Representing Huge Translation Models
Statistical Machine Translation

parallel text + alignment
Statistical Machine Translation

extract rules

parallel text + alignment
Statistical Machine Translation

parallel text + alignment
Statistical Machine Translation

parallel text + alignment

extract rules

score rules

load rules into memory

decoder
Statistical Machine Translation

parallel text + alignment

extract rules

score rules

load rules into memory

number of rules depends on corpus size...

decoder
Statistical Machine Translation

parallel text + alignment

extract rules

score rules

load rules into memory

... and model complexity

decoder
Statistical Machine Translation

parallel text + alignment

extract rules

score rules

filter rules for test set

load filtered rules into memory

decoding algorithm
Baseline Translation Model

• Hierarchical Phrase-based translation (Chiang 2007)
• 1M parallel sentences (27M words)
• GIZA++ alignments (Och & Ney 2003, Koehn et al. 2003)
  • alignments are dense
• Heuristics used to restrict number of extracted rules
• 67M rules, 6.1Gb of data
  • cf. 225M (Zens & Ney 2007), 55M (DeNeefe et al. 2007)
Some Possible Improvements

- 3.5M sentences (2.5M out-of-domain), 100M words
- Discriminatively trained alignments (Ayan & Dorr 2006)
  - Key difference: alignments are *sparse*
- *Loose* phrase extraction (Ayan & Dorr 2006)
Some Possible Improvements

- 3.5M sentences (2.5M out-of-domain), 100M words
- Discriminatively trained alignments (Ayan & Dorr 2006)
  - Key difference: alignments are sparse
- *Loose* phrase extraction (Ayan & Dorr 2006)
Some Possible Improvements

- 3.5M sentences (2.5M out-of-domain), 100M words
- Discriminatively trained alignments (Ayan & Dorr 2006)
  - Key difference: alignments are *sparse*
- *Loose* phrase extraction (Ayan & Dorr 2006)
Some Possible Improvements

• 3.5M sentences (2.5M out-of-domain), 100M words
• Discriminatively trained alignments (Ayan & Dorr 2006)
  • Key difference: alignments are sparse
• Loose phrase extraction (Ayan & Dorr 2006)
Some Possible Improvements

- 3.5M sentences (2.5M out-of-domain), 100M words
- Discriminatively trained alignments (Ayan & Dorr 2006)
  - Key difference: alignments are \textit{sparse}
- \textit{Loose} phrase extraction (Ayan & Dorr 2006)
Some Possible Improvements

- Rule extraction time: 77 CPU days
  - does not include sorting or scoring!
- Rules counted: 20 billion
  - 2 orders of magnitude larger than state of the art
- Estimated unique rules: 6.6 billion
- Estimated extract file size: 917Gb
- Estimated phrase table size: 600Gb
The Problem

- Current models are bounded by resource limitations.
- We’re already pushing the edge of what’s possible.
- Parallel data aren’t getting any smaller.
- Models aren’t getting any less complex.
The Solution

- Translation by pattern matching.
- Novel pattern matching algorithms.
  - Exploit ideas developed in bioinformatics, IR
- Support for tera-scale translation models.
Idea: Translation by Pattern Matching  
(Callison-Burch et al. 05, Zhang & Vogel 05)
Exact Pattern Matching

Input Pattern: it persuades him and it disheartens him
Exact Pattern Matching

Input Pattern  it persuades him and it disheartens him
=Query Pattern
Pattern Matching for Phrase-Based MT

Input Pattern  it persuades him and it disheartens him
Pattern Matching for Phrase-Based MT

Input Pattern    it persuades him and it disheartens him

Query Patterns   it
                 persuades
                 him
                 and
                 disheartens
                 it persuades
                 persuades him
                 him and
                 and it
                 it disheartens
                 disheartens him

                          it persuades him
                          persuades him and
                          him and it
                          and it disheartens
                          it disheartens him
                          it persuades him and
                          persuades him and it
                          him and it disheartens
                          and it disheartens him
                          it persuades him and it
                          persuades him and it disheartens
                          him and it disheartens him
Suffix Arrays

it makes him and it mars him, it sets him on and it takes him off.

