

Towards Efficiently Supporting Large Symbolic Declarative Memories

Nate Derbinsky

John E. Laird

University of Michigan

Bryan Smith

Ann Arbor, MI

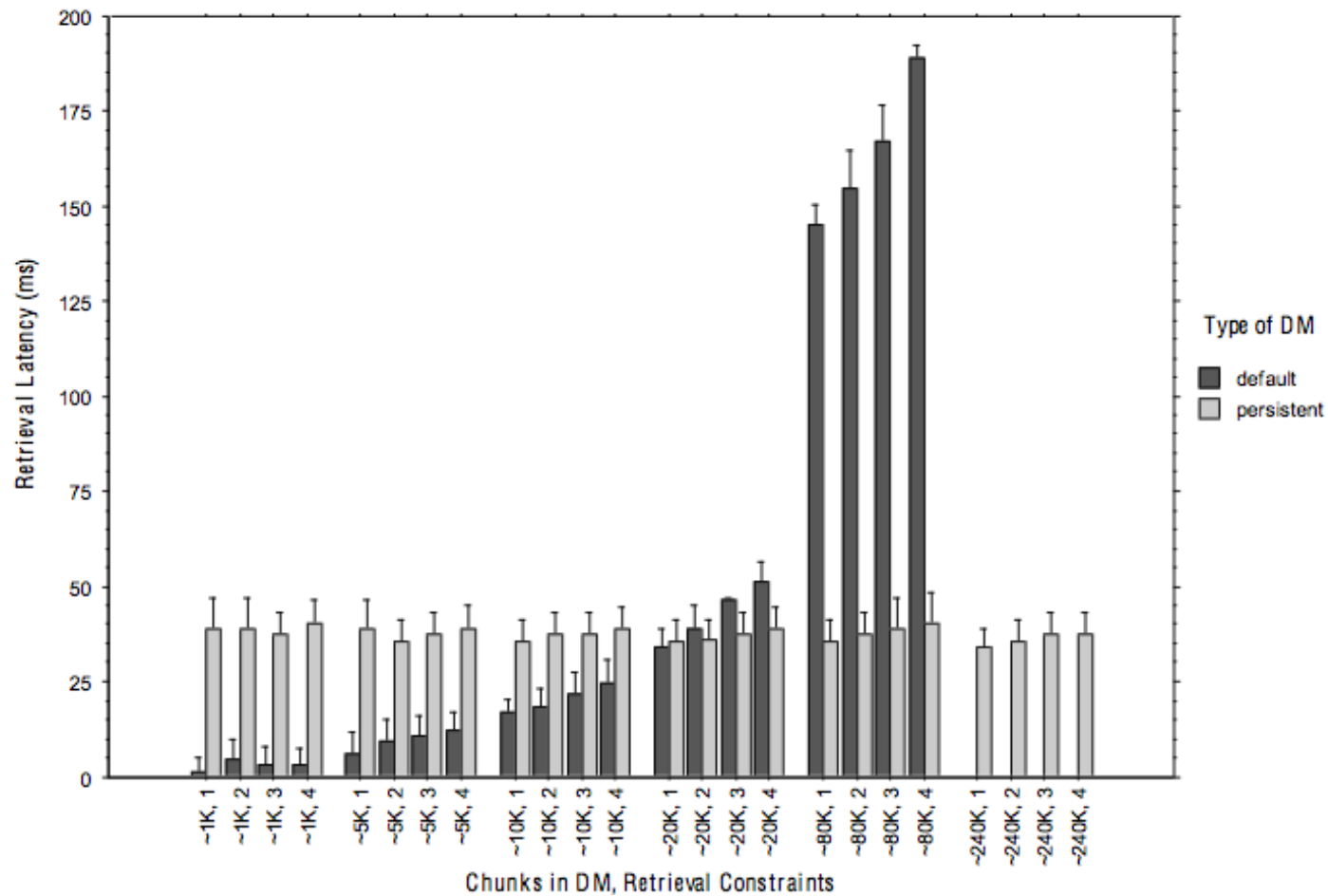
The Need for Large DMs

Typical cognitive models have very modest declarative memory (DM) requirements

Complex tasks require access to large stores of knowledge

SUMO	WordNet	Cyc
Ontology	Lexicon	“Common Sense”
4.5K classes 250K facts	212K word senses 820K facts	500K concepts 5M facts

