MapReduce Model에 기반한 도서 추천 시스템의 설계 및 구현*

임찬식* 이원재* 이하나** 이세화** 이상준*
*숭실대학교 컴퓨터학부
**광운대학교 컴퓨터소프트웨어학과
chanshik@gmail.com, onejae@gmail.com,
hanawasborn@gmail.com, sehwa4444@gmail.com
sangjun@ssu.ac.kr*

Design and Implementation of a Book Recommendation System based on the MapReduce Model

Chanshik Lim*, Wonjae Lee*, Hana Lee**, Sehwa Lee**, Sangjun Lee*
*School of Computing, Soongsil University
**Department of Computer Software, Kwangwoon University

요 약

하루에도 수많은 도서가 출판되는 현실에서 사용자가 원하는 목적에 맞는 도서를 찾아 읽기는 어려운 일이다. 본 논문에서는 방대한 분량의 도서 데이타를 바탕으로, MapReduce 모델을 활용하여 도서들 사이의 연관 관계를 추출하였다. 추출한 연관 관계 DB를 이용하여 사용자에게 서로 관련 있는 도서를 추천해줄 수 있는 시스템을 개발하고자 한다.

1. 서 론

오늘 날에는 수많은 도서들이 출판되고 있으며, 급속도로 변화하는 사회상과 정보기술 그리고 문화발전에 힘입어 책 또한 분야와 종류가 날로 방대해지고 있다. 새로운 문화나 지식에 대해 공부하고 싶을 때 손쉽게 얻을 수 있는 것이 도서로부터의 도움이지만, 많이 출판되어 나오기 때문에 어디서부터 시작해야할지 막막한 것이 사실이다.

독자들이 도서를 선택하는데 도움을 주기 위해서 아마존[1]과 같은 인터넷 도서 판매 사이트에서는 고객들의 구매 패턴을 이용하여 연관성이 있는 도서를 추천해준다. 뿐만 아니라 아마존의 데이타베이스에 포함된 메타데이타, 즉 책 제목, 저자, 장르, 사용자 코멘트, 개요 등의 메타 정보를 분석하여 사용자에게 도서를 추천해주는 시스템도 연구되었다[2].

본 논문에서는 도서 안에서 언급된 다른 도서의 제목을 찾는 방법으로 서로간의 연관성을 찾아내어 이를 기반으로 도서 추천 시스템을 구현하였다. 대부분의 도서에서는 저자가 직접 저술한 다른 도서를 언급하거나, 특정한 분야를 더욱 자세히 다루는 도서를 추천해주는 경우가 많다. 이는 한정된 지면에서 모든 걸 다 다룰 수 없기 때문인데, 이렇게 언급된 도서를 원래 도서와 함께 보여준다면 계속해서 읽어나갈 도서를 선택할 때 도움이 될 것이다.

도서들 사이의 연관 관계를 추출하려면 방대한 분량의 도서 전문을 검색해야 하며, 이를 위해 본 논문에서는 다수의 노드로 구성된 클러스터에서 MapReduce[3] 분산처리 프로그래밍 기법을 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서 도서 추천 시스템을 위한 관련 연구에 대해 살펴보고, 3장에서 제안된 시스템의 전체적인 내용을 설명한다. 마지막으로 4장에서 결론을 맺는다.

*이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (2010-0015874)

2. 관련 연구

2.1 도서 메타 정보를 이용한 도서 추천 시스템

기존 연구 내용 중에서 아마존의 도서 DB정보와 기계 학습 알고리즘을 이용하여 추천 도서 정보를 추출하는 알고리즘이 있다[2]. 먼저 Amazon subject search

기능을 이용해서 책 정보 URL의 목록을 얻어온다. 얻어온 URL의 웹 페이지를 다운로드하고 페이지에 산재되어있는 정보들을 구조화된 정보로 가공한다. 도서명, 저자, 개요, 리뷰, 독자 코멘트 등의 정보를 추출하고 이를 기계 학습 알고리즘을 이용해 몇 개의 카테고리로 분류한다. 이를 기반으로 사용자에게 도서를 추천하게 되며, 사용자는 추천받은 도서 목록에 대해 연관성 정도를 입력할 수 있다. 또한 이 점수는 더 좋은 연관관계 추출에 사용된다. 아마존에서 수집한 도서의 메타데이타를 기반으로 DB를 생성해두고, 사용자가 입력한 쿼리 문자열을 메타 데이타 안에서 검색하여 도서를 추천해주는 시스템도 존재한다[4]. 또한 웹상에서 사용자들이 서로의 지식을 기반으로 한 도서 추천 시스템에 대한 연구도 진행되었다[5].

