

11조 기말 프로젝트 보고서 

* 호텔 리뷰 데이터를 활용한 자연어 처리 및 모델 생성 -

|  |  |
| --- | --- |
| **과목명** | **딥러닝 분석** |
| **교수명** | **정원일 교수님** |
| **학 과** | **정보통계 보험수리학과** |
| **팀 원** | **조성찬(20110726)**  **황병호(20170845)**  **김시헌(20180728)** |
| **제출일** | **2023. 06. 11.** |

**목차**

[1. 개요 3](#_Toc137405979)

[**2. 데이터 전처리 4**](#_Toc137405980)

[**2.1 데이터 확인 4**](#_Toc137405981)

[**2.2 텍스트 데이터 전처리(Natural Language Processing, NLP) 5**](#_Toc137405982)

[**3. 모델 생성 7**](#_Toc137405983)

[**3.1 순환신경망(Recurrent Neural Network, RNN) 7**](#_Toc137405984)

[**3.2 Long Short Term Memory(LSTM) 8**](#_Toc137405985)

[**3.3 모델 생성 및 적합 9**](#_Toc137405986)

[**4. 데이터 탐구 및 모델 개선 11**](#_Toc137405987)

[**4.1.1 단어 빈도를 이용한 분석 11**](#_Toc137405988)

[**4.1.2 단어 영향도 조사 12**](#_Toc137405989)

[**4.1.3 K-means Clustering 13**](#_Toc137405990)

[**5. 결론 16**](#_Toc137405991)

# 1. 개요

최근 텍스트 데이터를 이용한 다양한 자연어 처리 서비스가 개발되면서, 인터넷을 통한 소비자 리뷰의 중요성이 더욱 부각되고 있다. 호텔 업계에서도 소비자 의견을 체계적으로 분석하고 경영 전략에 효과적으로 반영하는 것이 핵심 전략으로 떠오르고 있다.

본 보고서는 호텔 리뷰 데이터를 활용하여 자연어 처리 및 모델 비교에 대한 연구를 다루고 있다. 약 2만 건의 테스트 데이터와 14만 건의 데이터를 사용하여 분석을 수행하였다. 주요 목표는 다양한 통계적 방법을 활용하여 텍스트 데이터를 분석하고, 모델 간의 성능 차이를 비교한다.

자연어 처리 과정에서는 텍스트 데이터의 전처리, 특징 추출, 모델 학습 순서로 진행한다. 1점부터 5점까지의 평가 점수를 가지고 있는 텍스트 데이터를 바탕으로 자연어 처리를 진행한다. 전처리 과정에서는 정규화, 토큰화, 불용어 처리, 표제어 추출 등을 통해 텍스트 데이터를 정제하여 모델 학습에 사용한다.

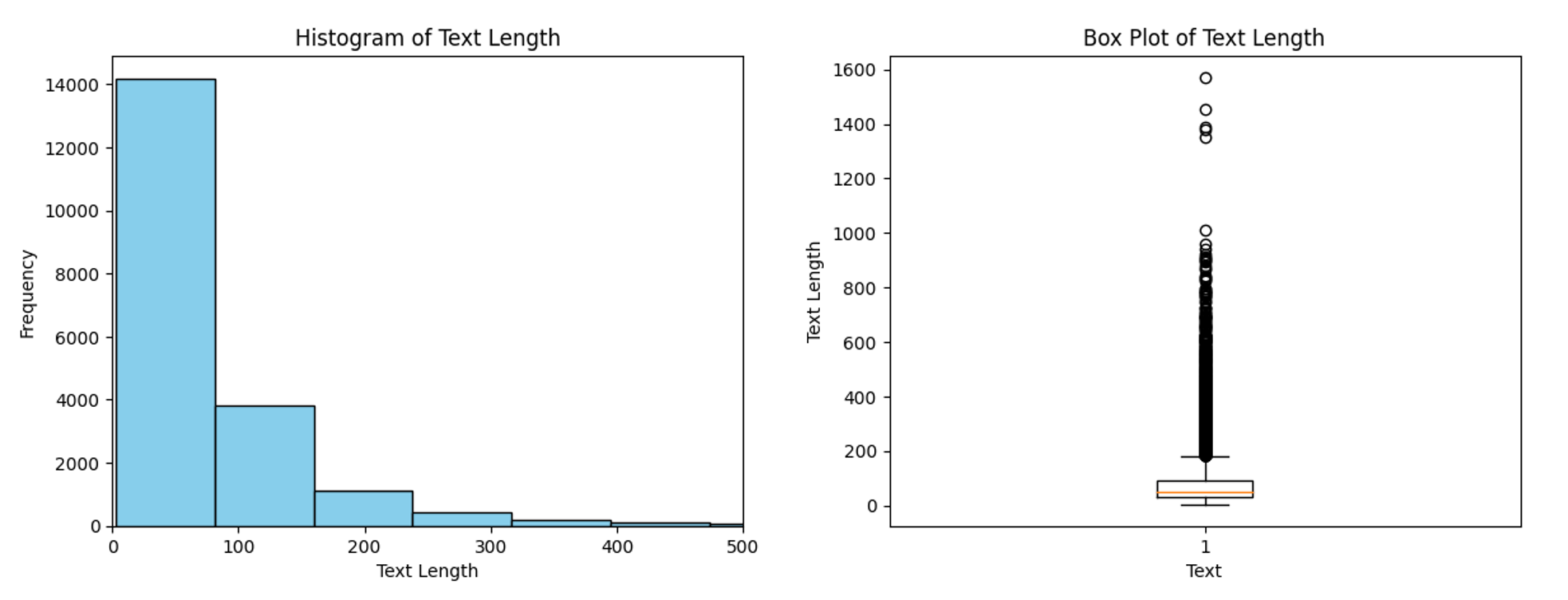
그 후, 딥러닝 모델인 순환신경망의 한 종류인LSTM를 활용하여 모델을 학습시킨 후 성능을 비교하였다. 뿐만 아니라 단어의 빈도수를 시각화 하고, 이를 활용해 성능 향상을 기대할 수 있도록 데이터를 가공하였다. 비지도 학습 방법 중 하나인 K-means 클러스터링을 이용하여 리뷰 데이터를 클러스터로 그룹화하고, 이를 시각화 하여 클러스터간 특징들을 추출한다.

