Fast Bag-Of-Words Candidate Selection in Content-Based Instance Retrieval Systems

Michał Siedlaczek [1] Qi Wang [1] Yen-Yu Chen [2] Torsten Suel [1]

[1] Department of Computer Science and Engineering
Tandon School of Engineering
New York University

[2] Blippar Inc.

December 12, 2018

Introduction

Problem Statement

NYU

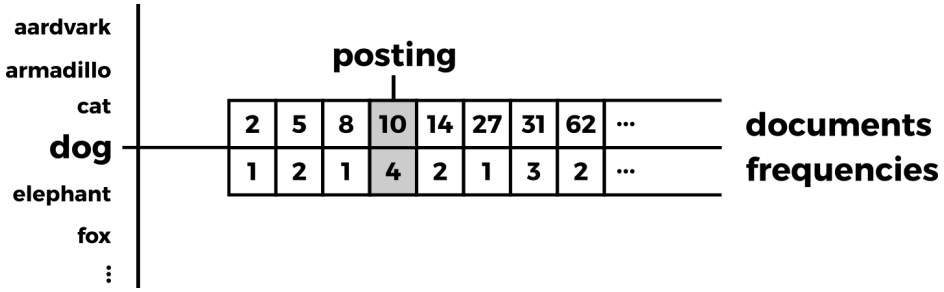
- Given a database of different types of images
  - Point phone camera at an object
  - Recognize it by finding its instance in the database
- Implemented as part of an Augmented Reality application
- General search in a broad domain

Content-Based Instance Retrieval

NYU

- **Given a picture, return its matching instance from database**
- **Bag-of-words retrieval**
  1. **Extract descriptors, robust against rotation, scaling, etc.**
     - **Convolutional Neural Networks (CNN) [Zheng 2017]**
     - **Scale-Invariant Feature Transform (SIFT) [Lowe 1999]**
  2. **Translate feature set into *visual words***
  3. **Use standard text search techniques to find candidates**
  4. **Rerank using a complex scoring method**

Inverted Index



Document Retrieval

NYU

1. Lists for query terms used to find matching documents
2. Matching documents scored to find top N candidates
3. Candidates re-ranked by a complex ranker (e.g., DNN or ML model) [Liu 2009, Wang 2010]
4. Top k < N results returned to user

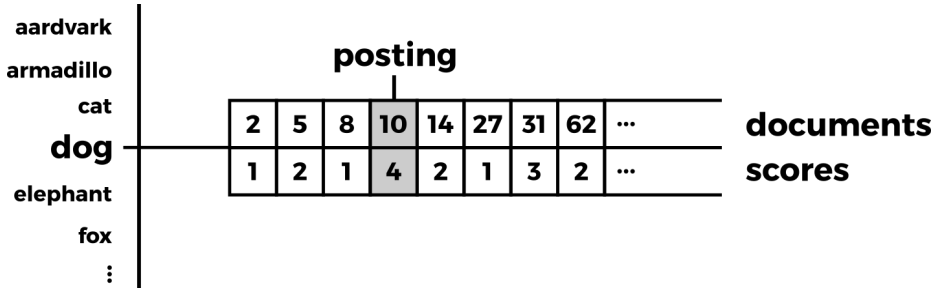
Document Retrieval

NYU

Our work:

- **Queries are pictures**
- **SIFT-generated descriptors translated to visual-word queries**
- **Partial scores stored in index and added up at query time**

