2019
Research Experience for Undergraduates

ReDAS Lab

Parallelizing Semi-Supervised Learning Algorithms with MapReduce

Nick Gauthier
Advisor: Dr. Rakesh Verma

Final Presentation
August 9th, 2019

UNIVERSITY of HOUSTON

Goal

- Introduce a faster and efficient way of using some Semi-supervised learning methods with big data.

Objectives

1. Identify which SSL algorithms have been previously parallelized.

2. Implement serial versions of selected algorithms.

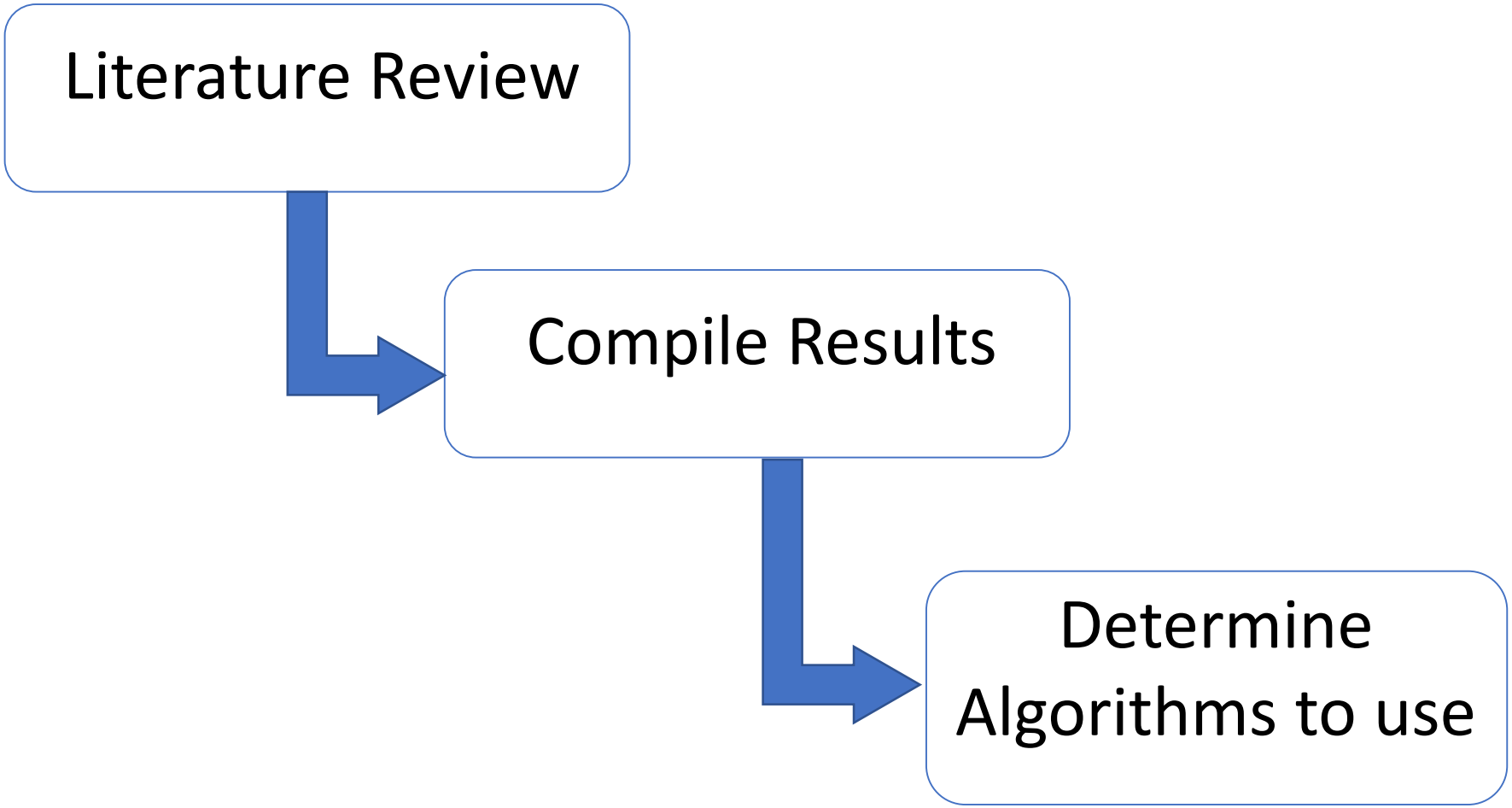
3. Convert serial implementation into a parallel version in MapReduce.

