Mathematical Formula Retrieval

Presentation by

Megha Chakraborty
Oshita Saxena
Yash Jain
Yash Raj

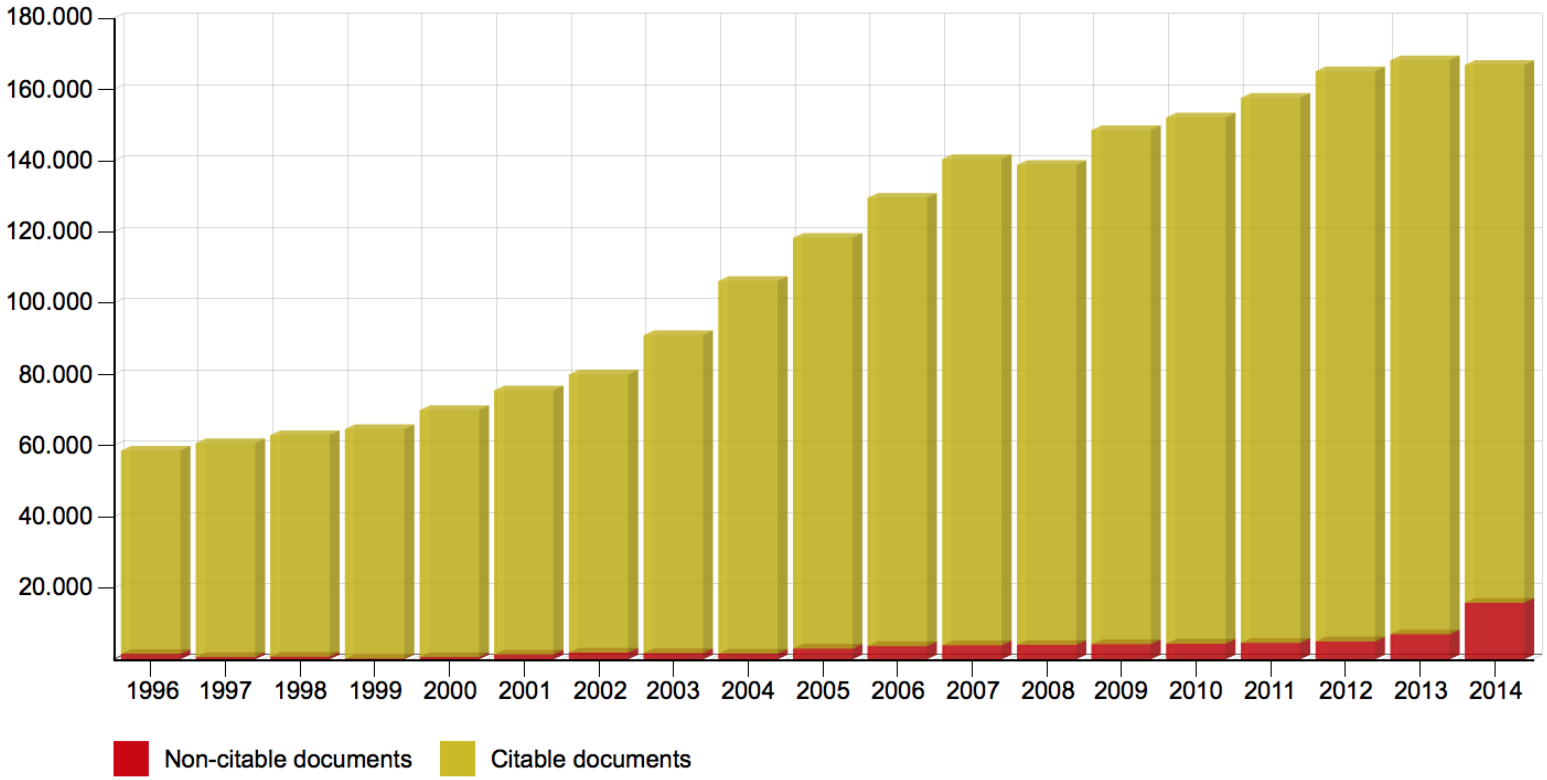
CHENNAI MATHEMATICAL INSTITUTE

Contents

1. Why Formula Retrieval?
2. Formula Representation
3. Retrieval Models
   - Text Based Model
   - Tree Based Model
   - Embedding Models

CHENNAI MATHEMATICAL INSTITUTE

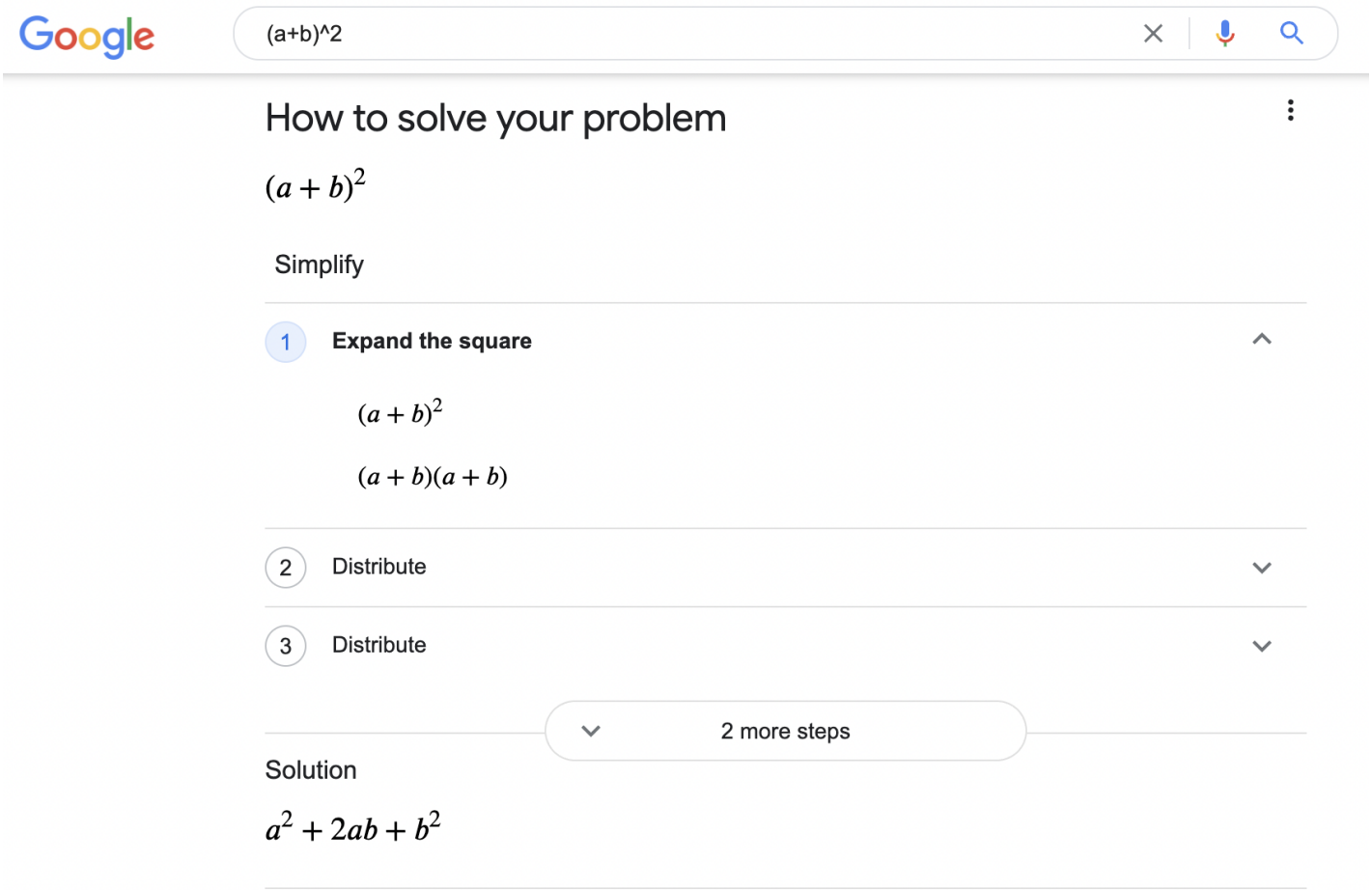
Why Formula Retrieval?