# 2. 데이터 전처리

# 데이터 확인

데이터 셋의 분포를 확인하여 간단하게 EDA를 진행한다. 모델 학습에 사용할 텍스트 데이터가 어떤 형태로 이루어져 있는지 확인하기 위해 각 텍스트가 가지고 있는 단어 개수를 파악한다.

약 2만개의 텍스트 데이터 중에 대부분의 데이터는 100개 이하의 단어로 이루어져 있다. 일부 텍스트의 경우 500개 이상의 단어로 구성되어 있지만 그 빈도수는 상대적으로 작아, 이후 처리 과정에서 제외시킨다. (그림 1)

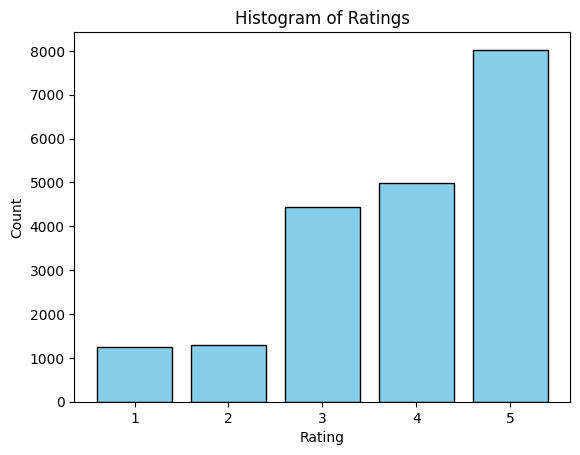


<그림 단어 길이 별 텍스트 데이터 셋 분포>

데이터 셋의 반응 변수(response variable)로 사용하는 평가 점수는 1점부터 5점까지 총 5개의 클래스에 대한 정수로 구성이 되어 있다. 5점에 가까울수록 좋은 평가를 남긴 리뷰이며, 1점에 가까울수록 나쁜 평가를 한 리뷰를 뜻한다.

대체로 긍정적인 평가를 한 리뷰가 가장 많으며 1점과 2점을 평가한 점수가 전체 데이터 셋에 12% 정도 수준만 차지하고 있어 다른 점수를 메긴 리뷰에 비해 상대적으로 적은 것을 확인할 수 있다. (그림 2)

이후 텍스트 데이터를 일부 가공하고, 기계가 이해할 수 있는 숫자 형태의 데이터로 변환하기 위한 과정을 진행한다. 이 과정은 자연어 처리(Natural Language processing, 이하 NLP)라 말하며, 텍스트 데이터가 가지고 있는 복잡한 구조를 단순한 벡터 형태로 가공하는 과정도 포함되어 있다.

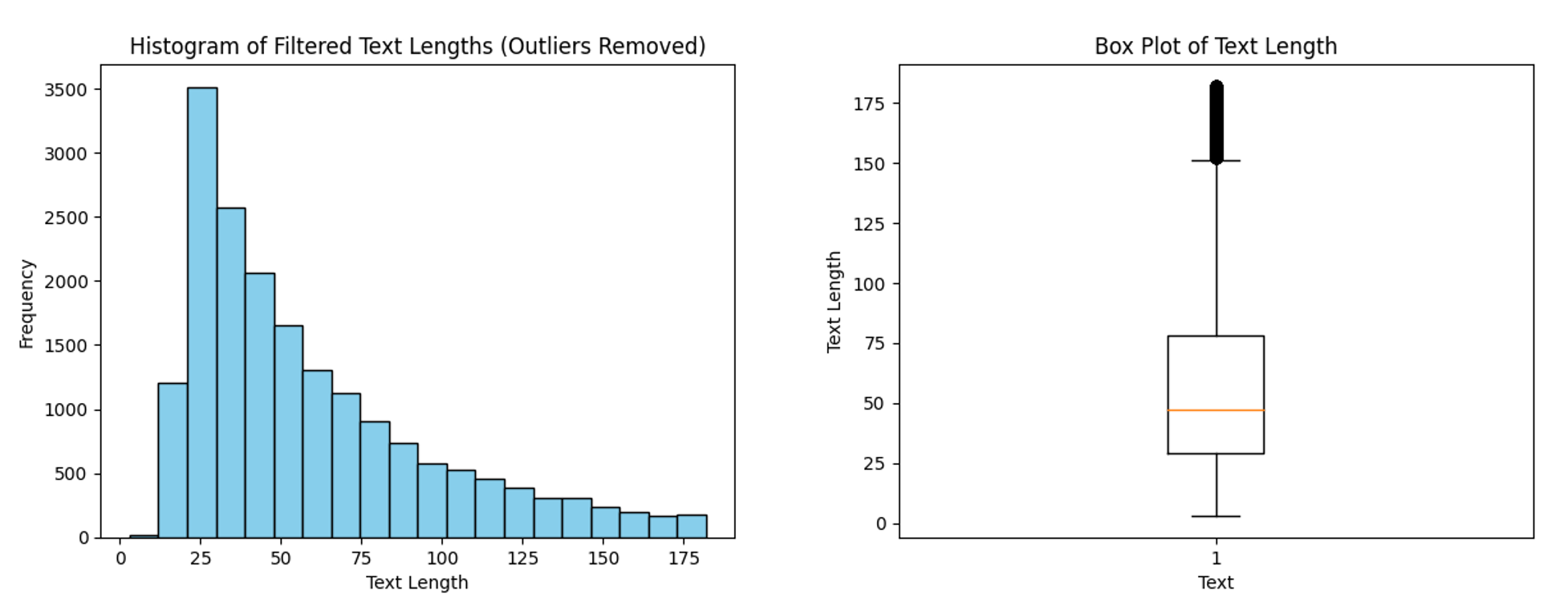


<그림 평가 점수 별 데이터 셋 분포>

# 텍스트 데이터 전처리(Natural Language Processing, NLP)

NLP 과정에 앞서서 앞서 확인한 데이터의 이상치 제거를 수행한다. 이전 과정에서 확인했던 텍스트 별 총 문장의 길이를 기준으로 하여, 특별히 긴 단어를 포함하고 있는 리뷰를 제외시킨다.