Objectives (cont.)

4. Test both on multiple datasets(if applicable on large datasets).

5. Compare output and runtime of parallel vs. serial implementations.

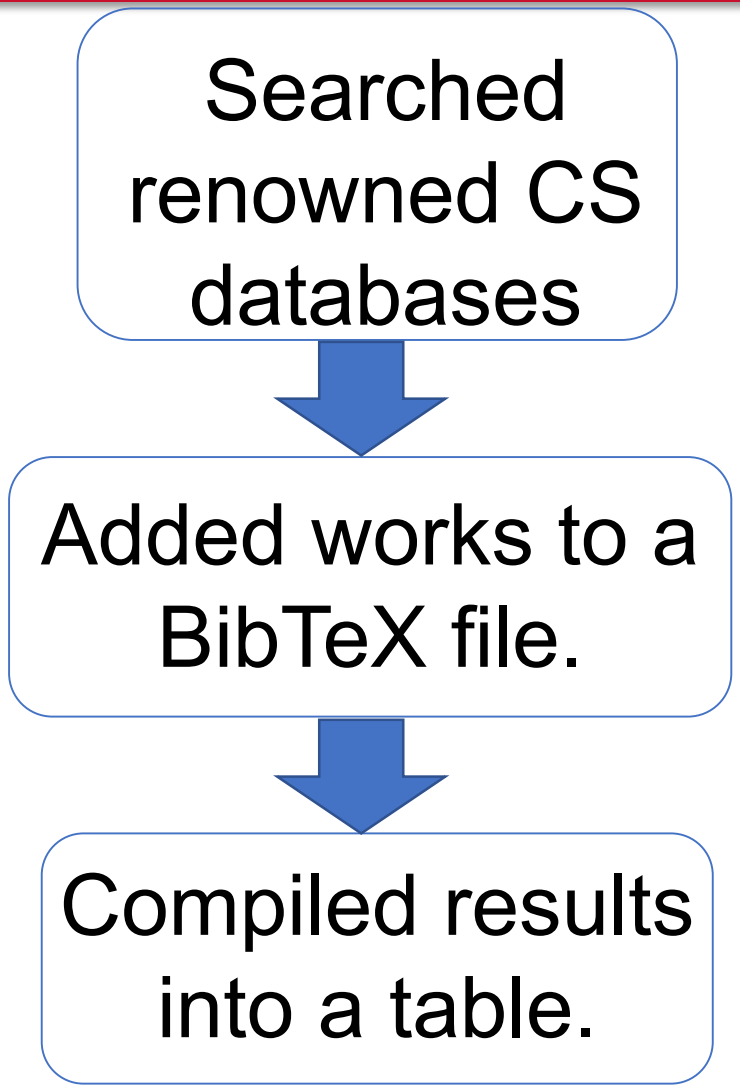
Objective 1: Tasks



Objective 1: Accomplishments

- Completed Lit. Review
- Created tables/references
- Determined which algorithms have and have not been parallelized.

