
Scaling Phrase-Based Statistical Translation to Larger Corpora and Longer Phrases

Chris Callison-Burch,
Colin Bannard and Josh Schroeder

LINEAR B



Talk Overview

- Motivation
- Suffix arrays for parallel corpora
- Speed-memory tradeoff
- Usefulness of sampling

Availability of Training Data

- SMT is unlike most other statistical NLP tasks
- Large amounts of training data for some languages
- EU, UN web sites can be harvested
 - 30 million words in 11 languages in Europarl corpus
- NIST Arabic-English, Chinese-English corpora
 - >100 million words in each language

Usefulness of Phrases in SMT

- Larger chunks of human translated text
- Directly handle multi-word items, such as idioms
- Less re-ordering needed
- Using longer phrases leads to better translation quality

Coverage of Phrases

- Coverage of NIST-2004 Arabic-English test set:

length	coverage	length	coverage
1	93.5%	6	4.70%
2	73.3%	7	2.95%
3	37.1%	8	2.14%
4	15.5%	9	1.99%
5	8.05%	10	1.49%

Table sizes

- NIST 2004 Arabic-English training data

phrase length	unique phrase pairs (million)	memory (gigs)
1	7.3	0.1
2	36	0.8
3	86	2.6
5	216	9.5
7	351	19.3
10	539	37.9

Ad Hoc Solutions

- Limit length of phrases
- Compute probabilities on disk
- Wait for test data, only extract those phrases

Our Solution: Intelligent Data Structure

- Retrieval of arbitrarily long phrases
- Uses less memory than table-based data structures
- **Suffix arrays** to index parallel corpus

Suffix Arrays

- Quickly find all instances of any phrase in a corpus
- Storage of index = size of corpus
- Total storage = 2 x corpus

How it works

Index of words:

Corpus

0	1	2	3	4	5	6	7	8	9
Spain	declined	to	confirm	that	Spain	declined	to	aid	Morocco

Initialized, unsorted
Suffix Array

Suffixes denoted by s[i]

s[0]	0	Spain declined to confirm that Spain declined to aid Morocco
s[1]	1	declined to confirm that Spain declined to aid Morocco
s[2]	2	to confirm that Spain declined to aid Morocco
s[3]	3	confirm that Spain declined to aid Morocco
s[4]	4	that Spain declined to aid Morocco
s[5]	5	Spain declined to aid Morocco
s[6]	6	declined to aid Morocco
s[7]	7	to aid Morocco
s[8]	8	aid Morocco
s[9]	9	Morocco