Scored Inverted Index



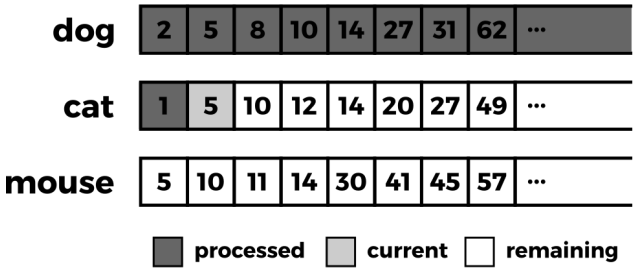
Text Retrieval Algorithms

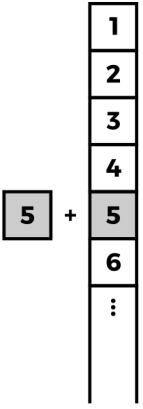
NYU

Exhaustive query processing

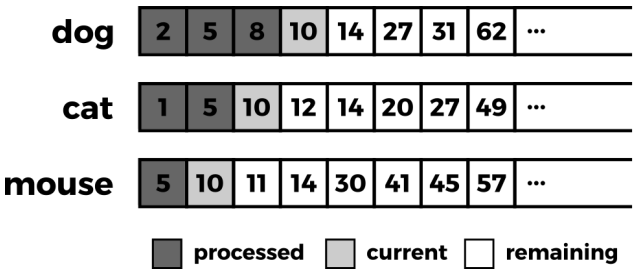
- Term at a time (TAAT)
- Document at a time (DAAT)
- Score at a time (SAAT)

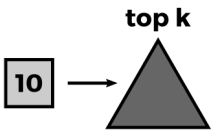
Term at a Time



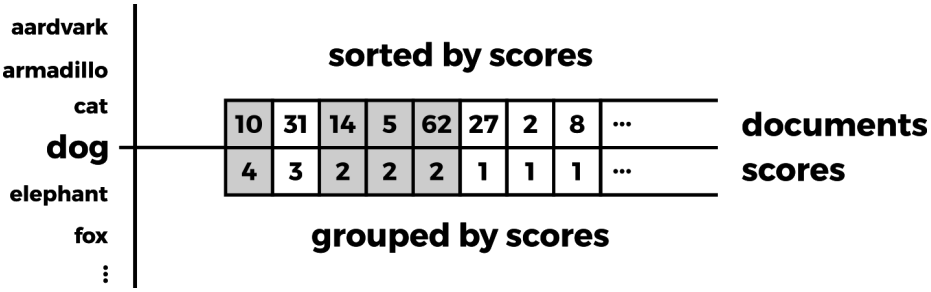


Document at a Time





Score at a Time



Safe Dynamic Pruning

NYU

Non-exhaustive processing

- **Threshold Algorithm [Fagin 1996]**
  - **Well known algorithm used in databases**
- **MaxScore [Turtle 1995]**
  - **Partitions terms/lists into essential and non-essential**
- **WAND [Broder 2003] (and variations)**
  - **Find *pivot* – a document to which all lists can be skipped without missing any top-k document**

Data Analysis

Data Analysis

NYU

Objective

Better understanding of how quantitative properties of bag-of-visual-words corpus and index may impact query efficiency.

Data Set Comparison

- **BoVW**
  - **subset of Blippar's production BoVW collection**
  - **sampled production queries**
- **Clueweb09B**
  - **standard IR text corpus**
  - **TREC 06-09 Web Query Track topics**