Douglass et al., ICCM 2009



Scaling to Large Declarative Stores

	Douglass et al.
Problem Formulation	Empirical
Methods & Analysis	System Dependent
Implementation	PostgreSQL+ACT-R
Matching	Symbolic*
Chunk Activation	Disabled*
Evaluation	WN-Lexical*, 240K chunks

Scaling to Large Declarative Stores

	Douglass et al.	This Work	
Problem Formulation	Empirical	Empirical Formal	←
Methods & Analysis	System Dependent	System Independent	←
Implementation	PostgreSQL+ACT-R	SQLite+Soar*	
Matching	Symbolic*	Symbolic	
Chunk Activation	Disabled*	<i>Locally Efficient</i>	
Evaluation	WN-Lexical*, 240K chunks	WN-Lexical, 820K chunks Synthetic, 3.6M chunks	←

Symbolic ACT-R DM Example

Chunk

(S-105261088-1 ISA S

SYNSET-ID 105261088

W-NUM 1

WORD "roach"

SS-TYPE "n"

SENSE-NUMBER 1

TAG-COUNT 0

)

Buffer Request

+retrieval>

ISA S

WORD "roach"

– SS-TYPE "v"

Symbolic Formulation: Chunk

Chunk

(S-105261088-1 ISA S
SYNSET-ID 105261088
W-NUM 1
WORD "roach"
SS-TYPE "n"
SENSE-NUMBER 1
TAG-COUNT 0
)

Formulation

- *Chunk* [id] as a set of symbolic *slot-value* pairs
- DM as a set of chunks

Symbolic Formulation: Buffer Request

Buffer Request

+retrieval>

ISA S

WORD "roach"

– SS-TYPE "v"

Formulation

- Constrained form of *subset* query on *set-values*
 - Known to be linear in DM

Efficient Support: Meaning

Naïve (chunk scan)

- Time: linear in DM
- Space: linear in DM

Efficient

- Time: sub-linear in DM
- Space: linear in DM

Holds for broad variety of
DM/queries

Efficient Support: Implementation

Inverted Index

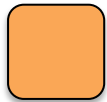
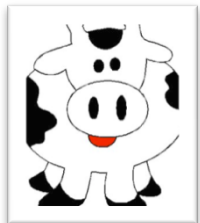
- On chunk addition, organize w.r.t. slot-value pairs
- On query, hash directly to slot-value pair of interest

Statistical Query Optimization

- Maintain statistics w.r.t. slot-value pair occurrence frequency in DM
- Re-order buffer request to minimize expected search