Alphabetically sorted

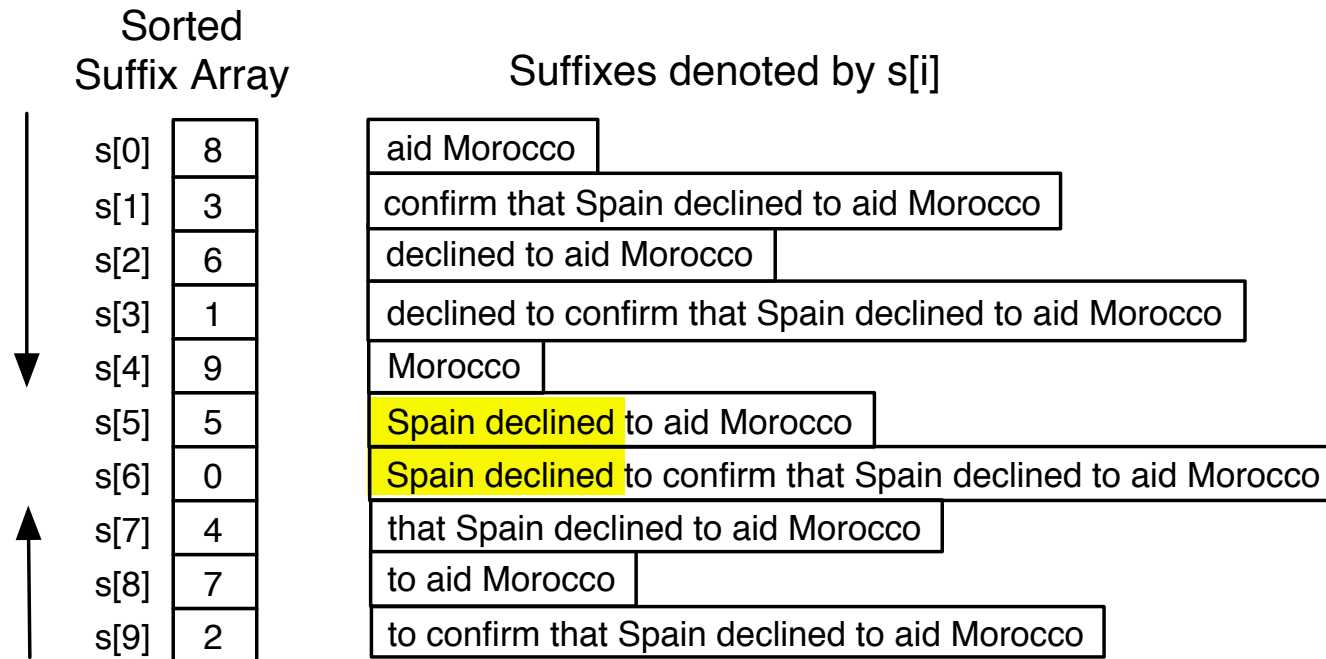
Sorted
Suffix Array

s[0]	8
s[1]	3
s[2]	6
s[3]	1
s[4]	9
s[5]	5
s[6]	0
s[7]	4
s[8]	7
s[9]	2

Suffixes denoted by s[i]

aid Morocco
confirm that Spain declined to aid Morocco
declined to aid Morocco
declined to confirm that Spain declined to aid Morocco
Morocco
Spain declined to aid Morocco
Spain declined to confirm that Spain declined to aid Morocco
that Spain declined to aid Morocco
to aid Morocco
to confirm that Spain declined to aid Morocco

Fast Find



Applied to Parallel Corpora

- Index source and target corpora
- Extract phrase alignments on the fly
- Allows retrieval of arbitrary length phrases
- Less memory, but greater complexity

On The Fly Lookup

Sorted Suffix Array

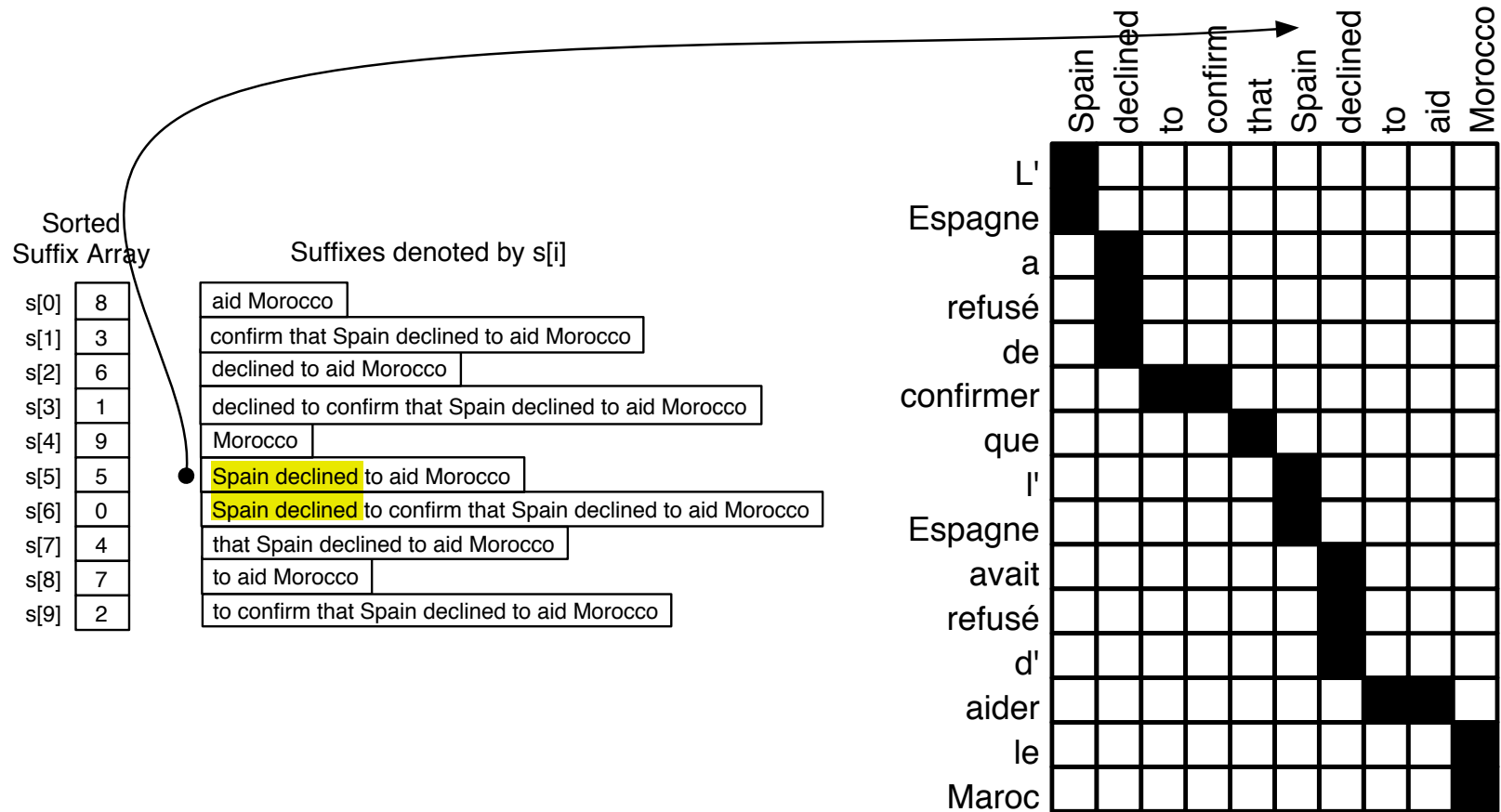
s[0]	8
s[1]	3
s[2]	6
s[3]	1
s[4]	9
s[5]	5
s[6]	0
s[7]	4
s[8]	7
s[9]	2

