Parallelizing Semi-Supervised Learning Algorithms with MapReduce

ReDAS

Nick Gauthier

Advisor: Dr. Rakesh Verma

COMPUTER SCIENCE

Motivation

- Semi-supervised learning(SSL) algorithms are slow when working with large-scale data.
- Some SSL methods have already been parallelized, but not all.

Goal

- Parallelize SSL algorithms utilizing MapReduce.
- Learning the research paradigm.

Objectives

- Introduce an efficient and significantly faster way to work with large-scale data for some SSL methods by parallelizing them within the MapReduce framework.

Expected Impact

- Parallelize semi-supervised algorithms that have not yet been parallelized.
- Improving the runtime and efficiency of a semi-supervised algorithm by removing bottlenecks.

Deliverables

- Report
- Poster presentation
- Documentation of the process
- Potentially software

Methods: Objective 1

- Study the algorithms
- Write pseudocode for the typical semi-supervised method
- Convert the typical semi-supervised pseudocode to MapReduce pseudocode

Methods: Objective 1

- Implement the MapReduce pseudocode with Python 3.
  - Mapper
  - Combiner
  - Reducer
  - Driver program.

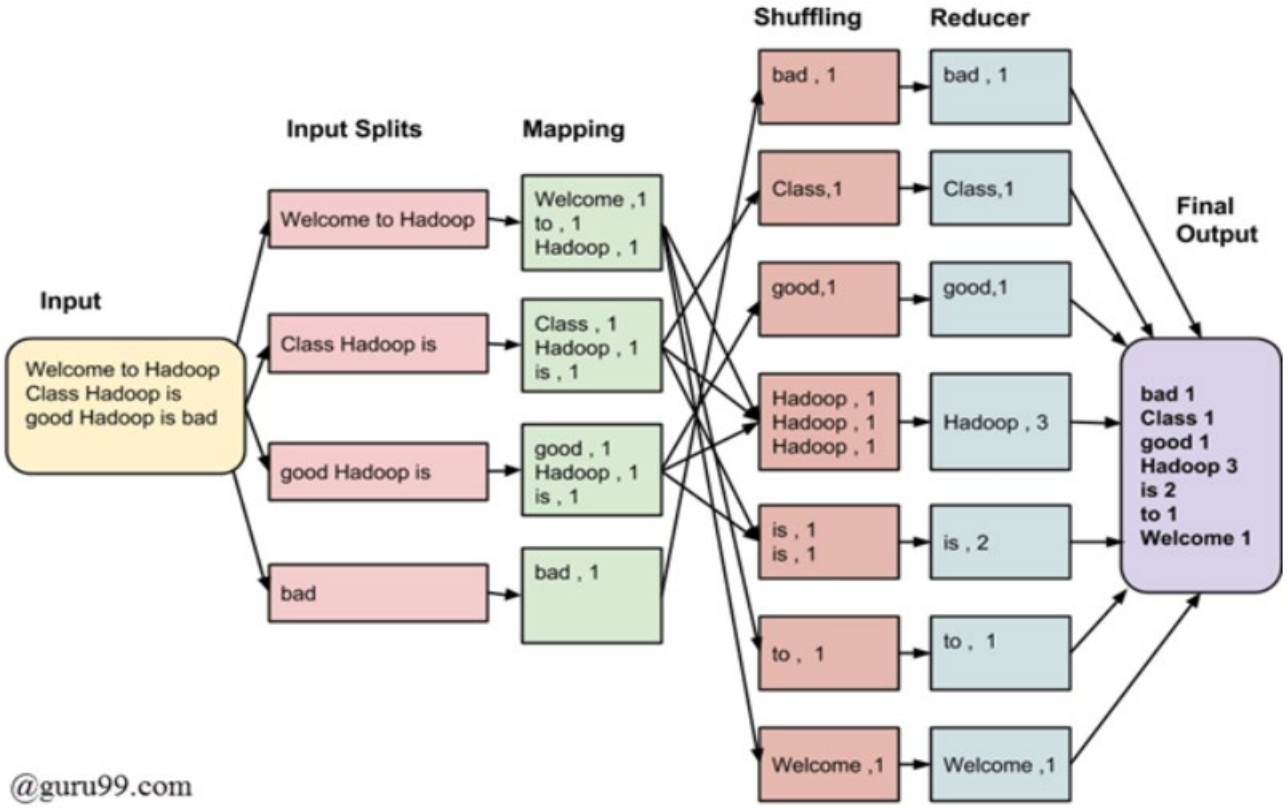
Methods: Objective 1

- Test, debug, test again, etc. until the bugs are worked out.
- Repeat testing until conclusion is reached.

Methods: Objective 1

- Example: Semi-Supervised Expectation Maximization (SS-EM)
  - Define parameters
  - E-Step
    - Assign expected labels
  - M-Step
    - Calculate probability of newly assigned labels
  - Repeat E & M step until convergence is reached.

Methods: Objective 1



https://www.guru99.com/introduction-to-mapreduce.html

Results: Objective 1

- I am currently in the stage of converting the semi-supervised pseudocode to MapReduce pseudocode.

Remaining Work

- I still have yet to write and implement the MapReduce code with Python 3.
- Testing the code once implemented.
- Create report, poster, and documentation.

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.