Text $T$
Suffix Arrays

it makes him and it mars him, it sets him on and it takes him off. #

Text $T$

4 it mars him, it sets him on and it takes him off. #

Suffix 4
Suffix Arrays

it makes him and it mars him, it sets him on and it takes him off. #
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

0 it makes him and it mars him, it sets him on and it takes him ...
1 makes him and it mars him, it sets him on and it takes him off. #
2 him and it mars him, it sets him on and it takes him off. #
3 and it mars him, it sets him on and it takes him off. #
4 it mars him, it sets him on and it takes him off. #
5 mars him, it sets him on and it takes him off. #
6 him, it sets him on and it takes him off. #
7 , it sets him on and it takes him off. #
...
Suffix Arrays

it makes him and it mars him, it sets him on and it takes him off. #
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

3 and it mars him, it sets him on and it takes him off. #
12 and it takes him off. #
2 him and it mars him, it sets him on and it takes him off. #
15 him off. #
10 him on and it takes him off. #
6 him, it sets him on and it takes him off. #
0 it makes him and it mars him, it sets him on and it takes him ...
Suffix Arrays

it makes him and it mars him, it sets him on and it takes him off. #
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

3 and it mars him, it sets him on and it takes him off. #
12 and it takes him off. #
2 him and it mars him, it sets him on and it takes him off. #
15 him off. #
10 him on and it takes him off. #
6 him, it sets him on and it takes him off. #
0 it makes him and it mars him, it sets him on and it takes him ...
4 it mars him, it sets him on and it takes him off. #
...
Suffix Arrays

it makes him and it mars him. it sets him on and it takes him off. #

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Text $T$

3 12 2 15 10 6 0 4 8 13 1 5 16 11 9 14 7 17 18

Suffix Array $SA$
Suffix Arrays

it makes him and it mars him . it sets him on and it takes him off . 

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Text $T$

3 12 2 15 10 6 0 4 8 13 1 5 16 11 9 14 7 17 18

Suffix Array $SA$

him and it

Query Pattern $w$
Suffix Arrays

it makes him and it mars him. it sets him on and it takes him off. #

Text $T$

| 3 | 12 | 2 | 15 | 10 | 6 | 0 | 4 | 8 | 13 | 1 | 5 | 16 | 11 | 9 | 14 | 7 | 17 | 18 |

Suffix Array $SA$

him and it

Query Pattern $w$
Suffix Arrays

it makes him and it mars him. it sets
him on and it takes him off.

Text $T$

| 3 | 12 | 2 | 15 | 10 | 6 | 0 | 4 | 8 | 13 | 1 | 5 | 16 | 11 | 9 | 14 | 7 | 17 | 18 |

Suffix Array $SA$

him and it

Query Pattern $w$
Suffix Arrays

it makes him and it mars him. it sets him on and it takes him off. #

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Text $T$

$3 \ 12 \ 2 \ 15 \ 10 \ 6 \ 0 \ 4 \ 8 \ 13 \ 1 \ 5 \ 16 \ 11 \ 9 \ 14 \ 7 \ 17 \ 18$

Suffix Array $SA$

him and it

Query Pattern $w$
Suffix Arrays

it makes him and it mars him. it sets him on and it takes him off. #

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Text $T$

3 12 2 15 10 6 0 4 8 13 1 5 16 11 9 14 7 17 18

Suffix Array $SA$ $O(|w| \log |T|)$

him and it

Query Pattern $w$
Suffix Arrays

it makes him and it mars him. it sets him on and it takes him off.

Text $T$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 3 | 12 | 2 | 15 | 10 | 6 | 0 | 4 | 8 | 13 | 1 | 5 | 16 | 11 | 9 | 14 | 7 | 17 | 18 |

Suffix Array $SA$  

$O(|w| \log |T|)$  

$O(|w| + \log |T|)$ (Manber & Myers, 93)

him and it

Query Pattern $w$
Suffix Arrays

it makes him and it mars him . it sets him on and it takes him off . #

Text $T$

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
|  3 | 12 |  2 | 15 | 10 |  6 |  0 |  4 |  8 | 13 |  1 |  5 | 16 | 11 |  9 | 14 |  7 | 17 | 18 |

Suffix Array $SA$

$O(|w| \log |T|)$

$O(|w| + \log |T|)$ (Manber & Myers, 93)

$O(|w|)$ (Abouelhoda et al., 04)