Example: DM

DM

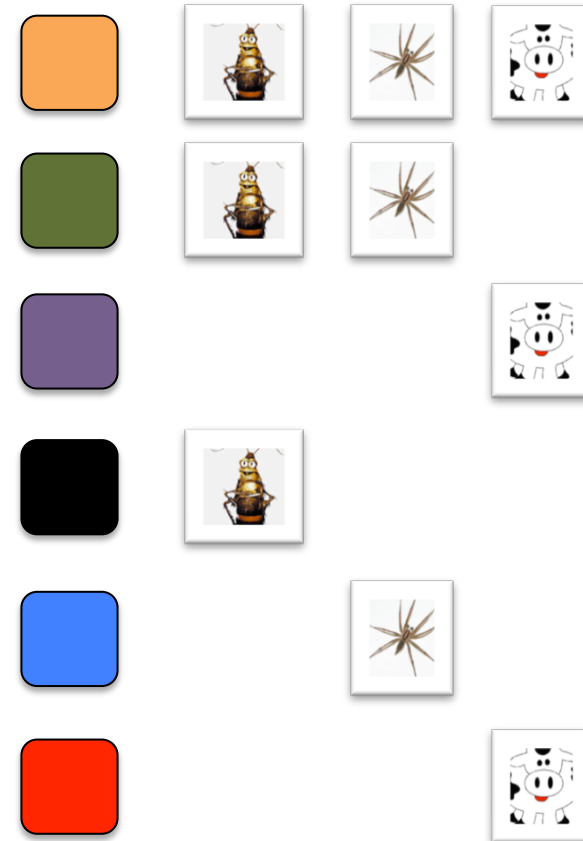


Example: Inverted Index

DM



Inverted Index

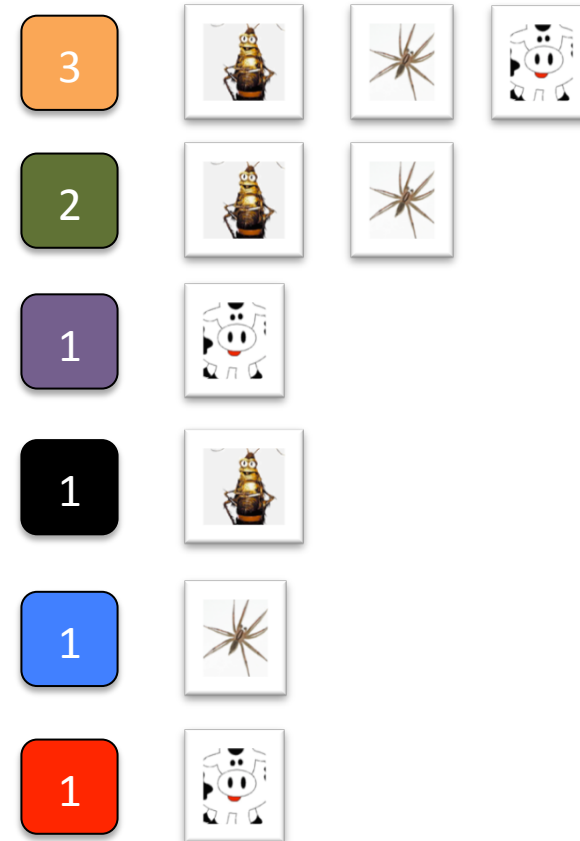


Example: Statistics

DM

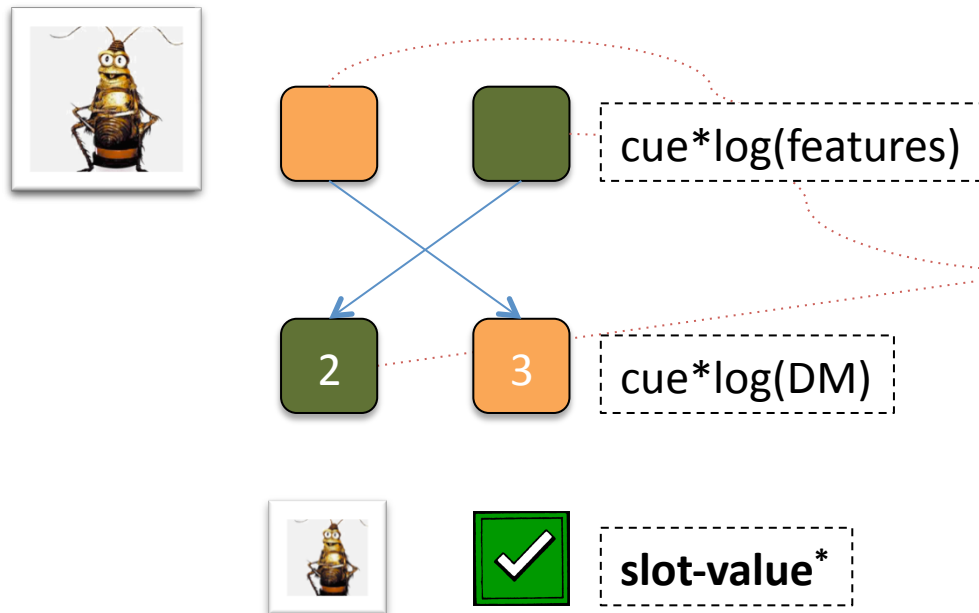


Inverted Index



Example: Buffer Request

Buffer Request



Inverted Index



Efficient Activation Bias

Extended problem formulation and implementation to *locally efficient* activation bias update functions

- Update can affect at most constant chunks
- Update must be sub-linear in DM
- Likely captures base-level approximation and permanent noise (not transient noise, partial matching, spreading)

Validated approach on SUMO, WordNet, OpenCyc

See paper for additional detail

Evaluation

- Implemented as Semantic Memory in Soar
- SQLite 3, 4GB RAM, 2.8GHz Core 2

	WordNet (WN-LEXICAL)	Synthetic
Source	Curated	Generated
DM Size	820K chunks (~400MB)	5K – 3.6M chunks (3MB – 2GB)
Purpose	Large, ecologically valid. Compare to Douglass et al.	Exhaustive benchmarking on arbitrarily large DMs

WordNet (1)

Purpose

- Retrieval time independent of slot-value selectivity
- Activation bias efficient in under-specified cues

100 random cues, single slot-value constraint

- 10 trials

~0.2 msec. ($\sigma=0.0216$)

WordNet (2)

Purpose

- Scale to larger cues
- Compare to Douglass et al.

