

# SNA

## Witcher Network 분석

데이터마이닝이론및응용  
에이쁠원하조

2020147024 김우영

2020251009 김혜리

2020195053 노가원

2020147018 조윤영

# RQ

소설<더 워쳐시리즈>를 원작으로 하여 현재 시즌 2까지 방영된 넷플릭스 오리지널 드라마 <워쳐> 시리즈는 탄탄한 구성으로 사랑받는 드라마이다. 하지만 방대한 소설 내용 때문에 드라마만을 시청한 시청자들은 인물간의 관계를 이해하기 어렵다는 평가가 있다. 우리는 소설 1~8권까지의 인물 관계 데이터를 활용하여 <워쳐>의 인물 관계를 분석하고자 한다. 하지만 단순히 1~8권 까지의 내용을 네트워크 분석하는 것이 아니라 시즌1에 해당하는 4권까지의 관계와 소설 전체에 해당하는 관계를 각각 분석하여 시즌1, 시즌2에서 변화하는 인물의 양상을 살펴보고자 한다. 또한 각 중심성 지수가 소설 내용이 진행됨에 따라 변화하는 양상을 바탕으로 인물 간에 어떠한 변화가 있는지, 내용의 흐름이 어떻게 진행되는지 확인해보고자 한다.

## Data, 변수 설명

witcher\_network.csv는 다음과 같은 변수를 가지고 있고 우리는 소설 1,2,3,4권에 해당하는 data\_half와 소설 전권의 데이터인 data\_full로 나누어서 비교하여 분석을 진행했다.

Source : <더 워쳐시리즈>의 등장인물

Target : 등장인물 중 Source의 인물과 연결되어 관계가 있는 인물

Type : 연결의 종류인 Directed/Undirected를 구분한다.

Weight : 두 등장인물이 상호작용한 횟수로서 관계의 깊이를 나타내는 수치라고 볼 수 있다.

book : 처음으로 <더 워쳐시리즈>에서 등장한 책의 권수.

## Outlier 제거

두 데이터 각각에서 다른 노드들과 어떠한 연결도 없는 노드는 제거하였다.

data\_full -> 'Vreemde', 'Morteisen' 제거

data\_half -> 'Voymir', 'Sigismund' 제거

df_f			df_f.describe()			df_h			df_h.describe()		
	Source	Target	Weight				Source	Target	Weight		
0	Velerad	Geraht	1		Weight	0	Velerad	Geraht	1		Weight
1	Geraht	Foltest	5	count	2383.000000	1	Geraht	Foltest	5	count	1118.000000
2	Foltest	Geraht	4			2	Foltest	Geraht	4		
3	Adda	Geraht	1	mean	3.406630	3	Adda	Geraht	1	mean	3.762075
4	Geraht	Ostrit	4	std	6.451621	4	Geraht	Ostrit	4	std	7.011866
...	...	...	...			...	...	...	...		
2595	Lytta	Ciri	2	min	1.000000	1146	Aen	Rudiger	1	min	1.000000
2596	Lytta	Triss	1	25%	1.000000	1147	Mistle	Rudiger	1	25%	1.000000
2597	Triss	Dandelion	1			1148	Rudiger	Mistle	1		
2598	Triss	Yarpen	1	50%	1.000000	1149	Mistle	Asse	2	50%	2.000000
2599	Dandelion	Mistle	1	75%	3.000000	1150	Kayleigh	Asse	1	75%	3.000000
2383 rows x 3 columns						1118 rows x 3 columns					
				max	88.000000					max	88.000000

```
df_f = data_full[['Source', 'Target', 'Weight']]
df_f.drop_duplicates(subset=['Source', 'Target', 'Weight'], inplace=True)

#these characters were way off to the side which made the rest of the network
df_f = df_f.drop(df_f.index[df_f['Source'] == 'Vreemde'])
df_f = df_f.drop(df_f.index[df_f['Source'] == 'Morteisen'])
df_f = df_f.drop(df_f.index[df_f['Target'] == 'Vreemde'])
df_f = df_f.drop(df_f.index[df_f['Target'] == 'Morteisen'])
```

