2018 Database System Project #3

Text mining with MongoDB

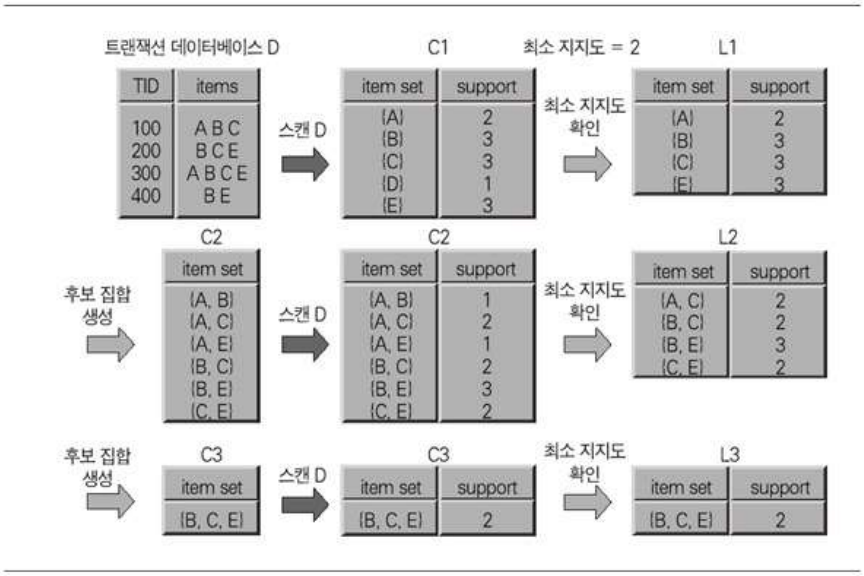
1 문제 정의

본 프로젝트에서는 텍스트 마이닝 기법 중 하나인 Apriori Algorithm을 이용하여 제공된 뉴스 기사들을 분석하고 뉴스 기사에서 주로 쓰이는 단어들을 알아내는 프로그램을 작성한다. 또한 뉴스 기사 분석을 위해 비정형 데이터를 다루기 쉬운 NoSQL 기반 데이터베이스인 MongoDB를 사용함으로써 NoSQL 데이터베이스의 사용법을 익힐 뿐 아니라 관계형 데이터베이스와 NoSQL 데이터베이스간의 차이점을 인식하는 것을 목적으로 한다.

텍스트 마이닝이란 비정형 데이터 마이닝의 유형 중 하나로 자연어 처리 기술과 문서 처리 기술을 적용하여 유용한 정보를 추출, 가공하는 것을 목적으로 하는 기술이다. 실생활에서 만들어 지는 대부분의 문서는 형태로 보관되며 제목, 저자, 출판 날짜 등과 같은 구조적인 특징들과 문서의 요약, 내용과 같은 크기가 일정하지 않은 비 구조적 요소들을 포함하기에 반 구조적 데이터로 분류된다. 응용 분야로는 Risk management, Knowledge management, Cybercrime prevention, Customer care service, Business intelligence, Spam filtering 등이 있다.

Apriori 알고리즘은 연관 규칙을 찾아 주는 알고리즘 중에서 가장 먼저 개발됐고 또 가장 많이 쓰이는 알고리즘이다. 연관 규칙 분석이란 어떤 두 아이템 집합이 번번히 발생하는가를 알려 주는 일련의 규칙들을 생성하는 알고리즘이다.