10 random nouns, full sense (7 slot-value)

- 10 trials

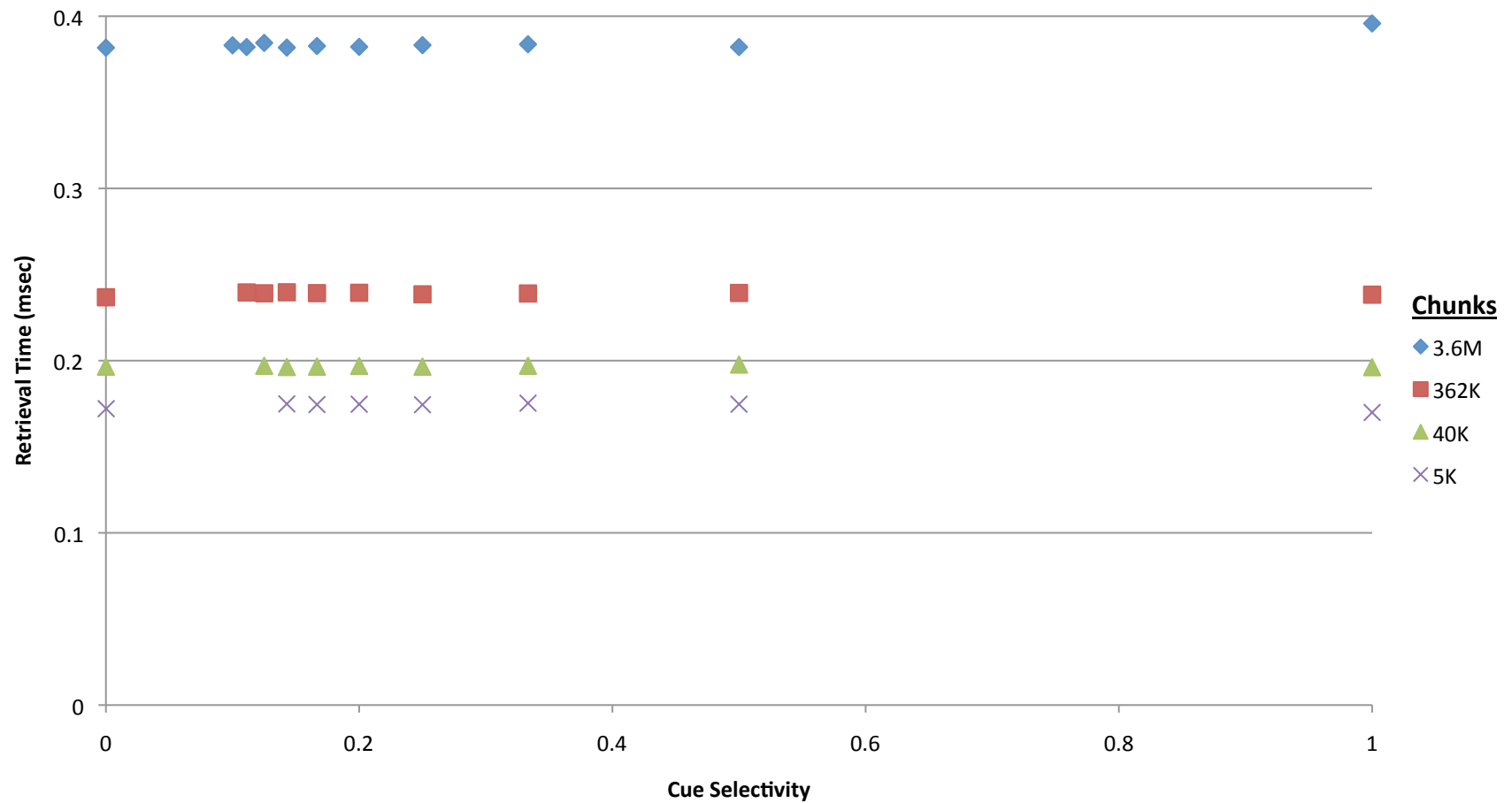
~0.3 msec. ($\sigma=0.0108$)