Number of Math publications every year

CHENNAI MATHEMATICAL INSTITUTE

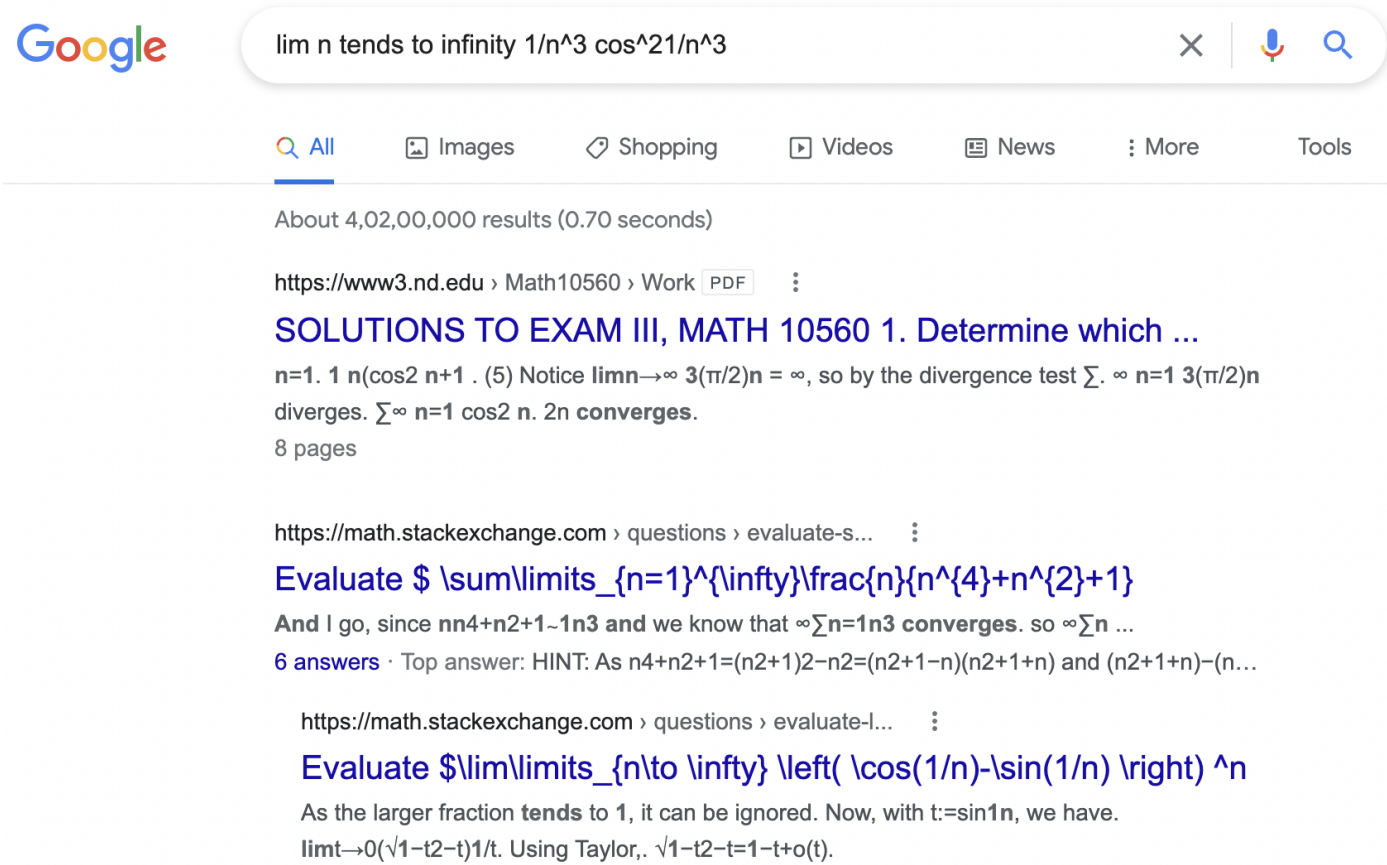
Why Formula Retrieval?



Search result for $(a+b)^2$

CHENNAI MATHEMATICAL INSTITUTE

Why Formula Retrieval?