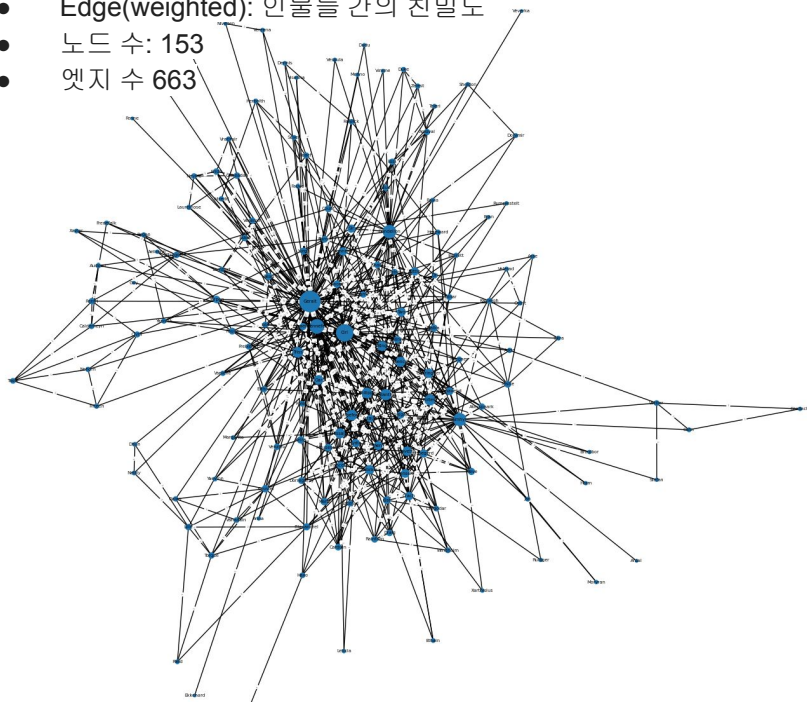
```
df_h = data_half[['Source', 'Target', 'Weight']]
df_h.drop_duplicates(subset=['Source', 'Target', 'Weight'], inplace=True)

#these characters were way off to the side which made the rest of the network
df_h = df_h.drop(df_h.index[df_h['Source'] == 'Voymir'])
df_h = df_h.drop(df_h.index[df_h['Source'] == 'Sigismund'])
df_h = df_h.drop(df_h.index[df_h['Target'] == 'Voymir'])
df_h = df_h.drop(df_h.index[df_h['Target'] == 'Sigismund'])
```

