Duplicate Document Detection Using Map-Reduce

By: Hani Khoshdel Nikkhoo

School of Computer Science
University of Waterloo

Monday, March 22, 2010

Introduction and Motivation

Near-duplicate documents usually cause:

- Storage waste
- Processing power waste
- User frustration

in Information Retrieval(IR) systems in general (e.g. HP[2]) , and in search engines in particular (e.g. Google[3]).

Statement of The Problem

- Finding near-duplicate documents in a collection of 50 million documents (ClueWeb09 Cat B [4])

- Definition of near-duplicate documents:

  Documents that are syntactically similar

- Challenge:

  Computationally intensive (in general $O(n^2)$)

General Solution

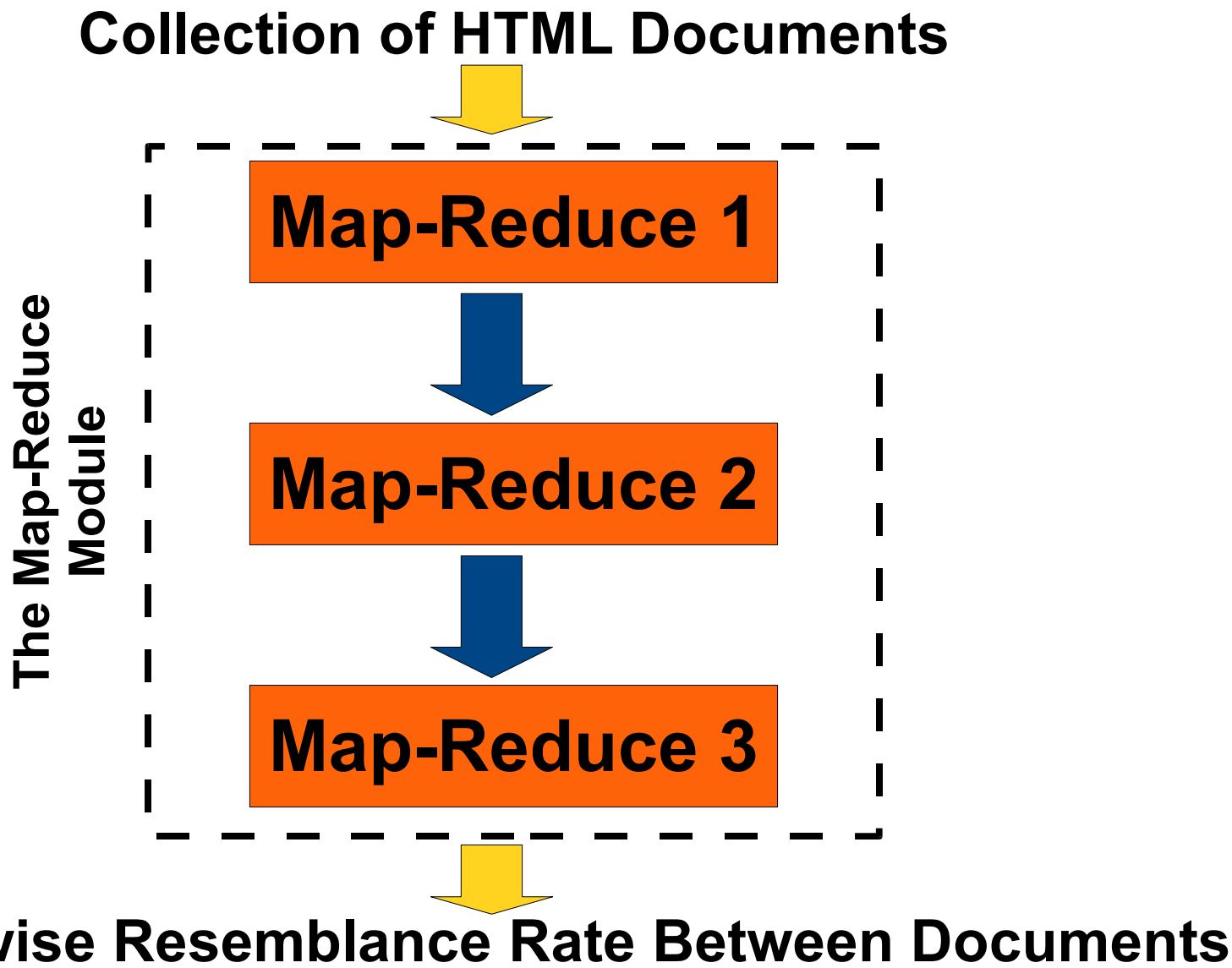
1. Split the document text into substrings called shingles