Objective 1: Methodology



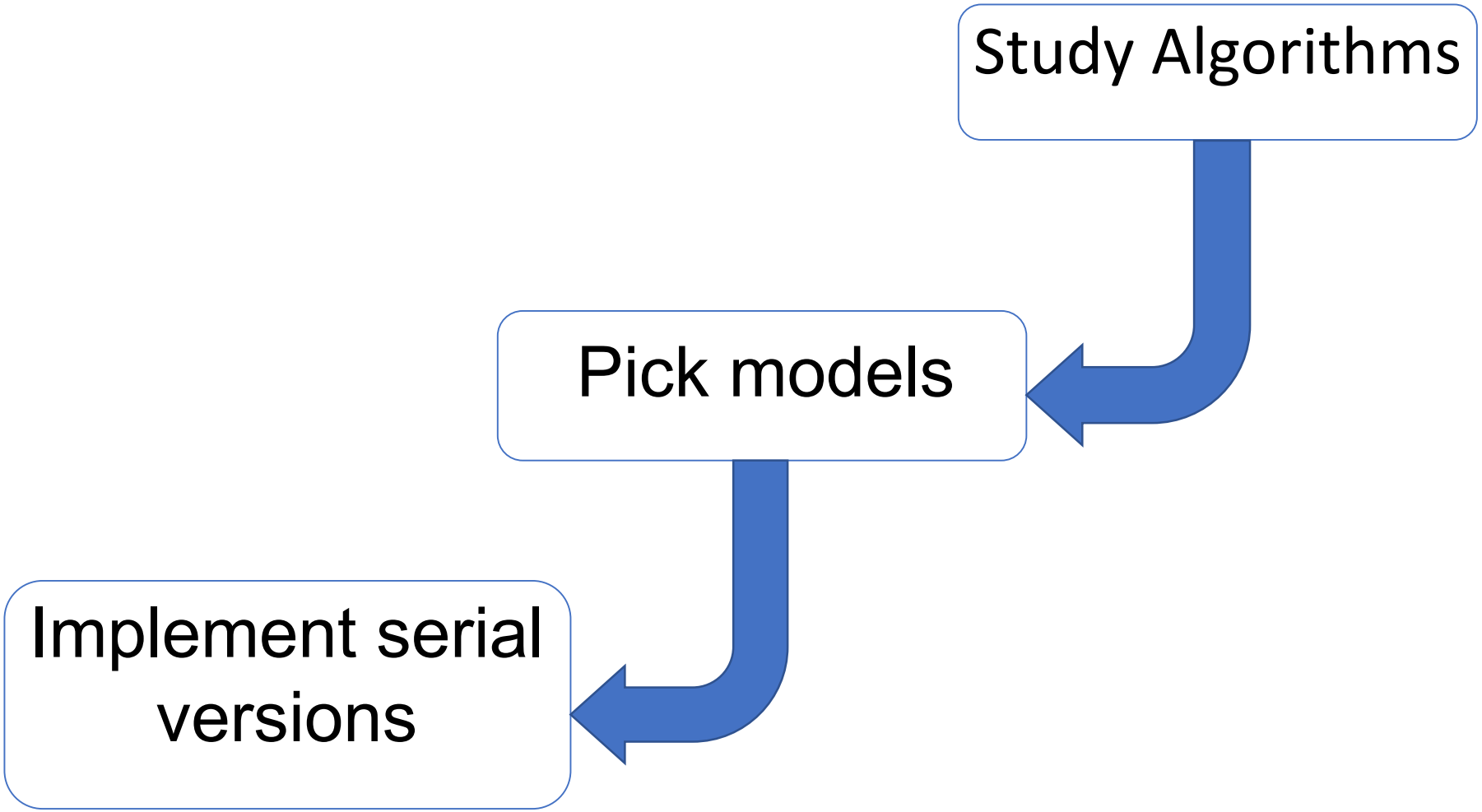
Objective 1: Results

S-S Algorithms	MapReduce	Apache Spark	Hive
Generative Models			
Self-Training			
TSVM/S3VMs		[2, 0 CTs, S3VM]	
Graph-Based	[1, UK CTs, PGL], [10, 1 CTs, MinHash], [12, 2 CTs], [13, 5 CTs], [16, 14 CTs], [21, 37 CTs, SSTF]		
Multiview	[9, 1 CTs, TT]		
SSR	[22, 1 CTs, MBSR]		
Label Propagation	[16, 14 CTs]	[15, 0 CTs]	
Multi-ant colonies clustering ensemble	[19, 1 CTs]		
SSPPCR	[20, 17 CTs]		
ELM	[3, 5 CTs, SS-ELM/PASS-ELM]		
FCMC	[4, 0 CTs, FCMC]		
PR	[6, 21 CTs, SSP]		
PLSA	[14, 9 CTs, PLSA]		
Non S-S Algorithms	[5, 5 CTs, SV-RF], [8, 16 CTs, k-MC/RDT]	[11, 0 CTs, UV-k-means]	

Objective 1: Discussion

- Most work is for Graph-based methods
- Some work for Generative Models
- Very little to no work for TSVMs
- Very little to no work for Self-Training algorithms.