## node와 edge 및 네트워크 그림

### Data\_Half

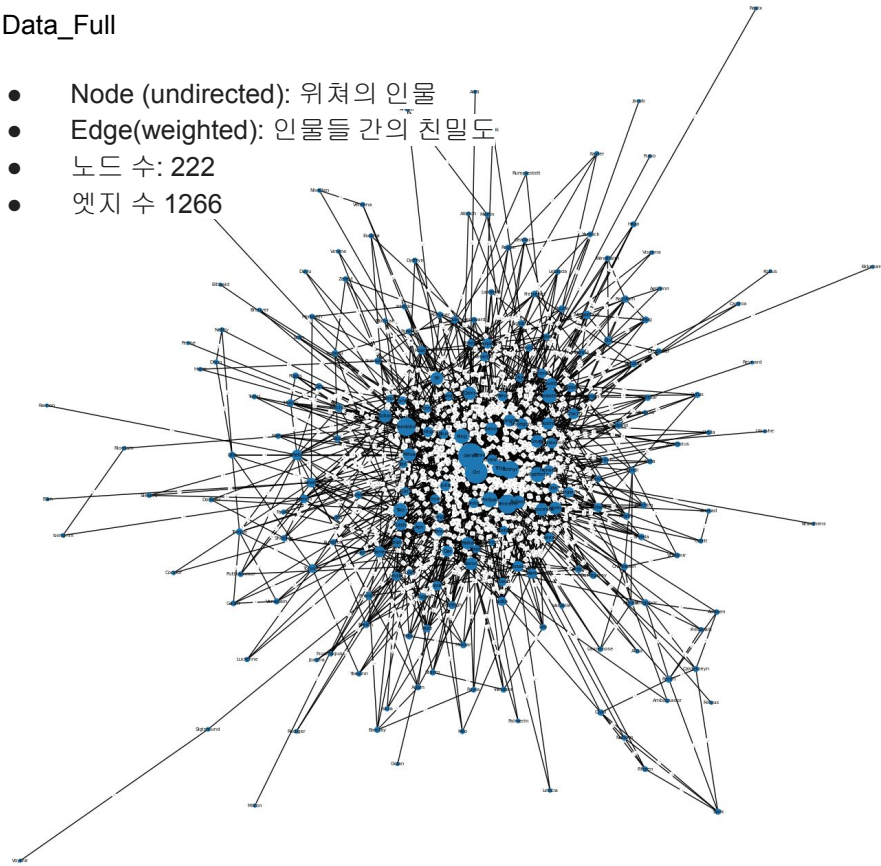
- Node(undirected): 위치의 인물
- Edge(weighted): 인물들 간의 친밀도
- 노드 수: 153
- 엣지 수 663



Geralt, Ciri와 같은 인물이 가장 큰 영향력을 갖는 것을 볼 수 있다. Dennis, Rennie, Ekkerhard 등의 인물은 중심에서 멀리 plot되어 있어 적은 영향력을 갖는 것을 볼 수 있다. 이는 소설에서 큰 비중을 갖지 않는 인물임을 의미한다.

### Data\_Full

- Node (undirected): 위치의 인물
- Edge(weighted): 인물들 간의 친밀도
- 노드 수: 222
- 엣지 수 1266



weighted를 사용하여 본 결과 Geralt, Ciri, Dandelion과 같은 인물이 가장 큰 영향력을 갖는 것을 볼 수 있다. 반면 Ramon, Ekkerhard 등의 인물은 중심 바깥에 작은 원으로 표현된 것을 보아 적은 영향력을 갖는 인물인 것을 알 수 있다. 또한 기준에 큰 영향력을 갖지 못했던 Dandelion이 비교적 큰 영향력을 갖게된 것을 볼 수 있다.

# 네트워크의 degree distribution 확인 및 density/diameter 등 확인

## 1. Data\_full



node와 연결된 모든 edge의 수인 degree를 통해 각 인물당 다른 인물들과의 연결 정도를 파악할 수 있었다. degree distribution을 확인한 결과 대부분의 node들의 edge는 0-40사이에서 분포해 있는 것을 알 수 있고, 이를 초과하는 node들의 경우 시리즈 내에서 상당히 영향력이 있는 인물을 즉 주인공에 준하는 인물들임을 예상할 수 있다. edge가 가장 많은 node들의 경우 그 수가 137,72,69,62로 다른 node들보다 edge수가 특별히 많음을 알 수 있었다.

네트워크에서 가장 멀리 연결된 pair의 거리인 diameter는 5로 edge수가 많은 것에 비해 diameter는 상대적으로 작음을 알 수 있다. 따라서 해당 node는 시리즈중 인물들과 큰 관계가 없는 주변 인물로 파악된다. density의 경우 0.0516으로 낮음을 알 수 있다. 이는 node들간 연결이 활발하지 않음을 의미한다. 따라서 edge수가 가장 많은 인물들을 제외한다면 다른 인물들간 연결이 크게 활성화 되지 않음을 의미한다. transitivity의 경우 0.257로 크게 높지 않음을 알 수 있다. 이는 네트워크에서 3이상의 관계에서 연결된 빈도가 크게 높지 않음을 알 수 있다. 이는 인물들이 군집을 형성하는 빈도가 낮다고 해석할 수 있겠다.

