의사결정 트리를 이용한 야구기사 작성 기법

김주봉, 고현영, 용상혁, 한연희*
한국기술교육대학교 컴퓨터공학부
e-mail:{gohyunyung, rlawnqhd3, yong632, yhhan}@koreatech.ac.kr

Generating baseball articles using decision tree

Hyun-yung Go, Ju-bong Kim ,Yong-sang Hyuk , Youn-Hee Han1)
School of Computer Science and Engineering
Korea University of Technology and Education

요 약

'야구경기 결과에 대해 자동으로 기사를 작성할 수 있는가'에서 본 논문에서는 야구 경기 데이터들을 기반으로 의사결정 트리기법을 사용하여 경기결과의 문맥과 기사작성에 필요한 요소들을 자동으로 추출해보았다. 그 결과 해당경기의 데이터를 가지고 객관적인 야구기사를 생산해 낼 수 있음을 도출해냈다.

1. 서론

최근 로봇 저널리즘이 다양한 분야에서 나타나고 있으며, 미국 및 영국의 주류 언론에서는 로봇이 쓴 기사를 볼 수 있으며 학계에서도 관련 논문이 출판되고 있으며, 로봇 저널리즘 프레임워크는 데이터 수집, 이벤트 추출, 중요 이벤트 선별, 기사의 무드 결정, 뉴스 기사 생성의 다섯 단계로 이루어진다고 한다[1]. 미국의 'LA Times'는 지진 보도 알고리즘 'Quake-bot'이 사실에 기반, 기자를 대체할 수 있을 정도의 수준으로 발전하였다. 로봇 저널리즘은 속보성이 강조되거나 단순 반복적인 작업이 요구되는 기사의 경우에 적합하다. 그 예로써, 야구경기의 결과 데이터를 통해 야구기사를 생성해내는 로봇 저널리즘에 대한 연구가 진행 중이다. 하지만 생성된 기사들이 경기의 박스스코어 및 승패에 관한 결과에 치중되어 있는 모습이 대부분이다[2].

본 논문에서는 의사결정트리(Decision Tree)기법을 사용하여, 경기 결과의 대략적인 흐름을 파악한 뒤, 기사 작성에 필요한 요소들을 자동으로 추출해 내는 기법을 소개한다. 박스스코어에 의해 단순히 승패 결과만을 이야기 하는데 그치지 않고, 기존 야구기사들의 경기흐름양상을 토대로 경기결과 데이터를 학습한 의사결정트리(Decision Tree)를 이용해 예측을 함으로써 기사의 전체적인 흐름양상을 잡는 문장을 도출하게 한다. 의사결정트리(Decision Tree)는 데이터 군집의 순수도(Purity) 및 불순도(Impurity)를 체크하며 클래스를 결정하는데, 여기서 데이터들의 군집(Cluster)은 야구경기결과 데이터가 되고, 순수도(Purity) 및 불순도(Impurity)는 흐름양상의 분포 정도를 가려내는 지표가 되는 동시에 클래스(Class)의 결정을 하게 된다. 그리고 클래스(Class)의 결정은 곧 야구 경기흐름양상을 판단하도록 하는 것이다. 이 결과, 기사의 전체적인 흐름양상을 잡는 문장을 도출하기 용이해지며, 기존의 야구 로봇저널리즘의 기사보다 다양한 문장을 사용할 수 있게 된다는데 있어 활용가치가 높음을 보인다.

