Adaptive Scalable Pipelines for Political Event Data Generation

Presenter
Ahmad Mustafa
University of Texas at Dallas
ahmad.mustafa@utdallas.edu

Andrew Halterman
Department of Political Science
Massachusetts Institute of Technology
ahalt@mit.edu

Phanindra Jalla, Yan Liang, Christan Grant
School of Computer Science
University of Oklahoma
{yliang, Phanindra.Jalla, cgrant}@ou.edu

Jill Irvine, Manar Landis
Women's and Gender Studies/International and Area Studies
University of Oklahoma
{jill.irvine, Manar.K.Landis-1}@ou.edu

Mohiuddin Solaimani
Department of Computer Science
The University of Texas at Dallas
mxs121731@ou.edu

NSF #1539302 RIDIR: Modernizing Political Event Data for Big Data Social Science Research

Outline

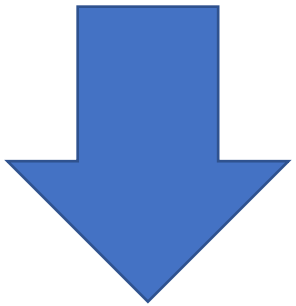
- Political Event Extraction
- Pipeline Architecture
- Kalman Filter
- Optimizations
- Summary

Political Events

- Governments, Social Scientists, and Political Scientist are interested in studying events around the world.
- Researchers spend spend many hours compiling and analyzing news documents.
- This grant came into an agreement with Lexis Nexis to obtain a large collection of US, multilingual, and International news documents.

Political Event Extraction

A town in western Sudan's South Kordofan state has been recaptured by Sudanese government forces from the rebel Sudan People's Liberation Army (SPLA)



Actor	Event	Target
Sudanese Military	Capture Territory	Sudans People’s Liberation Army

Goal

1. Create pipeline to help non-cs people extract political events.
2. Make the pipeline work on extremely large and small data.

Requirements

1. Should use MongoDB.
2. Must be deployable locally.

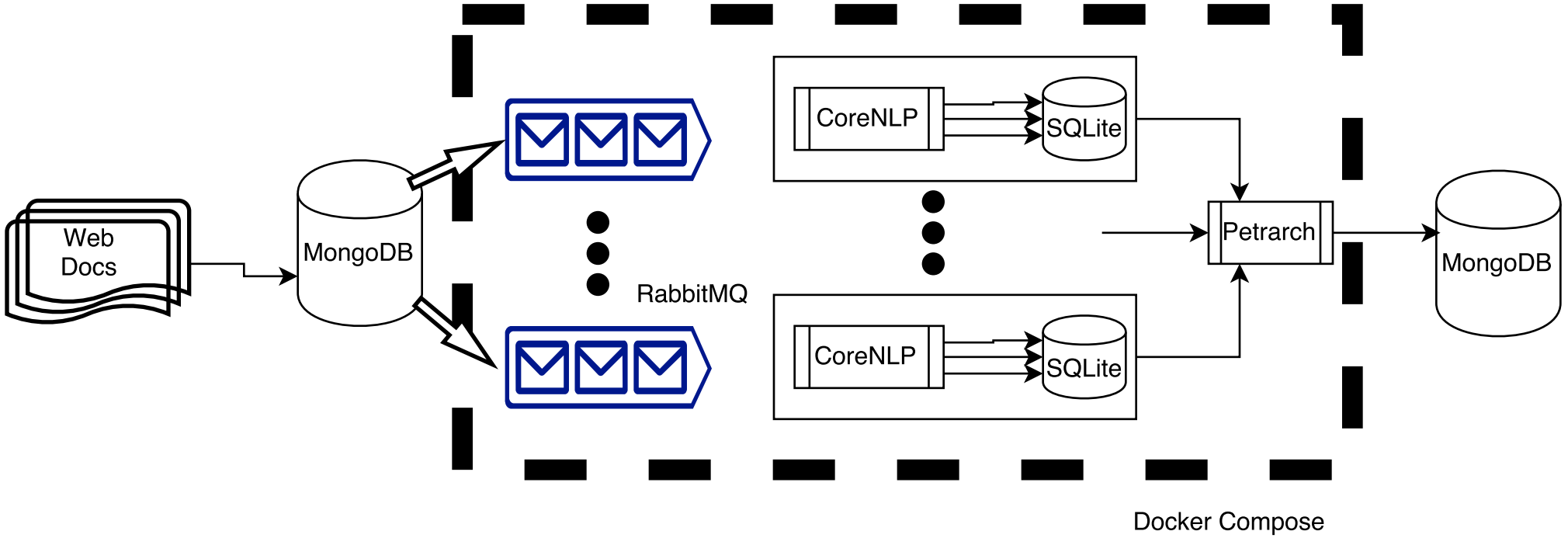
Biryani Pipeline

A pipeline to manage the scalability, execution, of event extraction.