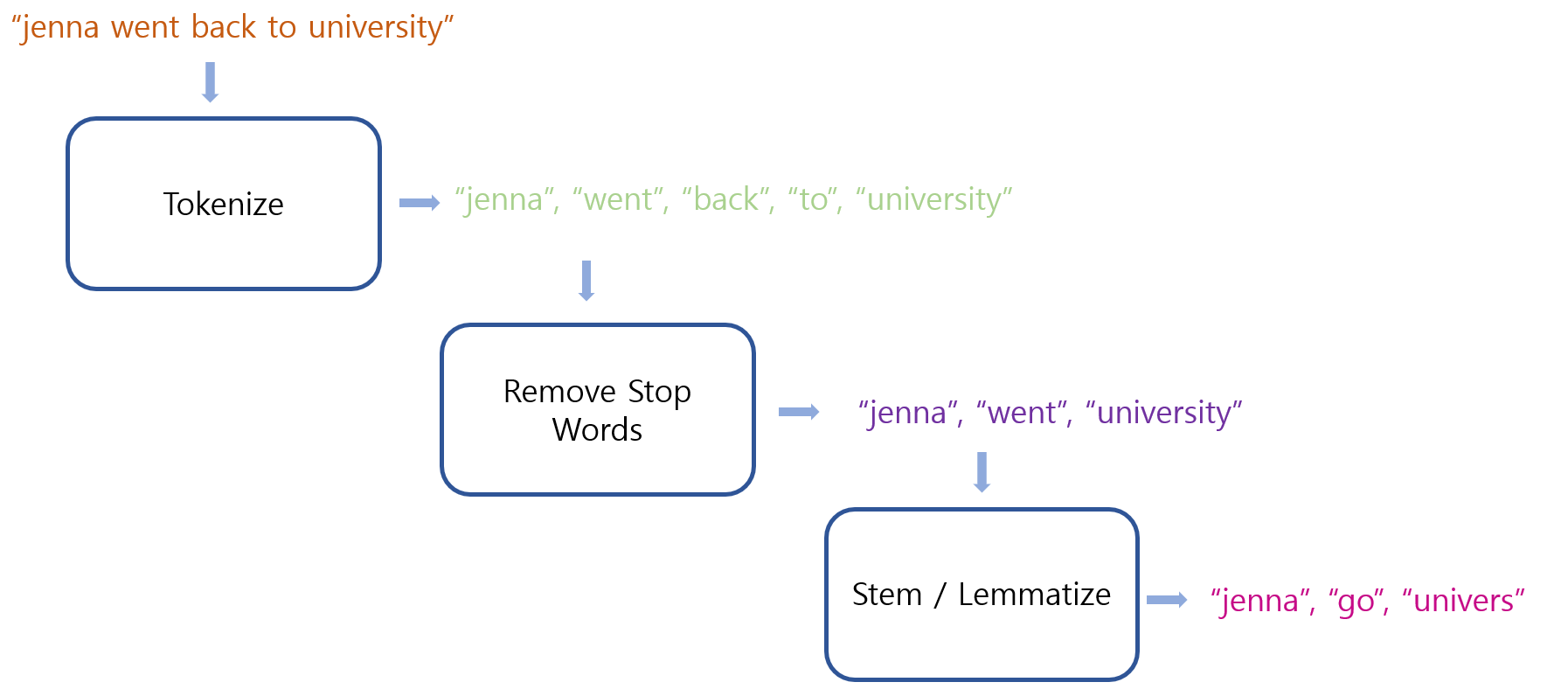
사분위수와IQR을 사용하여, 이상치 제거를 수행할 수 있다. 불용어 제거 처리를 하지 않고, 이상치를 제거한 결과25개 정도 단어를 사용한 텍스트가 가장 많았고, 대체로 20개에서 50개 단어 정도를 사용하는 것으로 확인할 수 있었다. (그림 3)



<그림 이상치 제거 후 단어 길이 별 텍스트 분포>

이후 NLP 과정은 아래와 같은 순서로 진행이 된다. (그림4)

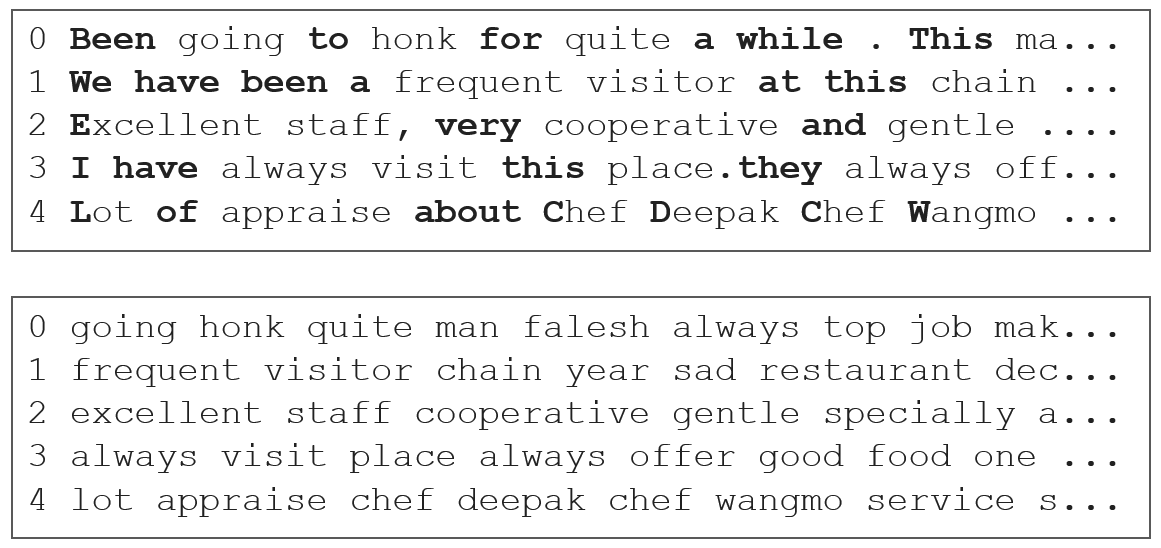
* **토큰화**
* **불용어 제거 및 표제어 추출**
* **시퀀스 생성 및 패딩**

****

<그림 토큰화, 불용어 제거, 표제어 추출 과정>

텍스트 전처리 과정 중 가장 기본이 되는 토큰화를 통해 텍스트 데이터를 단어로 분할하여 나눈다. 이 과정에서 정규표현식을 사용해서 이번 분석에서 제외할 특수 문자나 마침표 등을 제외하고, 나머지 단어들을 분할하여 리스트 형태로 저장한다.