해당 데이터는 undirected data로 reciprocity는 0이다

## 2.Data\_half



data\_half의 경우 data\_full과 마찬가지로 degree distribution을 확인한 결과 대부분의 node들의 edge는 0-40 사이에서 분포해 있는 것을 알 수 있고, 이를 초과하는 node들의 경우 data\_full과 마찬가지로 시리즈 내에서 상당히 영향력이 있는 인물들로 파악된다. 다만 크기가 큰 edge들의 경우 107,69,48인 것을 보아 시리즈2에서 주인공들이 더 많은 인물들을 만나며 시리즈1에 비해 영향력이 큰 node가 한개 적은 것을 보아 시리즈2에서 새로운 영향력이 큰 인물의 등장 또는 기존 인물중 한 인물의 비중이 커진 것임을 예상해볼 수 있다.

diameter는 5로 data\_full과 마찬가지로 해당 node는 시리즈중 인물들과 큰 관계가 없는 주변 인물로 파악된다. density의 경우도 역시 data\_full과 마찬가지로 0.057로 낮고 이는 node들간 연결이 활발하지 않음을 의미하며 영향력이 큰 인물들을 제외하면 다른 인물들 간 연결이 크게 활성화 되지 않음을 알 수 있다. transitivity의 경우 0.233으로 마찬가지로 크게 높지 않으며 네트워크에서 3이상의 관계에서 연결된 빈도가 크게 높지 않음을 알 수 있고 이는 인물들이 군집을 형성하는 빈도가 낮다고 해석할 수 있다.

# 네트워크 중심성 분석

시즌1, 소설 1,2,3,4권에 해당하는 네트워크 분석      시즌1, 시즌2, 소설 전권에 해당하는 네트워크 분석

```
degree centrality:
[('Geralt', 0.7039473684210525), ('Ciri', 0.4539473684210526), ('Vennefer', 0.3157894736842105)]
betweenness centrality:
[('Geralt', 0.5177182781865106), ('Ciri', 0.18284084725903282), ('Emhyr', 0.1060502338533477)]
closeness centrality:
[('Geralt', 0.7676767676767676), ('Ciri', 0.6333333333333333), ('Vennefer', 0.5846153846153846)]
eigenvector centrality:
[('Geralt', 0.3833791345535262), ('Ciri', 0.3063770518374445), ('Vennefer', 0.249740664222411)]
```

```
degree centrality:
[('Geralt', 0.6199095022624435), ('Ciri', 0.46153846153846156), ('Vennefer', 0.3257918552036199)]
betweenness centrality:
[('Geralt', 0.9917440422628955), ('Ciri', 0.14917506159328225), ('Dandelion', 0.07982894426906378)]
closeness centrality:
[('Geralt', 0.7198697068403909), ('Ciri', 0.638728323699422), ('Dandelion', 0.5800524934383202)]
eigenvector centrality:
[('Geralt', 0.302577502003062), ('Ciri', 0.28564285851311527), ('Vennefer', 0.22734753044606196)]
```

**degree centrality** 분석을 보면 ‘Geralt’가 시즌1, 2 모두에서 가장 높은 값을 보였고 이는 모든 시즌을 통틀어 ‘Geralt’가 중심 인물로서 모든 관계의 중심에 있다고 분석된다.

**betweenness centrality** 분석을 보면 여전히 ‘Geralt’가 가장 높은 값을 가지지만 시즌 2에서 그 값이 낮아진 것을 확인할 수 있다. 이는 시즌 1보다 시즌 2에서 ‘Geralt’가 중계자 노드로서의 영향이 감소했다고 분석 되는데 이것은 새롭게 등장한 ‘Dandelion’이 그 영향력을 가져간 것으로 분석해볼 수 있다. 혹은 시즌 2로 이야기가 진행됨에 따라 새로운 인물들이 등장하면서 네트워크의 양상이 바뀌었다고도 분석해볼 수 있다.

