# 줄거리에 기반한 TV 시리즈물 검색 시스템의 설계 및 구현

조진표[1], 천영우[2],김유섭[3],고영웅[4]
한림대학교 정보통신공학부[1 2 3 4]
hal96[1]@hallym.ac.kr,
frtbule[2]@unitel.co.kr,{yskim01[3],yuko[4]}@hallym.ac.kr

# Design and Implementation of TV Serial Drama Retrieval System Based on Synopsis

Jin Pyo Cho[1] , Young Woo Chun[2], Yu Seop Kim[3],Young Woong Ko[4]
Division of Information Engineering & Telecommunications,
Hallym University[1 2 3 4]

**요 약**

본 논문은 스토리에 기반하여 여러 편의 드라마 동영상 파일 중에서 사용자가 찾고자 하는 파일을 찾아주는 검색 시스템의 설계 및 구현에 대하여 기술한다. 기존의 동영상 검색 방식은 제목 및 주연 배우와 같이 제한적인 검색어에 의한 검색을 제공하고 있으나, 대부분의 드라마 파일은 영화 또는 다른 동영상 파일과 달리 여러 회로 나누어져 있기 때문에 기존의 주연 배우나 제목과 같은 정보만을 가지고 사용자가 원하는 파일을 검색하기 어렵다. 본 논문에서는 드라마의 제작사에서 제공하는 각 회당 스토리를 기본적인 불리안 모델과 결합시켜 사용자가 원하는 회차를 검색하는데 유용한 시스템을 설계 및 구현하였다. 본 논문에서 제시한 방식의 유용성을 보이기 위하여 실제 서비스 되고 있는 드라마를 대상으로 실험을 하였으며, 실험결과 높은 검색 능력을 보이고 있다.

Keyword : 드라마, 동영상, 줄거리 기반, 검색, 불리안 모델

## 1. 서론

인터넷의 발전과 더불어 방송국 또는 이동 통신 업체에서 유선 또는 무선으로 동영상 스트리밍(Streaming) 서비스를 제공하고 있다. 또한 인터넷에 널리 퍼져있는 P2P 사이트와 일부 포털 사이트에서는 동영상 파일의 다운로드를 제공한다.

현재 방송국 홈페이지 또는 이동 통신사 업체에서 제공하는 동영상 스트리밍 서비스는 사용자가 원하는 드라마를 선택한 후 결과 리스트에서 해당 회를 선택하거나, 검색어를 통해 검색을 하여야 한다. 이때 검색은 드라마 이름 또는 영화 이름, 주연 배우 이름 등과 같이 매우 제한적인 검색어에 의존적이다. 시리즈로 구성된 드라마는 같은 제목, 같은 주인공, 같은 제작사로 되어 있으며 여러 개의 파일로 구성되어 있다.

KBS 의 불멸의 이순신[1]과 같은 경우 100 여 개의 파일이 존재하며, MBC 의 대장금[2]과 같은 경우 마지막 특집 방송을 제외하면 54 개의 파일이 존재한다. 이와 같이 동일한 제목, 주인공, 제작사를 가지고 있는 파일에 대해서 기존의 검색 시스템을 사용하여 사용자가 원하는 파일을 검색하기란 매우 힘들다.

예를 들어 본 논문에서 실험 데이터로 사용된 드라마 대장금과 같은 경우 총 54 회로 이루어져 있으며, 54 개의 동영상 파일이 존재한다. 만약에 어떤 사용자가 자신이 원하는 스토리에 해당하는

* This work was supported by the Industry University Research Institute Consortium grant from the Small & Medium Business Administration