이후 문장에서 중요하지 않으며, 자주 등장하는 단어인 'a', 'the', 'is'와 같은 관사나 접속사는 문장의 의미를 해칠 수 있기 때문에 제거한다. nltk 라이브러리를 사용하여 사전에 정의된 불용어를 제거하고, 이후 표제어를 쉽게 추출할 수 있다. (그림 5)



<그림 자연어 처리 전후 비교, 표시된 단어가 제거 및 수정됐다.>

이후 텍스트 데이터의 단어를 사용해 단어 사전을 만들고, 이를 사용해 시퀀스 생성한다. 시퀀스는 단어 사전에서 인덱스를 추출해 텍스트 데이터를 정수로 이뤄진 벡터 형태로 가공한 데이터이다. One-hot Encoding에 비해 데이터 사이즈를 줄일 수 있으며, 단어의 등장 순서를 유지시킨다는 장점이 있다.

시퀀스를 생성함과 동시에 가장 긴 문장을 기준으로 데이터의 길이를 맞추는 패딩 과정을 진행한다. 이외에도 TF-IDF이나 Word2Vec 와 같은 다양한 NLP 관련 모델이 있지만, 초기 모델 생성 과정에서는 제외시켰다.

# 3. 모델 생성

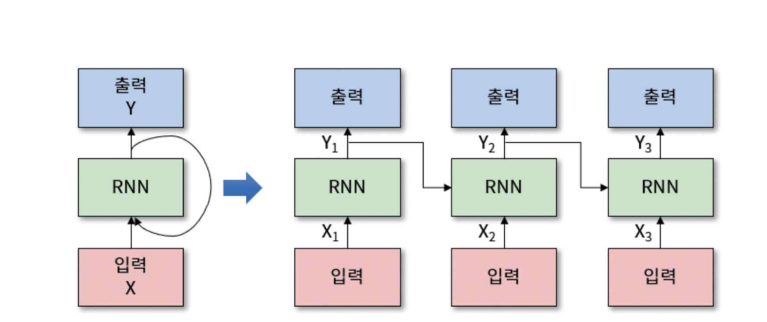
# 3.1 순환신경망(Recurrent Neural Network, RNN)

순환신경망(이하 RNN)은 딥러닝에서 사용하는 인공신경망의 한 종류로, 입력과 출력 사이에 시간적인 의존성이 있는 데이터를 처리하는데 유리한 구조를 가지고 있다. 텍스트 데이터 분석뿐만 아니라 시계열 예측, 주가 분석 등에서 사용하는 모델로 최근 유행하는 다양한 자연어 관련 모델의 근간이 되고 있다.

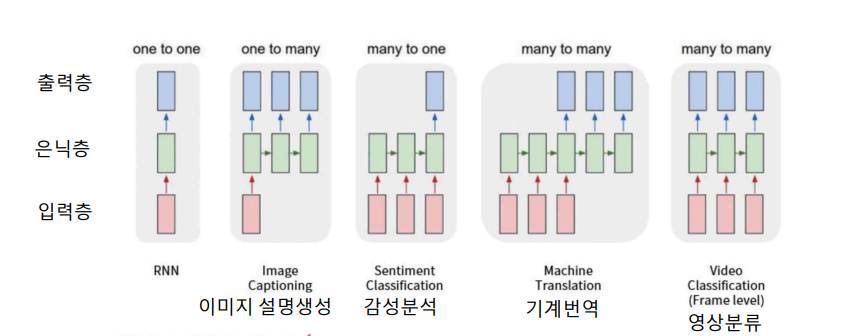
문장이나 문맥 등을 신경망의 데이터로 사용할 때에는 입력의 순서가 중요한데, RNN은 기존 신경망과 달리 이전 출력 값이 현재 결과에 미치는 ‘되먹임’ 구조를 가지기 때문에 용이하다.

즉, 인공신경망이 가지고 있는 은닉층의 유닛이 서로 네트워크를 이루고 있어 입력 데이터로 사용하고 있는 시퀀스의 순서를 네트워크가 이해하고 기억할 수 있는 구조로 이뤄져 있다. 이로써 문장의 각 단어의 출현 여부만을 학습시키는 것이 아니라 그 순서도 학습 결과에 영향을 끼칠 수 있는 구조이다. (그림 6)

또한, 입력과 출력의 길이 제한이 없기 때문에 다양한 형태의 네트워크를 만들 수 있는 강점이 있다. 본 프로젝트에서는 하나의 출력층을 사용하여 평점을 예측하는 감성분석 형태의 구조를 채택했다. (그림 7)



<그림 순환신경망 구조 풀이>



<그림 다양한 형태의 순환신경망>

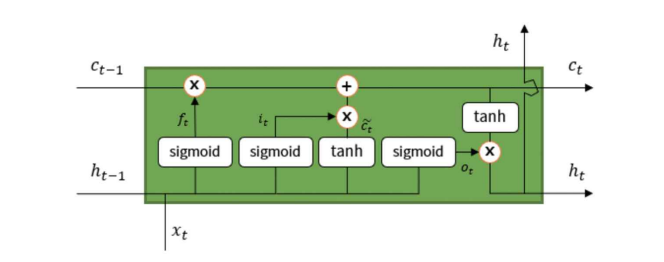
# 3.2 Long Short Term Memory(LSTM)

앞서 설명했던 RNN은 긴 시퀀스에서 나타나는 "장기 의존성 문제" 때문에 제한을 받는 경우가 있다. RNN모델 학습 시 순전파와 역전파(Forward&Backward Propagation) 과정에서 그래디언트가 지수적으로 감소 혹은 증가하면서, 가중치가 불안정하게 업데이트 되는 경우가 발생하기 때문이다.

신경망에 입력이 되는 데이터를 임베딩하여 사이즈를 줄였음에도 불구하고, 이 문제가 발생할 수 있어 모델의 성능을 하락시킬 수 있는 요인으로 작용한다. 실제로 사전 테스트 결과 동일 데이터에 대하여 SimpleRNN의 성능은 LSTM보다 소폭 낮은 것으로 확인하였다.

따라서 이런 문제를 개선한 LSTM을 본 프로젝트의 핵심 모델로 사용한다. LSTM 레이어는 simpleRNN에 비해 복합한 구조를 가지고 있는데, LSTM 셀 사이에만 공유되는 셀 상태(cell state)를 가지고 있다.