2.2 아마존의 오픈 API

아마존에서는 자신들의 사이트에서 판매되는 모든 상품에 대하여 Product Advertising API[6]라는 모델을 이용하여 정보를 제공한다. 제품에 대한 정보는 XML형태로 제공되어 원하는 정보만 선별해 사용할 수 있다. 본 논문에서는 도서의 제목과 저자에 대한 정보를 사용한다.

3. 제안 시스템의 설계 및 구현

본 시스템은 크게 도서의 연관관계를 추출하는 부분과 연관 도서를 검색하는 부분으로 구성된다. 또한 연관관계를 추출하는 부분은 크게 Crawler, TitleIndexer, EBookIndexer, RelationshipBuilder로 구성되어 있으며, 전체적인 데이타 흐름은 그림 1에서 확인할 수 있다.

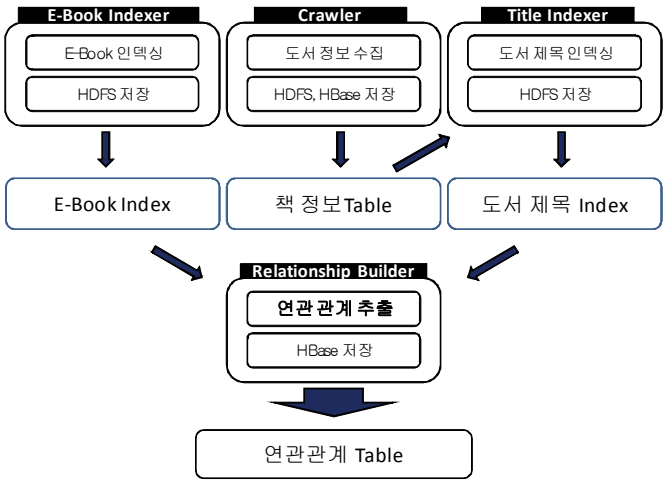

그림 1. 전체 시스템 구성도

E-Book에서 도서 제목을 검출하기 위해서는 우선 도서 DB를 구축하여야 한다. 특정 도서에서 다른 도서를 언급할 때 책 제목, 저자를 함께 언급하는 것에 주목하여, 아마존을 통해 1900년 1월 1일부터 현재까지 출판된 총 30만권의 도서 정보를 수집하였으며, 수집한 도서 제목을 이용해 연관 관계를 추출하였다.

3.1 도서 추천 시스템의 MapReduce 아키텍처

연관 관계를 추출하기 위해 사용한 분산처리 시스템은 구글[7]에서 제안한 MapReduce 프로그래밍 모델에 기초하였다. MapReduce 프로그래밍 모델에서는 커다란 데이타 집합을 처리하기 위하여 Map과 Reduce 함수를 정의하여 병렬로 분산시켜 처리한다.

본 시스템에서는 구글의 GFS[8], BigTable[9], MapReduce를 오픈소스로 구현한 Hadoop[10] 프레임워크를 사용하였다. Hadoop에서는 MapReduce 프로그래밍 모델과 HDFS[11], HBase[12]를 이용하여 구글 플랫폼과 동일한 역할을 할 수 있게 제공해준다.

MapReduce 프로그래밍 모델을 이용하여 구성한 모듈의 Map Task와 Reduce Task에 대한 <Key, Value> 데이타를 그림 2에서 확인할 수 있다.

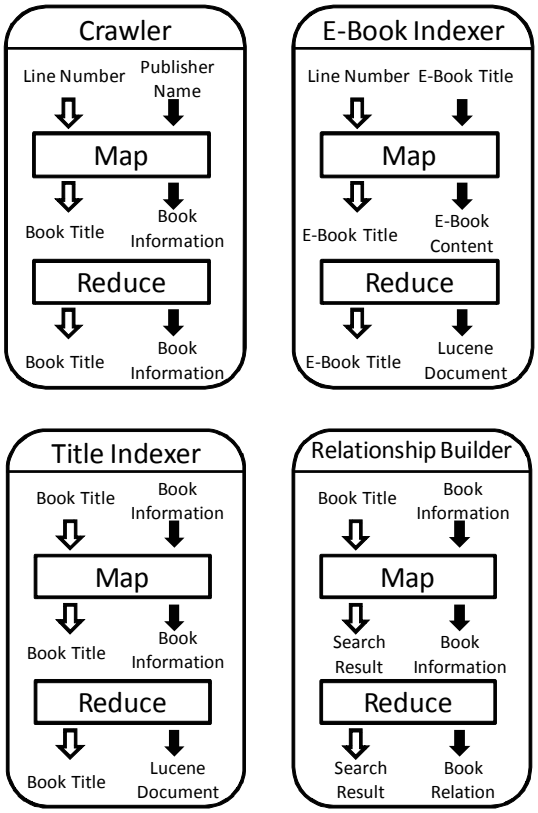

그림 2. 모듈별 MapReduce Tasks

3.2 PDF E-Book 데이타 추출 및 역 인덱스 생성

PDF 파일에서 텍스트를 추출해주는 PDFBox[13] 라이브러리를 이용하여, HDFS 상에 저장되어 있는 E-Book 데이타를 읽어오도록 구현하였다.