Suffixes denoted by s[i]

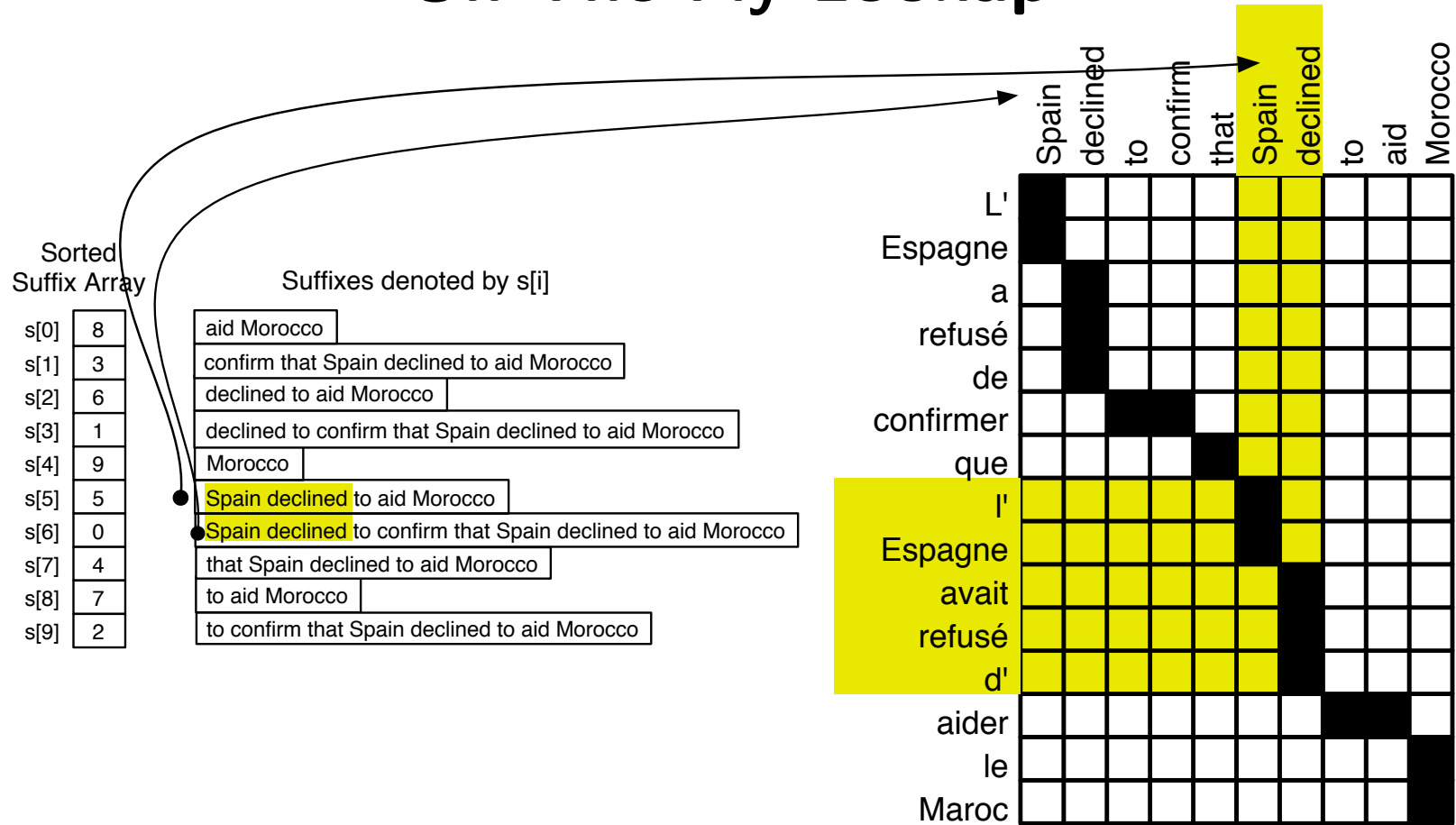
aid Morocco
confirm that Spain declined to aid Morocco
declined to aid Morocco
declined to confirm that Spain declined to aid Morocco
Morocco
Spain declined to aid Morocco
Spain declined to confirm that Spain declined to aid Morocco
that Spain declined to aid Morocco
to aid Morocco
to confirm that Spain declined to aid Morocco

