KIPS Trans. Softw. and Data Eng.
Vol.9, No.2 pp.61~68 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2020.9.2.61

Predicting Win-Loss of League of Legends Using Bidirectional LSTM Embedding

Cheolgi Kim† · Soowon Lee††

ABSTRACT

E-sports has grown steadily in recent years and has become a popular sport in the world. In this paper, we propose a win-loss prediction model of League of Legends at the start of the game. In League of Legends, the combination of a champion statistics of the team that is made through each player's selection affects the win-loss of the game. The proposed model is a deep learning model based on Bidirectional LSTM embedding which considers a combination of champion statistics for each team without any domain knowledge. Compared with other prediction models, the highest prediction accuracy of 58.07% was evaluated in the proposed model considering a combination of champion statistics for each team.

Keywords : League of Legends, Win-Loss Prediction, Machine Learning, Neural Network, LSTM

양방향 순환신경망 임베딩을 이용한 리그오브레전드 승패 예측

김 철 기† · 이 수 원††

요 약

e-sports는 최근 꾸준한 성장을 이루면서 세계적인 인기 스포츠 종목이 되었다. 본 논문에서는 e-sports의 대표적인 게임인 리그오브레전드 경기 시작 단계에서의 승패 예측 모델을 제안한다. 리그오브레전드에서는 챔피언이라고 불리는 게임 상의 유닛을 플레이어가 선택하여 플레이하게 되는데, 각 플레이어의 선택을 통하여 구성된 팀의 챔피언 능력치 조합은 승패에 영향을 미친다. 제안 모델은 별다른 도메인 지식 없이 플레이어 단위 챔피언 능력치를 팀 단위 챔피언 능력치로 임베딩한 Bidirectional LSTM 임베딩 기반 딥러닝 모델이다. 기존 분류 모델들과 비교 결과 팀 단위 챔피언 능력치 조합을 고려한 제안 모델에서 58.07%의 가장 높은 예측 정확도를 보였다.

키워드 : 리그오브레전드, 승패 예측, 기계 학습, 신경망, LSTM

1. 서 론

e-sports는 최근 전 세계적으로 대형 토너먼트 수가 급격하게 증가함에 따라 경제적 가치를 인정받았고, 미국과 중국을 비롯한 여러 국가에서 대규모 투자를 통하여 선도함에 따라 시청자와 이용자의 수 및 대중성 부분에서 꾸준한 성장을 이루었다. 또한 e-sports는 2018년 자카르타 팔렘방 아시안 게임의 시범종목으로 채택되는 등 단순한 게임을 넘어서 정식 스포츠 분야 중 하나로 인정받고 있다. e-sports의 장르로는 RPG(Role Playing Game), FPS(First Person Shooting), RTS(Real-Time Strategy), AOS(Aeon of Strife), 스포츠 등이 있다. 이 중 AOS 장르는 세계적인 규모의 대회를 거듭 개최하면서 e-sports 콘텐츠 시장과 게임 개발 시장의 발전에 기여하고 있으며 e-sports의 대표적인 장르로 자리매김 하고 있다. AOS는 MOBA(Multiplayers Online Battle Arena) 또는 ARTS(Action Real Time Strategy)로도 불리는데 명칭에서도 유추할 수 있듯이 여러 명의 플레이어가 적절한 전략을 수립하는 것이 경기 승리에 중요한 요소로 꼽힌다.

본 논문에서는 AOS 장르의 대표적인 게임인 리그오브레전드(League of Legends) 경기 시작 단계에서의 승패 예측 모델을 제안한다. 리그오브레전드는 5명씩 블루(blue) 팀과 레드(red) 팀으로 구성된 총 10명의 플레이어가 각각 다른 능력치(attack, defense, magic)를 갖는 챔피언(champion)이라고 불리는 게임에서의 가상 캐릭터를 선택하여 플레이한다. 승리의 조건은 상대팀 진영의 넥서스(nexus)라고 불리는 구조물을 먼저 파괴하는 것이다. 각 팀의 플레이어들은 경기 시작 단계에서 주어진 100여 가지의 챔피언 중 하나의 챔피언을 역할

※본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터 지원사업의 연구결과로 수행되었음(IITP-2019-2018-0-01419).
† 준 회 원 : 숭실대학교 융합소프트웨어학과 석사
†† 정 회 원 : 숭실대학교 소프트웨어학부 교수
Manuscript Received : August 23, 2019
First Revision : November 12, 2019
Accepted : December 4, 2019
* Corresponding Author : Soowon Lee(swlee@ssu.ac.kr)

과 조합을 고려하여 선택한다. 따라서 경기에 승리하기 위해서는 경기 시작 단계에서 어떤 플레이어가 어떤 챔피언을 선택할지, 팀의 챔피언 능력치 조합이 적절한지에 대하여 팀 단위 전략을 수립하는 것이 중요하다. 그러나 전문적인 도메인 지식 없이 각 플레이어가 100여 가지의 챔피언들을 대상으로 최적의 조합을 찾아내 전략을 수립하는 것은 매우 어려운 일이다.

본 논문에서는 리그오브레전드의 경기 시작 단계에서의 승패 예측을 위하여 Bidirectional LSTM(Long Short Term Memory) 임베딩 기반 딥러닝 모델을 제안한다. 제안 모델은 전문적인 도메인 지식 없이 플레이어들이 선택한 챔피언 능력치를 입력으로 팀 단위 챔피언 능력치 조합을 표현하는 임베딩 벡터를 추출하여 승부 예측에 유의미한 변수로 사용한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 소개하고, 3장에서는 데이터 및 모델을 제시한다. 4장에서는 실험의 결과를, 5장에서는 결론 및 향후 연구를 기술한다.

