**인공지능 assignment2**

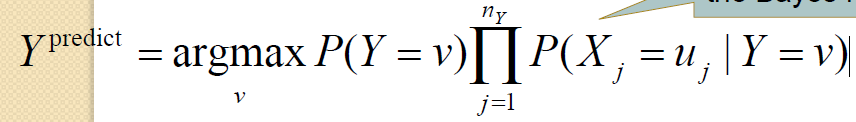
2013012041 안효민

**영화 리뷰 긍정/부정 분류하기**

* 사용 패키지

형태소 분석을 위해서 ‘Konlpy’패키지의 ‘Okt’ 태그를 사용했습니다.

* 사용기법

1. 확률이 모두 0이 되는 것을 방지하기 위해서 smoothing 기법을 사용했습니다.
2. Training data가 많은 관계로 확률들의 곱이 너무 작은 수가 나와 underflow가 발생하는 것을 방지하기 위해서 로그 값을 취해주었습니다.
3.  옆의 식을 바탕으로 긍정, 부정확률을 각각 계산하였습니다.

* 코드 동작 설명

데이터의 양이 많은 편이기에 자료구조 또는 태그 별로 수행시간을 알아보기 위해서 time을 체크했습니다.

read\_data 함수를 통해서 줄별로 분리된 데이터를 파일로부터 읽어들였습니다.

파일 실행파트에서 training data를 “ratings\_train.txt”에서 읽어들였고 tokenSet이라는 이름의 딕셔너리에 token의 종류와 해당 토큰이 긍정문 또는 부정리뷰에서 발견된 개수를 저장했습니다. 긍정토큰의 수와 부정토큰의 개수를 세어 마지막에 smoothing을 거쳐 긍정리뷰일 때 토큰이 나올 확률과 부정리뷰일 때 토큰이 나올 확률을 각각 계산해 주었습니다.

토큰은 기호와 외국어, 명사, 동사, 형용사들만 저장하고 나머지는 버렸습니다.

naivebayes\_classifier 함수에서 인자로 넣은 파일을 분류했습니다. 각 줄별로 위의 사용기법에 나온 식을 바탕으로 긍정인지 부정인지 조사했고 긍정이면 label에 1 부정이면 label에 0을 넣어주어 파일을 새로 작성했습니다.(ratings\_result.txt)

이 과정에서 underflow를 방지하기 위해서 log값을 취해주었으며 만약 긍정 부정 확률이 같을 경우 긍정으로 처리해주었습니다.

* 피드백

이번 과제는 무엇보다도 (1) 어떤 형태소를 분석에 사용할 것인지 (2) 추가적인 데이터 가공에 어떤 기법을 사용할 것인지 가 중요하다고 생각되어 여러 개의 형태소 분석기를 사용해 보았습니다. 처음에는 Kkma 태그를 사용하여 분석하였는데 사용한 결과 Kkma의 특성상 형태소를 좀 더 세분화하여 분류하기 때문에 비교적 정확한 결과를 얻을 수 있다는 것을 알았습니다. 하지만 정확한만큼 분석에 너무 많은 시간을 요했고, 무엇보다도 다른 파일들은 괜찮았지만 “ratings\_train.txt” 파일은 시간이 너무 길어서인지 너무 많은 공간을 잡아먹어서인지 알 수 없는 오류를 일으켰고 구글링을 해본 결과 Kkma태그의 자체적인 버그라는 사실을 알 수 있었습니다. 따라서 Kkma는 사용하기 힘들다는 것을 알았습니다.

차선책으로 Komoran을 사용해보았는데 Komoran은 Kkma보다는 널널한 분류를 했지만 속도에 있어서 월등히 발전된 모습을 보여주었습니다. 하지만 Komoran 또한 Kkma와 같은 오류를 일으켜서 사용할 수 없었습니다.

결국 Twitter를 사용하여 분석을 하였고 Twitter의 특성상 분석은 매우 빨랐지만 정확도에 문제가 있었습니다.

또한 과제를 진행하면서 특정 토큰이 긍정리뷰나 부정리뷰에서 한번도 나타나지 않았을 경우 확률이 0이 되어 결과가 무조건적으로 0이 나오는 현상을 발견하였습니다. 이 문제를 해결하기 위해서 smoothing과정이 필요하다는 사실을 알게 되었고 smoothing과정을 적용한 결과 모든 결과값이 0이 되거나 1이 되는 현상은 사라졌습니다.

또한 데이터셋이 매우 큰 만큼 확률이 굉장히 작아 확률들의 곱이 underflow를 일으킬 수 있다는 사실을 발견해서 log를 취한 값을 더하는 방식으로 해결했습니다.

처음 분류기를 실행시켰을 때는 토큰들과 확률을 클래스 형태로 저장하여 운용하였는데(클래스가 직관적이라고 생각해서 사용했습니다.) 매번 새로운 클래스를 만들고 또 해당 인스턴스들을 사용하는 것이 많은 cost를 소비한다는 사실을 알았습니다. 그래서 클래스 대신 좀 더 가벼우면서도 적합하다고 생각되어지는 딕셔너리를 사용해서 데이터들을 관리했습니다.

이번 과제를 수행하면서 처음으로 대용량의 데이터를 가공하고 이용하였는데 적은 데이터를 관리할 때와는 다르게 자료구조 설계에 많은 tradeoff를 고려해야 한다는 사실을 알게 되었습니다. 또한 분류의 정확성을 위해서는 좀 더 세분화된 형태소 분석이 유용하며 데이터 가공에 어떤 기법을 쓰느냐도 정확성에 큰 영향을 미친다는 것을 알게 되었습니다.

* Result