2. Hash the shingles

3. $resemblance(D_1, D_2) = \frac{|D_1 \cap D_2|}{|D_1 \cup D_2|}$

4. If $resebmblance(D_1, D_2) \geq \alpha$ we call D_1 and D_2 near-duplicate documents

The Map-Reduce Solution



Map-Reduce 1

- **Mapper 1**

  Input: .gz files including thousands of HTML Docs

  input → remove HTML tags → shingle generation →

  hashing → <DocID, $hash_1$>, <DocID, $hash_2$>, ...

  Output: <DocID, hash>

- **Reducer 1**

  Input: <DocID, hash>

  for each DOCID count total number of pairs

  Output: <hash, DocID-Size>

Map-Reduce 2

- **Mapper 2**

  Only sort and distribute

- **Reducer 2**

  <u>Input:</u><hash, DocID-Size>

$$\left\{\begin{array}{c}\langle hash, DocID_1 - \mathrm{Size}_1\rangle \\ \langle hash, DocID_2 - \mathrm{Size}_2\rangle \\ \langle hash, DocID_3 - \mathrm{Size}_3\rangle \\ \vdots\end{array}\right\} \rightarrow \left\{\begin{array}{c}\langle DocID_1 - \mathrm{Size}_1, DocID_2 - \mathrm{Size}_2\rangle \\ \langle DocID_1 - \mathrm{Size}_1, DocID_3 - \mathrm{Size}_3\rangle \\ \langle DocID_2 - \mathrm{Size}_2, DocID_3 - \mathrm{Size}_3\rangle \\ \vdots\end{array}\right\}$$

  <u>Output:</u> <$DocID_n$-$Size_n$, $DocID_m$-$Size_m$>

Map-Reduce 3

- **Mapper 3**

  Only sort and distribute

- **Reducer 3**

  <u>Input:</u> **<DocID$_n$-Size$_n$, DocID$_m$-Size$_m$>**

$$\left\{\begin{matrix}\langle DocID_1-\mathrm{Size}_1, DocID_2-\mathrm{Size}_2\rangle \\ \langle DocID_1-\mathrm{Size}_1, DocID_2-\mathrm{Size}_2\rangle \\ \vdots\end{matrix}\right\} \rightarrow \langle (DocID_1-\mathrm{Size}_1, DocID_2-\mathrm{Size}_2), \alpha\rangle \rightarrow$$

$$\left\langle DocID_1-DocID_2, \frac{\alpha}{\mathrm{Size}_1+\mathrm{Size}_2-\alpha}\right\rangle \rightarrow \langle DocID_1-DocID_2, resemblance-rate\rangle$$

- **<u>Output:</u>** **<DocID$_n$-DocID$_m$,resemblance (D_n, D_m)>**

References

(1)Jeffrey Dean and Sanjay Ghemawat. “Mapreduce: Simplified data processing on large clusters”. In Proceedings of Symposium on Operating Systems Design and Implementation (OSDI'04), pages 137-150, 2004.

(2)George Forman, Kave Eshghi and Stephane Chiocchetti. “Finding similar files in large document repositories”. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pages 394 - 400, 2005

(3)Monika Henzinger. “Finding near-duplicate web pages: a large-scale evaluation of algorithms”.Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 284 - 291, 2006

(4)“The ClueWeb09 Dataset”, http://boston.lti.cs.cmu.edu/Data/clueweb09/