2. 관련 연구

2.1 스포츠 승부 예측

스포츠 분야에서의 승부 예측은 과거부터 선수단과 방송사뿐만 아니라 일반 시청자들에게도 꾸준한 관심의 대상이었으며 관련된 많은 연구들이 이루어졌다. 먼저 통계적인 확률 모형을 이용한 연구를 찾아 볼 수 있는데, 포아송 확률 모형을 이용하여 K리그 축구 경기를 대상으로 결과를 예측하는 연구[1, 2]가 있고 Bradley-Terry 모형을 이용하여 한국 프로야구 경기 결과에 영향을 미치는 요인을 분석하고 한국 시리즈 우승확률을 예측하는 연구[3]가 있다. 통계적인 확률 모형을 이용한 연구와 더불어 데이터 마이닝 기법을 이용한 연구로는 로지스틱회귀분석과 의사결정나무를 이용하여 한국 프로야구의 승패를 예측한 연구[4]와 의사결정나무를 이용하여 한국 프로농구 경기에서 승패에 영향을 끼치는 요인과 패턴을 찾고 결과를 예측한 연구[5]가 있다. 최근에는 스포츠 데이터의 수집이 용이하여지면서 딥러닝 기법을 활용한 연구가 활발해졌는데 관련연구로는 딥러닝 기법을 이용하여 한국 프로야구의 승패를 예측하는 모델을 제안하는 연구[6, 7]가 있다.

2.2 리그오브레전드 승부 예측

리그오브레전드의 승부 예측과 관련된 연구는 게임이 시작되고 난 후 in-game 데이터를 이용하여 경기의 승패를 예측하는 연구와 경기 시작 단계에서 승패를 예측하는 연구로 나뉜다. in-game 데이터를 이용하여 승패를 예측한 연구로 로지스틱 회귀분석 및 인공신경망을 이용하여 예측한 연구[8]와 순환신경망을 이용하여 예측한 연구[9]가 있고, 경기 시작단계에서의 승패 예측 연구로 decision tree를 이용하여 승패를 예측한 연구[10]와 인공신경망을 이용하여 예측한 연구[11]가 있다.

2.3 순환신경망 임베딩을 이용한 연구

순환신경망 임베딩은 시간 또는 순서에 종속적인 여러 데이터 차원을 하나의 벡터 차원으로 표현하고자 할 때 사용되는데 주로 자연어처리 문제에서 사용된다. 개체명을 추출하는 연구에서 문자단위 벡터를 워드 단위 벡터로 임베딩하기 위하여 Bidirectional LSTM을 사용한 연구[12, 13]가 있고, 형태소 분석을 위하여 encoder 부분을 문자 단위 Bidirectional LSTM로 구현한 연구[14]도 있다.

3. 데이터 및 모델

3.1 수집 데이터

본 연구에서는 경기 시작단계에서 파악할 수 있는 데이터로부터 승패 예측을 진행한다. 리그오브레전드의 개발사인 라이엇게임즈는 연구자가 쉽게 게임 데이터를 사용할 수 있도록 API(Application Programming Interface)를 제공하고 있다. 본 연구에서 사용되는 게임 데이터는 라이엇게임즈 개발자 사이트[15] API를 통하여 수집하였다. 추가적으로 챔피언 승률 데이터는 fow.kr/champs[16]에서 수집하였다. 분석을 위하여 수집한 데이터는 리그오브레전드 게임의 여러 패치 버전 중 8.24 패치 버전에서 플레이 된 데이터이고, 한국에서 플레이 되었으며 5:5 랭크 게임으로 '소환사의 협곡' 맵에서 진행된 4050개의 게임 데이터이다.

3.2 데이터 전처리

수집된 데이터로부터 추출한 플레이어 별 속성 변수는 Table 1에서 설명된다. Table 1에서의 속성 변수는 플레이어 별 속성 변수로 각 변수에서 첨자 i는 플레이어를 의미하는데, 1부터 5까지는 블루 팀 플레이어를 6부터 10까지는 레드 팀 플레이어를 의미한다. 본 연구에서 다루는 게임에서는 라인(lane)과 롤(role)이라고 불리는 플레이어가 선택할 수 있는 게임에서의 역할이 존재하는데, 라인은 게임 초반 각 챔피언들이 전투를 수행하는 위치와 역할을 의미하고, 롤은 각 위치에서에서의 세부역할을 의미한다. API를 통하여 수집된 데이터는 먼저 라인에 따라 top, jungle, middle, bottom 순으로 정렬하였고 겹치는 라인이 있는 경우 롤에 따라 duo_support, duo, duo_carry, solo, none 순으로 정렬하였다.

Table 2는 팀 별 속성 변수에 대한 설명이다. Table 2의 변수에서 첨자 j는 팀을 의미하는데, 1과 2는 각각 블루 팀과 레드 팀을 의미한다.

Table 1과 2의 각 변수에 대하여 유형에 따라 다른 전처리 방식을 사용하였다. categorical type 변수의 경우 one-hot encoding 방식을 사용하였고, numerical type 변수와 vector type 변수에서는 원 데이터 값 간의 관계를 그대로 유지한 채로 측정단위에 종속된 문제점을 방지하기 위한 방법으로 min-max normalization을 사용하였다. vector type 변수의 경우 각 성분 별로 정규화하였다.

