MUSIC TALE

Computer programming

Prof. Kim

TEAM 6 김민지, 최민희, 이성규, 장홍석

June 6, 2016.

Table of Contents

1. 서론

저희 조는 지프의 법칙(Zipf's law)으로부터 영감을 받아 프로젝트를 시작하게 되었습니다. 미국의 문헌학자 George Kingsley Zipf 는 어떤 문헌에서든지 단어의 등장 빈도수는 그 단어의 순위에 반비례한다는것을 발견했습니다. 예를 들어, 'the', 'of', 'and' 순서대로 단어가 등장했다고 가정하면, 가장 많이 나타난 the 의 횟수는 K 번, of 는 K/2 번, 그리고 and 는 K/3 번 등장한다는 것입니다.

이 지프의 법칙이 음악에서도 발견된다는 연구 결과를 [1] 보고 지프의법칙을 이용해 문학과 음악을 연결해볼수 있을 것이라고 생각을 했습니다. 저희의 생각을 확인해보기 위해서 문헌을 input 으로 주었을 때 문헌과 가장 어울리는 기존의 음악을 선택하는 알고리즘과 문헌에 따라서 새로운 음악을 생성하는 알고리즘을 만들었습니다.

문헌과 음악을 연결시키기 위한 연결고리로 분위기, 빠르기, 단어와 음의 등장 빈도를 이용했습니다. 1 차적으로 분위기를 이용해 감성분석을 실시하고, 이후에 빠르기와 단어와 음의 등장 비율을 통한 빈도분석을 2 차적으로 이용했습니다. 분석을 위해 고전 소설 약 30 편, 음악 약 2000 곡을 데어터로 이용하였습니다. 정제된 단어를 많이 사용하여 분석이 용이하기 때문에 고전 소설을 선택하였고, 그 중에서 중심극한의 정리를 활용 할 수 있고, 성능을 저하시키지 않게 하기 위해서 30 편의 소설을 선택하였습니다. 음악의 경우에는 음악 분석 모듈인 music21 에서 기본적으로 제공하는 약 2000 곡을 이용하였습니다.

데이터 분석을 위해서는 stop_words, os, re, numpy, scipy.stats, matplotlib, music21 모듈을 이용하였습니다. stop_words, os, resms 모듈은 문헌에서 불용어를 제거하거나, 문헌의 크기를 받아오고, 정규식을 이용하여 단어들을 정제하기 위하여 사용했습니다. Numpy, scipy.stats, matplotlib 는 통계 분석을 하고, 중심 극한의 정리를 적용하고, 분석 결과물들을 그래프로 시각화하기 위해 사용했습니다. 마지막으로 music21 을 이용해서 음악들을 분석하고, 생성하는데 이용했습니다.

---

[1] Manaris, B, Hizrel, T, and Pharr, W,.2005. "Zipf's Law, Music Classification, and Aesthetics." Computer Music Journal. Spring2005, Vol. 29 Issue 1, p55-69.

2. 본론

a. 문헌 분석

문헌을 음악으로 옮기기 위해서 우선적으로 문헌과 음악의 구성요소를 분석하는 작업이 필요했습니다. 문헌의 구성요소들 중에 문헌의 분위기 문헌의 속도, 그리고 단어 등장의 비율을 분석에 이용했습니다.

1) 분위기

글의 분위기는 크게 긍정적 분위기와 부정적 분위기로 나누어서 분석했습니다. 가장 먼저 긍정적인 분위기를 나타내는 단어와 부정적 분위기를 나타내는 단어 약 300 개를 학습시켰습니다. 그 후에 단순하게 더 많이 등장한 감정의 단어들을 글의 분위기로 파악하지 않고, 부정단어 수/ 긍정단어 수를 글의 분위기를 파악하는 지표로써 사용했습니다.

분석을 할때에 두가지 문제점이 있었습니다. 첫번째 문제는 부정을 나타내는 단어들이 긍정을 나타내는 단어 앞에 접두사 dis 나 un 등이 첨가되어 두 단어가 중복적으로 계산이 된다는 것이었습니다. 저희는 부정적인 단어가 긍정적인 단어에 부정 접두사가 붙어 만들어졌다는 점에 착안하여 이 문제를 해결 했습니다. 단어가 dis, un 같은 부정 접두사가 붙은 부정적인 단어인지를 먼저 확인하고, 그 단어가 부정적인 단어가 아닐 경우에만 긍정 단어로 계산했습니다.