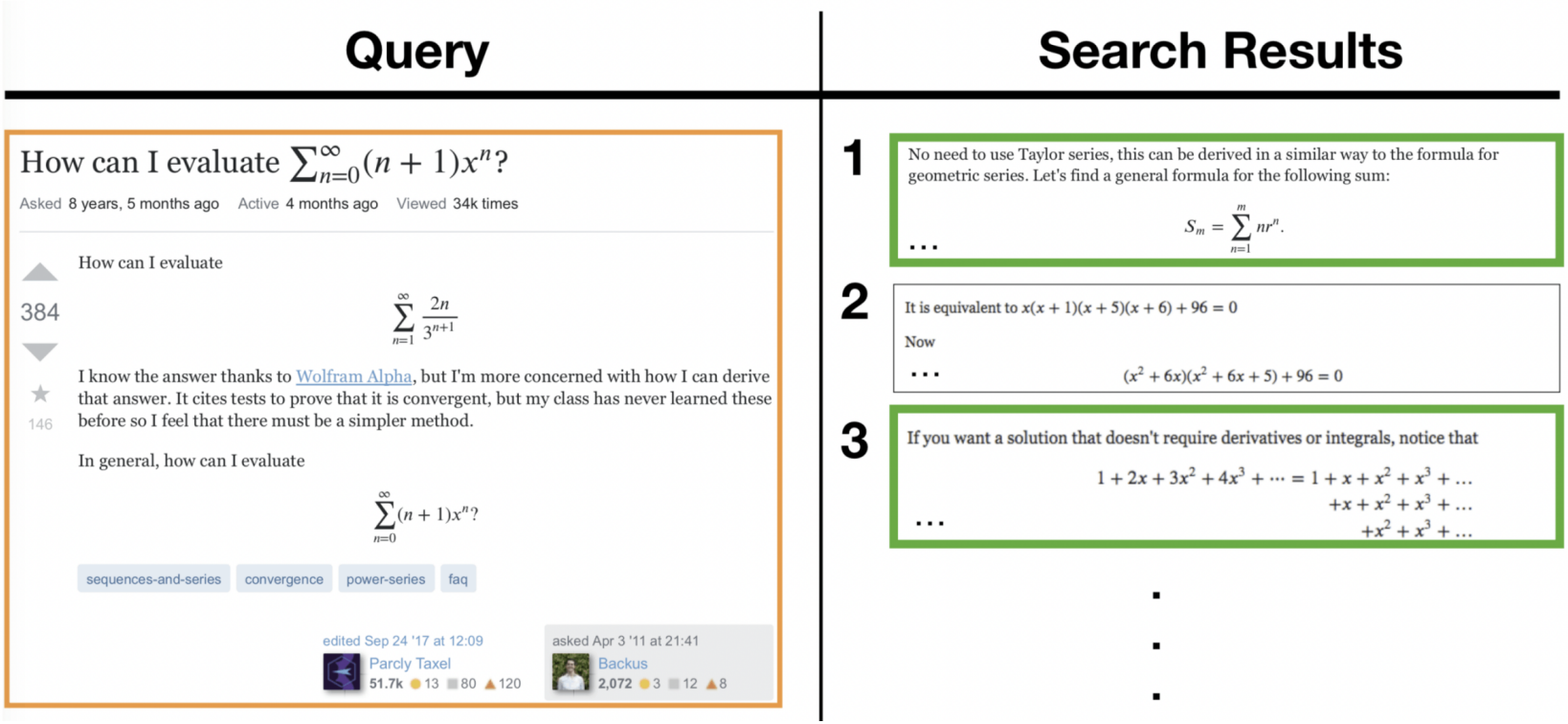


Search results for

$$\lim_{n\to+\infty} \frac{1}{n^3} \cos^2 \frac{1}{n^3}$$

CHENNAI MATHEMATICAL INSTITUTE

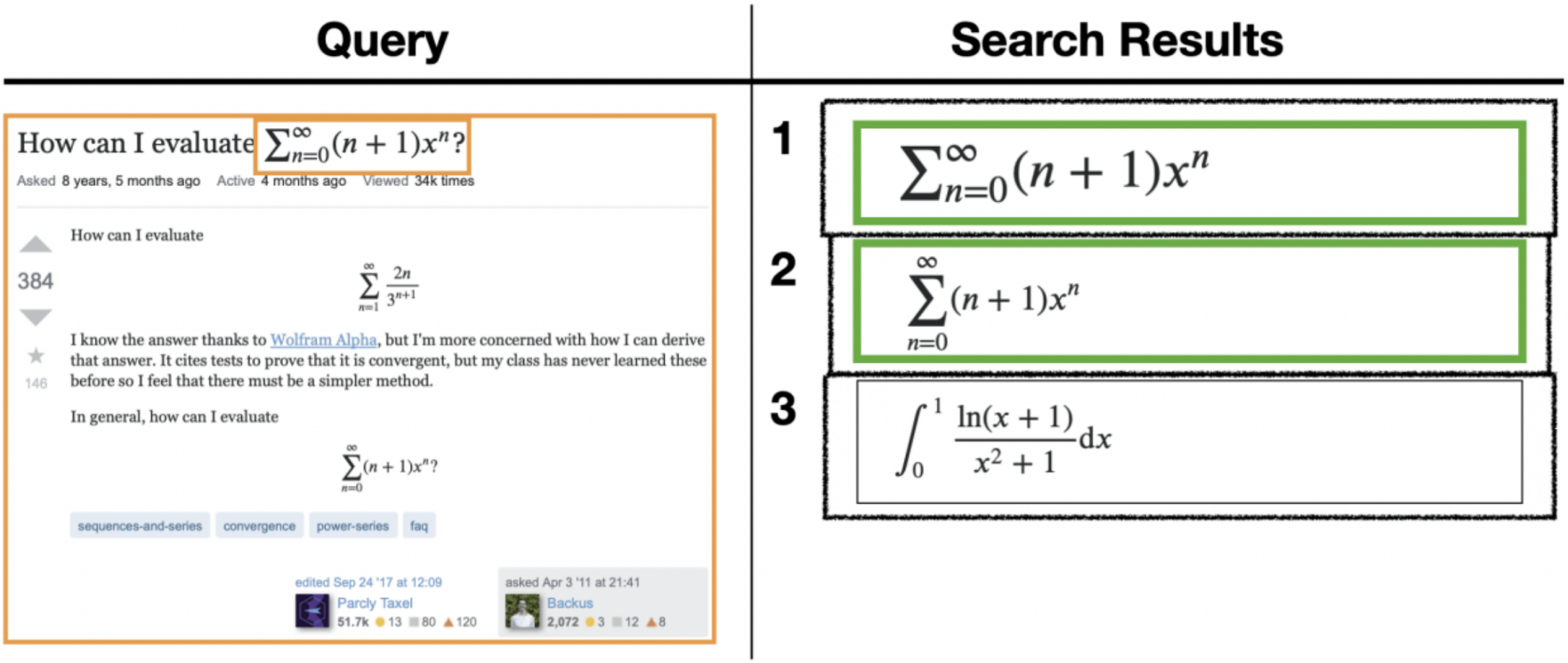
Why Formula Retrieval?



Answer Retrieval

CHENNAI MATHEMATICAL INSTITUTE

Why Formula Retrieval?



Formula Retrieval

Formula Representation

$$a^2 + b^2 = c^2$$

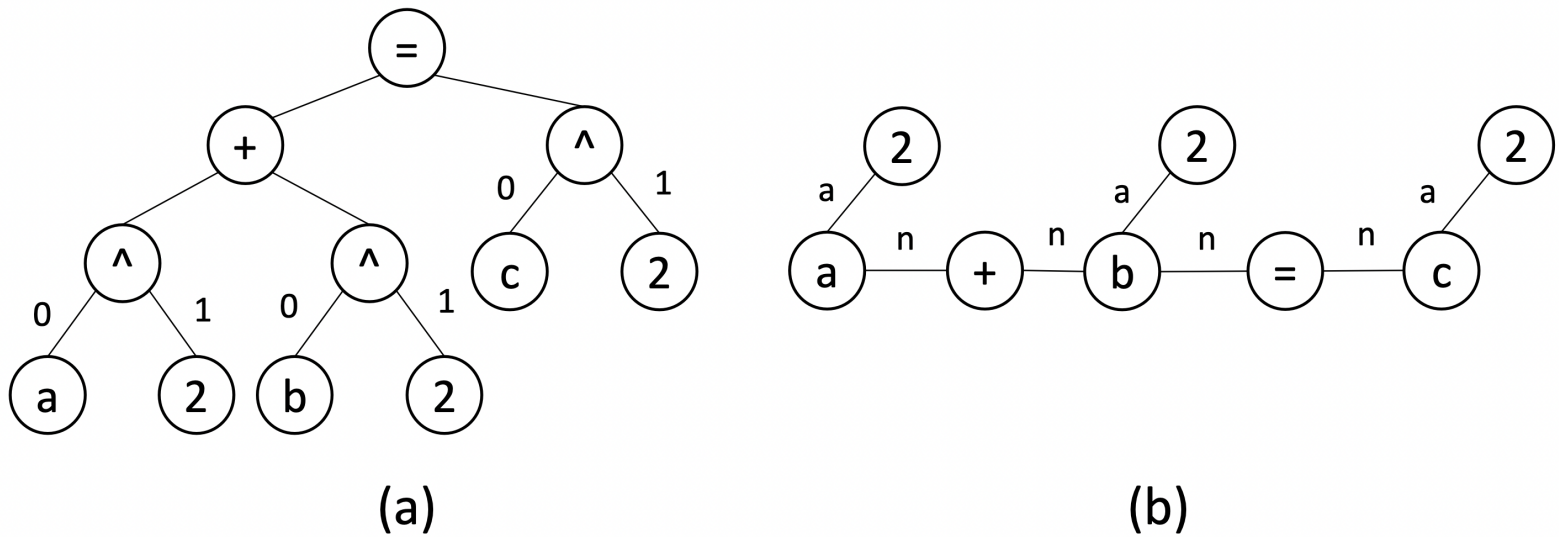


Figure: (a) Operator Tree and, (b) Symbol Layout Tree

Nodes and Edges

Node labels: **Type!**value

- Numbers (**N!**2)
- Variable names (**V!**x)
- Text fragments (**T!** *'lim'*)
- Fractions (**F!** $1/2$)
- Matrices (**M!** $()2 \times 3$)
- Wildcard (*)
- Commutative operator (**U!**)
- Non-commutative operator (**O!**)