3.3 변수 선택

고차원의 categorical type, vector type 변수의 경우 데이터의 차원이 증가할수록 해당 공간의 부피가 기하급수적으

Table 1. Player-specific Variable Description

Variable	Type	Description
$X^{WinLoss}_{i(i=1,...,10)}$	Categorical	Result of previous 5 games for player i
$X^{Tier}_{i(i=1,...,10)}$	Categorical	Tier of player i
$X^{Skill}_{i(i=1,...,10)}$	Categorical	Skill of player i
$X^{ChampStats}_{i(i=1,...,10)}$	Vector	The attack, defense and magic degrees of the champion chosen by the player i
$X^{WinRate}_{i(i=1,...,10)}$	Numerical	Overall win percentage of champion chosen by the player i
$X^{ChampRate}_{i(i=1,...,10)}$	Numerical	The selection ratio for the champion in the previous 15 games of player i
$X^{RoleRate}_{i(i=1,...,10)}$	Numerical	The selection ratio for the role in the previous 15 games of player i

Table 2. Team-specific Variable Description

Variable	Type	Description
$X^{TeamMastery}_{j(j=1,2)}$	Numerical	Number of mastery level players below the threshold of team j

로 증가하여 데이터의 밀도가 희박하여지는 문제를 야기할 수 있기 때문에 logistic regression 모델에서의 validation 정확도를 이용하여 변수의 사용여부를 판단하였다. Table 3은 numerical type 변수를 모두 사용한 조합에 특정 categorical type, vector type 변수를 추가하였을 때의 정확도를 보여준다.

Table 3을 통하여 차원 수가 증가함에 따라 전반적으로 validation 정확도가 감소하는 것을 확인할 수 있었고 categorical type 변수와 vector type 변수를 제외한 numerical type 변수만을 사용하였을 때 정확도가 더 높음을 확인할 수 있었다. 따라서 본 연구에서는 수집한 변수 중 numerical type 변수만을 선택하여 모델의 입력으로 사용하였다.

이후 numerical type 변수에서는 데이터의 분산은 최대한 보존하면서 주성분을 기저로 저차원 공간으로 변환하여 데이터의 정보를 효율적으로 추출하는 기법인 PCA(Principal Component Analysis)기법을 사용하여 변수를 추출하는 방법을 사용하였다.

Table 3. Validation Accuracy When Adding Categorical and Vector Type Variables

Variables used	Number of dimensions	Validation accuracy(%)
Only numerical variables	32	57.16
Numerical variables+$X^{ChampStats}_{i(i=1,...,10)}$	62	55.95
Numerical variables+$X^{Tier}_{i(i=1,...,10)}$	112	56.25
Numerical variables+$X^{Skill}_{i(i=1,...,10)}$	212	55.01
Numerical variables+$X^{WinLoss}_{i(i=1,...,10)}$	352	53.33

3.4 예측 모델

본 연구에서는 전문적인 도메인 지식 없이 플레이어들이 선택한 챔피언 능력치를 입력으로 팀 단위 챔피언 능력치 조합을 고려할 수 있는 Bidirectional LSTM 임베딩 기반 딥러닝 모델을 예측 모델로 제안한다. 또한 비교를 위하여 MLP (Multi Layer Perceptron) 임베딩, CNN(Convolution Neural Network) 임베딩 기반의 딥러닝 모델을 추가로 구현하였다.

1) 제안 모델

본 연구의 제안 모델은 시간 또는 순서에 종속적인 데이터 특성을 효율적으로 추출하는 Bidirectional LSTM을 이용하여 도메인 지식 없이 팀 단위 능력치 조합을 표현하는 vector를 추출하고 승패 예측 변수로 사용하는 모델이다.

a) LSTM 기반 팀 단위 챔피언 능력치 조합 임베딩

플레이어는 경기 시작단계에서 챔피언을 선택할 때, 이전에 선택된 챔피언만 고려하여 선택하는 것이 아니라 이후에 선택될 챔피언 또한 고려하여 선택한다. 단방향 LSTM의 경우 forward 정보만을 고려하기 때문에 현재 단계의 입력과 이전 단계의 정보만 이용할 수 있어 이후에 선택될 챔피언의 영향력이 고려되지 못하는 한계가 있다. 이러한 한계를 극복하기 위하여 본 연구에서는 forward와 backward 정보를 모두 고려할 수 있는 Bidirectional LSTM 모델을 임베딩 모델로 사용하였다.

본 연구에서는 해당 임베딩 기법을 이용하여 팀 단위 챔피언 조합을 하나의 벡터로 표현하기 위하여 각 팀에서 선택한 5개의 챔피언 능력치(attack, defense, magic)를 Bidirectional LSTM의 입력으로 사용하였다. 챔피언 능력치는 $X^{ChampStats}_{i(i=1,...,10)}$ 변수이며 각 수치 별로 1부터 10까지의 값을 갖는다. 표현된 임베딩 정보는 순서대로 선택된 챔피언들 간의 상호 의존성을 보다 효율적으로 고려하여 승패 예측에 도움이 되는 정보로 활용할 수 있다. 팀 단위 챔피언 능력치 조합 임베딩의 구조는 Fig. 1과 같다. Fig. 1에서와 같이 각 Bidirectional LSTM 모델의 양 끝 출력을 concatenation하여 양 팀의 챔피언 조합을 표현하는 벡터로 사용한다.