Query Pattern $w$
Suffix Arrays

it makes **him and it mars him**. it sets him on and it takes him off. #

Text $T$

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
</tr>
</tbody>
</table>

Suffix Array $SA$

$O(|w| \log |T|)$

$O(|w| + \log |T|)$ (Manber & Myers, 93)

$O(|w|)$ (Abouelhoda et al., 04)

Query Pattern $w$

on baseline model:

0.009 seconds / sentence

(not including extraction/scoring)
Problem: Phrases with Gaps

- Hierarchical phrase-based translation (Chiang 2005, 2007)

Input: it persuades him and it disheartens him

Source Phrase: it X him
Hierarchical Phrases: Phrases with Gaps

- Hierarchical phrase-based translation (Chiang 2005, 2007)

Input: it persuades him and it disheartens him

Source Phrase: it X him
Hierarchical Phrases: Phrases with Gaps

- Hierarchical phrase-based translation (Chiang 2005, 2007)

Input

Source Phrase

it X him
Hierarchical Phrases: Phrases with Gaps

- Hierarchical phrase-based translation (Chiang 2005, 2007)

Input: it persuades him and it disheartens him

Source Phrase: it X him
Hierarchical Phrases: Phrases with Gaps

- Hierarchical phrase-based translation (Chiang 2005, 2007)

Input: it persuades him and it disheartens him

Source Phrase: it X and X him
Problem Statement

Given an input sentence, efficiently find all hierarchical phrase-based translation rules for that sentence in the training corpus.
Pattern Matching for Hierarchical PBMT

Input Pattern: it persuades him and it disheartens him
Pattern Matching for Hierarchical PBMT

Input Pattern it persuades him and it disheartens him

Query Patterns

it persuades him
it persuades him and it
it persuades him and it disheartens
it persuades him and it disheartens him
it persuades him and it disheartens him
it persuades him and it disheartens him
it persuades him and it disheartens him
it persuades him and it disheartens him
it persuades him and it disheartens him
it persuades him and it disheartens him
Pattern Matching for Hierarchical PBMT

Input Pattern  it persuades him and it disheartens him

Query Patterns  

<table>
<thead>
<tr>
<th>it X and</th>
</tr>
</thead>
<tbody>
<tr>
<td>it X it</td>
</tr>
<tr>
<td>it X disheartens</td>
</tr>
<tr>
<td>it X him</td>
</tr>
<tr>
<td>persuades X it</td>
</tr>
<tr>
<td>persuades X disheartens</td>
</tr>
<tr>
<td>persuades X him</td>
</tr>
<tr>
<td>it persuades X it</td>
</tr>
<tr>
<td>it persuades X disheartens</td>
</tr>
<tr>
<td>it persuades X him</td>
</tr>
<tr>
<td>it X and it</td>
</tr>
<tr>
<td>it X it disheartens</td>
</tr>
<tr>
<td>it X disheartens him</td>
</tr>
<tr>
<td>it X and X him</td>
</tr>
<tr>
<td>persuades him X disheartens</td>
</tr>
<tr>
<td>persuades him X him</td>
</tr>
<tr>
<td>persuades X it disheartens</td>
</tr>
<tr>
<td>persuades X disheartens him</td>
</tr>
<tr>
<td>him and X him</td>
</tr>
<tr>
<td>him X disheartens him</td>
</tr>
<tr>
<td>it persuades him X disheartens</td>
</tr>
<tr>
<td>it persuades him X him</td>
</tr>
<tr>
<td>it persuades X it disheartens</td>
</tr>
<tr>
<td>it persuades X disheartens him</td>
</tr>
</tbody>
</table>
Pattern Matching for Hierarchical PBMT

Input Pattern
it persuades him and it disheartens him
it X and it disheartens
it X it disheartens him
persuades him and X him
persuades him X disheartens him
persuades X it disheartens him
it persuades him and X him
it persuades him X disheartens him
it persuades X it disheartens him
it X and it disheartens him

Query Patterns
Pattern Matching for Hierarchical PBMT

Input Pattern
it persuades him and it disheartens him

Query Patterns
it X and it disheartens
it X it disheartens him
persuades him and X him
persuades him X disheartens him
persuades X it disheartens him
it persuades him and X him
it persuades him X disheartens him
it persuades X it disheartens him
it X and it disheartens him
it X and it disheartens him