Edge labels

- **SLTs**: edge labels give spatial arrangement of symbols
  - Next (**n** or $\rightarrow$)
  - Within (**w** or $\boxdot$)
  - Above (**a** or $\uparrow$)
  - Below (**b** or $\downarrow$)
  - Under (**U**)
- **OPTs**: edge labels indicate argument position
  - $0/1$

CHENNAI MATHEMATICAL INSTITUTE

Approaches for Formula Retrieval

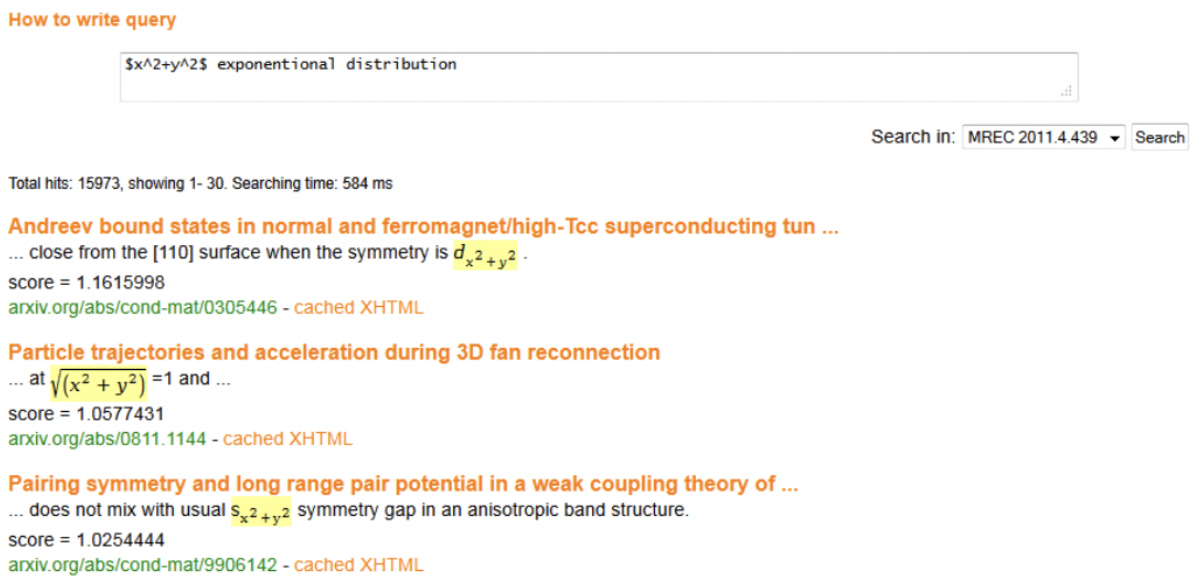
- **Text-based** (formulae are converted to a string such as LaTeX)
- **Tree-based** (formulae are represented as trees like SLTs/OPTs)
- **Embedding-based** (formulae are represented as trees and then converted to vectors)

MIaS: A Text Based Model

MIaS: A Text Based Model

MIaS-Math Indexer and Searcher

- Math-aware full-text search engine based on Apache Lucene
- Joins textual and mathematical querying
- Takes MathML or LaTeX as input



CHENNAI MATHEMATICAL INSTITUTE

MIaS: A Text Based Model

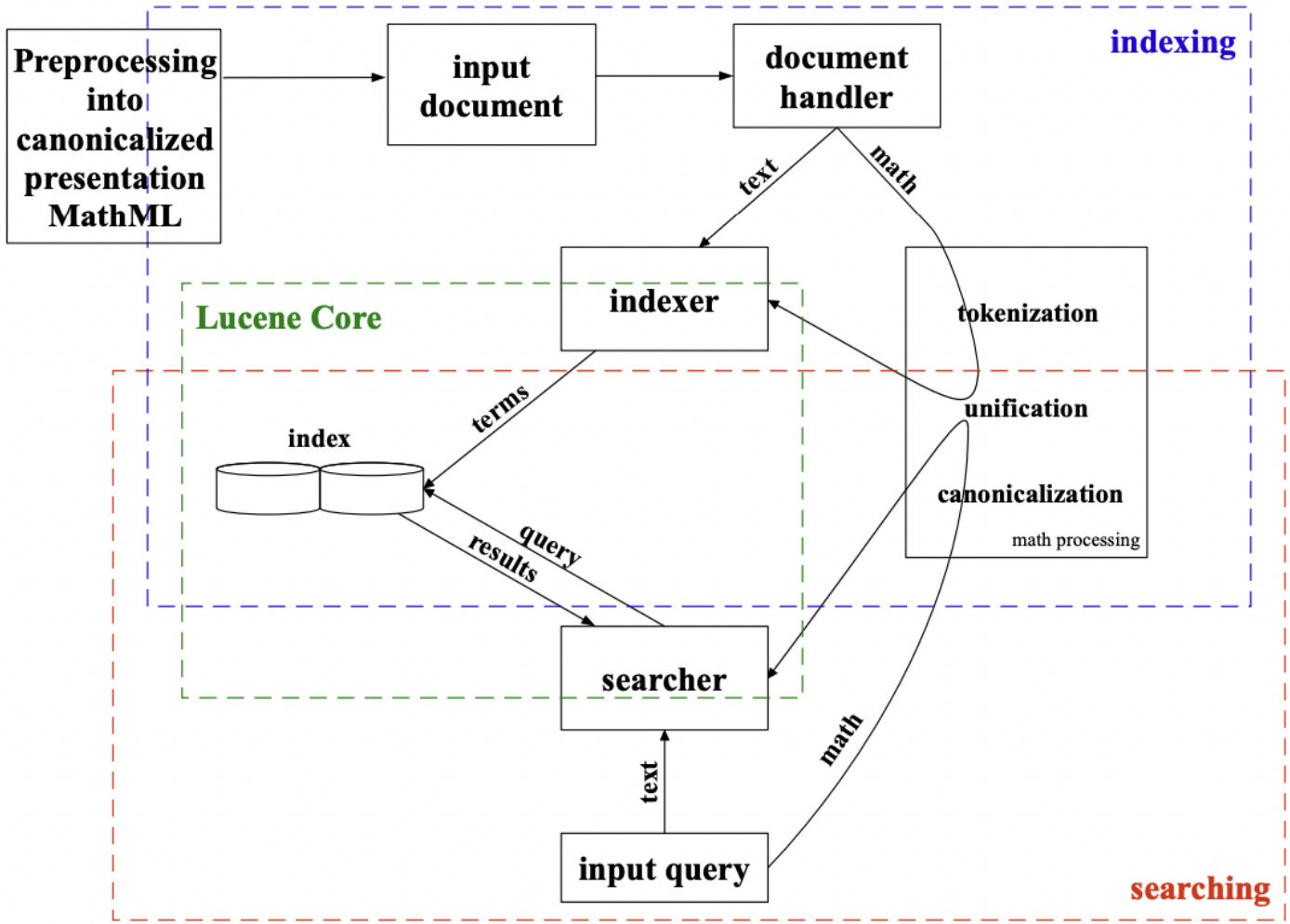
- In text retrieval: index word stems only instead of word forms
- From text to math: have multiple representations of a formula (with different 'near synonyms') put in the index
- LaTeX notation for querying
- MathML for indexing
- Uses Lucene's standard practical scoring function:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ in } q} \left(\text{tf}(t \text{ in } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d) \right)$$