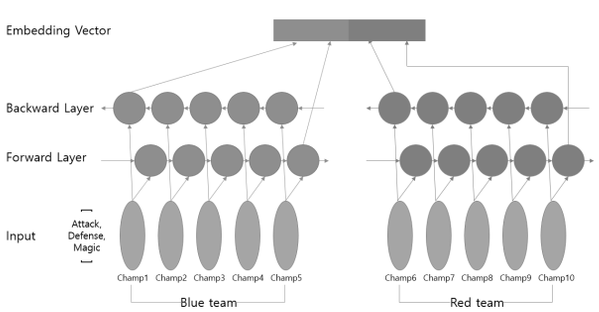

Fig. 1. Structure of the Bidirectional LSTM Embedding Model

b) 제안 모델의 구조도

본 연구에서 제안하는 모델의 구조도는 Fig. 2와 같다. 제안 모델의 구조도를 살펴보면 플레이어 별 챔피언 능력치를 입력으로 사용하여 Bidirectional LSTM 모델로 추출한 임

베딩 벡터와 numerical type 속성 변수를 입력으로 PCA를 통하여 추출한 변수가 신경망의 입력으로 들어간다. 이후 신경망의 hidden layer를 거쳐 추상화된 2차원의 출력 값이 softmax 함수를 통하여 승, 패 확률로 변환된다.

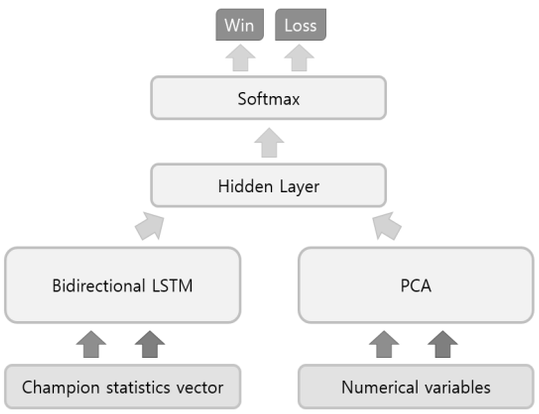

Fig. 2. Architecture of the Proposal Model

2) 임베딩 기반 딥러닝 모델

본 연구에서는 제안 모델과의 비교를 위하여 선택된 챔피언 순서에 상관없이 팀 단위 능력치 조합 정보를 추출할 수 있는 MLP 임베딩 기반 딥러닝 모델과 선택된 챔피언 간의 시너지를 고려하여 팀 단위 능력치 조합을 추출할 수 있는 CNN 임베딩 기반 딥러닝 모델을 추가로 구현하였다.

a) MLP 기반 팀 단위 챔피언 능력치 조합 임베딩

선택된 챔피언 순서에 상관없이 팀 단위 능력치 조합 정보를 추출하기 위한 방법으로 MLP 기반 임베딩 모델을 구현할 수 있다. MLP 기반 임베딩 모델의 구조는 Fig. 3과 같다.

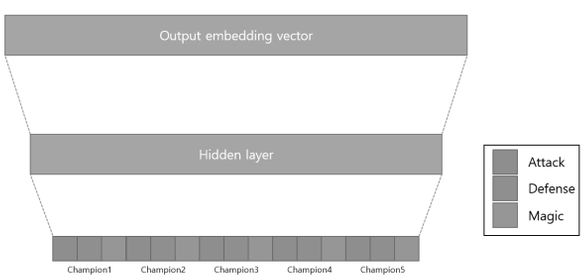

Fig. 3. Structure of the MLP Embedding Model

해당 임베딩 모델에서의 입력으로 제안 모델과 동일하게 각 팀에서 선택한 5개의 챔피언 능력치(attack, defense, magic)가 사용된다. 해당 임베딩 모델은 입력으로 들어온 챔피언 능력치를 hidden layer를 거쳐 추상화시키고 팀의 능력치 조합을 표현하는 vector로 출력한다. 각 팀의 능력치 조합으로 출력된 vector를 concatenation하여 승패를 예측하는 변수로 사용한다.

b) CNN 기반 팀 단위 챔피언 능력치 조합 임베딩

선택된 챔피언 간의 시너지를 고려하여 팀 단위 능력치 조합을 추출하기 위한 방법으로 CNN 기반 임베딩 모델을 구현할 수 있다. CNN 기반 임베딩 모델의 구조는 Fig. 4와 같다.

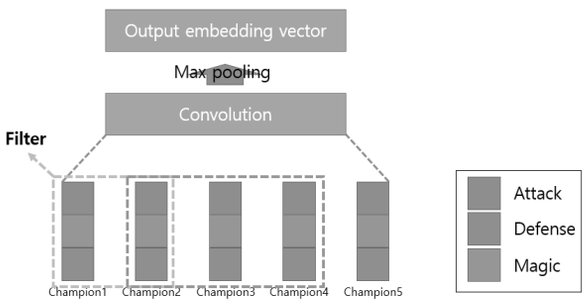

Fig. 4. Structure of the CNN Embedding Model

해당 임베딩 모델에서도 입력으로 제안 모델과 동일하게 각 팀에서 선택한 5개의 챔피언 능력치(attack, defense, magic)가 사용된다. CNN 기반 임베딩 모델에서의 각 filter는 convolution 연산을 통하여 챔피언 2명의 시너지, 3명의 시너지 등 챔피언 간의 시너지를 고려한다. 해당 임베딩 또한 각 팀의 능력치 조합으로 출력된 vector를 concatenation하여 승패를 예측하는 변수로 사용한다.