2. 야구 경기 데이터 기반 경기결과 흐름양상 추출

본 논문에서는 야구 경기흐름양상을 판단하기 위하여 데이터 마이닝(Data Mining) 기법 중 하나인 '의사결정트리(Decision Tree)'를 사용하였다. '의사결정트리(Decision Tree)'는 군집 데이터(Cluster Data)의 지도학습을 통하여 특정 데이터 셋(Data Set)의 여러 가지 성질을 분석하고, 군집 데이터(Cluster Data) 내의 유사한 성질을 가진 소그룹으로 분류해 내거나 예측하는데 사용한다. 종류는 출력변수로서 크게 연속형 데이터(Continuous Data)와 범주형 데이터(Categorical Data)로 나뉜다. 본 실험에서는 그 종류 하나인 'CART (Classification And Regression Tree, CART)'를 사용하였다. CART는 앙상블 방법의 일종이며 입력된 자료로부터 한 개 이상의 결정 트리를 생성한다. 초기 앙상블 방법인 Bagging (or Bootstrap aggregating) 결정 트리는 반복적으로 교체 과정을 수행하는 것과 함께 훈련 데이터를 재 샘플링하고, 합의 예측을 위한 트리를 선택하는 것으로 다수의 의사 결정 트리를 생성한다[3].

위 정의에서 트리의 학습을 위한 훈련 데이터(Training Data)는 야구경기결과의 박스스코어를 사용하였다. 표 1에서 볼 수 있듯이, 2600여개의 훈련 데이터에 각각 14개의 속성 컬럼(f_1, f_2, …, f_{14})을 준비하였다. 속성 컬럼(Attribute Column) $f_1 \sim f_{12}$에는 이닝별 점수의 차(원정팀의 이닝별 점수 - 홈팀의 이닝별 점수)를 차례로 갖게 하고, 속성 컬럼(Attribute Column) f_{13}에는 총 점수의 차

1) 교신 저자: 한연희 (한국기술교육대학교)

(원정팀의 총 점수 - 홈팀의 총 점수)를 갖게 하였다. 마지막 속성 컬럼(Attribute Column) f_{14}에는 4가지의 클래스(Class)를 갖도록 하였는데, 표 2에 자세히 기술하였다. 클래스(Class)가 의미하는 바는, 트리의 결정 값(Decision Value)을 의미하는 동시에, 해당 경기 승리 팀과 패배 팀 간의 경기양상을 뜻한다.

<표 1> 데이터 세트

#	f_1	f_2	…	f_{13}	f_{14}(Class)
0	1	-2	…	-1	3
…					
2599	-1	-2	…	-4	1

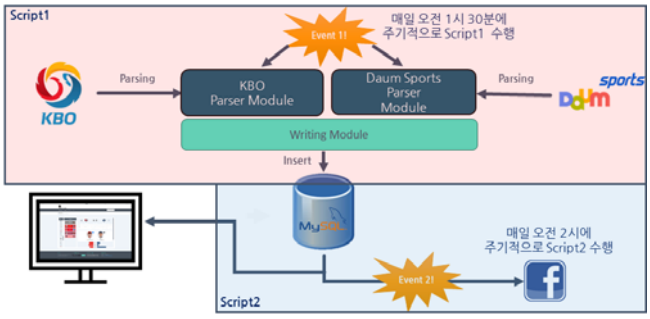
<표 2> 클래스별 결정 값

클래스(Class)	결정 값(Decision Value)
Class : 1	무난한 승리
Class : 2	접전 끝의 승리
Class : 3	역전 승리
Class : 4	완벽한 승리
Exception	무승부와 영봉승

본 실험에서 사용하는 의사결정트리(Decision Tree)는 범주형(Categorical) 출력변수를 사용하는 동시에, 최적의 가지(Branch) 분리를 선택하고자 지니지수를 사용하는데, 지니지수는 최소가 되는 트리 가지의 속성 분리를 선택한다. 실험에서 사용하는 지니지수의 정의는, $\sum_{f}^{n}(2\times Pf^2)$, 속성(Attribute)마다 클래스(Class)에 속하는 f번째 클래스(Class)의 가짓수에서 f번째 클래스(Class)의 전체수로 나눈 값들의 합이다. 이렇게 각 클래스마다의 불순도가 가장 낮은 지니지수 값을 사용함으로써, 수많은 트리 가지를 분리하는데 사용되고, 최적의 분리를 선택하게 된다. 그리하여 나온 클래스(Class)의 결정 값(Decision Value)이 야구 기사 전체의 흐름양상을 잡는 역할을 한다.