Data Analysis. 1: Query Lengths

NYU

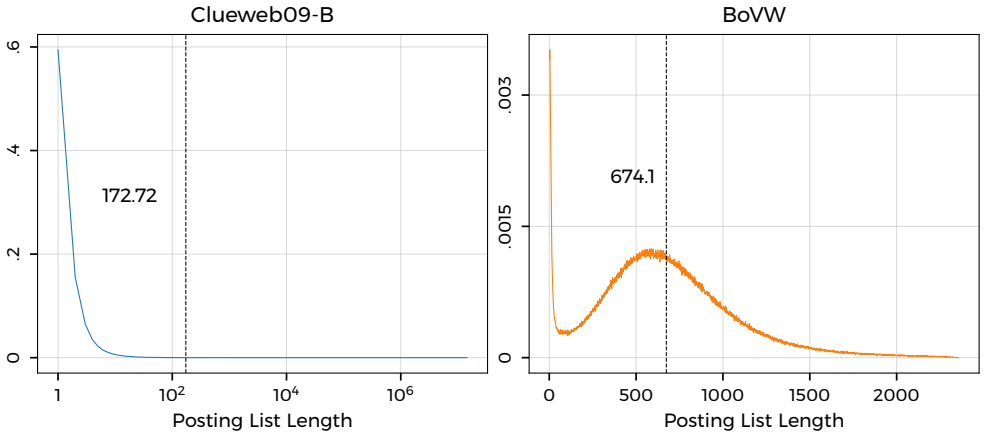
Average Query Lengths

BoVW	272
Clueweb09B	2.7

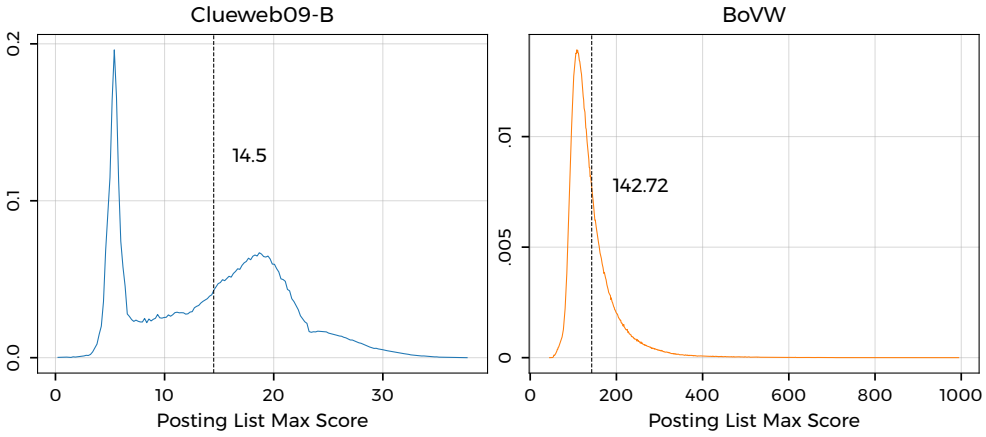
Significance

- Large overhead of selecting a posting list during processing in BoVW
- DAAT methods slow down significantly

Data Analysis. 2: Posting List Lengths



Data Analysis. 3: Posting List Max Scores



Data Analysis. 4: Length/Max Scores Correlation

NYU

- Clueweb09-B
  - strong negative correlation (-0.66)
  - Inverted Document Frequency: common words penalized by scoring functions
- BoVW
  - almost no correlation (0.06)

Significance

Potentially less advantage for dynamic pruning methods such as Max-Score.

Data Analysis. 5: Query Term Footprint

NYU

Query Term Footprint

The fraction of the query terms actually contained in the average top-k result.

Clueweb09B

- **60% – 95% depending on queries**

BoVW

- **1.1% for production queries**
- **Conjunctive queries impossible**
- **Negative impact on Max-Score algorithms — few non-essential lists to skip**

Data Analysis. 6: Index Size

NYU

Clueweb09-B

- 50 mln documents
- billions documents in real life

BoVW

- 2.6 mln documents
- about an order of magnitude more in production
- far fewer documents than most large text collections

Data Analysis. 7: Accumulator Sparsity

NYU

Clueweb09-B

- ~15% documents with non-zero scores

BoVW

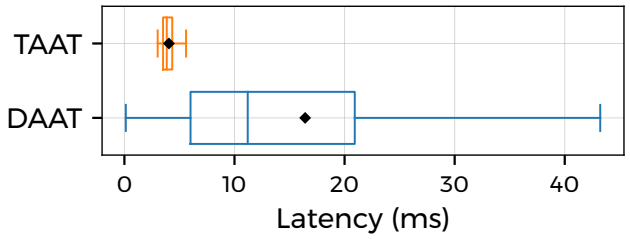
- ~8% documents with non-zero scores
- potential to improve accumulating and aggregating scores in TAAT processing

DAAT v TAAT

DAAT v TAAT

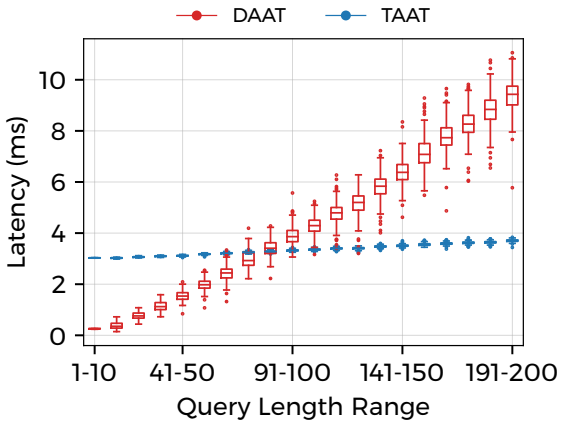
NYU

Results on BoVW



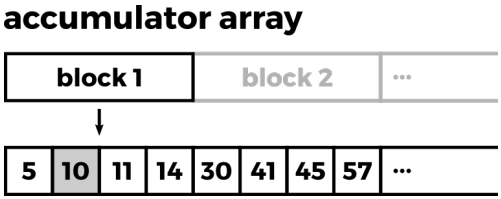
- **~75% of DAAT instructions select next posting list**