~100x faster than Douglass et al. on 3x larger DM*

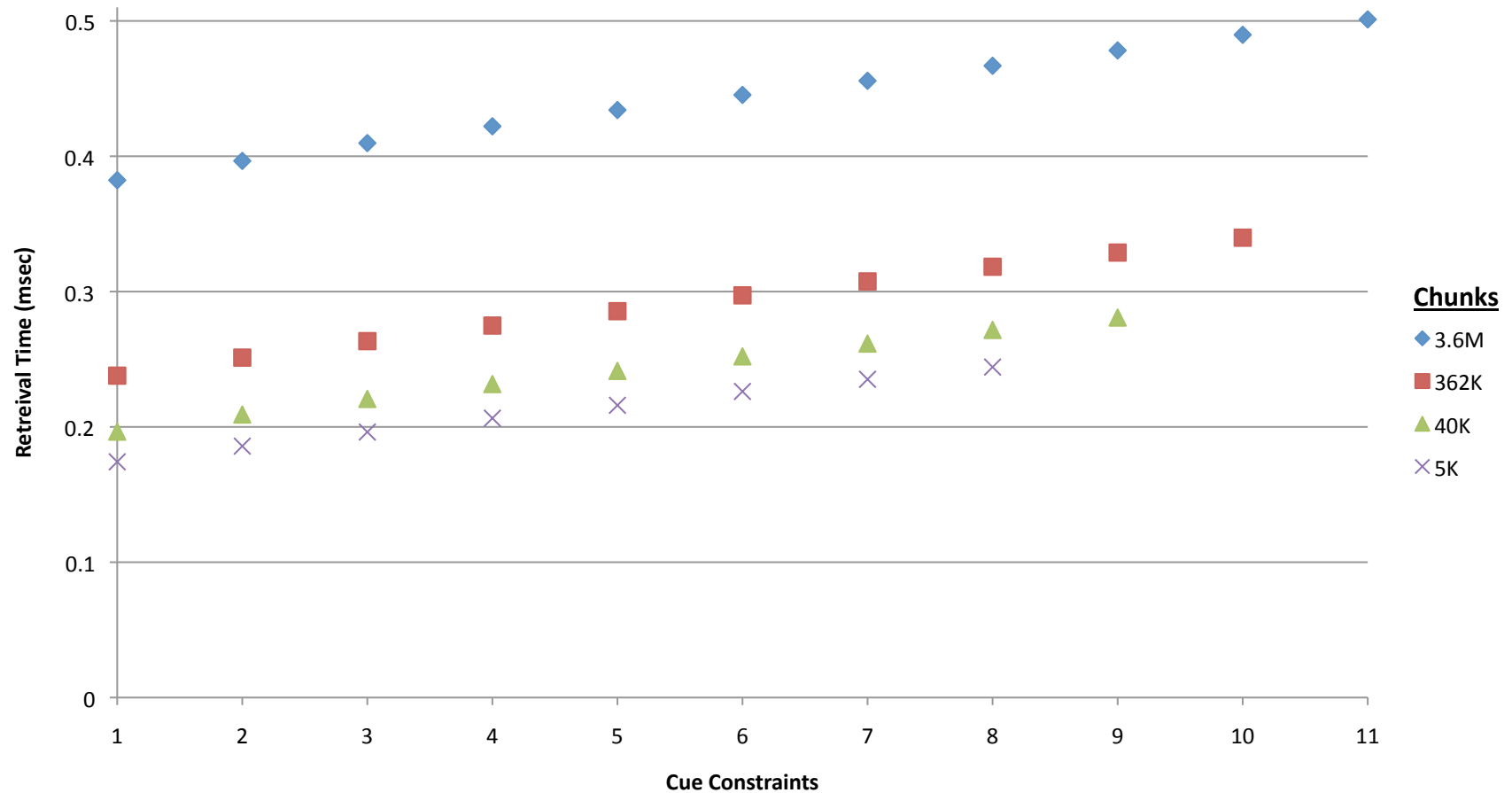
Synthetic: Data Sets

k	Chunks: k!	Slot-Value: [k+1]!	Store Size (MB)
7	5K	40K	3.00
8	40K	362K	27.81
9	362K	3.6M	291.95
10	3.6M	39.9M	2048.00