3. 시스템 구성 및 구현

본 논문에서는 그림 1과 같이 Python 언어를 사용하여 데이터를 추출하고 기사를 작성하는 스크립트1(Script 1)과 작성된 기사를 웹페이지와 페이스북(Facebook) 타임라인(Time Line)에 포스팅하는 스크립트2(Script 2)로 구성하였다. 스크립트 1은 경기가 있었던 날의 익일 오전 1시 30분에 운영체제 시그널 핸들러(Signal Handler)에 의해 실행되며, 스크립트 2는 같은 날 새벽 2시에 마찬가지로 시그널 핸들러(Signal Handler)에 의해 자동으로 실행된다. 경기결과의 문맥을 파악하기 위한 학습은 Python의 scikit-learn 모듈에서 제공하는 머신러닝 기법을 사용했다. 작성된 기사는 MySQL Database에 글을 저장하고, Facebook 봇 계정이 Database에서 글을 읽고 담벼락에 글을 남기는 순서로 동작한다. 이러한 모든 과정이 자동으로 동작하도록 Database를 갱신하는 스크립트와 Facebook에 글을 등록하는 두개의 스크립트는 일정한 시간간격을 두고 Signal호출을 통해 동작하게 된다[4].



(그림 1) 시스템 구조도

3-1. 데이터 추출

기사를 작성하기 전에 가장 먼저 필요한 작업은 데이터를 추출해오는 작업이다. 데이터 추출 대상은 한국프로야구 위원회(이하 KBO), Daum 스포츠이다. KBO와 Daum 스포츠 모두 대부분의 페이지들이 GET 방식의 접근을 허용하지만 Ajax 통신을 사용하는 Daum 스포츠는 조금 번거로운 과정을 거쳐야 한다.

1) KBO

BeautifulSoup는 HTML이나 XML파일에서 데이터를 추출하는 Python 언어용 라이브러리로 CSS selector문법을 이용하여 원하는 데이터를 추출할 수 가 있다.

파싱된 데이터는 Dictionary형태로 3개의 key를 갖는다. 각 key에 대한 설명은 표 3에 기술되어 있다.

<표 3> KBO에서 추출한 데이터의 구조

key	value
boxScore	팀별 이닝점수, 투수, 타자기록
rank	해당 경기의 순위
situation	이닝별 경기상황

2) Daum 스포츠

Daum 스포츠는 대부분의 태그가 클라이언트의 동작에 따라 Javascript 호출로 추가되는 동적인 데이터들이기 때문에 URL만을 가지고 html을 파싱하기는 불가능하다. 이를 위해 테스팅 프레임워크(Seleniuim)를 사용하여 가상의 브라우저를 띄우고 해당 브라우저를 통해 URL로 직접 이동해야만 Javascript가 적용된 html을 얻어올 수 있다. 파싱된 데이터는 Dictionary 형태로 12개의 key를 갖는다. 각 key에 대한 설명은 표 4에 자세히 기술 하였다.

<표 4> Daum스포츠에서 추출한 데이터의 구조

key	value
stadium	경기장 이름
accumulation	팀의 연승
batRecord	팀의 타수기록
criticalInning	승부처 이닝
criticalInningVOD_Url	팀의 승부처 이닝에 해당하는 동영상 URL
keyPlayer	키플레이어 역할을 한 타자
rank	팀의 순위
seasonStat	팀별 선발투수의 정보(이름,승수,패수,평균자책,볼넷허용률)
startingLineup	팀의 타자명단(포지션,이름,평균타율)
win_lose	팀의 전적(승,무,패)

3-2. 기사내용 설계

기사를 작성하기 위해서는 전형적인 기사의 흐름을 설계할 필요가 있다. 본 논문에서는 경기결과에 대해 객관적인 정보를 빠르게 전달함이 목표이므로 중립적인 시점에서 기사를 작성하기로 한다.