This is a variant of *approximate* pattern matching (Navarro ‘01)
Pattern Matching with Gaps

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>2</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Query pattern $\alpha$

him $X$ it

and it mars him, it sets him ...
and it takes him off. #
him and it mars him. it sets ...
him off. #
him on and it takes him off. #
him, it sets him on and it ...
it makes him and it mars ...
it mars him, it sets him on ...
it sets him on and it takes ...
it takes him off. #
makes him and it mars him ...
Pattern Matching with Gaps

Query pattern $\alpha$

him X it

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>2</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>4</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

and it mars him, it sets him ...
and it takes him off. #
him and it mars him. it sets ...
him off. #
him on and it takes him off. #
him, it sets him on and it ...

it makes him and it mars ...
it mars him, it sets him on ...
it sets him on and it takes ...
it takes him off. #

makes him and it mars him ...
Pattern Matching with Gaps

Query pattern $\alpha$

him X it

<table>
<thead>
<tr>
<th></th>
<th>3</th>
<th>and it mars him , it sets him ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td></td>
<td>and it takes him off . #</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>him off . #</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td><strong>him and it mars him . it sets</strong></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td><strong>him , it sets him on and it</strong></td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>it makes him and it mars ...</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>it mars him , it sets him on ...</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>it sets him on and it takes ...</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>it takes him off . #</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>makes him and it mars him ...</td>
</tr>
</tbody>
</table>
Pattern Matching with Gaps

Query pattern $\alpha$

him X it

Subpatterns $w_i$

him

it

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>2</td>
<td>15</td>
<td>10</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>13</td>
<td>1</td>
<td>:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

... and it mars him, it sets him ...

and it takes him off. #

him and it mars him. it sets ...

him off. #

him on and it takes him off. #

him, it sets him on and it ...

it makes him and it mars ...

it mars him, it sets him on ...

it sets him on and it takes ...

it takes him off. #

makes him and it mars him ...
Pattern Matching with Gaps

Query pattern $\alpha$

him X it

Subpatterns $w_i$

him

it

and it mars him, it sets him ...

him and it mars him. it sets ...

him off. #

him on and it takes him off. #

him, it sets him on and it ...

it makes him and it mars ...

it mars him, it sets him on ...

it sets him on and it takes ...

it takes him off. #

makes him and it mars him ...

...
Pattern Matching with Gaps

<table>
<thead>
<tr>
<th>Query pattern $\alpha$</th>
<th>Occurrences $n_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>him $X$ it</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subpatterns $w_i$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>him</td>
<td>3</td>
</tr>
<tr>
<td>it</td>
<td>12</td>
</tr>
</tbody>
</table>

and it mars him, it sets him ...

and it takes him off. #

him and it mars him. it sets ...

him off. #

him on and it takes him off. #

him, it sets him on and it ...

it makes him and it mars ...

it mars him, it sets him on ...

it sets him on and it takes ...

it takes him off. #

makes him and it mars him ...

...
Pattern Matching with Gaps

and it mars him, it sets him ...
and it takes him off. #
him and it mars him. it sets ...
him off. #
him on and it takes him off. #
him, it sets him on and it ...
it makes him and it mars ...
it mars him, it sets him on ...
it sets him on and it takes ...
it takes him off. #
makes him and it mars him ...
Pattern Matching with Gaps

\[
\begin{array}{c|c|c}
2 & 0 & 0 \\
15 & 4 & 4 \\
10 & 8 & 8 \\
6 & 13 & 13 \\
\end{array}
\]
Pattern Matching with Gaps

(2, 4)
(2, 8)
(2, 13)
(6, 8)
(6, 13)
(10, 13)
Pattern Matching with Gaps

RILMS (Rahman et al., 06)
Pattern Matching with Gaps

RILMS (Rahman et al., 06)

linear in number of occurrences of subpatterns: $O(\sum_i n_i)$
Baseline Timing Result

221 seconds per sentence

compare: 0.009 seconds per sentence for contiguous phrases
Complexity Analysis

contiguous

$$\sum_w (|w| + \log |T|)$$

137  5  27

discontiguous

$$\sum_{\alpha=w_1 \ldots w_I} \sum_{i=1}^I (|w_i| + \log |T| + n_i)$$

2825  3  5  27  82069
Complexity Analysis

contiguous

\[ \sum_{w} (|w| + \log |T|) \]