	Spain	declined	to	confirm	that	Spain	declined	to	aid	Morocco
L'	■									
Espagne	■									
a		■								
refusé		■								
de			■							
confirmer			■	■						
que					■					
l'						■				
Espagne							■			
avait								■		
refusé									■	
d'										■
aider									■	
le										■
Maroc										■

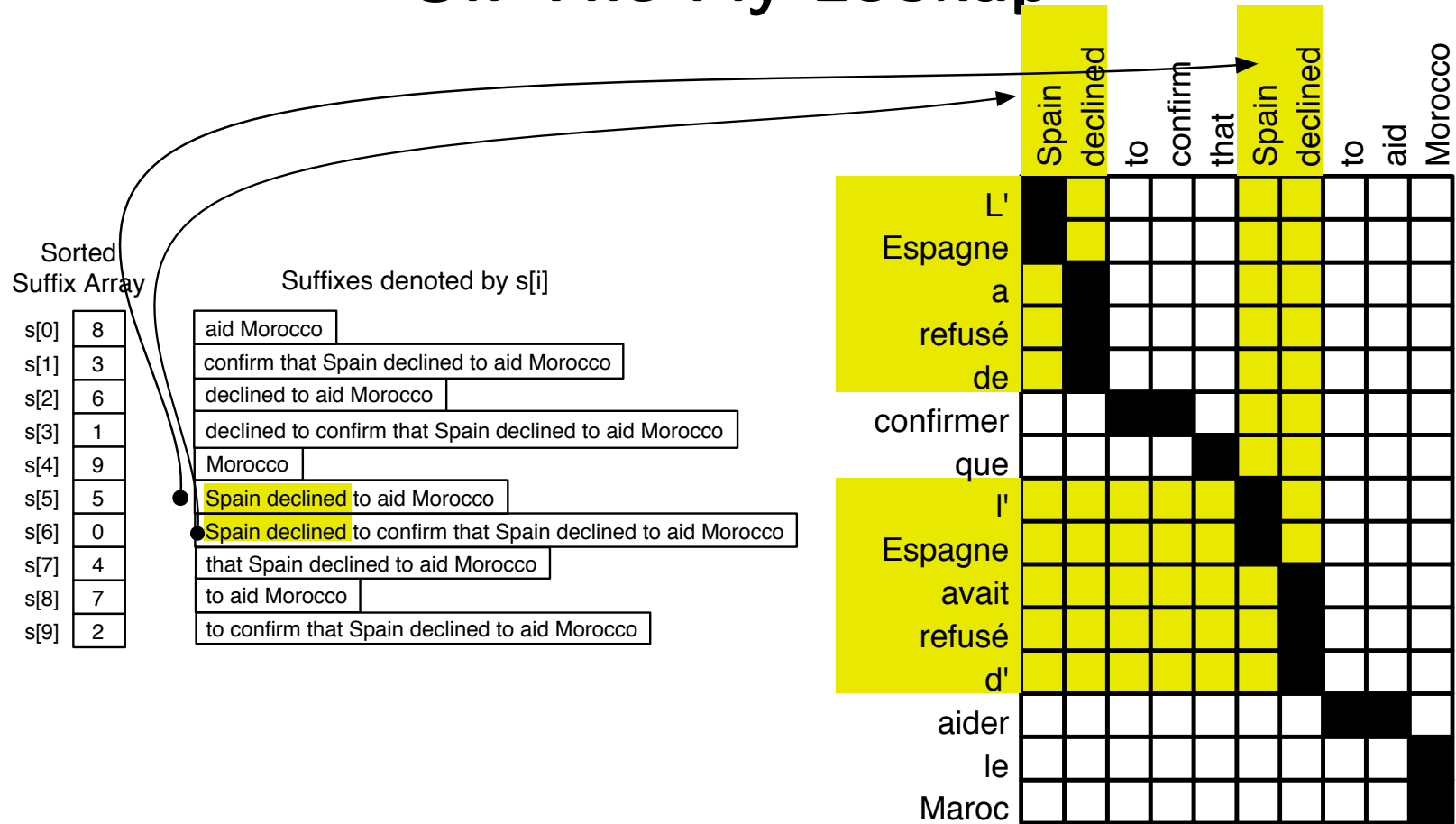
On The Fly Lookup



On The Fly Lookup



On The Fly Lookup



Memory reduction

- Memory = 2 * corpus + word alignments
- To index NIST 2004 data = 2 Gigabytes
- Less than to store phrases of length 3!