Objective 2: Tasks



Objective 2: Accomplishments

- Researched Semi-supervised Expectation Maximization (SS-EM)
- Chose Multinomial Naïve Bayes (MNB) as the model for SS-EM
- Using an open source serial implementation of MNB with SS-EM.

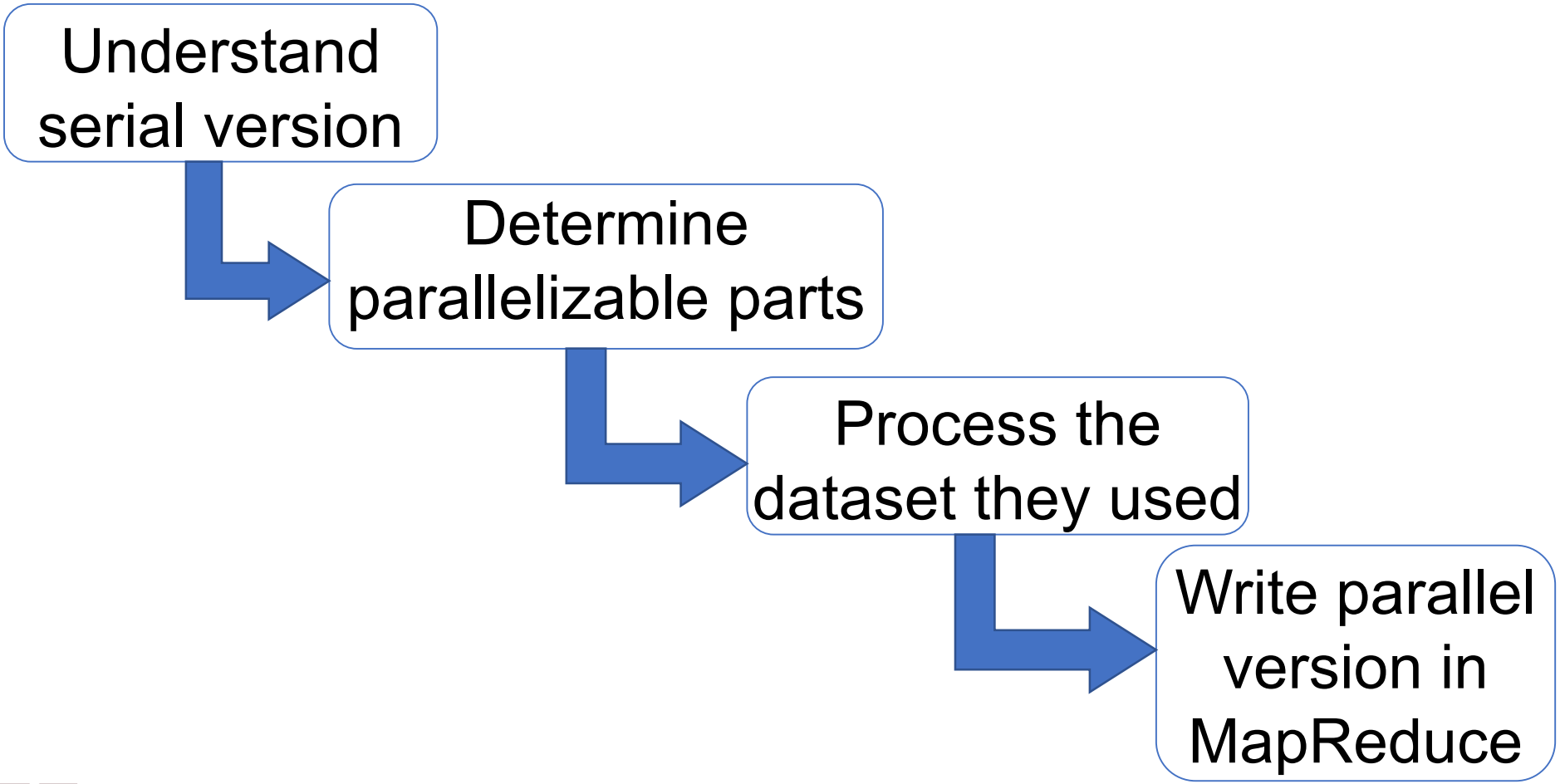
Objective 2: Methodology

Studying SS-EM

Difficulty in selecting a model

Using open source MNB with SS-EM code for serial version.

