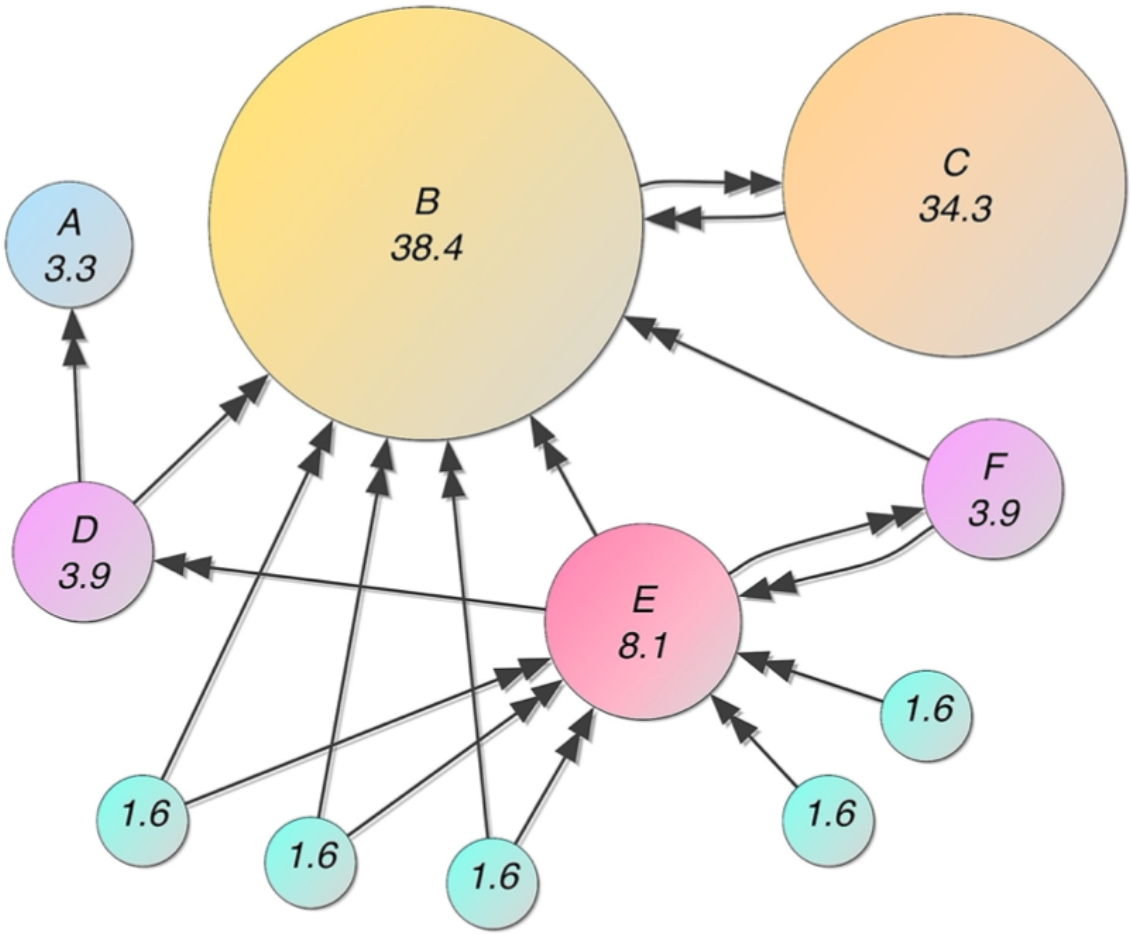
1. 알고리즘 구현

1-1. 그래프 자료구조

우선, PageRank 를 구현하기 위하여, 그래프를 저장할 때 어떤 자료구조를 활용하면 좋을지 고민해 보았다. PageRank 의 탄생 배경에 따라 웹 페이지를 노드로, 하이퍼링크를 간선으로 고려하였을 때, 웹 페이지를 나타내는 노드의 개수는 무수히 많다. 만약 이를 Adjacency Matrix 로 구현한다면, $O(V^2)$ 의 공간 복잡도로, V 가 커질수록 활용되지 않는 메모리 공간은 더욱 많이 존재하게 될 것이다. 따라서, 메모리를 효율적으로 활용하기 위해선 Adjacency list 를 활용하는 것이 최선이라는 판단을 하였다.

1-2. 발생 가능한 문제점

알고리즘 구현을 완료 후, 이를 실험해 보고자 예시로 주어진 그래프를 활용하였다.



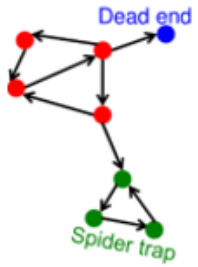
다음은 매개변수 walking length n 의 값을 10000 으로 두고, 알고리즘을 실행한 결과이다.