두번째 문제는 일반적으로 긍정 단어의 수가 부정 단어의 수가 훨씬 많다는 것이었습니다. 따라서 미리 30 개의 문헌을 학습시켜 평균과 표준 편차를 구하였습니다. 이 과정에서 표본의 크기가 커질 수록 표본들의 평균과 표준편차는 정규 분포를 보인다는 중심극한정리가 적용될 것이라고 가정했습니다. 이렇게 구한 평균과 표준편차를 이용해서 input 으로 준 문헌에 나타나는 단어들의 비율 중에 어느 것이 더 높은지 비교하는 방법을 선택했습니다. 결과적으로 평균보다 부정 단어 비율이 높으면 -1 로, 낮으면 1 로 결과값을 반환하는 알고리즘을 제작했습니다.

2) 빠르기

다음으로 한 문장이 얼마나 빠르게 끝나는지를 문장의 속도로 보기로 하였습니다. 따라서 전체 문헌의 길이와 한 문장이 끝날 때 나오는 구두점들(?, !, .)의 수를 비교하여 그 비율을 문장의 속도로 보았습니다. 전체 문장을 다 읽어오는 알고리즘은 속도가 굉장히 느렸습니다. 그래서 문헌의 용량은 전체 문헌의 길이와 비례한다는 것을 이용해서 읽어오려는 문헌의 용량 대비 구두점의 개수를 계산했습니다.

하지만 이렇게 구한 비율 자체로는 그 글이 다른 글들에 비해 얼마나 빠른

글인지를 알 수 없었습니다. 따라서 글의 분위기를 정할 때 학습시킨 30 개의 문헌들을 이용해서 주어진 글의 빠르기를 구했습니다. 이러한 문헌의 빠르기도 문헌의 분위기를 분석했을 때처럼 중심극한의 정리에 의해 정규분포를 따른다고 가정하였습니다. 텍스트의 빠르기는 0~1 의 값이 나오는 누정정규분포 함수 값을 통해, 빠를수록 1 에 가까워지도록 알고리즘을 제작했습니다.

3) 단어 빈도 비율

마지막으로 문헌에 등장하는 단어 등장 빈도의 비율을 구했습니다. 소설에 있는 단어들을 딕셔너리 구조를 이용해서 등장 빈도를 세고, 전체 등장한 단어의 수로 나누어서 비율을 구했습니다. 문헌을 한줄씩 읽으면서 공백을 기준으로 단어를 구분하여 단어를 키 값으로 딕셔너리에 추가했습니다. 문헌에 쓰인 단어들 중에서 'the' 나 'a' 혹은 기본 전치사와 같이 의미가 없는 불용어들은 제거 했습니다.

단어들의 빈도를 다 구한 후에 딕셔너리에 저장된 단어 빈도에 따라 내림차순으로 정리하고, 단어 빈도를 전체 단어수로 나누어 전체 값에 대한 비율을 알아내었습니다. 많이 등장할 수록 1 에 가깝고, 적게 등장할 수록 0 에 가까운 비율을 가지게 됩니다.

b. 음악분석

음악을 분석하기 위해 music21 에서 기본적으로 제공하는 악보들을 이용했습니다. 약 2000 곡이 넘는 곡들을 매번 분석을 하는 것은 너무 시간이 오래 걸려서 사용자가 작곡가를 설정하면 해당 작곡가의 음악들만 분석 하도록 시스템을 구현하였다.

1) 분위기

음악의 분위기는 장조와 단조를 이용해 분석했습니다. 음악의 요소들 중에서 음악의 밝음과 어두움을 가장 잘 들어내는 요소는 장조와 단조이기 때문입니다. 대부분의 장조는 밝고, 경쾌한 느낌을 주고, 단조는 어두운 느낌을 줍니다. 그렇기 때문에 음악이 장조인 경우에 1 을 반환하게 하고, 단조인 경우에는 -1 을 반환하게 했습니다.

2) 빠르기

두 번째로 음악의 빠르기는 악보의 빠르기 값(tempo) 값을 이용하였습니다. ♩=60 와 같이 표시된 메트로놈 표시는 1 분에 4 분 음표를 60 번 연주하라는 것으로 값이 높을 수록 더 빠른 음악입니다. 문헌 분석에서

빠르기를 분석할 때 느리면 0 의 값을 가지고, 빠르면 1 의 값을 가지게 했으므로, 유의미한 비교를 위해서 음악의 빠르기에도 0 에서 1 의 값을 가지게 했습니다.

