Efficient Indexing of Repeated $n$-Grams

Samuel Huston$^1$
Alistair Moffat$^2$
W. Bruce Croft$^1$

$^1$Center for Intelligent Information Retrieval, University of Massachusetts Amherst

$^2$University of Melbourne
Outline

• Definitions and Motivation

• Single CPU Indexing
  – “Standard” one pass indexing approach
  – Benefits of a second pass

• Massive Data
  – Parallel implementations
  – Fixing the problem

• Experiments using 1.5TB of English text
$n$-Gram Indexes

• Definition:
  – An $n$-gram is a sequence of $n$ adjacent words
  – A word may be represented as a variable length string or as an integer
  – For speed of execution and simplicity of analysis we use a 32 bit integer to represent each word

• Motivation for $n$-gram indexes
  – Searching for similar documents
  – Identification of reused, duplicated or plagiarized text
  – A variety of linguistic tasks; parse trees, POS tagging, ...
Repeated $n$-Gram Indexes

• In most of these tasks infrequent $n$-grams are uninteresting
  – Only index $n$-grams that occur at least $m$ times
  – Using a minimum threshold $m$ will help improve retrieval speed by significantly reducing the size of the final index
  – As $n$ increases, fraction of $n$-grams that are infrequent also increases
Outline

• Definitions and Motivation
• Single CPU Indexing
  – “Standard” one pass indexing approach
  – Benefits of a second pass
• Massive Data
  – Parallel implementations
  – Fixing the problem
• Experiments using 1.5TB of English text
Disk Based One Pass

Indexing Algorithm

• For each $n$-gram in sequence
  – Annotate with document position and any other weightings
  – Pass to sort phase

• Sort to collect like $n$-grams

• Index creation
  – Discard $n$-grams that do not appear $m$ times
  – Write out $n$-gram posting lists
**Disk Based One Pass**

- Space Complexity
  - Input data size: $N$ symbols
  - Intermediate data size: $(n+1) \times N$ symbols
  - Output data size << intermediate data size

- For example
  - $n = 10, \ N = 1 \text{ Trillion} = 4 \text{ TB}$
  - Intermediate space required up to **44 TB**

Huston 2011
Disk Based One Pass

• Time Complexity
  – Number of items to be sorted does not change with $n$
  – But as $n$ increases, intermediate data grows larger
  – Additional time is spent reading and writing larger temporary files that contain intermediate data
Peak Disk Requirements

- With any one pass algorithm
  - All $n$-grams must be passed to sort phase
  - Nothing can be discarded
- What about two passes?

---

Huston 2011
HASH BASED Filter

• During a first pass collect a $b$ bit hash value for each $n$-gram

• The size of this file is now independent of $n$

• Any hash value that occurs fewer than $m$ times corresponds to $n$-grams that can be discarded
HASH BASED Filtered Indexing

Pre-processed Corpus

First Pass

File 1
- Sort + check
- m-frequent hash values

File 2

File 3
- n-gram + positions filtered against File 2
- Sort + check
- m-frequent n-gram index

All hash values

Huston 2011
Multiple Passes?

• SPEX (Bernstein and Zobel 2006)
  – An iterative filtering approach
  – Observe that each $n$-gram contains two $n-1$-grams

  The quick brown .... ➔ The quick brown ....

• A series of filters is built, $k = 1, ..., n-1$ using $n-1$ passes

• Compared to the HASH BASED approach
  – File 1 is eliminated
  – File 2 is directly created in memory
  – But the corpus is read $n$ times
  – And File 3 might dominate space usage anyway

Huston 2011
### Space Usage for Hash Based Two Pass

<table>
<thead>
<tr>
<th>$n$</th>
<th>$f$</th>
<th>File 1</th>
<th>File 2</th>
<th>File 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>$3/4$</td>
<td>34 GB</td>
<td>13 GB</td>
<td>96 GB</td>
</tr>
<tr>
<td>5</td>
<td>$5/9$</td>
<td>34 GB</td>
<td>10 GB</td>
<td>107 GB</td>
</tr>
<tr>
<td>8</td>
<td>$1/2$</td>
<td>34 GB</td>
<td>9 GB</td>
<td>144 GB</td>
</tr>
</tbody>
</table>

- **Space usage**
  - File 1 stores a hash value for each $n$-gram
  - File 2 stores a hash value for each unique repeated $n$-gram
  - File 3 stores $n$-gram and location for each repeated $n$-gram

- **$N = 8$ Billion 32-bit symbols, 32GB of data**
- **$b = 34$ bits**
- **$f$** is the fraction of the sequence to be indexed
  - Values of $f$ are estimated using TREC Clueweb-B
Distributions of Repeated \( n \)-grams in TREC ClueWeb-B

% of Sequence

\( n \)

1 Billion Symbols or Words

Repeat

Multi

Single

Huston 2011
Single CPU Runtime, $n=8$, $m=2$
Outline

• Definitions and Motivation
• Single CPU Indexing
  – “Standard” one pass indexing approach
  – Benefits of a second pass
• Massive Data
  – Parallel implementations
  – Fixing the problem
• Experiments using 1.5TB of English text
**Distributed HASH-BASED TWO-PASS**