A	B	C	D	E	F	-	-	-	-	-
0	0.4999	0.4998	0.0001	0.0001	0	0	0.0001	0	0	0

의도했던 결과와는 다른 결과가 도출되는 것을 확인하였다. 간단하게 파악해 보면, A 와 B 의 노드는 서로를 가르키는 간선만 존재하므로, 다른 간선으로 나가지 못하고, 서로에 in-degree 값만을 증가시키기 때문이다.

그게 왜 문제일까? PageRank 에서 Random Walker 는 실제 웹 서핑을 하는 사람의 행동을 모델링한 것이라는 점을 생각해 볼 필요가 있다. 앞선 결과에 대하여 생각해보면, 웹 서퍼가 A 와 B 라는 두가지 웹 사이트 중 하나에 도달하는 순간부터, A 와 B 웹 사이트 외에는 다른 사이트로 갈 수 없다는 것인데, 실제 웹 서퍼가 작은 루프에 빠져 들었을 때에 그 루프를 나가지 못하고 맴돌 가능성은 거의 없다. 실제 웹 서퍼는 뒤로가기, 새로 검색하기 등을 통해 다시 새로운 노드로 나아갈 것이다.

결국, 이와 같이 알고리즘 구현 상 발생 가능한 문제점이 존재하며, 이를 크게 2 가지로 볼 수 있다. 이는 다음 그림에서 확인할 수 있다.



1-2-1. Dead End

어떤 페이지들은 out-links 를 갖지 않아, Random Walker 는 갈 곳이 없게 된다.

1-2-2. Spider Trap

Random Walker 가 트랩에 빠지는 현상. 모든 out-links 가 그룹 내에 존재하여, 그룹 외에 외부 링크로 나가지 못한다.

1-3 해결책

이를 해결할 수 있는 방법으로, 알고리즘에 jump 혹은 teleport 라는 동작을 추가할 수 있다. 이웃 탐색을 멈추고 완전히 다른 곳으로 이동할 확률 q 를 추가하는 것이다. Random Walker 의 각 시점의 스텝마다, 두가지 옵션을 갖도록 한다.

- (1-q)의 확률로 기존의 규칙에 따라 이웃 노드를 탐색
- q 의 확률로 임의의 노드로 점프. 이 경우에는 이웃하지 않은 노드라도 무작위로 선택하여 이동

이 때, Dead End 에서는 q = 1 의 확률로 바로 다른 노드로 이동하도록 한다.

그렇다면, Dead End 상황 외에, 각 시점에 적용되는 q 의 값을 어떻게 결정해야 하는가? 앞선 1-2 에서 Random Walker 가 실제 웹 서퍼의 행동을 모델링한 것이라는 점을 생각해 보았듯이, jump 동작 또한 실제 웹 서퍼의 행동을 모델링한 것이라는 점을 생각해 볼 필요가 있다.

The Anatomy of a Large-Scale Hypertextual Web Search Engine - Sergey Brin and Lawrence Page 논문에 소개된 damping facotr 라는 요소는 '어떤 마구잡이로 웹서핑을 하는 사람이 그 페이지에 만족을 못하고 다른 페이지로 가는 링크를 클릭할 확률' 을 의미하며, 이는 (1 -q) 의 확률로 이웃 노드를 탐색하는 동작과 같다. 반대로, 나머지 q 의 확률의 jump 동작은, 해당 서퍼가 링크 클릭을 멈추고 해당 페이지를 살펴보는 동작인 것이다. 즉, 해당 Random Walker 가 도착점에 도달하였기 때문에, 새로운 Random Walk 동작이 다시 시작되는 것이라고 해석할 수 있다.

논문에서는 이러한 damping factor 를 0.85 로 두었다고 소개하였으며, 이에 따라 이번 실험에 사용되는 q 값을 0.15 의 값으로 결정하였다.

해당 해결책을 알고리즘에 적용하여, 1-2 에서 예시로 활용한 그래프에 매개변수 n = 10000, q = 0.15 의 값으로 알고리즘을 실행한 결과이다.

A	B	C	D	E	F	-	-	-	-	-
0.0332	0.3809	0.3384	0.0399	0.0835	0.0427	0.0172	0.016	0.0168	0.0157	0.0157

알고리즘이 의도했던 대로 동작한 것을 확인할 수 있다.