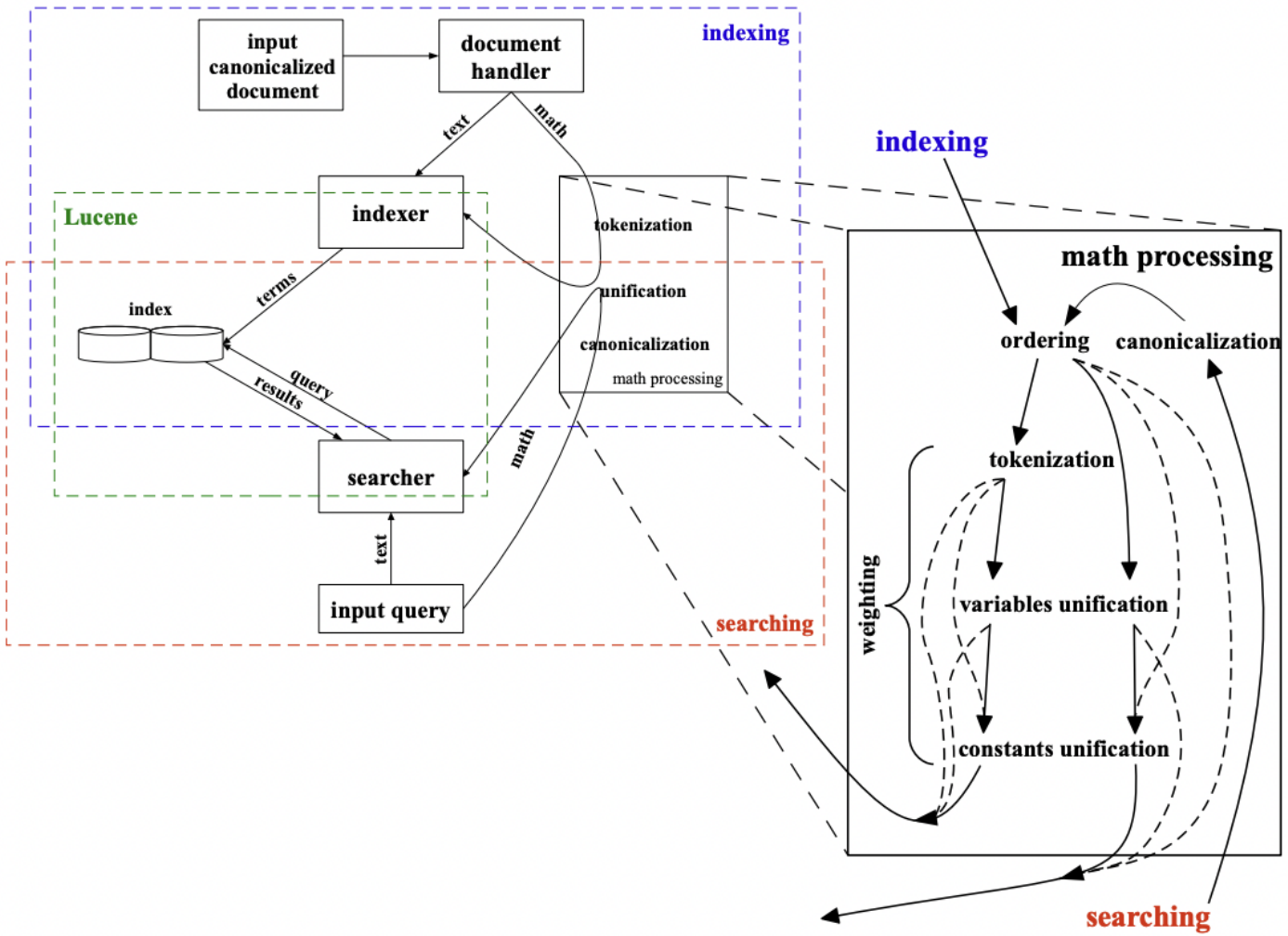
CHENNAI MATHEMATICAL INSTITUTE

MIaS: A Text Based Model

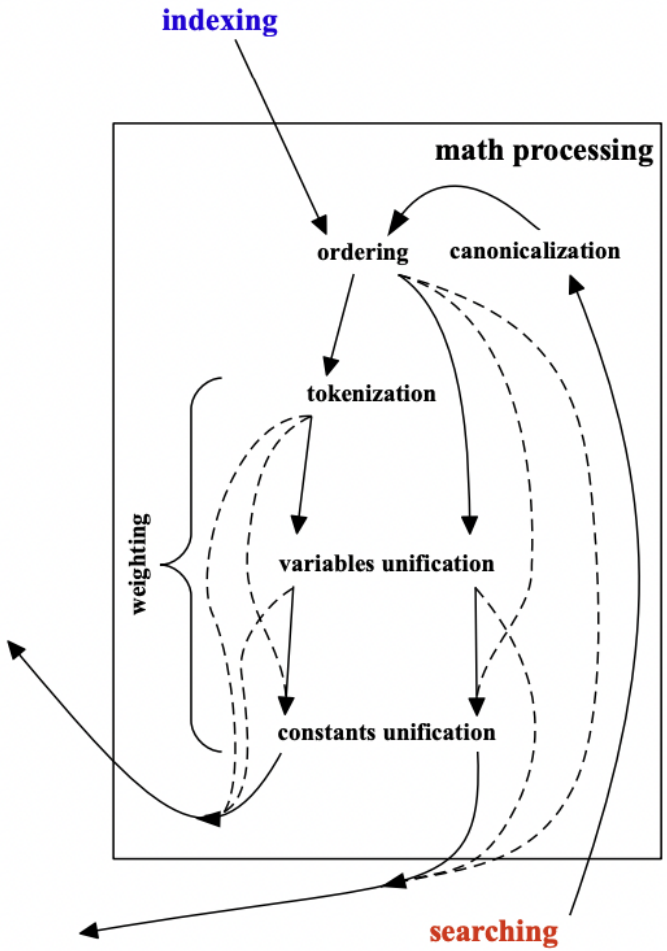
MIaS Framework



Indexing Framework



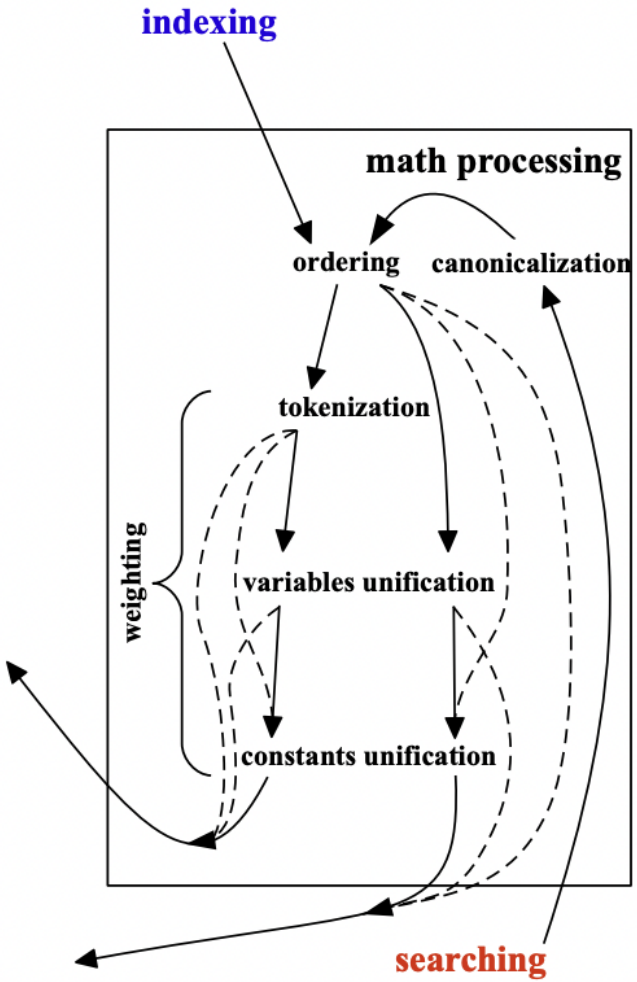
Index Processing

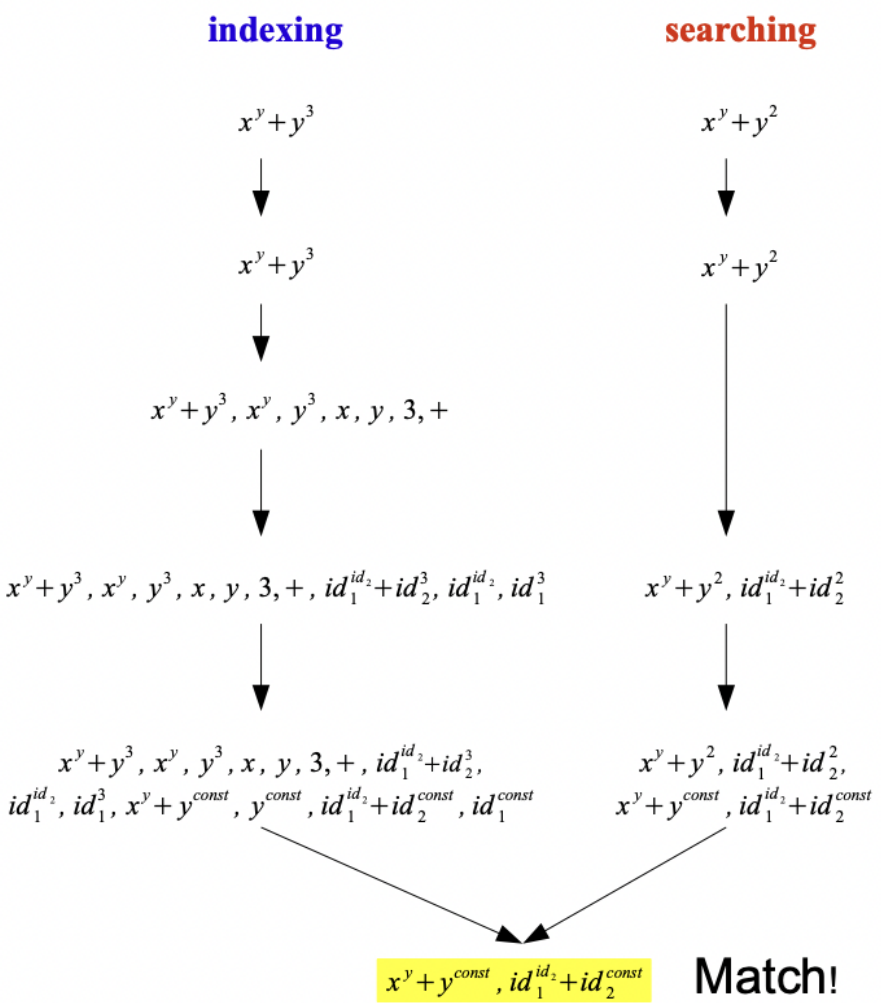


CHENNAI MATHEMATICAL INSTITUTE

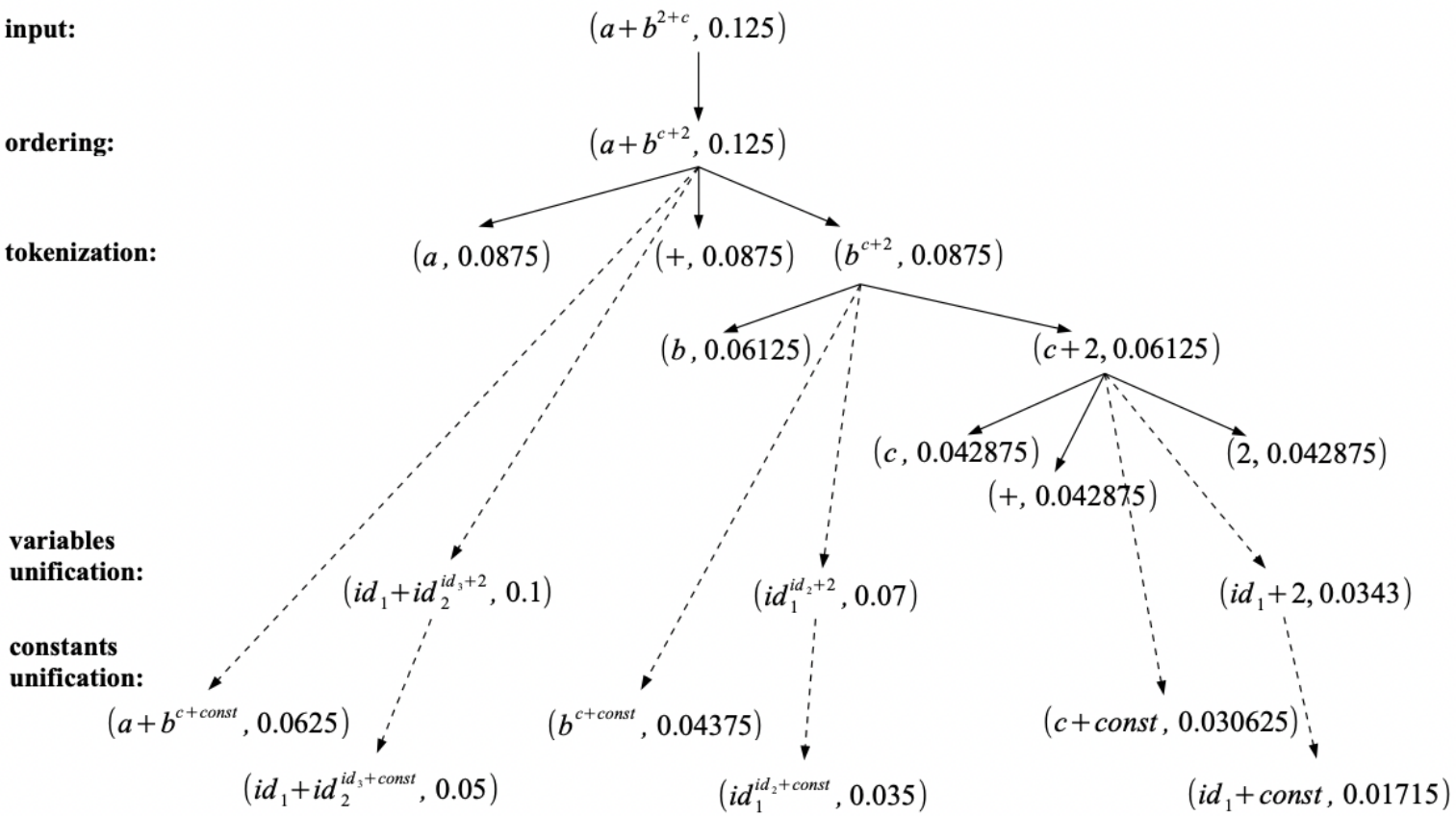
MIaS: A Text Based Model

Example



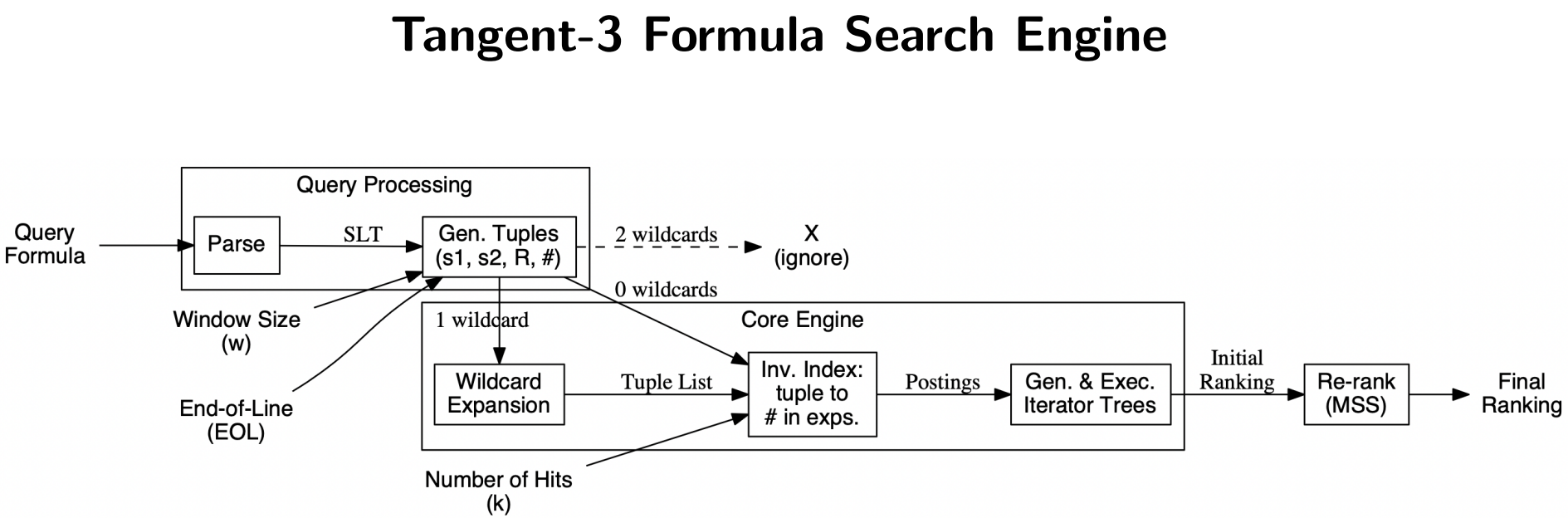


Formula Processing Example – Subformula Weighting



Tangent-3: A Tree Based Model

Tangent-3: A Tree Based Model



Tree to Tuples

SLT Tuples				OPT Tuples			
(Parent,	Child,	Path,	PFR)	(Parent,	Child,	Path,	PFR)
(**V!** a,	**N!** 2,	a,	-)	(**U!** eq,	**U!** plus,	0,	-)
(**N!2**,	eob,	n,	a)	(**U!** plus,	**O!** SUP,	0,	0)
