Tree Path Tokens. In *Woorking Notes of CLEF.*