정보만 알고 그 스토리가 몇 회 인지 알지 못할 경우 기존의 검색 방법(드라마 이름, 배우 이름)으로 검색을 한다면 검색 결과는 54 개의 파일이 될 것이며 사용자는 54 개의 파일 중에서 자신의 원하는 파일을 찾기 위해 최악의 경우 54 개의 동영상 파일을 재생해 보거나 제작사에서 제공하는 54 개의 스토리 파일을 열어 보아야 하는 문제가 있다. 그리고 대부분의 스트리밍 서비스를 제공하는 사이트 또는 일부 P2P 사이트는 유료로 운영이 되며, 무료로 운영된다 하더라도 모든 파일을 열람해 보는 것은 사용자에게 큰 부담이 될 수 있다. 이런 이유로 본 논문에서는 보다 효율적인 검색 시스템을 제안하고 이에 대한 설계 및 구현에 대하여 기술한다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서는 본 연구와 관련된 대내외 연구 결과에 대해서 기술하며, 3 장에서는 본 연구에서 제안하는 검색 기법을 서버와 클라이언트의 관점에서 기술한다. 4 장에서는 실제 널리 사용되는 데이터를 이용하여 실험을 수행하고 그 결과를 기술한다. 마지막으로 5 장에서 결론을 맺고 향후 연구 내용에 대해서 기술한다.

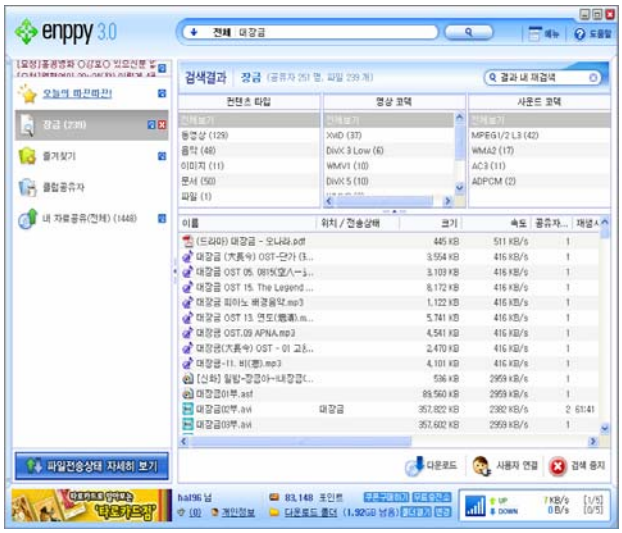
## 2. 관련 연구

본 논문에서 제안하는 검색 시스템은 불리안 모델과 벡터[3][4], 확률 모델[5]과 같은 여러 가지 정보 검색 모델 중에서 불리안 모델을 기반으로 구현하였다. 불리안 모델은 집합론과 불리안 대수에 기반한 단순 모델로서, 정보 검색 시스템의 일반 사용자도 쉽게 이해할 수 있는 틀을 제공하며, 질의는 명확한 의미를 가진 불리안 표현으로 정의된다. 고유의 단순성과 간결한 형식 때문에 불리안 모델은 옛부터 선호되어 왔으며, 많은 상용 검색 시스템에 채택되어 왔다. 불리안 모델의 장점으로는 모델의 명확한 형식과 단순성이며, 단점으로는 정확한 정합 때문에 검색 문헌이 너무 작거나 많다는 점이다. 이러한 단점을 보안하는 모델에는 퍼지 집합 모델[6]과 확장 불리안 모델[7]이 있다.

동영상 스트리밍 서비스를 제공하는 방송국 사이트는 검색어의 입력 없이 단순한 html 링크를 통하여 검색이 이루어 진다. 예를 들어 대장금과 같은 경우 www.imbc.com 에 접속하여 좌측 중앙에 있는 “ 종영 드라마” 메뉴를 클릭, 드라마 제목으로 정렬된 리스트에서 대장금 드라마를 선택 후 “ 다시보기” 메뉴를 선택 후 각 회차 별로 제공되는 스토리 파일을 하나씩 확인해 보아야 한다. 이렇게 검색어를 사용할 수 없는 검색 방법은 검색어를 처리할 필요가 없기 때문에 시스템 성능에는 효율적이지만 드라마의 수가 많아질수록 사용자에게 많은 불편함을 안겨 줄 것이다.