Complexity

- Memory v. speed tradeoff
- Complexity for table lookup = unit time $O(1)$
- Complexity for suffix array parallel corpus = much greater

Complexity Detailed

Sorted Suffix Array

s[0]	8
s[1]	3
s[2]	6
s[3]	1
s[4]	9
s[5]	5
s[6]	0
s[7]	4
s[8]	7
s[9]	2

log(n) ↓

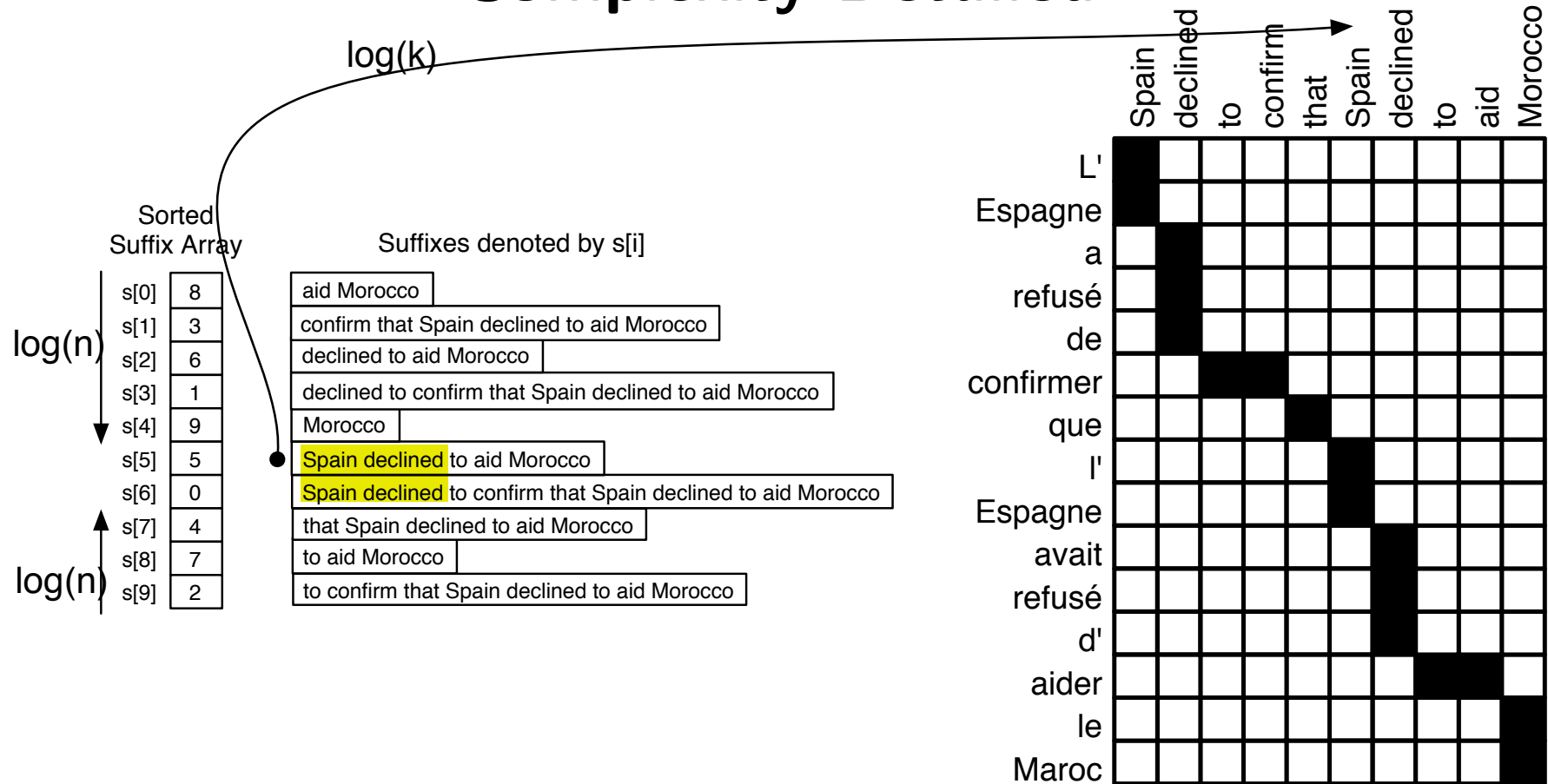
↑ log(n)

Suffixes denoted by s[i]