- Massive Data **requires** distributed processing
  - E.g. MapReduce or Sun Grid Engine

---

**Diagram**

- **Pre-processed Corpus**
  - **First Pass**: $n$-gram hash values
  - **Second Pass**: $m$-frequent $n$-grams and positions
  - **Merge**: $m$-frequent hash values
  - **Sort + Check**: $m$-frequent $n$-gram index
  - **File 1**
  - **File 2**
  - **File 3**

---

Huston 2011
HASH BASED Distributed Algorithm

• All parts of the filter (File 2) must be replicated across all processing nodes

• As the input data sequence increases
  – The size of the hash filter will increase with the final vocabulary size
  – Eventually the filter will grow too large to fit into RAM
  – And filter is randomly accessed...

• Hash based methods cannot be scalable!
LOCATION BASED Filtering

• Corpus locations are used to determine which $n$-grams to discard
  – No longer need actual hash values in application of the filter

• Filter is applied piece-wise

• Filter is distributed evenly across processing nodes
  – Each node processes a contiguous subset of locations
Space Usage of LOCATION BASED Filtering

<table>
<thead>
<tr>
<th>$n$</th>
<th>$f$</th>
<th>File 1</th>
<th>File 2</th>
<th>File 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3 / 4</td>
<td>68 GB</td>
<td>25 GB</td>
<td>96 GB</td>
</tr>
<tr>
<td>5</td>
<td>5 / 9</td>
<td>68 GB</td>
<td>20 GB</td>
<td>107 GB</td>
</tr>
<tr>
<td>8</td>
<td>1 / 2</td>
<td>68 GB</td>
<td>18 GB</td>
<td>144 GB</td>
</tr>
</tbody>
</table>

- File 1 now stores $N$ hash values and locations
- File 2 now stores a location for each repeated $n$-gram
- File 3 still stores $n$-gram and location for each repeat
- Peak disk usage determined by File 3
Outline

• Definitions and Motivation
• Single CPU Indexing
  – “Standard” one pass indexing approach
  – Benefits of a second pass
• Massive Data
  – Parallel implementations
  – Fixing the problem
• Experiments using 1.5TB of English text
Experimental Setup

• Corpus
  – ClueWeb-B TREC collection, 1.5 TB
  – Words translated into numerical values, document boundaries ignored

• Two separate computing platforms
  – Cluster of 32 Intel dual-core PCs, 2GB RAM / core
  – Cluster of 60 Intel 8-core PCs, 2GB RAM / core
  – Network attached storage is used in both cases

• Replication
  – Single CPU experiments: 10 times each
  – Distributed experiments: 5 times each

• Hash table implemented as in-RAM bit vector
  – 33 bit address used, 1GB total space, leaving 1GB for other use
Elapsed Runtime, $n=8$, $m=2$
Space Usage, $n=8$, $m=2$
TREC ClueWeb-B Experiments

- Experimental Setup and Collection Information

<table>
<thead>
<tr>
<th></th>
<th>Half</th>
<th>Whole</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpus Size (TB)</td>
<td>0.715</td>
<td>1.46</td>
</tr>
<tr>
<td>Sequence Length (x 10^9)</td>
<td>20.15</td>
<td>40.7</td>
</tr>
<tr>
<td>Numerical Data Size (GB)</td>
<td>75.1</td>
<td>150.7</td>
</tr>
<tr>
<td>Grams required (n, m)</td>
<td>8, 2</td>
<td>8, 2</td>
</tr>
<tr>
<td>Processors (p)</td>
<td>50</td>
<td>101</td>
</tr>
<tr>
<td>N / p (x 10^6)</td>
<td>402.9</td>
<td>402.9</td>
</tr>
</tbody>
</table>
Full TREC ClueWeb-B Experiments

<table>
<thead>
<tr>
<th></th>
<th>Disk Based</th>
<th>Location Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpus Size</td>
<td>Half</td>
<td>Half</td>
</tr>
<tr>
<td>Elapsed Time (sec x 10³)</td>
<td>8.5</td>
<td>12.38</td>
</tr>
<tr>
<td>Peak Disk Usage (GB)</td>
<td>417.4</td>
<td>159.9</td>
</tr>
<tr>
<td>Final Index Size (GB)</td>
<td>103.0</td>
<td>103.0</td>
</tr>
</tbody>
</table>

**Disk Based** is faster
# Full TREC ClueWeb-B Experiments

<table>
<thead>
<tr>
<th>Corpus Size</th>
<th>Disk Based</th>
<th>Location Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elapsed Time (sec x 10^3)</td>
<td>8.5</td>
<td>12.38</td>
</tr>
<tr>
<td>Peak Disk Usage (GB)</td>
<td>417.4</td>
<td>159.9</td>
</tr>
<tr>
<td>Final Index Size (GB)</td>
<td>103.0</td>
<td>103.0</td>
</tr>
</tbody>
</table>