LSTM 셀에는 입력(Input), 삭제(Forget), 출력(Output) 게이트가 포함되어 있다. 입력 게이트와 삭제 게이트는 Sigmoid 활성화 함수를 사용하여 현재 셀과 이전 셀의 입력에서 어떤 정보를 유지하고 삭제할지 결정한다. 삭제 게이트의 출력에서 발생한 정보는 LSTM 레이어 셀을 가로지르며 전달되기 때문에(Ct-1, Ct), 장기 의존성 문제를 해결할 수 있는 것이다. 출력 게이트에서는 입력 게이트와 삭제 게이트의 정보를 바탕으로 Tanh 활성화 함수를 통해 값을 출력한다. 이 방식은 simpleRNN과 동일하다. (그림 8)



<그림 셀로 나타낸 LSTM의 계산 흐름>

# 3.3 모델 생성 및 적합

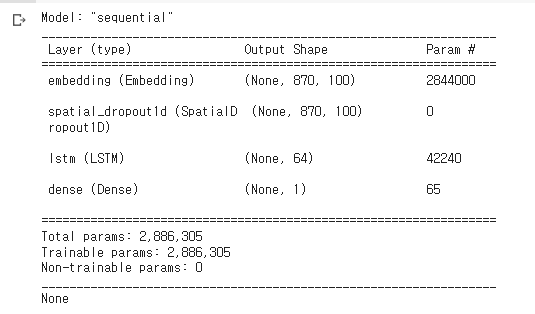
이제 본격적으로 데이터를 학습하기 위한 모델을 생성한다. 이전 전처리 과정에서 생성한 시퀀스를 사용하기 위해 Keras에서 제공하는 Embedding 레이어를 사용한다. 임베딩 차원을 100으로 설정한다.

임베딩은 단어를 고정된 길이의 밀집 벡터로 표현하는 과정으로, 단어 간의 의미적 유사성을 고려하여 단어를 표현한다. 모델의 첫 번째 레이어에 배치하여, 정수로 이뤄진 시퀀스를 입력 받아 임베딩하여 길이를 축소하여 압축된 벡터로 변환하는 역할을 한다. 또한 SpatialDropout1D층을 추가하여 입력 데이터에 대한 드롭아웃을 적용하여 과적합을 방지한다.

다음엔 LSTM 레이어를 추가한다. 유닛의 수는 64로 설정하였다. 마지막에는 1개의 유닛을 가지는 출력층을 배치하고, 선형 활성화 함수를 사용했다.

반응변수에 해당하는 평가 점수를 정규화하여 0부터 1사이의 값을 가지도록 했기에 1개의 유닛만 사용했다. 만약 5개의 유닛을 사용한다면, 활성화 함수로 softmax를 사용해야 한다.

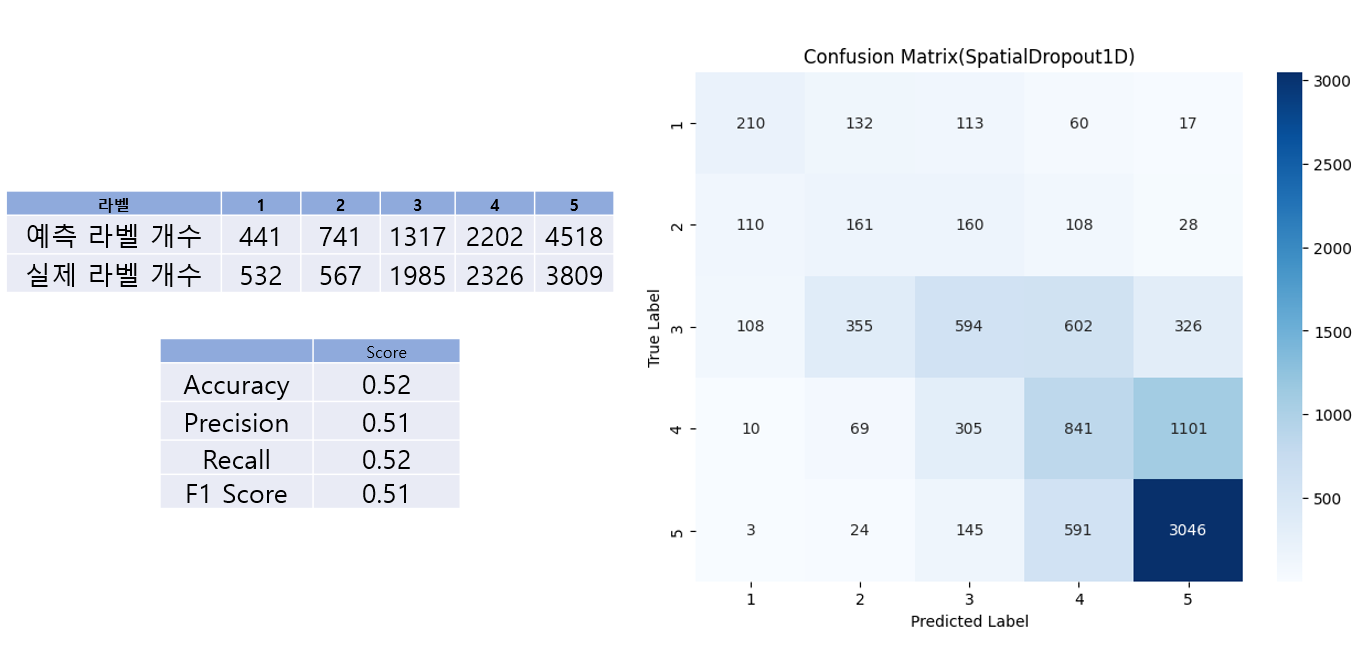
모델을 학습을 위한 Loss 함수로 평균제곱오차 (Mean Square Error, MSE)를 사용하였고, 옵티마이저로는 아담(Adam)을 사용했다.