[그림 1] MBC 대장금 검색

또한, 동영상 다운로드 서비스를 제공하는 사이트에서는 단순히 드라마의 이름과 회차 정보만으로 검색이 이루어 진다. 예를 들어 “ enppy” [8]와 같은 동영상 P2P 다운로드 사이트는 검색창에 해당 드라마의 이름과 회차를 입력하는 것으로 검색이 이루어 진다. 대부분의 동영상 다운로드 서비스에서는 검색어를 입력하여 검색하는 방식을 사용하고 있지만 검색어가 드라마의 이름과 회차 번호로 한정되어 있으며, 몇몇 동영상은 저작권에 의해 금지어에 등록되어 검색이 되지 않는 단점이 있다.

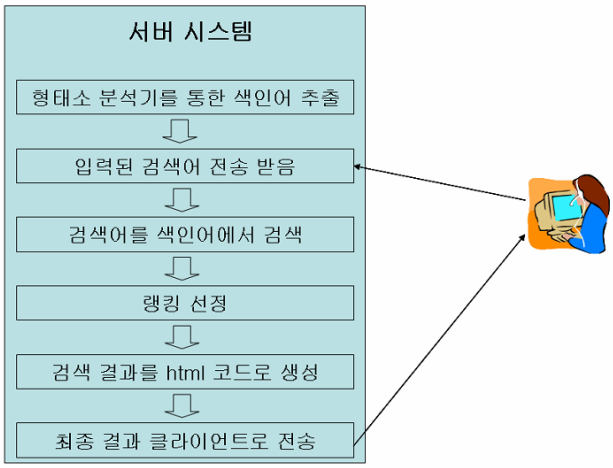



[그림 2] 동영상 P2P 사이트 검색 화면

## 3. 검색 시스템 구성

본 논문에서 구현한 스토리 기반 검색 시스템은 크게 서버와 클라이언트로 구성된다. 스토리 기반 검색 시스템의 기본 구조를 간략히 설명하면 다음과 같다.

먼저 클라이언트에서 사용자가 원하는 검색어를 입력한다. 서버는 동영상 제작사에서 제공하는 스토리 파일을 형태소 분석기를 이용하여 색인어를 추출한다. 그리고 색인어를 기준으로 색인어가 출현한 문서와 해당 문서에서 색인어의 출현 횟수로 데이터를 재구성 한다. 이 데이터를 이용하여 사용자가 입력한 검색어와 불리안 모델을 기반으로 검색한다. 전체적인 시스템 흐름도는 다음과 같다.



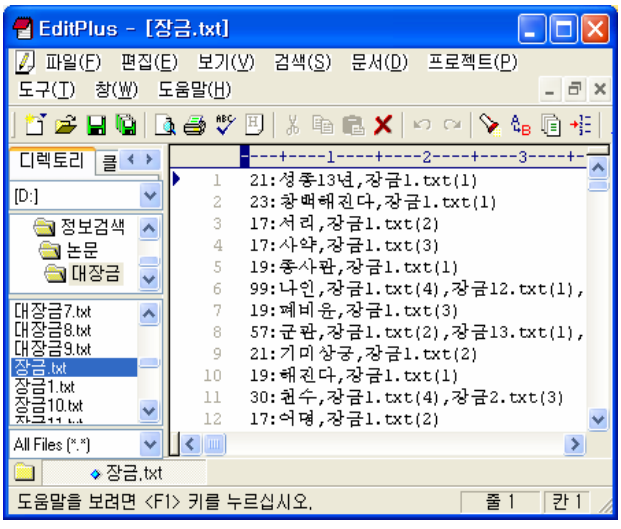

[그림 3] 스토리 기반 검색 시스템 흐름도

### 3-1 서버 시스템 구성

#### 3-1-1 기초 데이터 생성

서버 시스템에서 스토리 기반의 검색 서비스를 제공하기 위해서는 먼저 드라마 각 회에 해당하는 스토리 파일을 형태소 분석기[9]를 이용하여 색인어를 추출한다. 추출된 색인어는 드라마의 각 회당 하나의 파일로 생성되며 색인어와 해당 파일에 색인어의 발생 횟수에 대한 정보를 가지고 있다. 이렇게 생성된 파일들에는 일부 색인어가 중복되어 나타나는 경우가 발생 하므로 색인어를 기준으로 해당 색인어가 나타나는 스토리 파일과 그 파일에서의 색인어 발생 빈도를 포함하는 하나의 새로운 파일을 생성한다.