기사내용은 5가지의 요소로 나뉘며 다음 표 5에 구체적인 설명이 기술되어 있다.

<표 5> 기사의 5가지 구성 요소

구분	내용
Head	기사의 제목
Introduction	경기의 날짜, 팀명, 승리투수에 관한 내용, AWAY팀과 HOME팀의 점수
Main1	승부처 이닝에 있었던 타자들의 활약
Main2	경기 후 승리 팀과 패배 팀의 순위변동 여부와 연승의 내용
Ending	순위변동과 관련해서 팀의 게임차로 포스트시즌 진출여부

특히, Introduction과 Main2의 경우 의사결정트리의 결과로 분류된 경기 흐름에 따라 문장의 서술어구가 승리, 접전 승, 완승, 역전승을 구분하여 서술하게 된다.

3-3. 문장의 다양화

매번 똑같은 단어선택과 어투의 사용은 기사의 질을 떨어뜨리고 진부한 기사라는 인식을 갖게 한다. 이는 사용자에게 흥미를 유발시키지 못하고 효율성도 떨어뜨리게 된다. 이를 보완하기 위해 다양한 문구를 임의로 삽입하는 패턴으로 기사가 작성되도록 구현 했다.

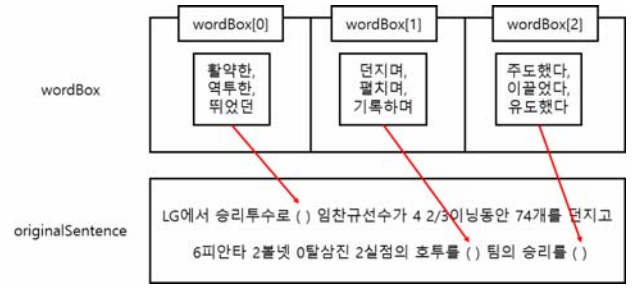
기사를 구성하는 5가지 요소는 각각의 문장뭉치를 가지고 있다. 문장뭉치는 List형태로 사용자가 기사에서 사용될 문장을 원소로 넣어주게 된다. Introduction의 첫 번째 문장의 서술 어구를 예로 들어 표 6에 기술하였다.

<표 6> Introduction의 첫 번째 문장에서 사용 가능한 서술 어구

의사결정 트리결과	문장뭉치
1	'승리하였다.', '승리를 거머쥐었다.', '승리를 가져갔다.', '승리를 챙겨갔다.', '1승을 챙겨갔다.'
2	'접전끝에 승리를 하였다.', '엎치락 뒤치락 끝에 승리를 가져갔다.', '팽팽한 싸움끝에 승리를 챙겨갔다.'
3	'역전에 성공했다.', '역전극을 만들어냈다.', '판을 뒤엎는데 성공했다.', '역전의 드라마를 성공시켰다.'
4	'완승을 하였다.', '큰 격차로 승리했다.', '완전히 게임을 압도했다.', '압도적인 승리을 보여주었다.'

조건문을 통해 퀄리티 스타트(Quality Start)와 같이 특정 조건을 만족시키는 경우에 해당하는 문장을 추가할 수도 있다. 이와 더불어 같은 의미에 대해서 여러 단어가 사용될 수 있도록 메서드를 구현했다. 이 함수의 원형은 String changeWithParam(String originalSentence, List... wordBox)와 같다.

이 메서드를 통해 그림 2와 같이 바꾸고자 하는 문장(originalSentence)의 괄호를 찾아 각 괄호의 순서대로 wordBox에서 임의로 하나를 골라 채워 넣는 방식으로 동작한다.