예를 들어, 소비자가 산 물건의 목록을 가지고 있는 데이터베이스가 있다고 생각해 보자. 한 소비자마다 구매한 물건의 목록이 하나 혹은 여러 개일 수 있는데, 한 소비자마다 산 물건의 목록을 가진 데이터를 하나의 트랜잭션(transaction)이라고 하면, 여러 트랜잭션이 모여서 거대한 데이터베이스가 된다. 패턴이라는 것은 물건들의 조합으로 생각하면 쉽다. 예를 들어, 사과를 산 손님은 몇 명이고, 사과와 귤을 같이 산 손님은 몇 명인지 알고 싶을 때, <사과>, <사과, 귤> 이런 것들을 반복적으로 나타날 수 있는 패턴이라고 생각하면 된다. 하지만 결국 의미 있는, 가치 있는 정보가 되기 위해서는 저런 패턴들이 일정 수 이상 반복해서 나타나야 할 것이다. 그래서 많이 반복되는 패턴만을 골라 내기 위해 minimum support threshold(minsup)을 사용한다. 예를 들어, 트랜잭션이 5개이고 minsup이 40%라면, 5*40%=2, 즉 2번 이상 반복되는 패턴을 찾겠다는 말이 된다. 이런 조건을 가지고 데이터베이스에서 minsup이상의 반복이 나타나는 모든 길이의 패턴을 전부 찾는 알고리즘이 Apriori 알고리즘이다. 아래의 그림은 Apriori 알고리즘의 한 예이다.

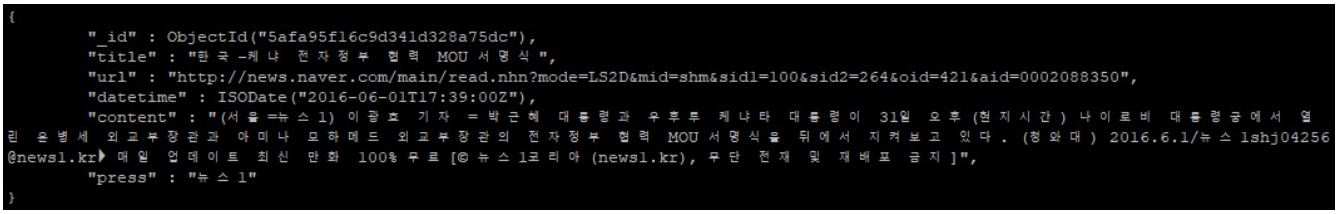


MongoDB는 C++로 작성된 오픈 소스의 문서 지향(Document-Oriented)적 데이터베이스이며 기존의 RDBMS의 한계를 극복하기 위해 만들 어진 새로운 형태의 데이터 저장소이다. 관계형 데이터베이스가 아니므로 RDBMS처럼 고정된 스키마 및 JOIN 연산이 존재하지 않으며, Document라고 불리는 기본 데이터 구조 단위로 이루어진다. 모든 데이터 구조는 한 개 이상의 key-value 쌍으로 이루어진다.

```
{
    "_id": ObjectId("5099803df3f4948bd2f98391"),
    "username": "ironman",
    "name": { first: "Tony", last: "Stark" }
}
```

2 요구사항

본 프로젝트에서는 제공된 뉴스기사에 대해 형태소 분석을 통해 문서를 키워드의 집합으로 분할한 후 Apriori 알고리즘을 수행한다. 다음은 MongoDB에 저장된 뉴스기사에 대한 예제 화면이다. 아래 예제 화면을 참고하여 문서의 구조를 분석하고, 2.1 ~ 2.4까지의 작업을 진행한다.



2.1. MongoDB 기본 질의

다음의 질의를 작성하고 질의문과 결과를 보고서에 기입하시오.

1) 전체 기사의 개수를 구하시오.

2) '연합뉴스' 사에 실린 기사들의 개수를 구하시오.

3) 2016년 6월 1일부터 2016년 6월 12일까지 실린 기사 개수를 구하시오.

4) 전체 기사를 날짜 별로 정렬해서 가장 빠른 날짜와 늦은 날짜의 기사 제목을 출력하시오.

5) '연합뉴스' 사에 실린 기사 혹은 2016년 6월 5일부터 2016년 6월 20일까지 실린 기사의 개수를 구하시오.

2.2. 뉴스 기사 전처리 과정

2.2.1. 형태소 분석 및 불용어 처리

형태소 분석은 텍스트 마이닝 시 자연어 처리의 첫 번째 단계로 입력 문자열을 형태소 열로 바꾸는 작업을 말한다. 형태소란 의미의 최소 단위로써 더 이상 나눌 수 없는 가장 작은 단위의 의미 요소를 말한다. (1)모든 뉴스 기사에 대해 제공된 형태소 분석 소스 코드와 불용어 리스트 파일을 이용해 텍스트 분석에 불필요한 단어(불용어)를 제거하고, 형태소 열이 추가된 상태로 데이터베이스를 update한다. 또한, (2)사용자로부터 뉴스 기사의 url을 입력 받아 해당하는 뉴스 기사의 형태소들을 출력 해준다.