위 작업은 스토리 기반 검색 시스템을 구현하기 위해 선행되어야 하는 작업으로 사용자가 입력한 검색어를 검색할 기초 데이터가 된다. 데이터의 형식은 다음과 같다. 첫 번째 숫자는 ' :' 문자 이후로 한 라인의 바이트 수를 나타낸다. 그 다음 ' :' 은 단순한 구분자 이며 구분자 이후 색인어가 위치하게 된다. 색인어 다음에는 색인어가 출현한 문서의 이름과 ()안에 색인어가 해당 문서에 출현한 횟수를 나타낸다. 색인어와 문서, 문서와 문서는 ' ,' 로 구분된다.



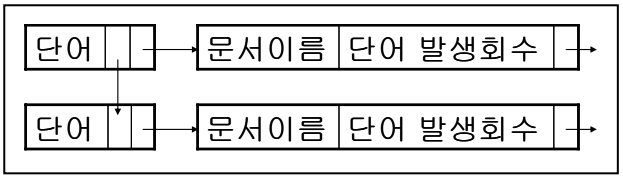

[그림 4] 기초데이터 형식

#### 3-1-2 검색어 검색

데이터 파일이 생성된 후에 서버에서 사용자의 검색어를 이용하여 해당 동영상 파일을 검색하고 랭킹을 부여하는 작업을 수행하게 된다.

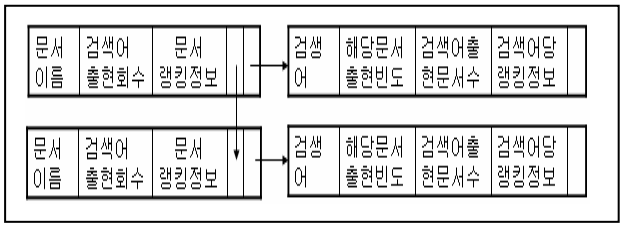
사용자가 클라이언트에서 검색어를 입력하는 경우, 검색 서비스의 요청이 발생하면 서버 시스템은 앞에서 생성한 데이터 파일의 색인어를 기준으로 하여 서버의 메모리에 연결 리스트(linked list) 형태로 적재한다.

색인어를 기준으로 생성한 연결 리스트의 구조는 그림 5 와 같다.




[그림 5] 색인어 기준 데이터 구조

그림 5 의 자료 구조가 완성되면 사용자로부터 입력 받은 검색어와 그림 5 의 자료 구조의 색인어를 비교하여 일치하는 색인어에 대한 문서이름 및 단어 발생회수, 랭킹을 부여하기 위해 필요한 기타 다른 정보를 생성하여 그림 6 와 같은 연결 리스트를 생성한다.




[그림 6] 검색 결과 데이터 구조

그림 6 의 자료구조는 검색된 단어를 포함하는 문서를 기준으로 생성된다. 위 자료 구조는 문서 이름, 검색어 출현횟수, 문서 랭킹 정보, 검색어 해당 문서 출현 회수, 검색어 출현 문서 수, 검색어 랭킹 정보를 포함한다.

여기서 문서 이름은 검색어를 포함하고 있는 스토리 파일의 이름이며, 검색어 출현 횟수는 해당 문서에 사용자가 입력한 검색어가 몇 개나 출현했는지 그 수를 의미한다. 그리고 문서의 랭킹 정보는 해당 문서에 출현한 각각의 검색어에 대한 랭킹 정보의 값을 더한 값이며, 검색어는 사용자가 입력한 검색어 중에서 해당 문서에 출현한 검색어를 의미한다. 해당 문서 출현 회수는 문서 파일을 기준으로 해당 검색어의 출현 회수를 나타내고, 검색어 출현 문서 수는 해당 검색어가 전체 스토리 파일 중에서 몇 개의 파일에 출현했는지를 나타낸다. 마지막으로 검색어 랭킹 정보는 해당 문서에 검색어가 출현한 회수 나누기 검색어가 출현한 문서의 수 이다.