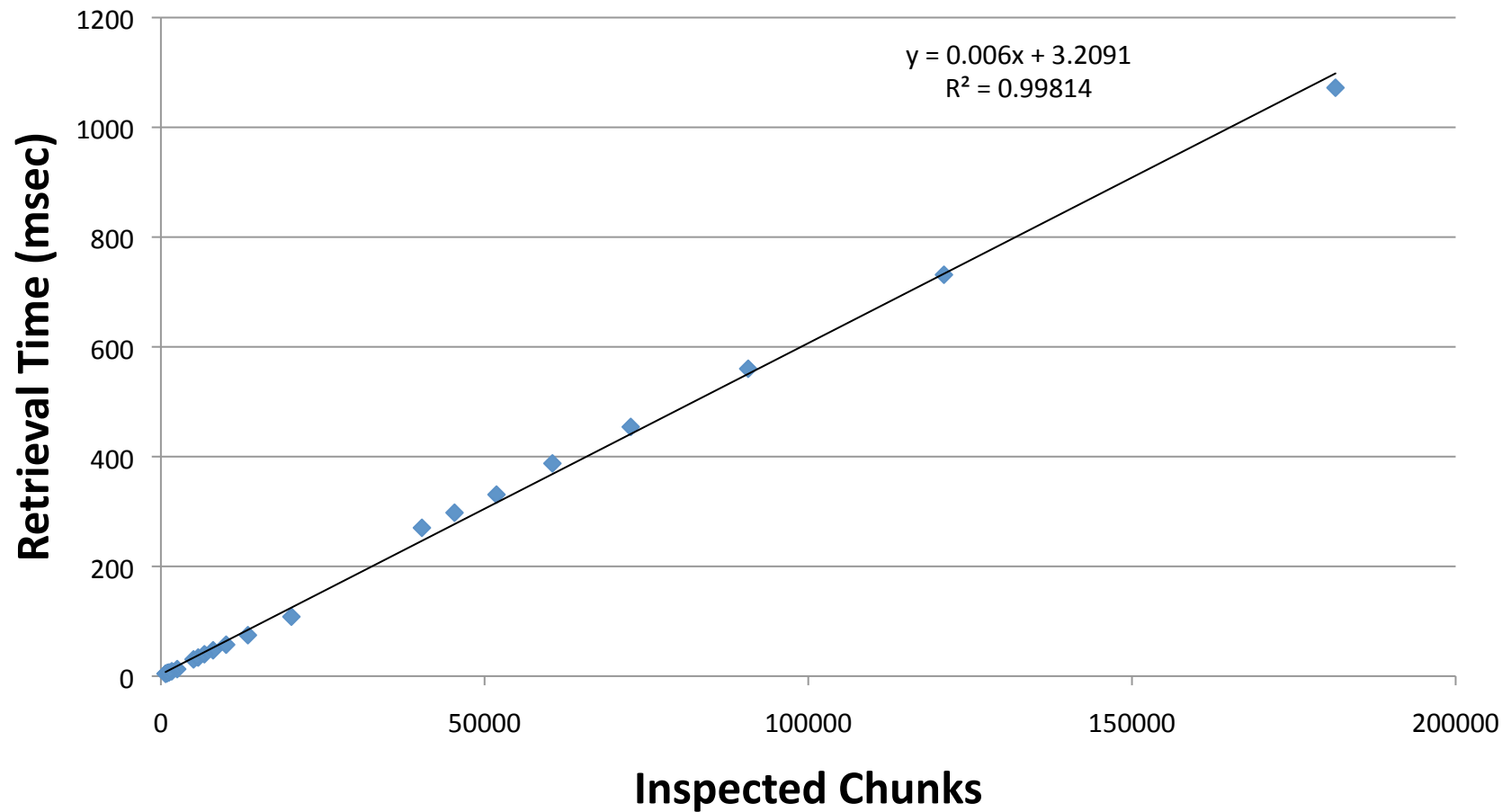
Synthetic: Selectivity Sweep



Synthetic: Successful Cue Sweep



Synthetic: Worst-Case Failure



Summary

Contributions

- Incremental extension in scaling to large DMs
 - Formal problem analysis
 - System-independent, efficient implementation
 - Efficient support for class of activation bias
 - Thorough evaluation

Future Work

- Bound worst-case performance
 - Parallelism
 - Heuristic search
- Expand efficient support for activation bias

Questions?

THANK YOU!