2.2.2 한 기사 내의 형태소 집합 구하기

형태소 집합은 각 기사 들마다 포함되어 있는 형태소들로 이루어진 집합이다. 형태소 분석 단계에서와 마찬가지로 (3)모든 뉴스 기사에 대해 각 기사에 나오는 단어들을 확인하고, 그 단어들을 집합으로 만들어 새로운 collection에 저장한다. 새로운 collection 이름은 "news_wordset"으로 한다. 그 후, 마찬가지로 (4)사용자로부터 뉴스 기사의 url를 입력 받아 해당하는 뉴스 기사의 Word set을 출력 해준다.

2.3 Apriori 알고리즘 구현

2.3.1 min sup을 만족 시키는 frequent itemset 생성

본격적으로 Apriori 알고리즘을 수행하기 위해 2.2.2에서 만든 형태소 집합을 이용해 frequent itemset을 만든다. 형태소 집합에 속한 하나의 단어의 support값이 minimum support count값보다 크다면 이 단어를 frequent item 이라한다. (5)frequent 1-itemset, frequent 2-itemset,frequent 3-itemset을 형성하고 DB에 저장하는 프로그램을 작성하라. DB이름은 candidate_L(number)로 한다. (ex. candidate_L2)　　(min_sup = 10%)

```
> show collections
candidate_L1
candidate_L2
candidate_L3
news
news_freq
news_wordset
system.indexes
```

```
> db.candidate_L2.findOne()
{
        "_id" : ObjectId("5b07ad256c9d34361e0c2b09"),
        "item_set" : [
                "자 신 ",
                "생 각 "
        ],
        "support" : 11
}
```

2.4.2 strong 연관 규칙을 생성

만들어 진 frequent itemset을 이용해 strong 연관 규칙을 만들어 낸다. 연관 규칙은 X=>Y로 표기한다. Strong 연관 규칙이란 threshold를 넘는 confidence를 가지는 규칙들을 의미한다. Confidence는 X라는 상품을 구매한 건 수 가운데 Y도 같이 구매한 비율을 의미하며 조건부 확률 P(Y|X) 로 나타낸다.

TID	List of item_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3

예를 들어, 위의 dataset에서 frequent itemset I = {I1,I2,I5} 일 때, I의 공집합이 아닌 부분 집합은 {I1,I2}, {I1, I5}, {I2,I5}, {I1}, {I2}, {I5}가 있다. 이들의 confidence를 구하면 (min conf = 80%)

I1∧ I2 → I5, confidence = 2(I1∧ I2∧I5 의 count값)/4(I1∧ I2 의 count값) = 50% (< min conf)

I1∧ I5 → I2, confidence = 2/2 = 100% (> min conf)

I2∧ I5 → I1, confidence = 2/2 = 100% (> min conf)

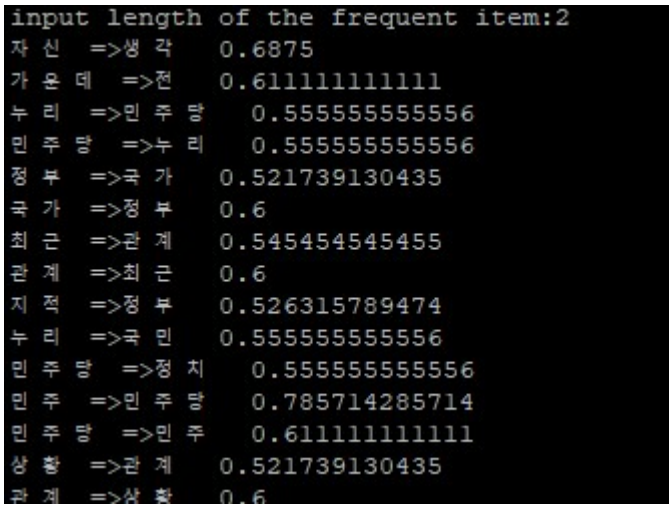
I1→ I2 ∧ I5, confidence = 2/6 = 33% (< min conf)