DAAT v TAAT: Query Lengths



TAAT Optimizations

TAAT Optimizations: Aggregation (A)



- Keep max of each block while traversing
- Before aggregating a block, check if max is higher than the current threshold

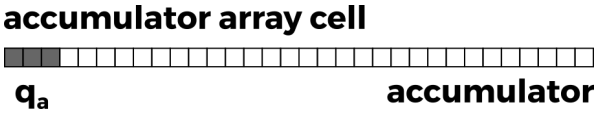
TAAT Optimizations: Prefetch (P)

NYU

- ~50% accumulator access instructions miss L1 cache
- We hint CPU to prefetch accumulators ahead of time
- Additionally, we hint that it can be evicted right after the write instruction

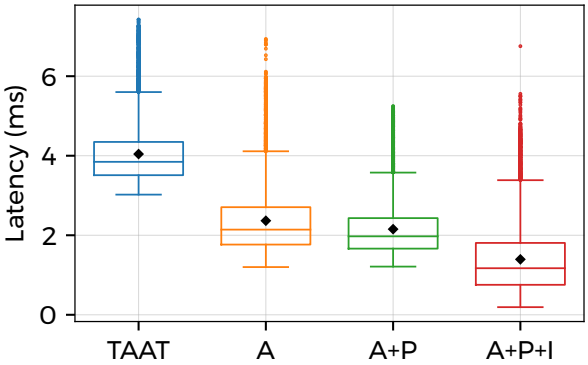
TAAT Optimizations: Accumulator Initialization (I)

NYU



- **A cyclic query counter q of size m**
- **At traversal, if $q_a < q$, the accumulator is overwritten, and $q_a \leftarrow q$**
- **Otherwise, we increase the accumulator**
- **At $q = 0$, we erase the accumulator before traversal**

TAAT Optimizations



Early Termination

Safe Early Termination

NYU

- We analyzed mechanics behind safe early termination techniques:
  - Threshold Algorithm
  - WAND
  - MaxScore
- Data proves those techniques to be inefficient

Safe Early Termination

NYU

Threshold Algorithm

On average, the stopping condition occurs after processing 98% of postings.

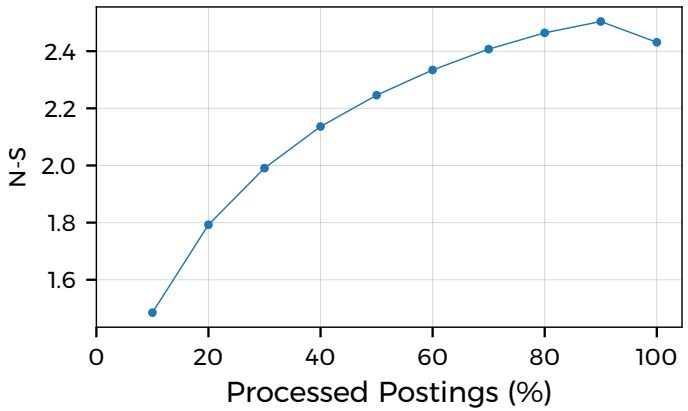
MaxScore

Given the real final threshold, 97% of terms (98% of the postings) are essential on average.

WAND

Almost 80% of the postings have to be visited on average, and over 70% have to be evaluated.

Unsafe Score at a Time



Conclusions

NYU

- CBIR bag-of-words collection and queries are much different from textual
- This impacts the efficiency of known retrieval algorithms
- TAAT outperforms DAAT due to query length
- TAAT can be further optimized to neutralize its drawbacks
- Tested early termination techniques fail in our type of scenario

Q&A

References

[Broder 2003] Broder, Carmel, Herscovici, Soffer, Zien. *Efficient query evaluation using a two-level retrieval process*

[Fagin 2001] Fagin, Lotem, Naor. *Optimal aggregation algorithms for middleware*

[Lowe 1999] Lowe. *Object recognition from local scale-invariant features*

[Turtle 1995] Turtle, Flood. *Query evaluation: strategies and optimizations*

[Zheng 2017] Zheng, Yang, Tian. *SIFT meets CNN: A decade survey of instance retrieval*