Uses Docker and Docker-compose to facilitate execution of threads and batches.

Optimization allow the system to adapt to changes in system configuration and workload.

Biryani Architecture



Optimizations: Kalman Filter

- Optimal estimation algorithm
- Inputs:
  - Number of documents read at a time (batch).
  - Observation: Time taken for a batch.
- Output:
  - Updated batch size.

Algorithm 1 Pseudocode for Kalman 0.75 filter based optimization

$P = 1$
$Q = 10^{-1}$
$K = 0.0$
$R = 0.1^{0.75}$
while batch **in** stream **do**
  $P' = P + Q$
  Z = **len**(batch)
  $K = \frac{P+Q}{P+Q+R}$
  $X' = X + K * (Z - X)$
  $P = (1 - K) * P'$
end while

Experiment Environment

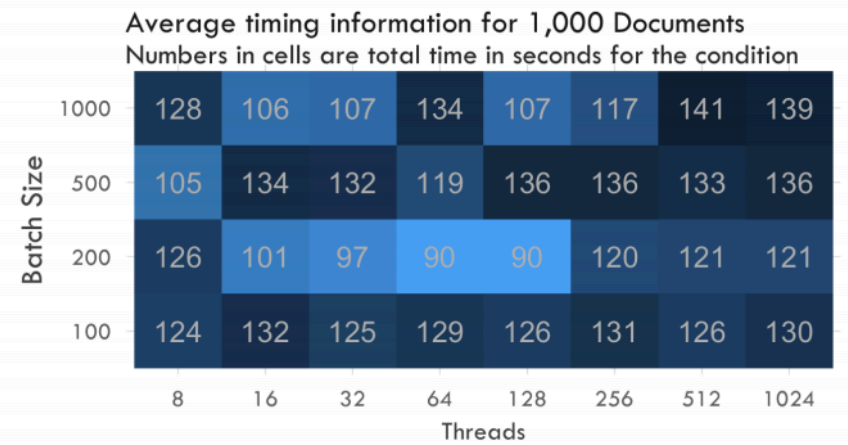
- Machines
  - **Processing machine**: Intel® Core™ i7-6950X CPU @ 3.00GHz CPU with 20 total cores and 126 GBs of RAM
  - **Data Storage**: Intel® Xeon® CPU X5687 @ 3.60GHz with 16 total cores and 96 GBs of RAM
  - **Laptop**: Intel® Xeon® CPU @ 2.30GHz, 4 cores, and 16GB of RAM.
- Data Sets
  - English Gigaword corpus (4 Million Documents, 12 gigabytes)

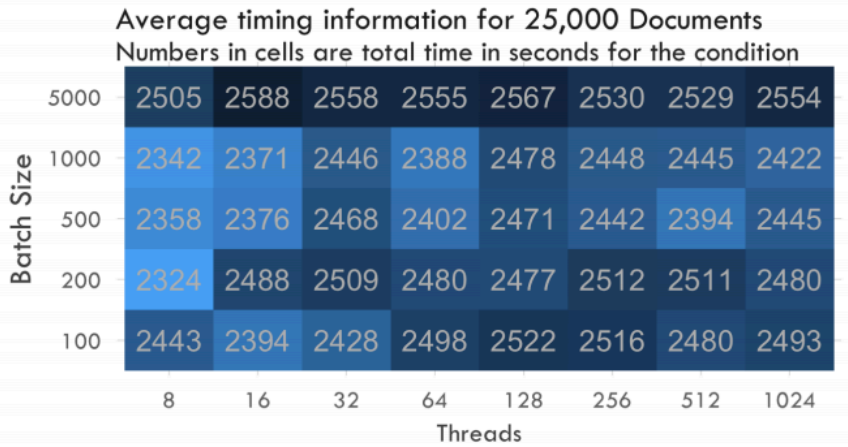
Experiments

- Experiment 1: Optimal and thread batch size.
- Experiment 2: Kalman Filter performance.
- (Creation of the Terrier Dataset)

Thread and Batch Size

- The experiments measures total run time for 1K and 25K documents.
- (Lighter and smaller means faster)
- We see clear optimal choices for batch sizes and thread size that change over time.
- Thread count affected timing less than batch.
- This motivates a system to adapt to changing the workload.



