Practical Work 4: Word Count

Pham Tuong Ngan

Subject: Distributed Systems

1 Introduction

This report documents the implementation of a Word Count application using a custom MapReduce framework.

2 Why

I chose Python for its simplicity. It allows for a concise implementation of the MapReduce workflow (Map, Shuffle, Reduce) without the boilerplate of heavier frameworks, focusing directly on data processing.

3 How

The implementation follows three phases:

1. **Mapper:** Tokenizes input lines, removes punctuation, and yields `(word, 1)` pairs.
2. **Shuffle:** Groups emitted pairs by key, transforming lists of pairs into dictionary entries (e.g., `'word': [1, 1]`).
3. **Reducer:** Sums the counts for each word to produce the final frequency.

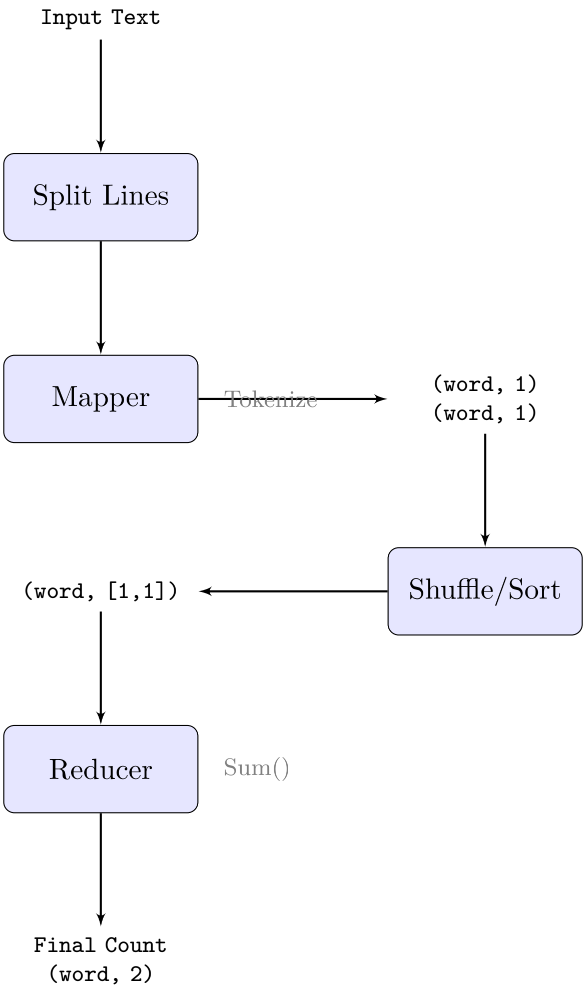

Figure 1: Data Flow of the Python MapReduce Implementation