(**V!** a,	+,	n,	-)	(**O!** SUP,	**V!**a,	0,	00)
⋮				⋮			

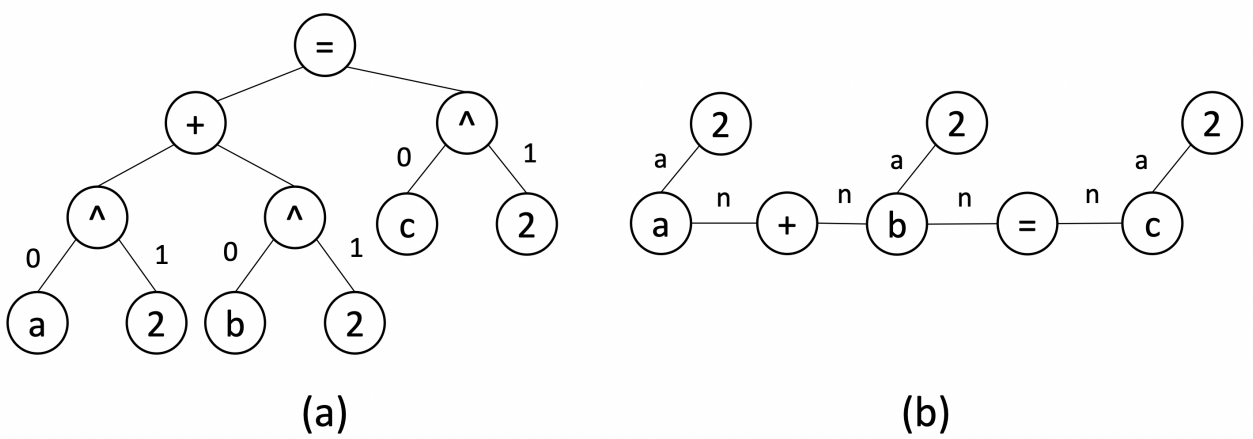


Figure: (a) Operator Tree and, (b) Symbol Layout Tree

CHENNAI MATHEMATICAL INSTITUTE

PFR Path-from-root

Indexing

Inverted index built over document-formula-tuple relationships

- D_f: formula $\rightarrow$ formID
- D_t: tuple $\rightarrow$ tupleID
- D_d: document $\rightarrow$ docID
- D_w: wildcardtuple $\rightarrow$ wildcardtupleID
- P_t: tupleID $\rightarrow$ (formID, count)$^+$
- P_f: formID $\rightarrow$ (docID, position)$^+$
- P_w: wildcardtupleID $\rightarrow$ (tupleID)$^+$

Greedy wildcard expansion

Behavior	Query	Match
Unrestricted	$x + *$	$x+1$ $x+y+z+sin(x)$
	$y + x+z = \frac{\pi}{4}$	
	e^*	$f(x) = e^{x+1} + 2$