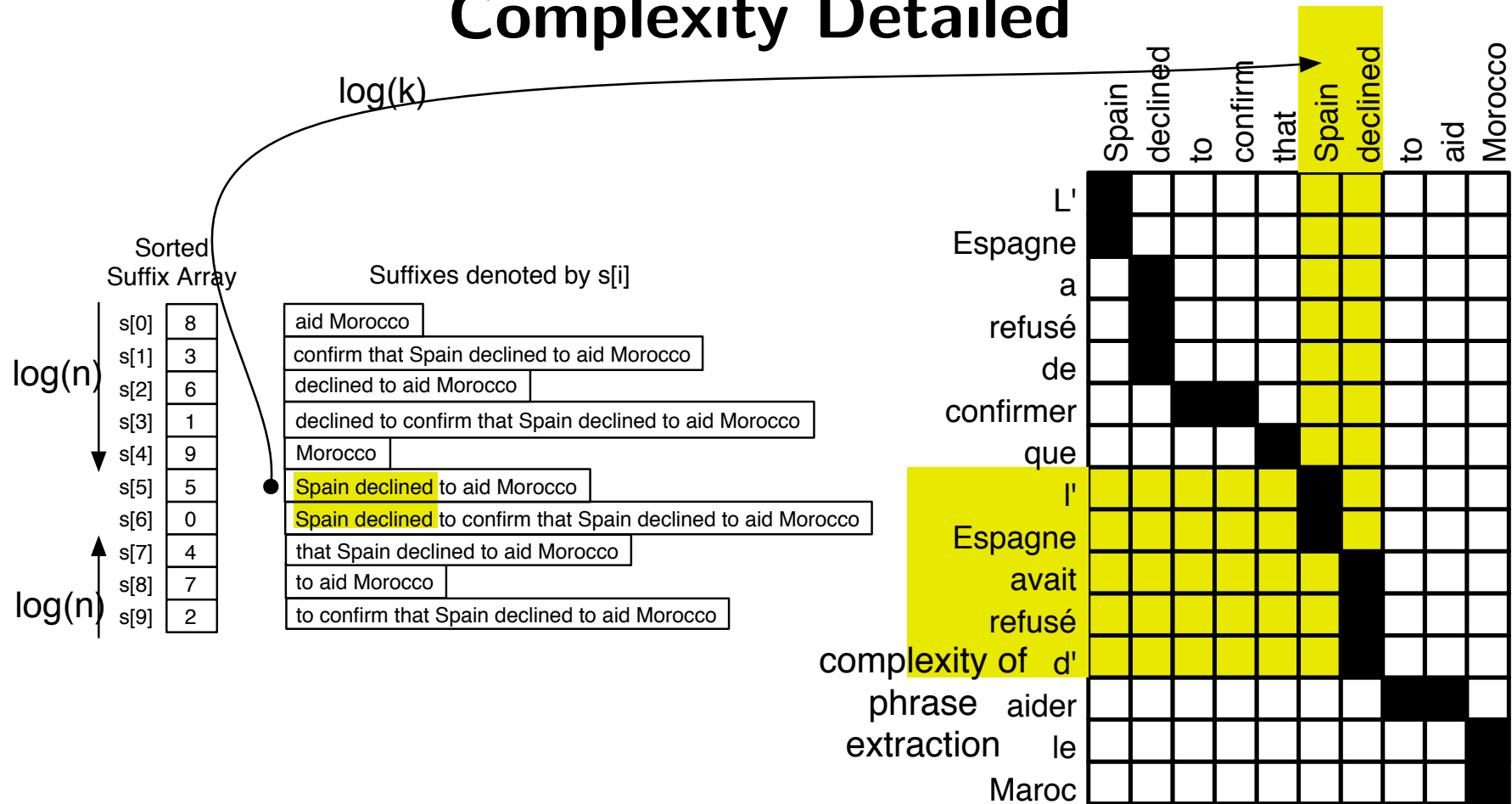
aid Morocco
confirm that Spain declined to aid Morocco
declined to aid Morocco
declined to confirm that Spain declined to aid Morocco
Morocco
Spain declined to aid Morocco
Spain declined to confirm that Spain declined to aid Morocco
that Spain declined to aid Morocco
to aid Morocco
to confirm that Spain declined to aid Morocco

	Spain	declined	to	confirm	that	Spain	declined	to	aid	Morocco
l'	■									
Espagne										
a		■								
refusé										
de										
confirmer			■	■						
que										
l'						■				
Espagne										
avait							■			
refusé										
d'										
aider								■	■	
le										
Maroc										■