Objective 3: Tasks



Objective 3: Accomplishments

- Determined parts which can be parallelized
- Making significant progress on pre-processing the data
- Still have yet to finish the parallel implementation

Objective 3: Methodology

Created pseudocode to work through the logic

Learning python & libraries, Hadoop framework, and API for job submission / Input and Output

Pre-processing dataset for Hadoop's HDFS

Objective 3: Methodology (cont.)

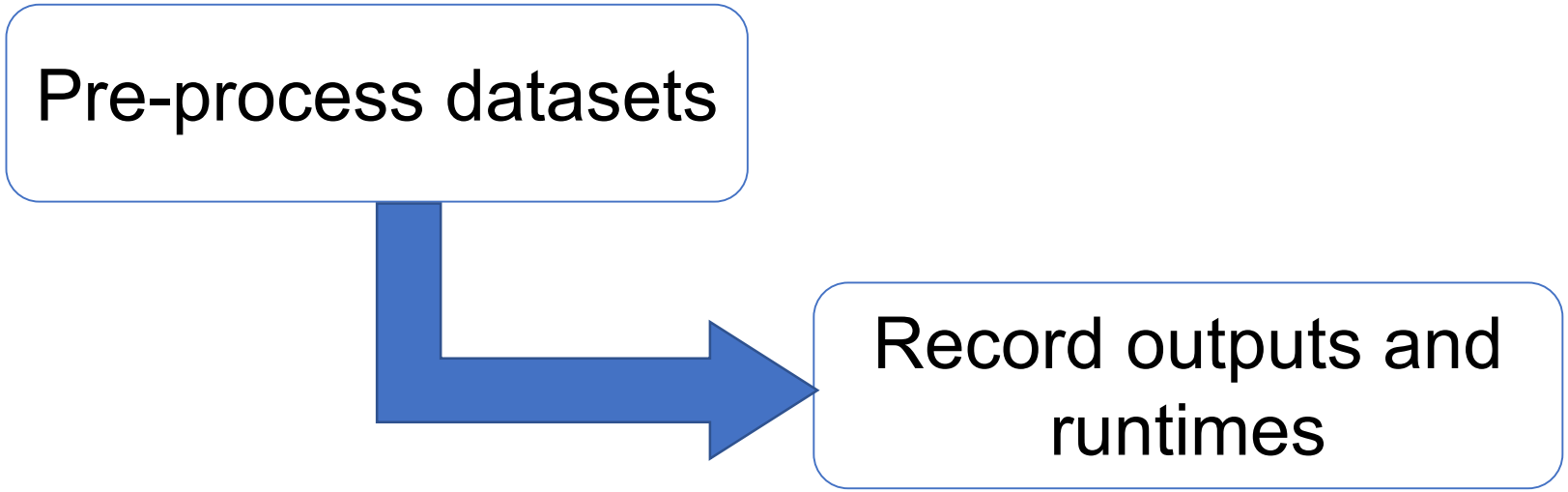
One document in 20NewsGroups dataset

```
['I was wondering if anyone out there could enlighten me on this car I saw\n'
 'the other day. It was a 2-door sports car, looked to be from the late 60s/\n'
 'early 70s. It was called a Bricklin. The doors were really small. In '
 'addition,\n'
 'the front bumper was separate from the rest of the body. This is \n'
 'all I know. If anyone can tellme a model name, engine specs, years\n'
 'of production, where this car is made, history, or whatever info you\n'
 'have on this funky looking car, please e-mail.',
 'A fair number of brave souls who upgraded their SI clock oscillator have\n'
 'shared their experiences for this poll. Please send a brief message '
 'detailing\n'
 'your experiences with the procedure. Top speed attained, CPU rated speed,\n'
 'add on cards and adapters, heat sinks, hour of usage per day, floppy disk\n'
 'functionality with 800 and 1.4 m floppies are especially requested.\n'
 '\n'
 'I will be summarizing in the next two days, so please add to the network\n'
 "knowledge base if you have done the clock upgrade and haven't answered this\n"
 'poll. Thanks.']
```