Restricted by children	$*^2+1$	$x^2 + y^2+1$ $x^2 + y+1$ $x^2 + (y+z)^2+1$
Restricted by binding	$*1*^2+*1*+1$	x^2+x+1 $(x+1)^2+(x+1)+1$ x^2+y+1
Horizontal expansion (right)	$x + *+1$	$x+y+1$ $x+y+z+1$ $x+y-z+1$ $x+\frac{1}{2+y}-3z+1$
Horizontal expansion (left)	$*+1$	$x+y+z+1$ $\alpha = f(x+y+1, x^2)$ $f(x,y) = \frac{1}{x+y+1}$

Core Engine

Candidate Selection

The engine uses Dice's coefficient over tuples as a simple ranking algorithm, counting number of tuples that overlap between query and candidate formula

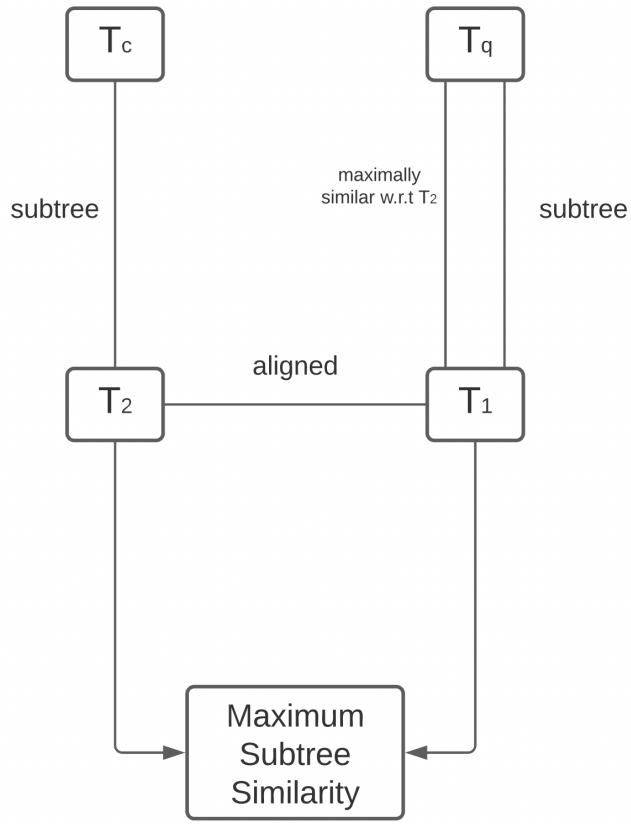
Dice's Coefficient

Given query tree T_q and candidate tree T_c with tuple sets F_q and F_c, Dice's coefficient of similarity is given by $\frac{2|F_q \cap F_c|}{|F_q|+|F_c|}$