<그림 Model 요약>

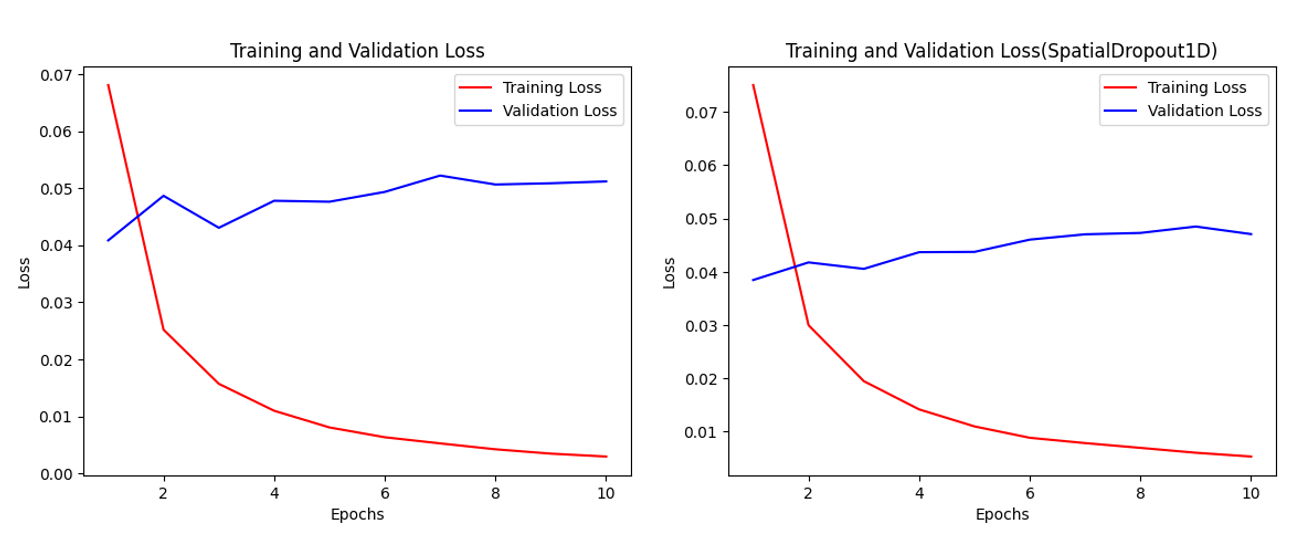
모델 학습은 Epoch 10회로 비교하여 학습하였고, 배치 사이즈는 32로 설정하였다. 출력층의 활성 함수를 선형 함수를 사용했기 때문에, 예측 결과에 대해서는 5개의 클래스로 변환하는 별도의 함수를 사용했다. 약 2만 개의 데이터 중에서 짝수 행의 데이터를 사용해 학습을 진행했으며, 홀수 행의 데이터를 사용해서 예측 및 성능을 확인했다.

일부 데이터에 대해서 잘못된 분류를 했지만, 1점인데 5점으로 분류하거나, 5점인 평가를 1점으로 분류한 치명적인 오류는 전체 예측 결과에 0.2% 수준이다. (그림 10)



<그림 모델 예측 결과>

또한 모델 생성시 사용했던 SpartialDropout 레이어에 대한 효과를 비교했다. SpatialDropout1D는 RNN 또는 LSTM 레이어에서 입력 데이터에 대해서 시간 축을 기준으로 드롭아웃을 적용하여 시퀀스 데이터의 전체적인 패턴을 학습하고 과적합을 줄이는 역할을 한다. 실제로 해당 레이어를 적용시 Validation Loss 0.05이하로 소폭 감소하는 것을 확인할 수 있었다. (그림 11)



<그림 Validation Loss 비교, 드롭아웃 미적용(좌), 적용(우)>

# 

# 4. 데이터 탐구 및 모델 개선

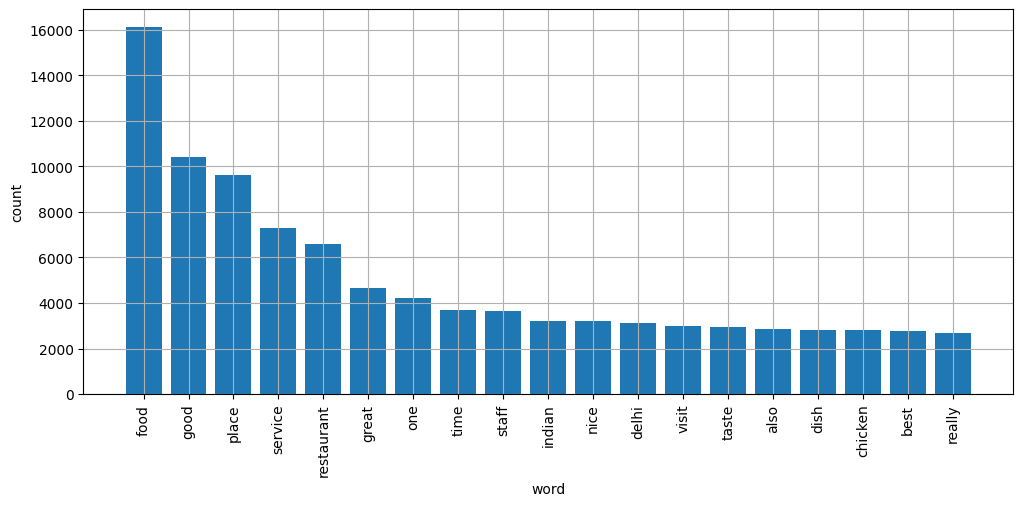
모델 학습에 더 좋은 결과를 만들기 위해 주어진 데이터에 대한 추가 분석을 진행한다. 단어의 발생 정도를 파악하여 모델 학습에 영향을 미칠 수 있는지 확인하고, 이를 시각화한다.

또한 비지도 학습인 K-means Clustering를 활용하여, 각 클러스터가 어떤 정보의 단어를 가지고 있는지 확인한다.

# 4.1.1 단어 빈도를 이용한 분석

전체 데이터 셋이 4점과 5점이 차지하는 비율이 높기 때문에, 모델 학습 시 일부 단어가 더 자주 등장하게 될 가능성이 있다. 이로 인해 해당 단어에 대한 가중치가 높게 평가되어, 전체적인 모델 성능의 하락 가능성이 존재한다.