4. 실험 결과

본 연구에서는 제안 모델과 logistic regression, SVM (Support vector Machine), 단순 신경망, MLP 임베딩 기반 딥러닝 모델, CNN 임베딩 기반 딥러닝 모델과의 예측 정확도를 비교하였다. 모델을 구현하는데 있어서 logistic regression과 SVM, PCA 모델은 Python의 Scikit-Learn 패키지를 이용하여 구현하였고, MLP, CNN, Bidirectional LSTM과 신경망 모델은 Tensorflow 패키지를 이용하여 구현하였다.

training set과 validation set, test set은 6:2:2 비율로 나누어 사용하였으며, 승패 예측 모델의 성능 평가를 위하여 5-fold cross validation을 사용하였다. 각 예측모델에서 grid search를 통하여 5-fold의 validation 정확도가 최대가 되는 parameter set을 찾고 그 때의 test 정확도를 해당 모델의 최종 정확도로 평가한다.

4.1 Logistic Regression 모델의 실험 결과

Table 4는 3.2절에서 언급한 'Number of mastery level players below the threshold of team j에 대한 변수 $X_{j(j=1,2)}^{TeamMastery}$를 추출할 때 임계치를 변경하여가며 validation 정확도를 비교한 표이다.

Table 4. Validation Accuracy by Threshold in the Logistic Regression Model

Threshold	2	3	4	5
Validation accuracy(%)	56.42	56.84	57.16	57.04

API를 통하여 수집한 데이터에서 구분할 수 있는 마스터리 레벨은 1부터 4까지였고 임계치를 4로 하였을 때 가장 높은 validation 정확도를 확인할 수 있었다.

본 연구에서는 수집된 데이터에서 직관적으로 추출할 수 있는 변수 이외에 도메인 지식을 토대로 팀 단위 챔피언 능력치(attack, defense, magic)의 분산이 승패에 영향을 끼칠 것이라는 가정을 바탕으로 '팀 j의 각 플레이어가 선택한 챔피언능력치를 element-wise sum 한 후 분산을 구한 값'에 해당하는 팀 단위 numerical type 변수 $X_{j(j=1,2)}^{VarTeamStats}$를 추가적으로 추출하였다. 해당 모델에서 변수를 추가하여 validation 정확도를 측정하였을 때 57.21%로 사용하지 않았을 때보다 더 높은 정확도를 보였다.

PCA 또한 validation 정확도를 이용하여 최적의 차원 수를 결정하였으며, 차원 수를 30으로 축소하였을 때 validation 정확도가 57.43%로 가장 높았다. validation 정확도를 토대로 선택한 parameter로 logistic regression 모델을 학습하고 test set에서 예측하였을 때의 정확도는 57.06%였다.

4.2 SVM(Support Vector Machine) 모델의 실험 결과

SVM 모델에서 또한 validation 정확도를 기반으로 최적의 kernel과 C value, PCA 차원 수를 결정하였다. kernel은 linear kernel을 사용하였을 때, C value는 157, PCA 차원은 28차원에서 57.83%로 가장 높은 validation 정확도 보였다. validation 정확도를 토대로 선택한 parameter setting에서 SVM 모델을 학습하고 test set에서 예측하였을 때의 정확도는 57.09%였다.

4.3 단순 신경망 모델의 실험 결과

실험에 앞서 속성 변수에 내재된 정보를 효과적으로 추출하는 신경망 모델의 특성을 고려하여 변수 선택과정을 다시 한 번 확인하였다. Table 5는 hidden layer node 수를 1024로 충분히 확보한 신경망 모델에서 numerical type 변수를 모두 사용한 조합에 특정 categorical type, vector type 변수를 추가하였을 때의 정확도를 보여준다.

Table 5. Validation Accuracy in Neural Network Model When Adding Categorical and Vector Type Variables

Variables used	Validation accuracy(%)
Only numerical variables	58.49
Numerical variables+$X_{i(i=1,...,10)}^{ChampStats}$	58.02
Numerical variables+$X_{i(i=1,...,10)}^{Tier}$	56.64
Numerical variables+$X_{i(i=1,...,10)}^{Skill}$	56.10
Numerical variables+$X_{i(i=1,...,10)}^{WinLoss}$	54.07

기존의 변수 선택 결과와 동일하게 numerical type 변수만 사용하였을 때 가장 높은 정확도를 보였다. 또한 모든 변수를 사용하였을 때 54.07%의 validation 정확도를 보였다. 따라서 해당 모델과 신경망을 기반으로 하는 모델에서 numerical type 변수만 사용한다.

해당 모델은 팀 단위 챔피언 능력치 조합 임베딩을 포함하지 않는 단순한 모델이다. Table 6, 7, 8은 각 parameter 변화에 따른 validation 정확도를 보여준다.