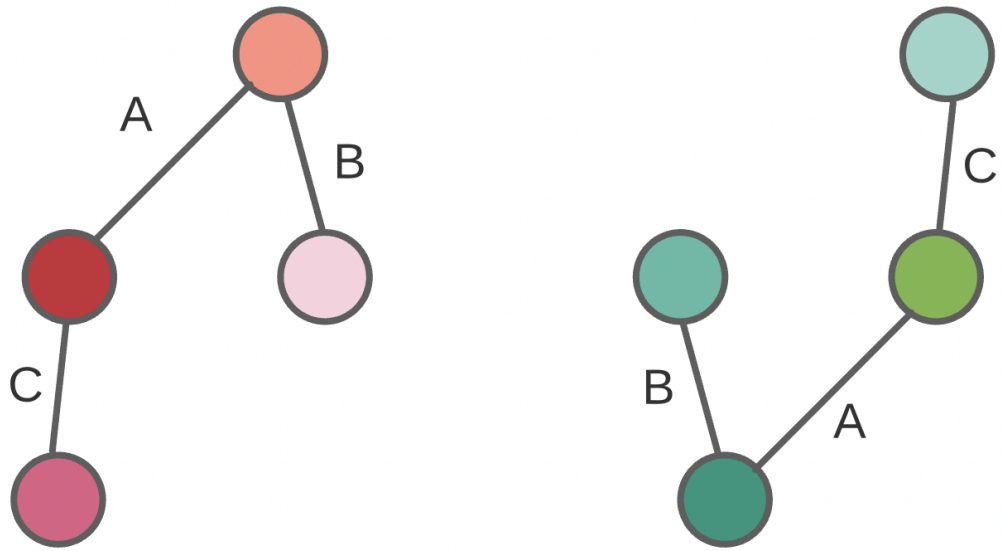
CHENNAI MATHEMATICAL INSTITUTE

Re-ranking

- T_q: query tree
- T_c: candidate tree
- T_1: subtree of T_q
- T_2: subtree of T_c



Aligned Subtrees



- Similar structure
- Nodes may not match

Maximally Similar Subtree

- Let m = # matching nodes between aligned trees T_1 and T_2
- Dice's coefficient = $\frac{2m}{|T_1|+|T_q|}$
- = measure of similarity of T_1 to T_q with respect to T_2
- T_1 is maximally similar to T_q if root of T_1 can be matched with root of T_2 and has maximum dice's coefficient

CHENNAI MATHEMATICAL INSTITUTE

Re-ranking

	Initial Ranking	Re-ranked (MSS)
Query 1: $f_*(z) = z^2 + c$		
1.	$f_c(z) = z^2 + c$	$f_c(z) = z^2 + c$
2.	$f_c(z) = z^2 + c.$	$\mathbf{P}_c(z) = z^2 + c$
3.	$f(z) = z^2 + c$	$f_c(\mathbf{x}) = \mathbf{x}^2 + c$
4.	$f_0(z) = z^2$	$f_c(z) = z^2 + c.$
5.	$f_c(z) = z * z + c$	$f(z) = z^2 + c$

	Initial Ranking	Re-ranked (MSS)
Query 2: $\sum_{*2*}^{*1*} * = \sum_{*2*}^{*1*} *$		
1.	$E = \sum_i^N E_i$	$\sum_{i=1}^d a_i = \sum_{i=1}^d b_i$
2.	$G_{net} = \sum_i \sum_{i=1}^N$	$\sum_{i=1}^N d_i = \sum_{i=1}^N \lambda_i.$
3.	$\sum_i^{N_1} p_i = \sum_j^{N_2} p_j$	$\sum_{n=0}^{\infty} a_{\sigma(n)} = \sum_{n=0}^{\infty} a_n.$
4.	$\sum_{i=1}^n x_i k_i = \sum_{i=1}^n x_i$	$\sum_i^{N_1} p_i = \sum_j^{N_2} p_j$
5.	$= \sum_{k=1}^n a_k$	$\sum_{n=0}^{\infty} a_n = \sum_{n \in N} a_n.$

CHENNAI MATHEMATICAL INSTITUTE

Tangent-CFT: An Embedding Based Model

Introduction to Embedding Model: Tangent-CFT

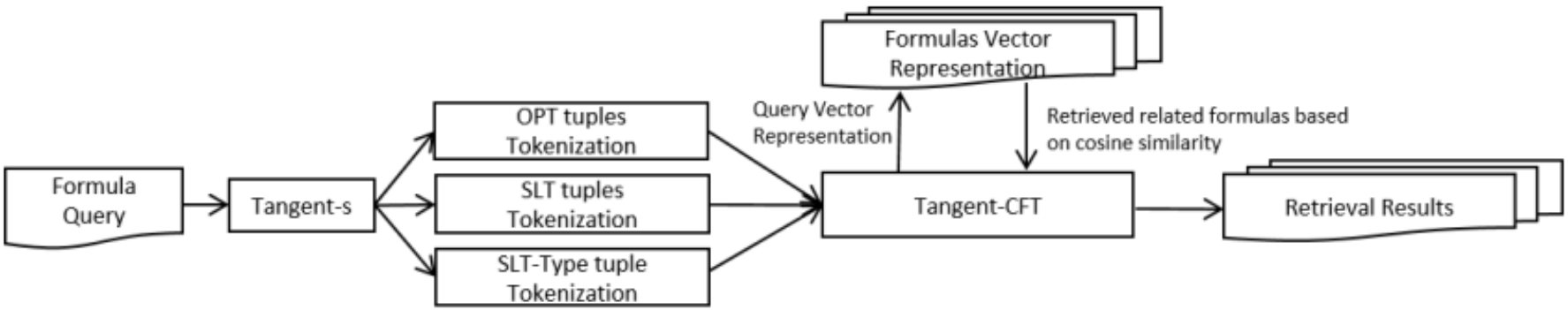
- Tangent Combined with fastText
- n-gram EM for OPT and SLT representations
- Current SOTA for partial matching on NTCIR-12 formula task
- Drawback: loses structural information- less effective for near-exact matches

Similarity Features for Embedding Models

- Provide dense vector representations - provide better matching for partially relevant formulae
- Originally:
  - 3 n-gram embedding models are trained separately on SLT, OPT and SLT Type trees
  - Final representation was sum of 3 embedding vectors
- For Learning to Rank Models:
  - Produce similarity scores separately for each of the three representations
  - Combine these using LtR model

Tangent-CFT

Framework



Generating the Embedding Vectors

1. Tuple sequence generation
2. Tokenizing formula tuples: each character (token) is encoded using a unique identifier
3. Applying fastText to formulae
4. Combining SLT and OPT embeddings (we are avoiding this step)

Thank You!

CHENNAI MATHEMATICAL INSTITUTE