라벨을 구분하지 않고 단어 별 빈도수를 파악하여 상위 20개의 단어를 추출하여 분석하는 과정을 진행한다. (그림 12)



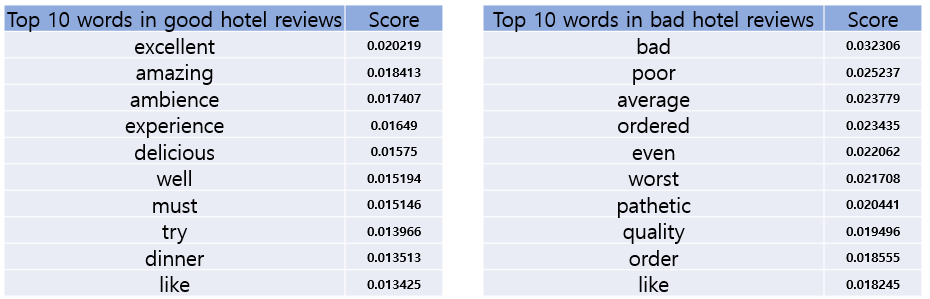
<그림 자주 발생하는 단어>

단어 목록을 추출한 결과, ‘good’, ‘great’, ‘nice’ 등 긍정적인 의미를 가지고 있는 단어와 ‘food’, ‘restaurant’, ‘staff’, ‘taste’ 등 호텔이 제공하는 음식이나 서비스에 대한 뜻을 가진 단어가 자주 등장하는 것을 확인할 수 있다. 전체 데이터 셋이 긍정적인 리뷰에 편향되어 있기 때문에, 부정적인 뜻을 지닌 단어는 확인할 수 없다.

# 4.1.2 단어 영향도 조사

텍스트가 가지고 있는 단어가 리뷰 평가에 끼치는 영향을 확인하기 위해, 데이터를 긍정, 부정으로 나누어서 분석을 진행한다. 앞서 분석한 상위 20개 단어를 전체 데이터 셋에서 제거를 한 뒤에1점과 2점의 평가를 한 리뷰는 부정, 4점과 5점은 긍정으로 구분하여 분석한다.

Sklearn의 TF-IDF (Term Frequency-Inverse Document Frequency)를 사용하여 단어 발생 빈도를 파악하고, 점수를 부여하여 단어별 영향도를 평가한다. TF-IDF는 텍스트 데이터에서 각 단어의 상대적인 중요도를 평가하는 데 사용되는 통계적인 방법이다. TF-IDF는 각 문서에서 단어의 등장 빈도인 TF (Term Frequency)와 단어가 얼마나 일반적인지를 나타내는 IDF (Inverse Document Frequency)를 곱하여 계산된다. 이를 통해 각 단어의 중요도를 파악할 수 있다. (그림 13)

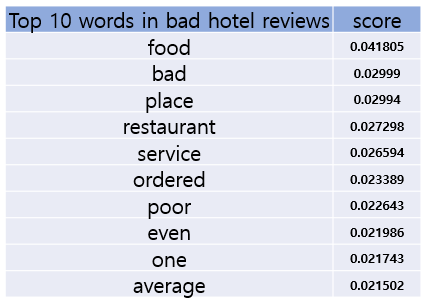


<그림 TF-IDF를 사용하여 단어 별 영향도 평가>

전체적으로 자주 발생하는 20개 단어를 제거 한 이후, 단어 별 영향도를 확인했다. 그 결과 두 클래스 별 단어의 성향이 확연히 다른 것을 볼 수 있다. 긍정적 리뷰에서는 ‘excellent’, ‘amazing’ 등의 단어가 점수가 높았고, 부정적 리뷰에서는 ‘bad’, ‘poor’ 등의 단어가 점수가 높은 것을 확인할 수 있다.

이전 과정에서 상위 20개 단어를 제거하지 않고 부정적 리뷰에서 TF-IDF 점수를 확인한 결과, ‘food’, ‘bad’, ‘place’ 등이 부정적 리뷰에 영향을 많이 미치고 있는 것을 확인할 수 있었다.

부정적 문장에서도 음식이나 분위기, 장소에 대한 단어가 많이 사용되는데, 긍정적인 리뷰에도 많이 포함되는 단어이다. 이는 모델 학습시에는 불리하게 작용할 가능성이 있는 것을 뜻할 수도 있다. (그림 14)

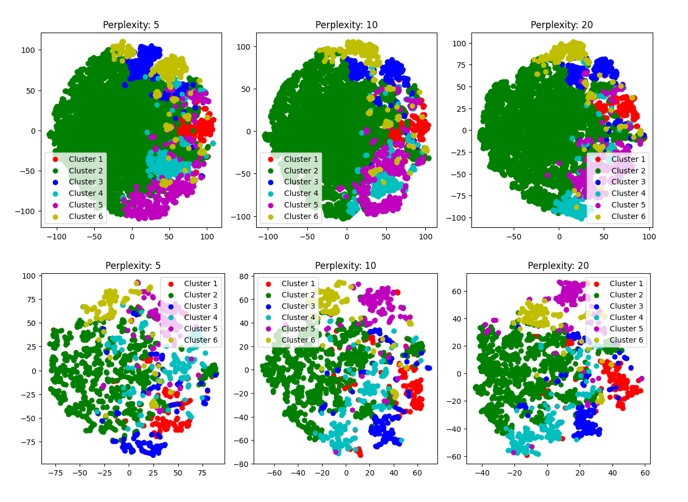


<그림 단어 제거를 수행하지 않은 부정적 리뷰 단어>

# 4.1.3 K-means Clustering

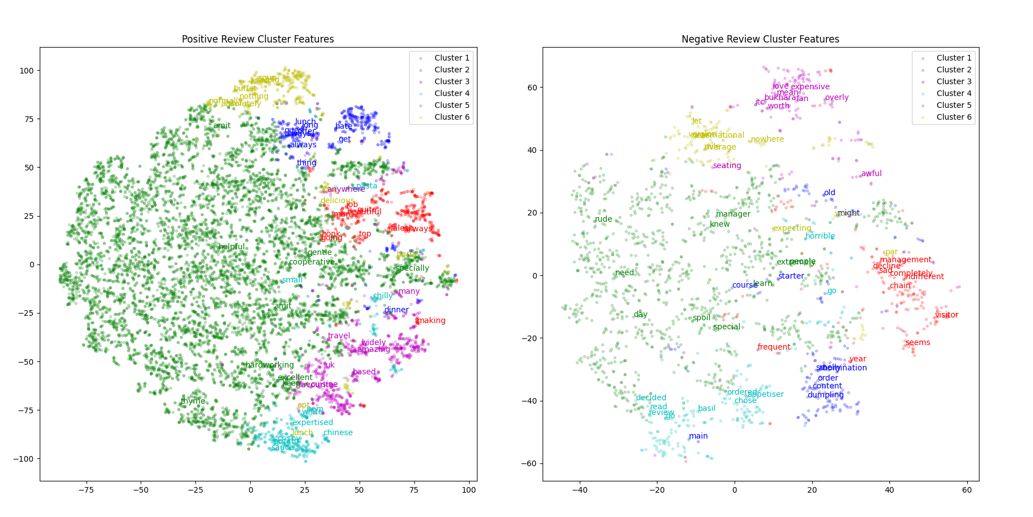
이외에도 라벨을 사용하지 않고, 각 시퀀스가 가지고 있는 데이터를 활용하여 비지도학습을 진행한다. 시퀀스는 단어 사전으로부터 추출한 정수형 데이터를 가지고 있기 때문에, K-means Clustering을 통해 각 시퀀스를 군집화 하고, 유의미한 클러스터가 있는지 확인한다.