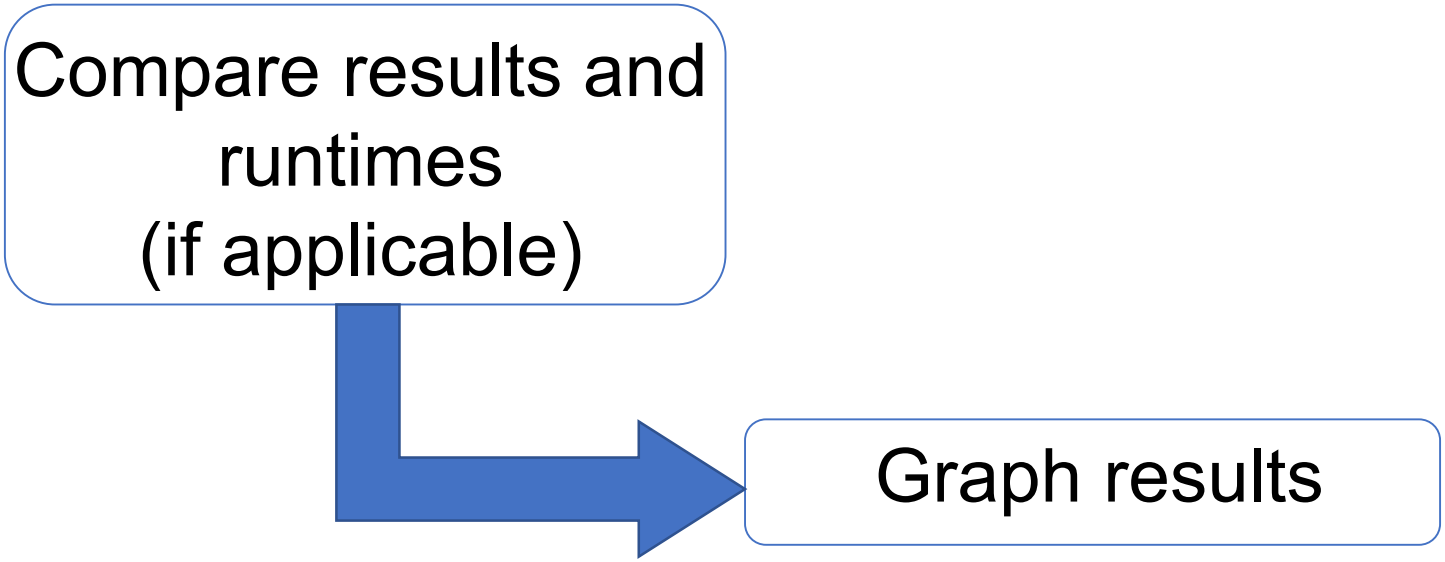
Some of Vectorized version of document

```
(0,  95844)     0.2085823901983838
(0,  97181)     0.11904550931918896
(0,  48754)     0.08965702221604545
(0,  18915)     0.14367199533485261
(0,  68847)     0.058045106782124184
(0,  88638)     0.05191891673444119
(0,  30074)     0.06757106887546716
(0,  37335)     0.17894975360023155
(0,  60560)     0.056847192482369406
(0,  68080)     0.08295129558221048
(0,  88767)     0.16403190525380046
(0,  25775)     0.4046469916999255
(0,  80623)     0.11038705606471814
(0,  88532)     0.1642840263996455
(0,  68781)     0.06213700528329297
(0,  31990)     0.09144124612366937
(0,  51326)     0.07095431715969416
(0,  34809)     0.1289860835335575
(0,  84538)     0.14219503554259302
(0,  57390)     0.11745787957135642
(0,  89360)     0.03000170182339287
(0,  21987)     0.04246364738118965
(0,  41715)     0.10166193380744531
(0,  55746)     0.12007927287527424
(0,  9843)      0.1806731975402963
:       :
```