**Disk Based** is faster

**Location Based** is more compact
# Full TREC ClueWeb-B Experiments

<table>
<thead>
<tr>
<th></th>
<th>Disk Based</th>
<th>Location Based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corpus Size</strong></td>
<td>Half</td>
<td>Half</td>
</tr>
<tr>
<td><strong>Elapsed Time (sec x 10^3)</strong></td>
<td>8.5</td>
<td>12.38</td>
</tr>
<tr>
<td><strong>Peak Disk Usage (GB)</strong></td>
<td>417.4</td>
<td>159.9</td>
</tr>
<tr>
<td><strong>Final Index Size (GB)</strong></td>
<td>103.0</td>
<td>103.0</td>
</tr>
</tbody>
</table>

**Disk Based** is faster

**Location Based** is more compact

**Location Based** also scales well
Discussion + Conclusions

• **Disk Based One Pass** – Non-Filtered approach
  – Clearly the fastest approach
  – Straight forward scalable parallelization
  – But disk consumption grows with $n$

• **Location Based Two Pass** – Filtered approach
  – Most efficient space usage – disk consumption not linked to $n$
  – Time scalable for single CPU and in a distributed environment
  – Designed for parallelization
  – Provides alternative compromise for finding repeated $n$-grams
Questions and Comments
SPEX Filtered Indexing

Pre-processed Corpus

Filter Construction Passes

1-Filter

2-Filter

... 

Final Pass

File 3

n-gram + positions filtered against File 2

Sort + Check

m-frequent n-gram Index

Huston 2011
Distributions of Repeated n-grams

Randomly generated Zipfian data

GOV2 English textual Data

Term Frequency vs. Term Rank in Frequency Table

Huston 2011
## Distributions of Repeated n-grams in TREC ClueWeb-B

<table>
<thead>
<tr>
<th>n</th>
<th>250 M Symbols / Words</th>
<th>1,000 M Symbols / Words</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Single</td>
<td>Multi</td>
</tr>
<tr>
<td>1</td>
<td>0.5</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>8.3</td>
<td>4.9</td>
</tr>
<tr>
<td>3</td>
<td>26.2</td>
<td>9.5</td>
</tr>
<tr>
<td>4</td>
<td>40.9</td>
<td>10.4</td>
</tr>
<tr>
<td>5</td>
<td>48.4</td>
<td>9.9</td>
</tr>
<tr>
<td>6</td>
<td>51.9</td>
<td>9.5</td>
</tr>
<tr>
<td>7</td>
<td>53.7</td>
<td>9.2</td>
</tr>
<tr>
<td>8</td>
<td>54.9</td>
<td>9.1</td>
</tr>
<tr>
<td>9</td>
<td>55.9</td>
<td>8.9</td>
</tr>
<tr>
<td>10</td>
<td>56.7</td>
<td>8.8</td>
</tr>
<tr>
<td>20</td>
<td>61.8</td>
<td>8.0</td>
</tr>
<tr>
<td>30</td>
<td>65.0</td>
<td>7.6</td>
</tr>
</tbody>
</table>

Huston 2011
# Full TREC ClueWeb-B Experiments

<table>
<thead>
<tr>
<th></th>
<th>Disk Based</th>
<th>Location Based</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Corpus Size (TB)</strong></td>
<td>0.715</td>
<td>0.715</td>
</tr>
<tr>
<td><strong>Grams required (n, m)</strong></td>
<td>8, 2</td>
<td>8, 2</td>
</tr>
<tr>
<td><strong>Sequence Length (x 10^9)</strong></td>
<td>20.15</td>
<td>20.15</td>
</tr>
<tr>
<td><strong>Numerical Data Size (GB)</strong></td>
<td>75.1</td>
<td>75.1</td>
</tr>
<tr>
<td><strong>Processors (p)</strong></td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td><strong>N / p (x 10^6)</strong></td>
<td>402.9</td>
<td>402.9</td>
</tr>
<tr>
<td><strong>Elapsed Time (sec x 10^3)</strong></td>
<td>8.5</td>
<td>12.38</td>
</tr>
<tr>
<td><strong>Peak Disk Usage (GB)</strong></td>
<td>417.4</td>
<td>159.9</td>
</tr>
<tr>
<td><strong>Final Index Size (GB)</strong></td>
<td>103.0</td>
<td>103.0</td>
</tr>
</tbody>
</table>
TREC ClueWeb-B
Recurrence Statistics, $n=8$, $m=2$

- Corpus Size (Billions of Symbols)
  - 1
  - 20.15
  - 40.7

- Correlation
  - Repeat
  - Multi
  - Single

Huston 2011
Future Problems

• Investigate higher values of $m$
  – Will change relative sizes of File 1 and File 3

• Other data sources are likely to produce different degrees of recurrence
  – How varied are other English corpora?

• Un-ordered $n$-gram variations may also be of interest
  – Likely to have fewer discarded $n$-grams

• $n$-gram index compression schemes
  – Vocabulary dominates the space usage