데이타 생성 작업을 단순화하기 위해서 E-Book의 파일 이름을 해당 도서의 제목으로 지정하였다. 해당 E-Book PDF 파일의 데이타는 Map Task 안에서 텍스트만 추출되어 저장된다. Reduce 단계에서는 추출된 텍스트와 제목 등의 메타 데이타를 모아 저장한다.

각 단어별 역 인덱스는 오픈소스 검색 엔진인 Lucene[14]을 이용하여 생성하였다. 전체 도서 텍스트 정보와 메타 데이타를 기반으로 하여 단어별 역 인덱스를 추출한다.

Map Task 단계에서는 문서별 텍스트 데이타를 입력으로 받아서 공백문자를 구분자로 하여 단어를 분리한다. 분리된 단어를 Key로 하고, 단어가 속해 있는 문서 제목을 Value로 하여 Reduce Task 단계의 입력으로 보낸다. Reduce Task 단계에서는 단어(Key)와 텍스트의 제목(Value)을 입력 받아서 HBase에 <Key, Value> 형태로 저장한다.

3.3 도서들 사이의 연관 관계 추출

최종적으로 얻어내는 연관 관계는 그림 3과 같다. 수집한 도서 제목, E-Book문서에서 얻어낸 텍스트, 그리고 해당 텍스트를 대상으로 생성한 역 인덱스를 기반으로 도서들 사이의 연관 관계를 추출하였다.

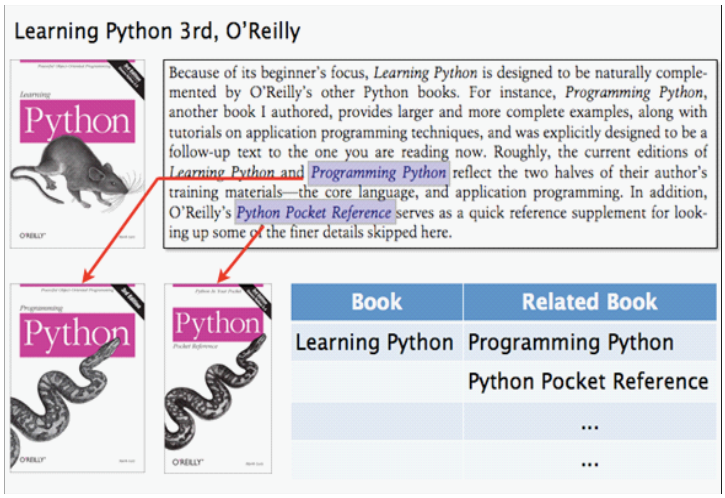

그림 3. 도서들 사이의 연관 관계

연관 관계는 E-Book의 본문 데이타와 E-Book의 제목, 저자 등을 연관 지어 찾아낸다. Map Task에서는 입력된 도서 제목을 인덱스 데이타를 이용해 검색하고, <도서 제목, 현재 검색 중인 E-Book> 결과를 Reduce Task로 보낸다. Reduce Task에서는 도서의 제목과 해당 제목이 언급된 도서 제목을 입력받아 E-Book이 참조하고 있는 도서 목록 데이타를 생성한다.

추출한 데이타의 정확도를 높이기 위하여 두 가지 필터링을 적용하였다. 그림 4에서 필터링 적용 순서를 확인할 수 있다.

첫 번째 필터링 방식은 도서 출간 날짜를 이용한 필터링이다. 특정 도서 이름을 검색하였지만, 해당 도서의 출판 일이 언급된 도서보다 빠르다면 이를 연관 관계로 추출하지 않고 그냥 무시하였다.

두 번째 필터링 방식은 도서의 카테고리를 이용한 것이다. 서로 같은 카테고리에 있는 도서가 아닌 서로 다른 카테고리에 있는 도서를 언급할 경우에도 연관 관계로 추출하지 않는다.