(그림 2) changeWithParam의 동작방식

4. 문제 해결 방안 및 구현 결과

본 장에서는 데이터 세트(Data Set)를 의사결정트리 생성에 제기 되었던 문제에 대한 해결방안과, 생성된 의사결정트리가 사용된 기사작성 시스템에 대하여 설명하고자 한다. 먼저 의사결정트리 생성 시에 눈여겨 보아야할 것은, 원하는 정보를 얻기 위해서 데이터의 특성을 명확히 알고, 데이터들을 어떻게 속성화 할 것인지 알아야 한다는 것이다. 우리는 초기에, 야구경기의 박스스코어 요소를 그

대로 속성화 시켰기에 의사결정트리의 결정클래스가 잘못 결정 되는 문제가 발생하였다. 원인을 분석한 결과, 생성되는 이진 트리의 각 노드가 가지는 속성(Attribute)정보 수가 많아 임계 값이 명확하지 않다는 데에 있었다. 임계값은, 각 노드가 어떤 결정 값에 포함될 수 있는지 순수도 정도를 나타내기 위한 척도이다. 그래서 속성의 수를 줄이며 명확성을 높이기 위해 속성간의 연관관계를 주었다. 그리하여, 14개의 속성을 갖춘 데이터 세트, 약 2600여개를 학습시켜 의사결정트리를 생성하였다. 결과적으로 야구경기의 양상 및 흐름을 분석해 낼 수 있었다. 의사결정트리는 해당 야구경기의 경기양상을 클래스 결정 값으로 도출해 내는데, 파이썬(Python)으로 만들어진 기사작성 알고리즘이 기사를 생성하고, 데이터베이스에 저장하는 역할을 한다. 또한 데이터베이스에 저장된 내용은 웹페이지와 페이스북(Facebook) 봇(Bot)계정에 포스팅 되는 시스템을 갖추도록 하였다.

(그림 3) 로컬 웹페이지 및 페이스북 타임라인

그림 3은 웹페이지와 페이스북 타임라인(Time Line)에 생성된 기사가 포스팅 되는 시스템을 구현한 것을 보여준다. 웹페이지는 로컬(Local)로 구성한 것인데, 정식 홈페이지는 현재 준비 중에 있다.

5. 결론

본 논문에서는 의사결정트리(Decision Tree)를 이용하여 파싱(Parsing)한 데이터를 기반으로 야구경기 전체에 대한 양상 및 흐름을 분석하였고, 이를 토대로 기사를 작성하는 시스템을 설계 하였다. 향후에는 의사결정트리(Decision Tree)를 이용하여 이닝을 분석하는 것 외에도 각 게임의 MVP (Most Valuable Player)를 검출해 내는, 세부적인 경기 내적인 요소에 관해서도 연구할 예정이다. 또한 대량의 데이터를 분석하고 패턴(Pattern)을 찾는 기계학습 알고리즘의 특성에 의해 경기 예측이 가능할 것이며 학계에서도 이에 대한 논문이 출판되어 있다[5]. 이에, 조금 더 많은 데이터를 수집하면서 기계학습에 대한 연구를 진행할 계획이다. 단순히 경기 뿐 만이 아닌 예측을 통하여 데이터로 알 수 없는 경기 외적인 요소가 기사에 등장할 수 있도록 좀 더 인간다운 로봇저널리즘을 완성 시킬 것이며 페이스 북 봇 계정을 통해 이용자의 피드백을 진행하여 수정 및 보완 할 계획이다.

참고문헌

[1] 김동환 (2015), “로봇 저널리즘: 알고리즘을 통한 스포츠 기사 자동 생성에 관한 연구” 한국언론학회, 제 59권 5호.

[2] [국내 저자 있음] 김대원. 미디어미래. (2015), “로봇 저널리즘을 넘보다”.

[3] Peter Harrington. (2012), "Machine Learning in Action".

[4] Swaroop C. H.(2004) “A Byte of Python”.

[5] 오윤학 (2014), “데이터마이닝을 활용한 한국프로야구 승패예측모형 수립에 관한 연구“ 한국연구재단, 제 40권 1호.