2. Page Rank 알고리즘 실행 결과

2-1. 가충지 없는 그래프에서의 PageRank

2-1-1. 무방향 그래프 (스타워즈)

- 파일

nodes: starwars-full-interactions-allCharacters-nodes.tsv
links: starwars-full-interactions-allCharacters-links.tsv

- 실행 결과

n = 40000, q = 0.15, Top 5

ANAKIN	OBI-WAN	C-3PO	PADME	LUKE
0.0406	0.0393	0.0332	0.0312	0.0283

2-1-2. 방향 그래프 (따릉이)

- 파일

nodes: station_names.tsv
links: bicycle_trips_all.tsv

- 실행 결과

n = 40000, q = 0.15, Top 5

옥수역 3 번 출구	청계천 생태교실 앞	여의나루역 1 번 출구 앞	고속터미널역 8-1 번, 8-2 번 출구 사이	뚝섬유원지연 1 번 출구 앞
0.0024	0.0022	0.0020	0.0019	0.0018

2-2. 가충지 있는 그래프에서의 PageRank

2-2-1. 무방향 그래프 (스타워즈)

- 파일

nodes: starwars-full-interactions-allCharacters-nodes.tsv
links: starwars-full-interactions-allCharacters-links.tsv

- 실행 결과

n = 40000, q = 0.15, Top 5

ANAKIN	HAN	OBI-WAN	C-3PO	R2-D2
0.0576	0.05687	0.05682	0.0509	0.0485

2-2-2. 방향 그래프 (따릉이)

- 파일

nodes: station_names.tsv
links: bicycle_trips_all.tsv

- 실행 결과

n = 40000, q = 0.15, Top 5

뚝섬유원지연 1 번 출구 앞	롯데월드타워 (잠실역 2 번 출구 쪽)	고속터미널역 8-1 번, 8-2 번 출구 사이	옥수역 3 번 출구	마포구민체육센터 앞
0.0028	0.0027	0.0024	0.0021	0.0020

2-3 실험결과 고찰

가중치가 있는 경우와 가중치가 없는 경우를 각각 실행해 보았다. 각각의 페이지 랭크가 의미하는 바는 스타워즈 데이터의 경우엔 비중이 높은 중요한 인물 , 따릉이 데이터의 경우에는 사용이 활발하게 이루어지는 중요한 구역이다. 두가지 데이터의 가중치 값은 각각 같이 출연한 횟수, 반납이 이루어지는 횟수로, 앞서 말한 중요도에 지대한 영향을 끼치는 요소이다. 따라서, 가중치를 활용하는 것이 보다 정확한 PageRank 값을 측정할 수 있다고 생각한다.

3. 왕좌의 게임 데이터를 활용한 인물의 PageRank 값 측정

앞선 2 번 실험을 진행하며 아쉬운 점으로, starwars 영화를 한번도 본 적이 없어, 해당 등장인물들에 대하여 전혀 모른다는 점과, 따릉이 데이터도 마찬가지로 어느 구역의 PageRank 값이 높다는 것을 직관적으로 받아들이지 못한다는 점이 있었다.

따라서, PageRank 의 결과를 직관적으로 받아들일 수 있으며, 흥미롭게 실험할 수 있는, 애청했던 미국 드라마 시리즈 왕좌의 게임(Game of Thrones)의 network 에 PageRank 를 적용하여 중요한 인물을 분석해 보고자 한다.

왕좌의 게임이 신선하고 재미있었던 이유는 주인공이 죽는다는 점이었다. 해당 인물이 드라마 시리즈에 주인공 이구나 생각이 들 때 즈음 가차없이 잔인하게 죽으며, 주인공이 바뀌는 과정이 생생하다. 이러한 왕좌의 게임의 특성을 활용하여 시리즈의 진행과정에 중간 중간의 PageRank 값을 측정하면, 주인공이 바뀌는 과정이 드러나는 좋은 예시가 될 수 있을 것이라고 생각했다.

3-1. 데이터 준비

https://www.kaggle.com/mmmarchetti/game-of-thrones-dataset 의 데이터를 활용하여 직접 작성한 알고리즘에 적용될 수 있도록 정제하였다 (dataset 에 함께 첨부). 드라마가 아닌, 책에서 추출된 데이터이지만, 같은 스토리를 공유하여 의도하는 실험에는 마찬가지로 적합하다 판단하여 사용하였다. 총 1 권부터 5 권까지의 데이터를 준비하였다.

- 파일

book*_nodes.tsv: 왕좌의 게임 *권에 나온 인물

book*_links.tsv: 왕좌의 게임 *권에 나온 인물들 사이의 공통 등장 빈도

3-2. 실험 결과

- book 1

n = 40000, q = 0.15, Top 5

Eddard-Stark	Robert-Baratheon	Tyrion-Linnister	Jon-Snow	Catelyn-Stark
0.07	0.0479	0.0473	0.0472	0.0363

예상했던 대로, Stark 가의 가주 Eddard-Stark 가 가장 높은 PageRank 값을 가졌다. 다음으로 PageRank 가 높은 인물은 당시에 칠왕국을 다스리던 국왕이던 Robert-Baratheon 이다.