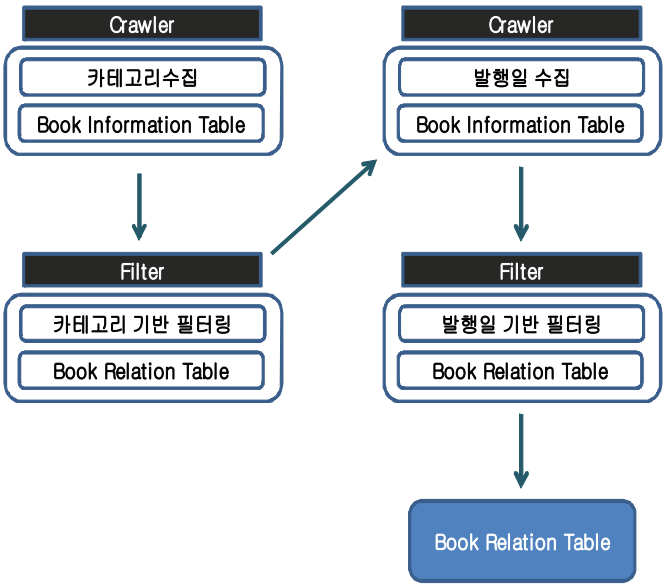

그림 4. 데이타 필터링의 시스템 흐름도

3.4 웹 사용자 인터페이스

MapReduce 프로그래밍 모델을 통하여 추출된 도서들 사이의 연관 관계는 JSP와 Silverlight를 통하여 사용자에게 제공된다. 그림 5의 화면과 같이, 검색창에 도서 제목을 입력하면 해당 도서와 연관되어 있는 다른 도서에 대한 정보를 바로 확인할 수 있다.

그림 5. 웹 사용자 인터페이스

4. 결론

본 논문에서는 도서 안에서 언급된 다른 도서의 제목을 찾는 방법으로 서로간의 연관성을 찾아내고자 하였다. 도서의 본문에서 언급된 다른 도서제목을 서로 간의 연관 관계 데이타로 판단하고 이를 추출하여 연관 관계 DB로 구축한 뒤, 사용자에게 제공한다.

연관 관계 DB를 구축하기 위해서 방대한 분량의 도서 전문을 검색해야하는 문제점이 있다. 이를 해결하기 위해 다수의 노드로 구성된 클러스터에서 MapReduce 프로그래밍 기법을 사용하였다.

도서들 사이의 연관 관계는 Hadoop 프레임워크 안에서 MapReduce 프로그래밍 모델을 통해 추출해낼 수 있고, 결과는 HBase 안에 <Key, Value> 형태로 저장된다. 사용자는 웹 페이지를 통해서 원하는 도서를 검색할 수 있고, 결과 화면을 통해 서로 연관 관계를 가지고 있는 도서를 확인할 수 있다. 도서의 내용을 직접 분석하여 서로 관계있는 도서를 추천해주기 때문에, 같이 읽으면 좋은 도서를 쉽게 찾을 수 있다.

5. 참고 문헌

[1] Amazon.com http://amazon.com

[2] Raymond J. Mooney, Loriene Roy, "Content-Based Book Recommending Using Learning for Text Categorization", Proceedings of the fifth ACM conference on Digital libraries, pages 195-204, 2000

[3] Sanjay Ghemawat, Howard Gobioff, Shun-Tak, "The Google File System", Proceedings of the nineteenth ACM Symposium on Operating systems principles, Pages 29-43, 2003

[4] Binge Cui, Xin Chen, "An Online Book Recommendation System Based on Web Service", Proceedings of the 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Vol. 7, Pages 520-524, 2009

[5] Chin-Yeh Wang, Fu-Hsiang Wei, Po-Yao Chao, Gwo-Dong Chen, "Extending e-Books with Contextual Knowledge Recommenders by Analyzing Personal Portfolio and Annotation to Help Learners Solve Problems in Time", Proceedings of the IEEE International Conference on Advanced Learning Technologies, Pages 306-310, 2004

[6] Amazon Product Advertising API http://docs.amazonwebservices.com/AWSECommerceService/latest/DG/index.html?rest-signature.html

[7] Google, http://google.com

[8] Jeffrey Dean, Sanjay Ghemawat, "MapReduce: simplified data processing on large cluster", Proceedings of The 6th Conference on Symposium on Operating Systems Design & Implementation, (OSDI 04), Usenix Assoc., Pages 137-150, 2004

[9] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Andrew Fikes, Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", Proceedings of the 7th Symposium on Operating Systems Design and Implementation, Pages 205-218, 2006

[10] Hadoop, http://hadoop.apache.org/core/

[11] HDFS, http://hadoop.apache.org/hdfs/

[12] HBase, http://hadoop.apache.org/hbase/

[13] PDFBox http://pdfbox.apache.org/

[14] Lucene, http://lucene.apache.org/