Table 6. Accuracy of the Neural Network Model According to the Number of Hidden Layer Nodes

Number of hidden layer nodes	4	8	16	32	64
Validation accuracy(%)	58.25	58.49	58.99	58.47	58.59

Table 7. Accuracy of the Neural Network Model According to Hidden Layer Drop out Ratio

Hidden layer drop out	0.6	0.7	0.8	0.9	1.0
Validation accuracy(%)	58.81	58.91	58.99	58.99	58.99

Table 8. Accuracy of the Neural Networkmodel According to the Number of PCA Dimensions

Number of PCA dimensions	27	28	29	30	31
Validation accuracy(%)	58.47	58.42	58.99	58.91	58.79

validation 정확도를 기준으로 실험 결과, 해당 모델의 parameter를 learning rate 0.001, hidden layer nodes 16, drop out ratio 0.8, PCA 차원 수 29로 하였을 때 58.99%로 가장 높은 validation 정확도를 보였다. 이때 drop out ratio는 0.0이 학습단계에서 모든 weight를 사용하지 않는 것이고 1.0이 모든 weight를 사용하는 것을 의미한다. 이는 다른 모델에서도 동일하게 적용된다. hidden layer drop out ratio가 0.8, 0.9, 1.0일 때 같은 validation 정확도를 보이는데, 각 parameter에서의 fold 별 정확도의 표준편차를 고려하여 표준편차가 가장 작은 parameter를 선정하였다. 표준편차는 drop out ratio가 0.8일 때 1.42, 0.9일 때 1.59, 1.0일 때 1.77로 0.8일 때 가장 작은 수치를 보였다. validation 정확도를 토대로 선택한 parameter setting에서 신경망 모델을 학습하고 test set에서 예측하였을 때의 정확도는 57.23%였다.

4.4 MLP 임베딩 기반 딥러닝 모델의 실험 결과

해당 모델은 MLP 임베딩 기법을 이용하여 팀 단위 능력치 조합을 고려한 모델이다. Table 9, 10, 11, 12, 13, 14는 각 parameter 변화에 따른 validation 정확도를 보여준다.

Table 9. Accuracy of the MLP Embedding Model According to the Number of Hidden Layer Nodes

Number of hidden layer nodes	64	128	256	512	1024
Validation accuracy(%)	59.14	59.53	59.88	59.43	59.41

Table 10. Accuracy of the MLP Embedding Model According to Hidden Layer Drop Out Ratio

Hidden layer drop out	0.6	0.7	0.8	0.9	1.0
Validation accuracy(%)	59.68	59.68	59.60	59.83	59.88

Table 11. Accuracy of the MLP Embedding Model According to MLP Drop Out Ratio

MLP drop out	0.1	0.2	0.3	0.4	0.5
Validation accuracy(%)	59.48	59.88	59.51	59.21	59.06

Table 12. Accuracy of the MLP Embedding Model According to the Number of MLP Nodes

Number of MLP nodes	8	16	32	64	128
Validation accuracy(%)	59.16	59.26	59.88	59.28	59.21

Table 13. Accuracy of the MLP Embedding Model According to the Number of MLP Output Nodes

Number of MLP output nodes	128	256	512	1024	2048
Validation accuracy(%)	59.21	59.48	59.88	59.43	59.19

Table 14. Accuracy of the MLP Embedding Model According to the Number of PCA Dimensions

Number of PCA dimensions	30	31	32	33	34
Validation accuracy(%)	59.80	59.70	59.75	59.88	59.88

validation 정확도를 기준으로 실험 결과, 해당 모델의 parameter를 learning rate 0.001, hidden layer nodes 256, hidden layer drop out ratio 1.0, MLP drop out ratio 0.2, MLP nodes 32, MLP output nodes 512, PCA 차원 수 34로 하였을 때 59.88%로 가장 높은 validation 정확도를 보였다. validation 정확도를 토대로 선택한 parameter setting에서 해당 모델을 학습하고 test set에서 예측하였을 때의 정확도는 58.05%였다.

4.5 CNN 임베딩 기반 딥러닝 모델의 실험 결과

해당 모델은 CNN 임베딩 기법을 이용하여 팀 단위 능력치 조합을 고려한 모델이다. Table 15, 16, 17, 18, 19는 각 parameter 변화에 따른 validation 정확도를 보여준다.

validation 정확도를 기준으로 실험 결과, 해당 모델의 parameter를 learning rate 0.001, hidden layer nodes 64, hidden layer drop out ratio 0.9, CNN drop out ratio 0.4, CNN filter 수 8, PCA 차원 수 33로 하였을 때 59.48%로 가장 높은 validation 정확도를 보였다. PCA 차원 수가 33, 34일 때 같은 validation 정확도를 보이는데, 각 parameter에서의 fold 별 정확도 표준편차를 고려하여 표준편차가 더 작은 parameter를 선정하였다. 표준편차는 PCA 차원 수가 33일 때 0.71, 34일 때 0.82로 33일 때 더 작은 수치를 보였다. validation 정확도를 토대로 선택한 parameter setting에서 해당 모델을 학습하고 test set에서 예측하였을 때의 정확도는 57.58%였다.

Table 15. Accuracy of the CNN Embedding Model According to the Number of Hidden Layer Nodes

Number of hidden layer nodes	16	32	64	128	256
Validation accuracy(%)	58.84	59.14	59.48	58.96	58.84

Table 16. Accuracy of the CNN Embedding Model According to Hidden Layer Drop Out Ratio

Hidden layer drop out	0.6	0.7	0.8	0.9	1.0
Validation accuracy(%)	59.33	59.43	59.36	59.48	59.26

Table 17. Accuracy of the CNN Embedding Model According to CNN drop Out Ratio

CNN drop out	0.2	0.3	0.4	0.5	0.6
Validation accuracy(%)	58.99	58.40	59.48	59.26	59.09

Table 18. Accuracy of the CNN Embedding Model According to the Number of CNN Filters

Number of CNN filters	2	4	8	16	32
Validation accuracy(%)	58.84	58.40	59.48	58.59	58.30

Table 19. Accuracy of the CNN Embedding Model According to the Number of PCA Dimensions

Number of PCA dimensions	30	31	32	33	34
Validation accuracy(%)	59.01	59.28	59.21	59.48	59.48

4.6 제안 모델의 실험 결과

본 논문에서 제안하는 Bidirectional LSTM 임베딩 기반 딥러닝 모델에서의 결과는 다음과 같으며, Table 20, 21, 22, 23, 24는 각 parameter 변화에 따른 validation 정확도를 보여준다.