선택한 작곡가가 썼던 음악들 중에 곡의 빠르기가 정규 분포에서 어느 위치에 있는지 계산하는 것입니다. 작곡가들이 여러 음악을 작곡했으므로 중심극한의 정리에 의해 음악들의 빠르기는 표준 정규 분포를 따르게 될 것이고, 가장 느린 음악은 0 의 값을 가지고, 가장 빠른 음악은 1 의 값을 가지게 되는 것입니다.

3) 음의 빈도 비율

마지막으로 음 등장 빈도의 비율을 계산하여 음과 옥타브를 키 값으로 딕셔너리에 값을 저장했습니다. 음과 옥타브 값을 모두 이용한 이유는 같은 음이더라도 옥타브에 따라서 다르기 때문입니다. 예를 들어 4 옥타브 도인 'C4'는 5 옥타브 도인 'C5'나 4 옥타브 레인 'D4'는 서로 다릅니다.

음악에서 등장한 키 값들을 내림차순으로 정리하고, 음의 등장 빈도를 전체 등장한 음의 빈도로 나누어서 비율을 구했습니다. 많이 등장할 수록 1 에 가깝고, 적게 등장할 수록 0 에 가까운 비율을 가지게 됩니다.

c. 음악 매칭

1) 분위기

위에서 분석한 문헌의 구성요소들과 음악의 구성요소들을 이용하여 2 단계에 거쳐서 문헌에 가장 적합한 곡을 매칭시켰습니다. 1 차적으로는 작곡가가 썼던 음악들 중에서 문헌의 분위기와 일치하는 곡들을 선정했습니다. 그리하여 밝은 느낌의 문헌은 장조의 음악들을 연결시켰고, 어두운 느낌의 문헌은 단조의 음악들을 연결시켰습니다.

2) 빠르기*빈도

;(문헌의 빠르기 순위-음악의 빠르기 순위)**2 * sum((각 순위의 문헌 단어-각 순위의 음악 음)**2) -> 값이 가장 작은 것이 유사함. 가장 작은 값을 가진 곡을 매칭.

문헌과 분위기가 일치하는 음악들을 대상으로만 빠르기와 빈도 비율을 이용해 가장 유사한 곡을 골랐습니다. 우선 두 자료의 빠르기 차이가 적은 것을 골랐습니다. 이를 위해 문헌의 빠르기와 음악의 빠르기를 빼고 그 값을 제곱했습니다. 문헌의 빠르기와 음악의 빠르기 모두 정규분포를 따르므로, 둘의 값을 빼서 유사도를 팡가합니다. 그 후에 두 값 사이의 차가 음수일 수 있으므로 두 값의 차에 제곱을 했습니다.

빈도의 비율에도 같은 방식을 이용했습니다. 다만, 문헌과 음악의 템포는 단 하나의 값이지만, 단어와 음의 빈도는 각 단어와 음의 수만큼 있으므로 두 값을 빼고, 그 값들을 더하는 sum of squares 를 이용했습니다. 단어와 음의 출현 빈도는 각각 지프의 법칙을 따르므로 유사도 측정을 위해 두 값을 뺄 수 있었습니다. 이때, 음의 수가 단어의 수보다 적은 경우에는 음의 개수에 맞춰 계산을 진행했고, 음의 수가 더 많은 경우에는 단어의 수에 맞춰 계산을 진행했습니다. 예를 들어, 문헌에서 많이 등장한 단어의 비율이 0.16, 0.08, 0.04, 0.02 이고, 음악에서 많이 등장한 음의 비율이 0.15, 0.07,0.03 이라면 0.02 값을 버린 (0.16-0.15)**2 + (0.08-0.07)**2 + (0.04-0.03)**2 의 값인 0.0003 을 가지게 되는 것입니다.

이렇게 각각 구한 (문헌의 빠르기-음악의 빠르기)**2 의 값과 sum((단어의 빈도-음의 빈도)**2)를 곱하여서 가장 작은 값을 가지는 음악을 문헌과 가장 유사한 분위기를 가진 문헌으로 선정했습니다. 왜냐하면 좌항과 우항 모두 두 값이 유사할 수록 0 에 수렴하고, 두 값의 차이가 클 수록 1 에 수렴하기 때문입니다. 그렇기 때문에 두 항을 곱한 값 역시 문헌과 유사할 수록 0 에 수렴하고, 차이가 클 수록 1 에 수렴하는 값을 가지게 됩니다. 이 중에서 가장 작은 값을 가진 음악을 문헌과 가장 유사한 음악으로 선정합니다.