Objective 4: Tasks



Objective 5: Tasks



Deliverables

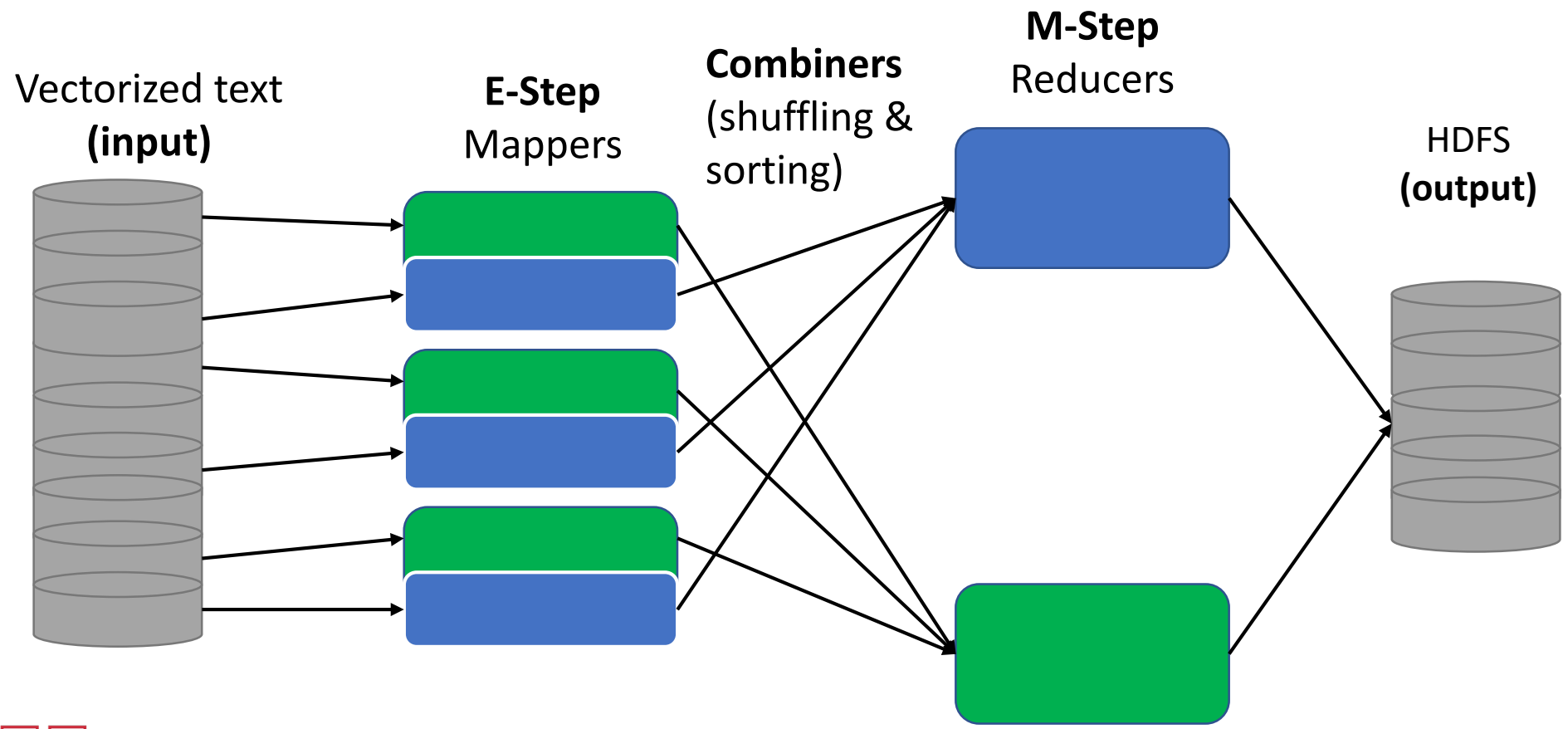
1. Poster

2. Detailed documentation of my progress and research log/notes.

3. Data pre-processing scripts with output files, and instructions on how to run/use them.

Future Work

- Finish parallelizing MNB and SS-EM.



Future Work (cont.)

- Testing the program on several larger datasets, and compare to serial version.
- Determine models for TSVM and Self-Training.
- Parallelize TSVM and Self-Training on MapReduce.
- Test TSVM and Self-Training on several larger datasets, and compare to serial versions.

Acknowledgements

The REU project is sponsored by NSF under award NSF-1659755. Special thanks to the following UH offices for providing financial support to the project: Department of Computer Science; College of Natural Sciences and Mathematics; Dean of Graduate and Professional Studies; VP for Research; and the Provost's Office. The views and conclusions contained in this presentation are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.