Table 20. Accuracy of the Proposed model According to the Number of Hidden Layer Nodes

Number of hidden layer nodes	16	32	64	128	256
Validation accuracy(%)	58.67	58.86	59.09	58.89	58.40

Table 21. Accuracy of the Proposed Model According to Hidden Layer Drop Out Ratio

Hidden layer drop out	0.6	0.7	0.8	0.9	1.0
Validation accuracy(%)	58.81	58.94	59.09	58.94	58.79

Table 22. Accuracy of the Proposed Model According to LSTM Drop Out Ratio

LSTM drop out	0.6	0.7	0.8	0.9	1.0
Validation accuracy(%)	58.67	58.74	58.89	58.86	59.09

Table 23. Accuracy of the Proposed Model According to the Number of LSTM Units

Number of LSTM units	4	8	16	32	64
Validation accuracy(%)	58.59	58.74	59.09	58.74	58.74

Table 24. Accuracy of the Proposed Model According to the Number of PCA Dimensions

Number of PCA dimensions	30	31	32	33	34
Validation accuracy(%)	58.74	58.72	58.57	59.09	59.01

validation 정확도를 기준으로 실험 결과, 해당 모델의 parameter를 learning rate 0.001, hidden layer nodes 64, hidden layer drop out ratio 0.8, LSTM drop out ratio 1.0, LSTM units 16, PCA 차원 수 33로 하였을 때 59.09%로 가장 높은 validation 정확도를 보였다. validation 정확도를 토대로 선택한 parameter setting에서 제안 모델을 학습하고 test set에서 예측하였을 때의 정확도는 58.07%였다.

4.7 예측 모델 별 결과 비교

Table 25는 각 모델 별 validation과 test 정확도를 보여준다. 예측 모델 별로 validation set을 이용하여 최적의 parameter를 찾고 test set에 대한 예측을 수행한 결과 제안 모델인 Bidirectional LSTM 임베딩 기반 딥러닝 모델에서 58.07%의 정확도로 가장 높은 예측 정확도를 보였다.

Neumann(2015)의 연구[10]에서는 decision tree를 이용하여 58%의 정확도를 보였다. 학습에 사용한 데이터가 다르기 때문에 직접적인 비교는 불가능 하지만 간접적으로 비교하였을 때 본 연구의 제안 모델에서의 정확도가 더 높은 것을 확인할 수 있다. Kim(2019)의 연구[11]와도 비교해 볼 수 있는데 Kim의 제안 모델 정확도는 60.25%로 본 논문의 제안 모델보다 높다. 이는 성능 평가방식의 차이 때문인데, Kim의 제안 모델을 본 연구에서의 성능 평가방식으로 평가했을 때 57.33%의 test 정확도로 평가되었고 Kim의 모델이 특정 데이터에 편향되어 있다고 해석할 수 있다. Kim의 연구는 본 연구와 동일한 데이터 및 자질과 fine tuning된 MLP 모델을 이용하여 경기 시작단계에서 승패를 예측한다. 동일한 성능 평가 방식으로 측정했을 때 본 연구의 제안 모델에서 58.07%로 더 높은 정확도를 보이는데, 이를 통하여 신경망 모델을 깊게 쌓아 fine tuning하는 것보다 제안 모델의 임베딩 기법을 이용하여 팀 단위 능력치 조합 변수를 고려하는 것이 성능 향상에 더 많은 도움을 주는 것을 확인할 수 있다.

Table 25. Accuracy Comparison by Prediction Model

Model	Validation accuracy (%)	Test accuracy (%)
Prediction as "Loss" for all blue teams	51.41	
Logistic regression	57.43	57.06
SVM	57.83	57.09
Neural network (without embedding)	58.99	57.23
Neural network (with MLP embedding)	59.88	58.05
Neural network (with CNN embedding)	59.48	57.58
Neural network (with Bidirectional LSTM embedding)	59.09	58.07

5. 결론 및 향후 연구

본 논문에서는 e-sports의 대표적인 게임인 리그오브레전드의 게임 데이터와 양방향 순환신경망 임베딩을 이용한 경기 시작 단계에서의 승패 예측 모델을 제안하였다. 제안 모델은 Bidirectional LSTM 임베딩 기반 딥러닝 모델이며, 팀 단위 챔피언 능력치 조합을 고려한 제안 모델과 팀 단위 챔피언 능력치 조합을 고려하지 않은 모델들을 비교하였을 때 제안 모델에서 58.07%로 가장 높은 예측 정확도를 보였다. 이것으로 별도의 도메인 지식 없이 Bidirectional LSTM 임베딩을 통하여 자동으로 추출된 팀 단위 챔피언 능력치 조합 정보가 경기 시작단계에서의 승부 예측에 도움이 되는 것을 확인할 수 있었다.

사용자는 제안된 모델을 이용하여 경기 시작 단계에서의 승패 확률을 토대로 챔피언 조합과 관련된 팀 단위 전략 수립에 도움을 받을 수 있다. 또한 본 논문에서 제안한 임베딩 방법을 활용하여 다른 팀 단위 스포츠 승패 예측 연구에 적용한다면 더 나은 정확도를 기대할 수 있다.