137 5 27

discontiguous

\[ \sum_{\alpha=w_1X\ldots Xw_I} \sum_{i=1}^{I} (|w_i| + \log |T| + n_i) \]

2825 3 5 27 82069
Exploiting Redundancy

Input Pattern  it persuades him and it disheartens him

Query Patterns
- it X and
- it X it
- it X disheartens
- it X him
- persuades X it
- persuades X disheartens
- persuades X him
- it persuades X it
- it persuades X disheartens
- it persuades X him
- it X and it
- it X it disheartens

it X disheartens him
it X and X him
persuades him X disheartens
persuades him X him
persuades X it disheartens
persuades X it disheartens him
persuades X disheartens him
persuades X disheartens him
persuades X disheartens him
persuades X disheartens him
Exploiting Redundancy

Input Pattern  it persuades him and it disheartens him

Query Patterns

it X and
it X it
it X disheartens
it X him
persuades X it
persuades X disheartens
persuades X him
it persuade X it
it persuade X disheartens
it persuade X him
it X and it
it X it disheartens

it X disheartens him
it X and X him
persuades him X disheartens
persuades him X him
persuades X it disheartens
persuades X disheartens him
persuades X disheartens him
him and X him
him X disheartens him
it persuade X him X disheartens
it persuade X him X him
it persuade X it disheartens
it persuade X disheartens him
it persuade X disheartens him
Exploiting Redundancy

Query Pattern

it persuades X disheartens him
Exploiting Redundancy

Query Pattern: it persuades X disheartens him
Maximal Prefix: it persuades X disheartens

(Zhang & Vogel 2005)
Exploiting Redundancy

Query Pattern: it persuades X disheartens him
Maximal Prefix: it persuades X disheartens
Maximal Suffix: persuades X disheartens him
Prefix Tree with Suffix Links
Timing Results

seconds/ sentence

Baseline

221
Timing Results

<table>
<thead>
<tr>
<th></th>
<th>Seconds/sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>221</td>
</tr>
<tr>
<td>Prefix Tree</td>
<td>177</td>
</tr>
</tbody>
</table>
Complexity Analysis

contiguous

\[
\sum_{w} (|w| + \log |T|)
\]

137 5 27

discontiguous

\[
\sum_{\alpha=w_1 X \ldots X w_I} \sum_{i=1}^{I} (|w_i| + \log |T| + n_i)
\]

2825 3 5 27 82069
Complexity Analysis

contiguous

\[ \sum_{w} (|w| + \log |T|) \]

137 \hspace{1cm} 5 \hspace{1cm} 27

discontiguous

\[
\sum_{\alpha=w_1X\ldots Xw_I} \sum_{i=1}^{I} (|w_i| + \log |T| + n_i)
\]

2825 \hspace{1cm} 3 \hspace{1cm} 5 \hspace{1cm} 27 \hspace{1cm} 82069
Empirical Analysis

computations (ranked by time)
Distribution of Patterns in Training Data

Pattern types (in descending order of frequency)
Distribution of Patterns in Training Data

Pattern types (in descending order of frequency)
Analysis of Problem

• The expensive computations involve at least one frequent subpattern. There are two cases.
  • A frequent pattern paired with an infrequent pattern
  • Two frequent patterns paired with each other
Frequent $\times$ Infrequent Subpatterns
Frequent × Infrequent Subpatterns
Frequent $\times$ Infrequent Subpatterns
Frequent × Infrequent Subpatterns
Double Binary Search
Baeza-Yates, 04
Double Binary Search
Baeza-Yates, 04

Queryset $Q$

Dataset $D$
Double Binary Search
Baeza-Yates, 04

Queryset $Q$  
Dataset $D$
Double Binary Search
Baeza-Yates, 04
Double Binary Search

Baeza-Yates, 04

Queryset $Q$

Dataset $D$
Double Binary Search
Baeza-Yates, 04

Queryset $Q$

Dataset $D$
Double Binary Search
Baeza-Yates, 04

Queryset $Q$  Dataset $D$
Double Binary Search

Baeza-Yates, 04

Upper bound complexity: $|Q| \log |D|$
Obtaining Sorted Sets
Obtaining Sorted Sets