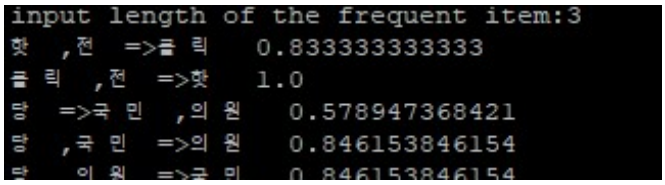
I2→ I1 ∧ I5, confidence = 2/7 = 29% (< min conf)

I5→ I1 ∧ I2, confidence = 2/2 = 100% (> min conf)

2번째, 3번째 그리고 마지막 연관 규칙이 strong 연관 규칙이 된다.

(6)본 프로젝트에서는 frequent n-th item set의 n을 입력 받았을 때 frequent n-th itemset에서의 strong 연관 규칙을 모두 출력하는 프로그램을 작성하여라.(min_conf = 50%)





3 사용 환경

서버: Host - dbpurple.sogang.ac.kr / Port - 22

운영 체제: Ubuntu 14.04.5 LTS

데이터베이스: Mongodb 3.0.14

사용 언어: python 2.7.6

라이브러리: pymongo, MeCab

서버 계정: db학번 (e.g. db20181234)

서버 비번: db학번 (e.g. db20181234)

데이터베이스 계정: db학번 (e.g. db20181234)

데이터베이스 비번: db학번 (e.g. db20181234)

4 제출물

4.1 기술 문서(보고서)(50점)

4.1.1 MongoDB 질의문 및 결과(40점)

데이터베이스 서버에 접속해 MongoDB 질의를 수행하고 결과 화면을 screenshot으로 첨부할 것.

4.1.2 RDB vs. NoSQL DB에 대한 비교(10점)

텍스트 마이닝을 수행할 때에는 RDB보다 NoSQL DB를 사용하는 것이 좋은가? 이에 대한 답을 하고 이유를 기술하시오(진행된 프로젝트와 관련 지어 기술).

4.2 Python 프로그램 파일(50점)

작성한 프로그램 소스 파일

5 제출방법

5.1 Hard Copy

기술 문서를 1부 출력하여 제출

5.2 Soft Copy

기술 문서와 Python 코드 파일을 압축하여 e-mail로 제출

db2018spring@gmail.com

파일 명 및 메일 제목의 양식은 다음을 따를 것.

기술 문서 파일 : DBprj#3_학번.docx (e.g. DBprj#3_20181234.docx)

Python 파일 : DBprj#3_학번.py (e.g. DBprj#3_20181234.py)

압출 파일 : DBprj#3_학번.zip (주의 : zip 이외의 다른 압축 형식은 받지 않음)

메일 제목 : DBprj#3_학번

6 제출 기한

Hard Copy: 6월 7일(목) 17:00시 전까지 AS916 앞 상자에 제출

Soft Copy: 6월 7일(목) 17:00시 전까지 e-mail제출

7 평가 기준

- 요구 사항들이 적절히 반영 되었는가
- 제출물이 정해 진 기한 내 제출 되었는가

8 기타

- **데이터베이스 접속 계정 및 방법은 추후 공지하고 관련 내용을 실습할 예정**
  실습일: 5/25(금) 질의 응답: 6/1(금)
- 입력 형식과 출력 형식은 자율에 맡기되 사용자 편의성을 고려해 구성할 것.
- Copy는 1회 적발 시 0점 처리, 2회 적발 시 과목 성적 F 처리
- 다음과 같은 경우 감점
  - 기한을 지키지 않은 경우.
  - E-mail이나 hard copy 중 한 방식으로만 제출 하면 50%감점
  - 첨부 파일의 압축이 손상되거나 바이러스가 있는 경우 0점 처리
  - 제출 양식을 지키지 않은 경우 제출물을 찾지 못하면 미 제출 처리 될 수 있음. 그 이외의 경우 10%씩 감점