### 3-1-3 랭킹 부여 방법

이렇게 생성된 자료 구조는 아직 랭킹을 부여하기 위한 자료를 가지고 있을 뿐 랭킹과는 무관한 순서로 나열되어 있다. 본 논문에서는 이 자료 구조에 다음과 같은 랭킹 선정 작업을 수행한다.

첫 번째 해당 문서에 출현한 검색어의 수를 기준으로 내림차순 정렬을 수행한다. 만약 여기서 해당 문서에 출현한 검색어의 수가 같은 또 다른 문서들이 존재한다면 두 번째로 문서 랭킹 정보를 내림차순으로 정렬 한다. 문서 랭킹 정보 계산에 사용되는 식은 다음과 같다.

$$R = \sum_{i=1}^{n} \frac{TF_i}{N_i}$$

n = 해당 문서에서 검색된 단어의 수.

TF = 해당 검색어가 해당 문서에 출현한 횟수

N = 해당 검색어가 출현한 문서의 수

R = 문서 랭킹 정보

위의 수식으로 계산된 결과 R 값을 이용하여 R 값이 큰 파일에 대하여 높은 랭킹을 부여한다.

이러한 방식으로 검색어를 포함한 줄거리 파일에 대한 검색과 랭킹부여 작업이 끝나면 그 결과를 토대로 결과를 클라이언트에 전송한다.

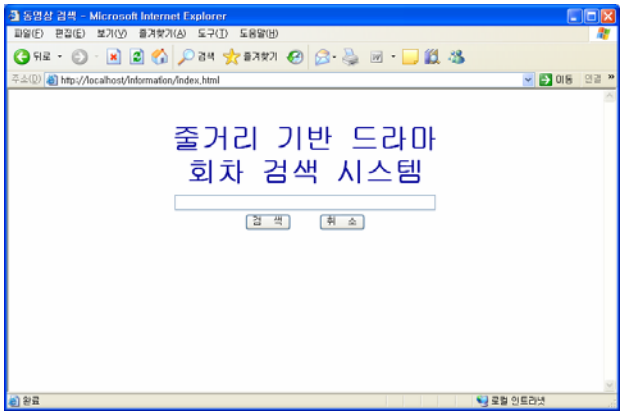
이때 전송되는 결과 데이터는 랭킹 순서에 맞추어 해당 스토리 파일의 회차와 해당 스토리 파일에 출현한 검색어, 랭킹 값 그리고 스토리 파일의 링크와 동영상 파일 링크를 포함하는 html 코드를 생성, html 코드를 클라이언트에 전송한다.

## 3-2. 클라이언트 시스템 구성

클라이언트는 PHP 를 이용하여 작성되었으며 사용자가 입력한 검색어를 서버 시스템으로 전송

및 검색된 결과를 사용자에게 보여주는 역할을 한다.

웹에서 검색 서버로 접속하면 아래 그림 7 과 같은 화면이 웹브라우져에 나타난다. 여기서 사용자가 원하는 스토리에 대한 검색어를 입력한 후 검색 버튼을 누르면 입력 받은 검색어를 서버로 전송하게 된다.




[그림 7]클라이언트 실행 화면

클라이언트에서 입력 받은 검색어를 서버로 전송하는 방법은 PHP 의 passthru 함수를 사용하여 검색어를 argument 로 하여 서버의 검색 프로그램을 실행하는 것으로 이루어 진다. 그리고 서버의 검색 프로그램에서 검색 결과로 생성한 html 코드를 웹브라우저에 출력해 준다.

클라이언트에 최종적으로 보여지는 검색 결과는 사용자가 입력한 검색어와 아래 항목을 리스트 형태로 보여준다.

- 해당 회차
- 해당 회차에서 검색된 검색어,랭킹 값
- 줄거리 파일 링크
- 동영상 파일 링크

다음 그림은 스토리 기반 드라마 검색 시스템의 결과 화면을 보여준다. 동영상은 윈도우 미디어 플레이어를 사용하였다.