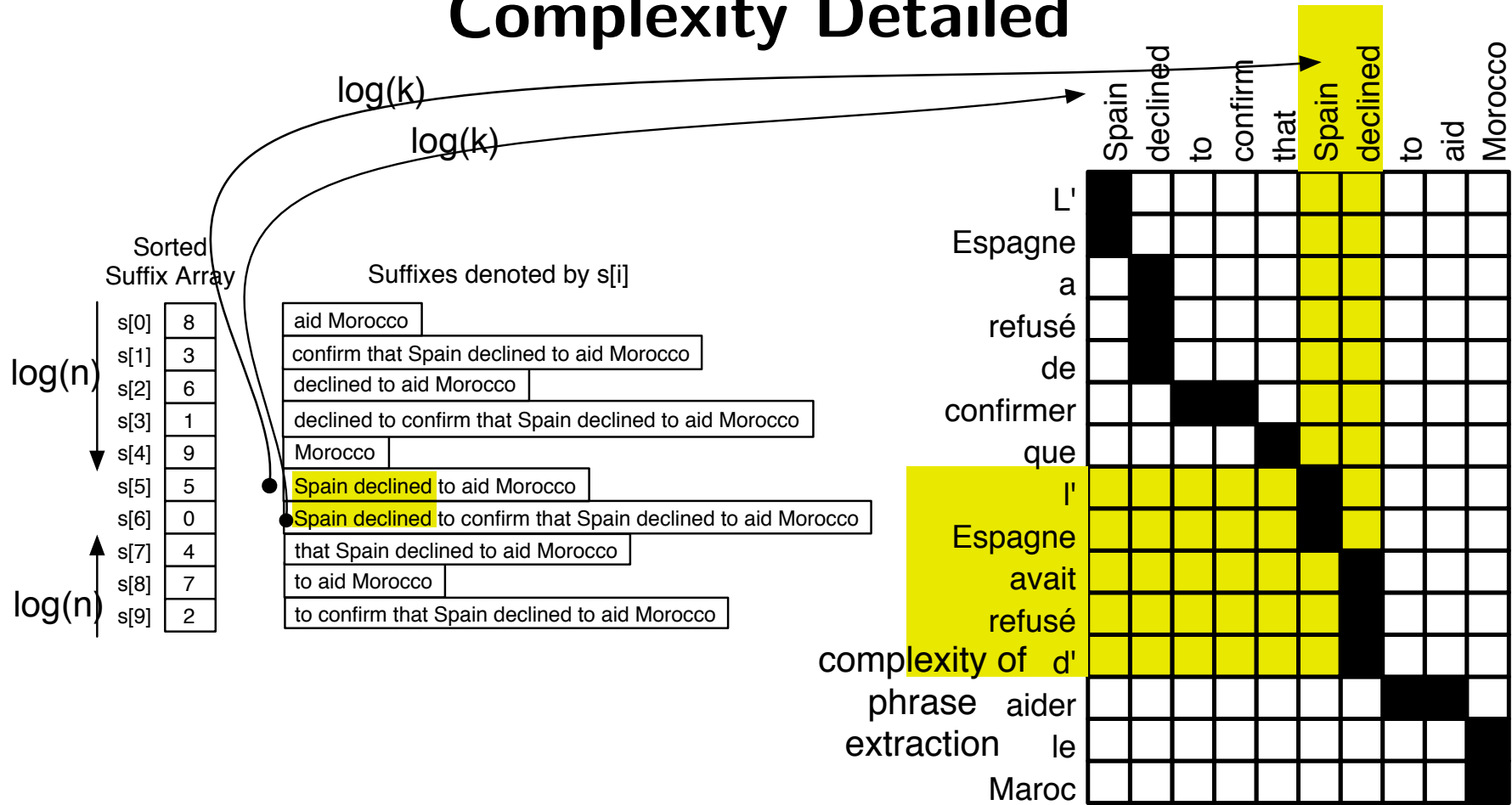
Complexity Detailed



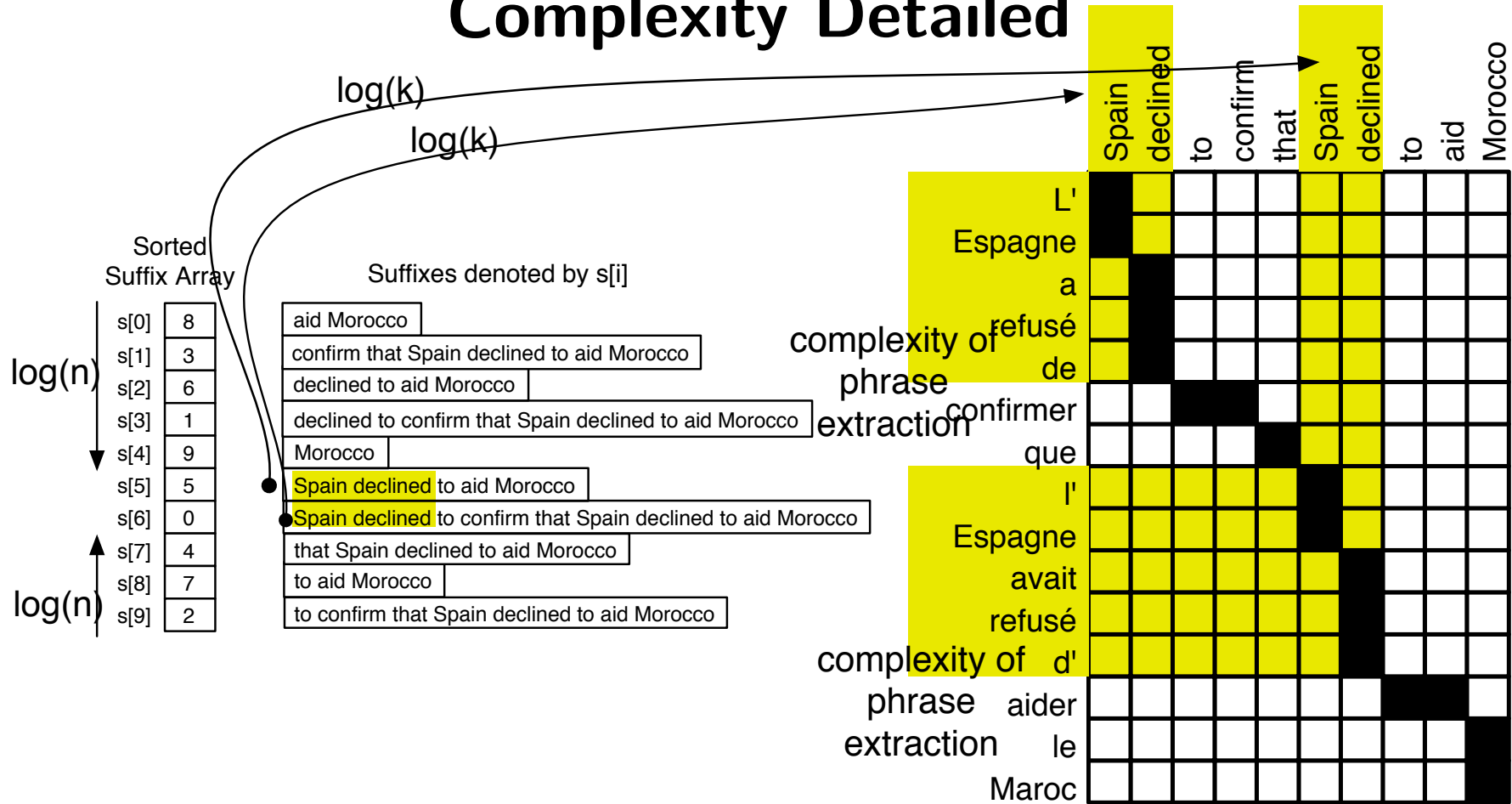
Complexity Detailed



Complexity Detailed



Complexity Detailed



Retrieval times

- Dominated by frequency

phrase	freq	computations	time (ms)
<i>respect for the dead</i>	3	80	24
<i>since the end of the cold war</i>	19	240	136
<i>the parliament</i>	1291	4391	1117
<i>of the</i>	290921	682550	218369

Coping Strategies

- Mixed data structure
 - Table for most frequent items
 - Suffix array handles the rest
- Increases memory, but still less than enumerating everything
- Alternatively: sampling

Sampling

- Need to see 300,000 instances of “the” to learn translations?
- Set some threshold
- Randomly choose phrases
- Estimate probability from that sample

Experiment

- Created suffix array parallel corpus for Europarl German-English
- Withheld 400 sentences with 50 words or less for testing
- Sampled at various cut offs
- Measured the effect on translation quality
- Compared against lookup time

Results

sample size	time	Bleu
unlimited	6279 sec	.290
50000	1051 sec	.289
10000	336 sec	.291
5000	201 sec	.289
1000	60 sec	.288
500	35 sec	.288
100	10 sec	.288

- Time reduced by orders of magnitude
- Translation quality nearly unchanged

Summary

- Effective data structure for dealing with large corpora
- Arbitrarily long phrases
 - Less memory than table storing 3 word phrases
 - One tenth memory of table with 8 word phrases
- Memory size / speed trade-off
- Sampling an effective counter measure
 - 100x speed up with nearly no loss in quality

Shameless self-promotion

- Linear B open source initiative
- Code for this (and much more!) available for research purposes
- E-mail us if interested: developers@linearb.co.uk

LINEAR B

Thank you!