d. 곡 생성

1) 변경 사전 생성

문헌과 음악들의 구성요소를 이용해서 문헌에 가장 잘 어울리는 음악을 선정하였습니다. 이후에는 문헌에 나타나는 단어와 음악에 나타나는 음을 빈도순으로 일대일 대응을 시켜줍니다. 예를들어서 한 문헌에서 ‘the’, ‘of’, ‘to’ 순서대로 등장을 했고, 음악에서 ‘C4’, ‘A4’, ‘G4’가 순서대로 등장을 했다고 보자. 그러면 {‘the’:’C4’,’is’:’A4,’to’:’G4’}처럼 단어와 음이 일대일 대응이 되는 사전을 만들 수 있습니다.

2) 음악 변경

문헌과 음악을 바탕으로 사전을 만든 이후에는 문헌에서 우리가 고른 단어가 나올 때마다 단어에 해당하는 음을 기존의 음악에 대입시켜 줍니다. 예를 들어, 위의 사전을 쓴다고 생각을 해봅시다. 'She is the most beautiful girl compared to any other girls.’란 문장이 있다면 사전에 없는 she 란 단어는 뛰어넘지만, is 에서는 위 사전에 따라서 A4 가 대입이 됩니다. 이와 마찬가지로 the, to 에서 각각 C4, G4 가 음악에 대입되는 식으로 음악을 변경해서 새로 생성합니다.

e. 제한점

하지만 저희가 사용한 방법은 곡을 매칭/생성하는데 한계가 있습니다. 첫번째로 저희가 사용한 표본은 소설 약 30 권, 음악 약 2000 개로 정확한 패턴 파악을 위한 학습에 있어서 한계가 있었습니다. 이를 위해서는 프로그램의 속도를 유지하면서도 분석에 이용하는 표본을 증가시켜야 합니다. 표본이 증가하면 정규분표를 구할때 오차 범위가 감소하므로 보다 정확한 패턴을 파악할 수 있을 것입니다.

두번째로 음악 매칭에 있어서, 다양한 자료들의 구성요소를 고려하지 못했습니다. 문헌같은 경우에는 그 글의 문체와 배경지식 등을 고려하지 못 했고, 음악 같은 경우에는 박자와 화음을 고려하지 못 했습니다. 글에 있어 문체는 그 글마다의 고유한 개성과 글쓴이의 색깔이 묻어 있어서, 작가마다 같은 내용을 전달해도 느낄수 있는 인상이 다르게 됩니다. 이 다양한 구성요소들을 더 많이 고려한다면 더 정확한 매칭 결과를 얻을 수 있을 것입니다.

마지막으로 악보를 완성하는 단계에서, 곡에 문헌의 전반적인 분위기를 고려하여 음을 매치시켰음에도 불구하고, 전체적으로 혹은 부분적으로 완벽하게 분위기 흐름을 반영하는데 한계가 있었습니다. 문헌을 다시 읽어 내려가면서, 빈도수를 기준으로 매치된 단어와 음을 그 순간 순간마다 기계적으로 바꾸게 되다보니, 음악의 전반적인 분위기에 문헌의 전반적인 분위기를 반영하지 못하는 한계가 있었습니다.

3. 결론

문헌과 음악에 공통적으로 지프의 법칙이 나타난다는 것을 이용하여 문헌을 음악에 매칭 시켜주고, 새로운 곡을 만들어 주는 프로젝트를 시작했습니다. 문헌과 음악의 많은 구성요소들 중에서 서로 유의미하게 비교할 수 있는 분위기, 빠르기, 빈도를 이용해서 곡의 유사도를 파악할 수 있었습니다. 이렇게 구한 유사도로 문헌에 가장 잘 어울리는 음악을 선택하고, 또 새로운 음악을 만들 수도 있었습니다.

비록 원하는 결과를 정확하게 얻을 수는 없었고, 모든 구성요소들을 분석해볼 수는 없었습니다. 하지만 유사도를 측정하기 위해 이용한 방법들은 다른 구성요소들에 이용한다면 더 정확한 결과를 얻을 수 있을 것입니다. 예를 들어, 감성 분석, 중심극한정리, sum of squares 등의 방법을 문헌의 문체나 배경지식 또는 음악의 박자나 화음을 분석하는데 이용한다면 더 좋은 결과를 얻을 수 있을 것입니다.