경기 승패에 영향을 미치는 요인으로 챔피언 정보 뿐 아니라 각 플레이어들의 역량 및 컨디션 또한 고려해 볼 수 있다. 본 연구에서는 각 플레이어들이 선택한 챔피언에 대한 정보와 조합에 중점을 두고 승패 예측 모델을 제안하였는데 위에서 언급한 플레이어의 역량과 컨디션을 효과적으로 고려하지 못하는 한계가 존재한다. 개선 방안으로 시계열 데이터를 기반으로 플레이어의 역량과 컨디션을 고려하는 방법을 생각해 볼 수 있다. 향후 연구에서는 각 플레이어들의 역량 및 컨디션을 고려할 수 있는 방법의 제안과 제안 방법이 예측 정확도 향상에 도움이 되는지 비교 분석이 필요하다.

References

[1] Hyun Seong and Woojin Chang, "Forecasting the Results of Soccer Matches Using Poisson Model," *IE interfaces*, Vol.20, No.2, pp.133-141, 2007.

[2] Hosang Kim, Jonghwan Won, and Dae-Ki Kan, "Modelling and Predicting 2010-2012 K-League Association Football Matches Using Poisson Model Analysis," in *Proceedings of the IEEK Conference*, pp.1572-1575, 2014.
[3] Young Gun Choi and Hyoung Moon Kim, "A Statistical Study on Korean Baseball League Games," *The Korean Journal of Applied Statistics*, Vol.24, No.5, pp.915-930, 2011.
[4] Cha Yong Kim, "A Win, Loss Predicting Model by Analyzing Professional Baseball Game," *Journal of Sport and Leisure Studies*, Vol.16, pp.807-819, 2001.
[5] Sea-Joong Kim, Chong-Kwan Heo, and Jae-Hyun Do, "Analysis on the Korean basketball league using data mining," *Journal of The Korean Society of Sports Science*, Vol.19, No.2, pp.1413-1420, 2010.
[6] Jong-Hun Kim, Kyung-Tae Kim, and Jong-Ki Han, "Big Data Analysis based on Deep Learning for Baseball Game Data," in *Proceedings of the Korean Institute of Communication Sciences Conference*, pp.262-265, 2015.
[7] Yeong-Jin Seo, Hyung-Woo Moon, and Yong-Tae Woo, "A Win/Lose prediction model of Korean professional baseball using machine learning technique," *Journal of the Korea Society of Computer and Information*, Vol.24, No.2, pp.17-24, 2019.
[8] Ji-Min Ku and Kyeonah Yu, "Design and Application of a Winning Forecast Model of the AOS Genre Game," *KIISE Transactions on Computing Practices*, Vol.23, No.1, pp. 37-44, 2017.
[9] Antonio Luis Cardoso Silva, Gisele Lobo Pappa, and Luiz Chaimowicz, "Continuous Outcome Prediction of League of Legends Competitive Matches Using Recurrent Neural Networks," 2018. http://www.sbgames.org/sbgames2018/files/papers/ComputacaoShort/188226.pdf.
[10] Alexander Neumann, "Developing a Model to Predict Match Outcomes in League of Legends," Barrett, The Honors College Thesis, Arizona State University, 2015.
[11] Cheolgi Kim and Soowon Lee, "Predicting Win-Loss of League of Legends at the Start of the Game using Machine Learning Techniques," in *Proceedings of the ISSAT DSBFI*. pp.28-30, 2019.
[12] Chi-Yun Song, Sung-Min Yang, and Sangwoo Kang, "Named-entity Recognizer Based on Bidirectional LSTM CRFs Using Improved Word Embedding Models And Dictionaries," in *Proceedings of the Korean Information Science Society Conference*, pp.699-701. 2017.
[13] Seung-Hoon Na, and Jinwoo Min, "Character-based LSTM CRFs for named entity recognition," in *Proceedings of Korean Institute of Information Scientists and Engineers*, pp.729-731, 2016.
[14] Jianri Li, EuiHyeon Lee, and Jong-Hyeok Lee, "Sequence-to-sequence based Morphological Analysis and Part-Of-Speech Tagging for Korean Language with Convolutional Features," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.44, No.1, pp.57-62, 2017.
[15] League of Legends Developers Web Site [Online], Available: https://developer.riotgames.com/.
[16] 리그오브레전드 챔피언 통계 [Internet], http://fow.kr/champs/.

김 철 기
https://orcid.org/0000-0002-1430-3833
e-mail : kimchelgi03@gmail.com
2018년 강원대학교 컴퓨터정보통신공학과 (학사)
2020년 숭실대학교 융합소프트웨어학과 (석사)
관심분야 : 인공지능, 머신러닝, 텍스트마이닝

이 수 원
https://orcid.org/0000-0001-5863-1188
e-mail : swlee@ssu.ac.kr
1982년 서울대학교 계산통계학과(학사)
1984년 한국과학기술원 전산학과(석사)
1994년 미국 University of Southern California 전산학과(박사)
2003년 ~ 2004년 한국정보과학회 인공지능연구회 운영위원장
2008년 한국정보과학회 소프트웨어 및 응용 논문지 편집위원장
2008년 ~ 2012년 한국BI데이터마이닝학회 부회장
1995년 ~ 현 재 숭실대학교 소프트웨어학부 교수
관심분야 : 데이터사이언스, 인공지능, 스포츠IT융합