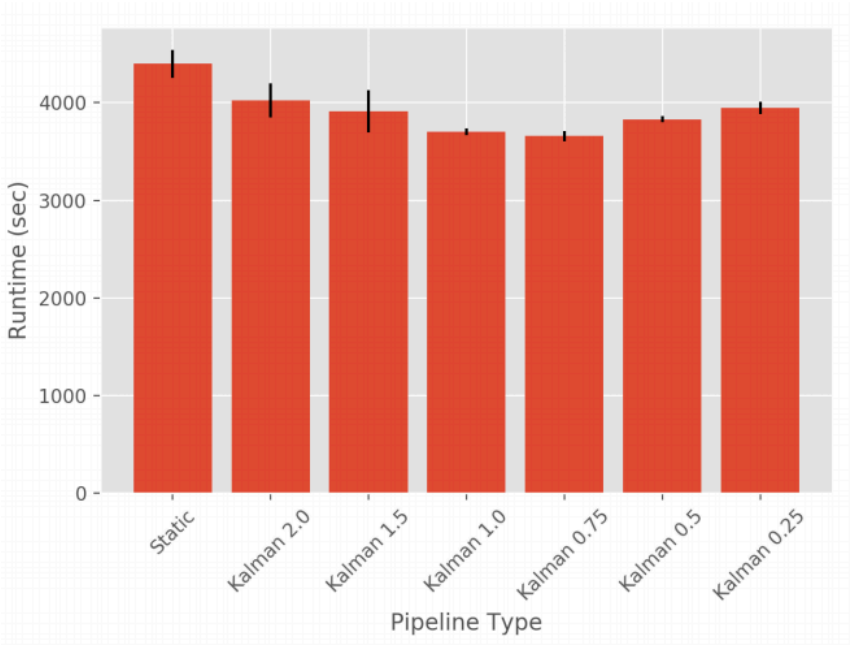


Kalman Filter Performance

- Adding a Kalman filter over the batch sizes shows a performance gain.
- Compared against a sorted, static data set (best possible performance)
- Performance gains of up to 20.33% on a *laptop-style* configuration.
- Some hyper parameter tuning of the Kalman filter is required. But defaults work well.

Pipeline	Docs	Run 1 (s)	Run 2 (s)	Avg (s)	% Gain
Static	150K	13,555	13,619	13,587	-
Kalman 0.75	150K	13,051	13,180	13,115	3.47 %

TABLE I
PERFORMANCE GAIN OF KALMAN FILTER APPROACH OVER 150,000 DOCUMENTS.



Summary

- We combined scalable container-based pipeline with adaptable .
- We added a optimization filter to automatically manage the processing.
- We used this pipeline to create one of the largest political event datasets available.
- Political scientist can also use Biryani to generate their own data sets.

Coming Spring 2018 …

Terrier

Temporally Extended Regularly
Reproducible International Event Records

Email jill.Irvine@ou.edu for access

Thank you!

Questions?

```
{
  "code": "171",
  "src_actor": "MKD",
  "month": "01",
  "tgt_agent": "",
  "country_code": "MKD",
  "year": "2015",
  "id": "572fa63c172ab8317c450234_2",
  "source": "",
  "date8": "20150106",
  "src_agent": "",
  "tgt_actor": "IND",
  "latitude": 41.96222,
  "src_other_agent": "",
  "quad_class": 4,
  "root_code": "17",
  "tgt_other_agent": "",
  "day": "06",
  "target": "IND",
  "goldstein": -9.2,
  "geoname": "Skopje ",
  "longitude": 21.62355,
  "url": ""
}
```


TERRIER Creation

- TERRIER is a machine-coded political event dataset covering 1979 to 2015
- Event data records the interactions between political actors that are reported in news text
- Terrier creation used Biryani and Birdcage
- ~ 900K sample is available now
- Uses the CAMEO ontology for actor/action representation
- It includes the complete archives of all major US and international newspapers and wire services going back to the 1970
- Will be the largest data set of its kind
- May 2018 Adding **Arabic** and **Spanish** (ISA)

References

- Biryani Figure https://commons.wikimedia.org/wiki/File:Bangladeshi_Biryani.jpg
- Protest https://www.flickr.com/photos/johnnysilvercloud/28476745294