[그림 8] 검색 결과 화면

## 4. 실 험

본 논문에서 제시한 시스템 모델의 유용성을 보이기 위하여 실제 널리 사용되는 데이터를 이용하여 실험을 하였다. 실험 데이터는 MBC 에서 제작한 대장금 드라마를 사용하였으며, 검색어에 대해서 어느 정도의 정확도로 해당 회차를 상위 랭킹에 보여주는 지에 초점을 두었다.

실험 방법은 다음과 같다. 먼저 본 논문에서 제시한 줄거리 기반 검색 시스템의 내부 동작 구조를 모르는 5 명의 피실험자를 선발 하였다. 이들에게 각각 대장금 32 회를 시청하도록 한 후 각각 5 단어로 구성된 검색어 집합을 만들도록 하였으며 이 데이터를 실험에 사용하였다.

실험 결과는 다음 표 1 과 같다.

| 실험자 | 랭킹 순위 | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| A | 31 | 34 | 47 | 33 | 32 | 49 | | | | | | | |
| B | 32 | 48 | 31 | 2 | 34 | 11 | 47 | 33 | 7 | 49 | 4 | 5 | 8 |
| C | 32 | 48 | 2 | 11 | 7 | | | | | | | | |
| D | 32 | 11 | 7 | | | | | | | | | | |
| E | 32 | 11 | 7 | | | | | | | | | | |

[표 1] 실험 결과 표

A 실험자의 검색어 집합으로 검색한 결과 6 개의 동영상 파일이 검색되었고 그 중에서 32 회는 5 위에 랭킹 되었으며, B 실험자는 13 개의 동영상 파일이 검색, 32 회는 1 위로 랭킹 되었다. C 와 D, E 실험자도 각각 5, 3 ,3 개의 동영상 파일이 검색되었으며, 모두 32 회는 1 위로 랭킹 되었다. 그리고 D 와 E 실험자는

서로 다른 검색어 집합을 생성 하였지만 결과는 동일한 것으로 나타났다.

## 5. 결 론

본 논문에서 구현한 스토리 기반 드라마 검색 시스템은 기존의 검색 시스템보다 풍부한 검색어를 제공하며 사용자가 원하는 스토리의 동영상 파일을 검색해 준다. 이것은 동영상 재생 서비스를 이용하는 사용자에게 보다 편리한 환경을 제공하여 동영상 재생 서비스를 사용하는 사용자의 증가를 통해 동영상 재생 서비스 발전에 기여할 수 있을 것이다.

향후 검색어가 단어로 한정되어 있는 문제에 대하여 검색어를 단어뿐만 아니라 문장으로 확장하여 드라마 동영상 파일을 검색할 수 있는 시스템을 구현할 예정이다. 또한 현재 검색 시스템은 매번 사용자가 검색을 요구할 때마다 서버의 검색 프로그램이 실행되므로 많은 사용자가 검색을 요청할 경우 성능이 좋지 않을 수 있다. 따라서 서버의 검색 프로그램을 데몬 형식으로 수정하고 검색에 필요한 자료구조를 미리 생성하여 메모리에 적재하는 방식으로 성능을 개선하고자 한다.

## 참 고 문 헌


[1]http://www.kbs.co.kr/drama/leesonshin/view/vod/vod.html

[2]http://www.imbc.com/broad/tv/drama/daejanggum/vod/index.html

[3]G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. Journal of the ACM, 15(1):8-36, January 1968.

[4]G. Salton. The SMART Retrieval System – Experiments in Automatic Document Processing. Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[5]S. E. Robertson and K. Sparck Jones. Relevance weighting of search terms. Journal of the American Society for Information Sciences, 27(3):129-146,1976

[6]Y. Ogawa, T. Morita, and K. Kobayashi. A fuzzy document retrieval system using the keyword connection matrix and a learning method. Fuzzy Sets and Systems, 39:163-179,1991

[7]Gerarad Salton, Edward A. Fox, and Harry Wu. Extended Boolean information retrieval. Communications the ACM, 26(11):1022-1036,November 1983

[8] http://enppy.entica.com/

[9] http://nlp.kookmin.ac.kr