시리즈의 시작이 Robert-Baratheon 이 Eddard-Stark 에게 왕의 핸드로의 임명을 제안하는 장면이었던 점을 고려하면, 이 두명의 등장인물이 높은 PageRank 를 가져가는 것은 자연스럽다.

의외로, 초반의 Jon-Snow 의 비중이 낮은 줄 알고 있었던 본인의 생각과는 달리, PageRank 값이 Top5 안에 들어와 있다는 점이 흥미롭다.

- book 2

n = 40000, q = 0.15, Top 5

Tyrion-Lannister	Joffrey-Baratheon	Arya-Stark	Bran-Stark	Jon-Snow
0.045	0.033	0.0304	0.0301	0.0286

전 권의 두 주요 인물에 죽음으로, 전체적인 서사가 시작된다. 이에 따라, Eddard-Stark 와 Robert-Baratheon 이 PageRank 의 순위권 밖으로 밀려났고, 기존의 3 위였던 Tyrion-Lannister 가 1 위가 되었다. Tyrion-Lannister 는 전체적인 서사의 진행 과정에 속하는 주요 인물로, 지속적으로 높은 PageRank 값을 가질 것으로 예상된다.

다음으로는, 국왕 Robert-Baratheon 에 죽음에 따라 친자(인줄 알았던) Joffrey-Baratheon 이 왕좌에 오르게 되었기에, 이에 따라 높은 순위권을 차지하는 것을 확인할 수 있다.

Jon-Snow 는 여전히 Top5 안에 속해있는 것을 확인할 수 있다.

- book 3

n = 40000, q = 0.15, Top 5

Jon-Snow	Tyrion-Lannister	Robb-Stark	Jaime-Lannister	Sansa-Stark
0.036	0.035	0.028	0.027	0.025

전 권에서 PageRank 2 위를 기록했던 국왕 Joffrey-Baratheon 은 독살로 인하여 사망하였고, 이에 따라 PageRank 순위권 밖으로 사라진 것을 확인할 수 있다.

북부의 벽 너머의 이야기가 본격적으로 시작되면서, 자연스럽게 Jon-Snow 의 비중이 올라가 PageRank 1 위를 차지하였다.

Robb-Stark 의 아버지의 복수를 위한 여정이 본격화 되며 비중이 높아진 것을 확인할 수 있다. 결국엔 죽음에 이르기 때문에, 다음 권 부터는 PageRank 의 순위권 밖으로 밀려날 것으로 예상된다.

- book 4

n = 40000, q = 0.15, Top 5

Cersei-Lannister	Jaime-Lannister	Brienne-of-Tarth	Samwell-Tarly	Margaery-Tyrell
0.0587	0.049	0.026	0.023	0.019

PageRank 의 전체적인 판도가 바뀌었다. Cersei-Lannister, Jaime-Lannister 의 순위가 높은 것으로 보아, Lannister 가의 스토리와 비밀이 밝혀지는 권인 듯 하다. 또 한, Margaery-Tyrell 이 등장하며, 킹스 랜딩에서 벌어지는 갈등이 심화되는 스토리를 주로 다루었던 권으로 예상된다.

외에도 Jon-Snow 의 조력자 Samwell-Tarly 가 등장하며, 벽 너머의 스토리 또한 심화되어 간다.

- book 5

n = 40000, q = 0.15, Top 5

Jon-Snow	Daenerys-Targaryen	Stannis-Baratheon	Tyrion-Lannister	Tyrion-Lannister
0.0609	0.0458	0.0338	0.0276	0.0246

다시금 Jon-Snow 의 PageRank 가 1 위로 올라오게 되었다. 여기서 눈여겨 볼 점은 Daenerys-Targaryen 의 PageRank 가 급격하게 올라왔다는 점이다. 이전까지는 먼 대륙의 잠깐의 스토리에 등장하던 인물인 Daenerys-Targaryen 의 비중이 중요해진 것이다.

이를 통해, 앞으로 진행되어질 중요한 스토리가 킹스 랜딩의 왕좌를 위한 싸움이 아닌, 벽 너머의 존재와의 싸움일 것이라는 예상할 수 있다.

3-3. 실험 고찰

왕좌의 게임을 재미있게 보았던 이유 중 하나로 초반에 메인 주인공을 쉽게 예측할 수가 없다는 점이 있었다. 후에 가장 중요한 등장인물이 될 Jon-Snow 와 같은 경우도, 처음 시리즈를 볼 때에는 전혀 예측하지 못하였었다. 허나, 이에 PageRank 알고리즘을 적용해 보았을 때, 해당 인물의 비중은 초반부터 꾸준히 높았다는 점을 확인할 수 있었다.

결론적으로, 이 실험을 통하여, PageRank 알고리즘을 적절히 활용한다면 쉽게 파악하지 못하고 있던 중요도 높은 노드 또한 정량화된 값으로 파악해 낼 수 있다는 점을 알 수 있었다.