**closeness centrality** 분석을 보면 다른 인물들은 시즌1, 2간의 차이가 미약하지만 ‘Vennefer’ 대신 ‘Dandelion’이 높은 **closeness centrality** 값을 가지는 것을 볼 수 있다. **degree centrality**는 ‘Vennefer’가 더 높은 값을 가지는 것으로 보아 내용의 핵심 전개를 담당하기 보다는 인물간의 연결에 초점을 두고 만들어진 친화력 좋은 인물일 가능성이 있다.

**eigenvector centrality** 분석은 대체적으로 비슷한 값을 가진 것을 볼 수 있다. 이것은 ‘Geralt’, ‘Ciri’, ‘Vennefer’가 서로 비슷한 노드들과 연결되어 있어서 평균적으로 값이 비슷해졌을 가능성도 있지만 소설이 하나의 주인공만을 강조하여 진행되는 것이 아니라 3명의 인물이 모두 주인공으로서 각각 큰 비중을 갖는 형태로 진행된다고도 분석해볼 수 있다. 또한 새로운 인물 ‘Dandelion’이 높은 값을 가지지 않는 것으로 보아 소설의 핵심 인물보다는 보조적인 친화력 좋은 인물일 가능성이 더 높을 것으로 분석된다.

## 한계점 및 개선안

### 한계점

- 본 연구는 드라마 <위쳐> 시청자들의 이해를 돕기 위한 목적을 가지고 진행되었다. 반면, 분석에 활용한 데이터는 원작 소설의 내용을 바탕으로 추출한 데이터이기에 소설과 드라마 사이의 세부적인 차이점은 반영하지 못한다는 한계가 있다.
- 소설 텍스트 데이터에서 추출한 데이터이기에, 표면상으로 드러나지 않고 전반적인 내용 속에 함의되어있는 관계를 표현하지 못했을 가능성이 있다.
- SNA는 일반적으로 해석 과정이 중요하기에, 소설 <더 위쳐시리즈> 또는 드라마 <위쳐>에 보다 많은 지식과 이해를 가진 상태가 아니었기에 심층적인 분석에 한계가 있었다.
- 중심성 분석의 특성상, 중심인물 몇 명 위주로 분석이 진행되었다는 한계가 있다.

### 개선안

- 드라마 <위쳐>의 대본 데이터를 이용해 분석을 진행해본다면 소설과 드라마의 차이점을 확인해볼 수 있을 것이다.
- 소설 <더 위쳐시리즈> 또는 드라마 <위쳐>에 대한 높은 이해도를 가진 이들의 의견을 구할 수 있다면 보다 심층적인 분석이 가능할 것이다.
- 중심 인물들을 제거하고 네트워크 분석을 새롭게 진행해보는 과정을 통해, 차순위로 중요한 인물들의 관계에 대해서도 파악해볼 수 있을 것이다.

## creative application areas

- 소설의 사회 연결망 분석 결과를 추후 텍스트 분석 인공지능이 소설의 내용을 정확히 파악할 수 있도록 적용하여 더욱 효과적인 학습 효과를 기대해 볼 수 있다.
- 관련하여 확장한다면, 특정 소설 텍스트를 입력하면 해당 소설 등장인물들의 네트워크 그림을 보여주고 분석해주는 인공지능 프로그램도 개발할 수 있으리라 기대한다. 이러한 프로그램은 장편소설이나 여러 개의 시즌을 가지는 드라마 등을 효과적으로 이해하는 데에 도움을 줄 수 있을 것이다.

## Reference

Sadasivan, Ava. "Witcher Network." *Kaggle*, 14 Jan. 2022, <https://www.kaggle.com/datasets/avasadasivan/witcher-network>.

Avasadasivan. "Witcher Network Analysis." *Kaggle*, Kaggle, 15 Jan. 2022, <https://www.kaggle.com/code/avasadasivan/witcher-network-analysis>.