Sort via Stratified Tree
(van Emde Boas et al. 1977)
Obtaining Sorted Sets

Sort via Stratified Tree
(van Emde Boas et al. 1977)

Problem: complexity increases to
\[ O(|Q| \log |D| + (|Q| + |D|) \log \log |T|) \]
Obtaining Sorted Sets

Problem: complexity increases to

Solution: cache sorted set in prefix tree

Sort via Stratified Tree
(van Emde Boas et al. 1977)

Problem: complexity increases to

\[ O(|Q| \log |D| + (|Q| + |D|) \log \log |T|) \]
Timing Results

<table>
<thead>
<tr>
<th></th>
<th>Seconds/ Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>221</td>
</tr>
<tr>
<td>Prefix Tree + double binary</td>
<td>177</td>
</tr>
</tbody>
</table>
Timing Results

seconds/ sentence

Baseline: 221
Prefix Tree: 177
+ double binary: 174
Obtaining Sorted Sets

Sort via Stratified Tree

Problem: sort complexity is still very high for very frequent patterns
Obtaining Sorted Sets

Solution: precompute the \textit{inverted index} for 1000 most frequent contiguous patterns
Timing Results

seconds/sentence

Baseline: 221
Prefix Tree: 177
+ double binary: 174
Timing Results

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Prefix Tree</th>
<th>+ double binary</th>
<th>+ inverted indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>seconds/sentence</td>
<td>221</td>
<td>177</td>
<td>174</td>
<td>44</td>
</tr>
</tbody>
</table>
Frequent \times Frequent Subpatterns
Problem:
There is no clever algorithm to solve this problem
Solution: Precomputation

it makes him and it mars him. it sets him on and it takes him off. #
Solution: Precomputation

it makes him and it mars him. it sets him on and it takes him off. #

Text

Most Frequent Patterns
it (4)           Precomputed Pattern Matches
it X him
him (4)         him X it
               it X it
               him X him
Solution: Precomputation

it makes him and it mars him. it sets him on and it takes him off. #

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Text

Most Frequent Patterns   Precomputed Pattern Matches

it (4)                   it X him
him (4)                  him X it
(0, 2)(0, 6)(13, 15)     (2, 4)(2, 8)(10, 13)
(4, 6)(4, 10)(8, 15)     (6, 8)(6, 13)
(0, 4)(0, 8)             (2, 6)(2, 10)(10, 15)
(4, 8)(4, 13)(8, 13)     (6, 10)(6, 15)
Timing Results

seconds/sentence

Baseline: 221
Prefix Tree: 177
+ double binary: 174
+ inverted indices: 44
Timing Results

- Baseline: 221 seconds/sentence
- Prefix Tree: 177 seconds
- Prefix Tree + double binary: 174 seconds
- Prefix Tree + inverted indices: 44 seconds
- Prefix Tree + precomp: 1 second
Analysis of Fixed Memory Usage

- Source Text: $|T|$
- Suffix Array: $|T|$
- Alignments: $|T|$
- Target Text: $|T|$
- Total Cost: 4 $|T|$
- For 27M words: about 700M
- including indices for 1000 words: about 2.1 Gb
  - for 100 words: 1.1Gb, increases time to 1.6 secs/sent
Longer Spans, Longer Phrases

Maximum Span Length

Maximum Phrase Length

BLEU

BLEU
The Tera-Scale Translation Model

- Task: NIST Chinese-English 2005
- Baseline Model: 30.7
- Tera-Scale Model: 32.6
- All modifications contribute to overall score
- With better language model and number translation:
  - Baseline Model: 31.9
  - Tera-Scale Model: 34.5
Open Questions

- Can we improve speed?
- Can we improve memory use? *Compressed self-indexes*?
- Uses for arbitrarily large translation models?
  - Context-sensitive models (Chan et al. 2007, Carpuat & Wu 2007)
  - Factored models (Koehn et al. 2007)
  - Syntax-based model (DeNeefe et al. 2007)
- What other algorithms can we use from bioinformatics?
Thanks

Acknowledgements:
David Chiang, Chris Dyer, Philip Resnik