클러스터의 개수는 6개로 지정한다. 시퀀스가 가지고 있는 단어의 개수 차이로 인해 발생하는 문제를 제거하기 위해 시퀀스의 길이를 10으로 제한하여 수행한다. (그림 15)

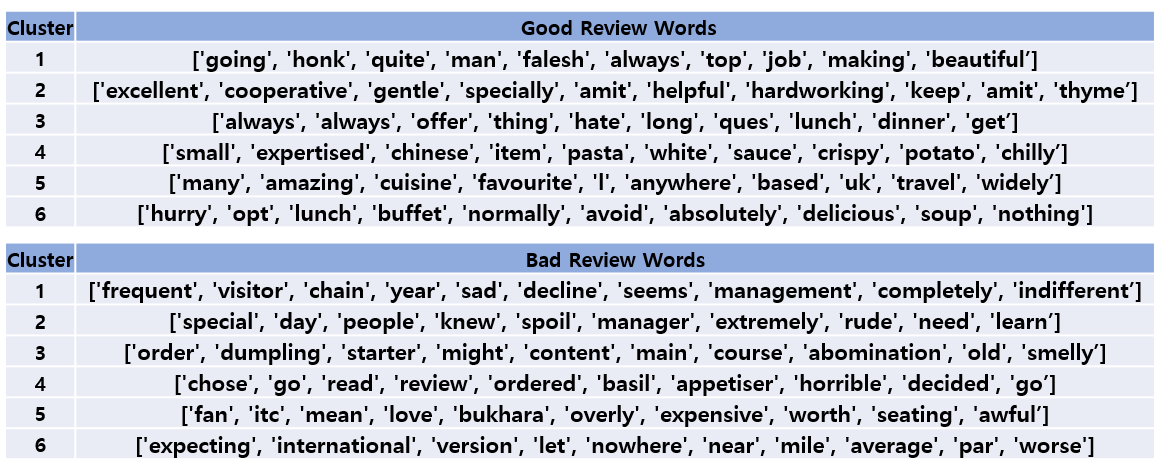


<그림 perplexity별 K-means Clustering 수행 결과, 긍정(상) 부정(하)>

Perplexity를 조정하여 5, 10, 20일 때 클러스터링의 결과를 확인할 수 있다. 미미한 차이를 보이지만, 20일 때 일부 클러스터에 군집화가 잘 관찰되는 경향이 있다. 이후 해당 Perplexity로 설정하여 각 클러스터가 가지고 있는 시퀀스를 추출하여 비교한다. (그림 16, 그림 17)



<그림 Perplexity 20일 때 Clustering 결과와 단어 시각화, 긍정(좌), 부정(우)>



<그림 K-means Cluster을 통해 얻은 시퀀스>

긍정 리뷰에 대한 클러스터링 결과, 전체적으로 음식과 분위기에 대한 단어가 많이 포함이 되어 있음을 확인했다. 시퀀스를 가지고 클러스터링을 진행했기 때문에, 실제로 입력으로 사용했던 시퀀스를 추출했다.

클러스터에 포함된 시퀀스를 확인하여, 각 클러스터가 어떤 뉘앙스의 글들을 포함했는지 확인하는 수단으로 사용할 수 있다. 군집화가 잘된 긍정 리뷰 클러스터 3의 경우 ‘hate’, ‘long’과 같은 단어를 포함한 시퀀스를 확인할 수 있는데, 이는 평점과 무관한 내용의 텍스트가 작성되었을 가능성이 있다.

부정 리뷰의 경우 클러스터 모두 부정적인 뜻을 지닌 단어를 포함하는 시퀀스를 확인할 수 있었으며, 음식에 대한 평가(3, 4번 클러스터)와 서비스에 대한 평가(1, 2번 클러스터)로 나뉜 것을 확인할 수 있었다.

추가적인 데이터 수집단계나 라벨이 없는 데이터에 대해서 작업이 필요할 때 K-means Clustering 과 같은 비지도 학습으로 진행한다면, 적은 리소스로 분류할 수 있을 가능성도 존재한다.

# 5. 결론

호텔 리뷰 데이터를 바탕으로 자연어 처리(NLP)를 진행하고, 순환신경망을 사용하여 딥러닝 모델을 생성하였다. 또한 단어 빈도 및 영향도 탐색, K-means Clustering을 활용한 비지도 학습을 수행하여, 텍스트 데이터가 가지고 있는 특징을 파악했다.

텍스트 데이터가 가지고 있는 가장 큰 특징인 단어와 문맥을 벡터 형태의 데이터로 전환하고, 이를 위한 효과적인 모델을 채택하는 것이 본 프로젝트의 핵심 과제였다.

다양한 순환신경망 중에서 장기 의존성 문제를 해결할 수 있는 LSTM이 가장 성능이 좋았으며, 다른 통계 모델과 비교하는 것이 추후 과제로 남아있다.

이번 프로젝트를 통해 최종적으로 생성한 모델과 자주 등장하는 단어 제거와 같은 데이터 전처리 방법을 사용하여 약 14만개의 텍스트 데이터에 적용했다.

* Training Loss: 0.0083, Validation Loss: 0.02, Test Loss: 0.03
* Accuracy: 0.59, Precision: 0.56, Recall: 0.59, F1 Score: 0.56

2만 개의 테스트 데이터를 사용했을 때보다 성능이 소폭 상승하였고, Validation loss가 줄어든 것을 확인할 수 있었다. 하지만 이는 데이터 처리에 의한 결과보다 데이터의 크기에 의한 영향일 가능성이 존재한다. 따라서 추가적인 모델 수정 및 학습으로 향상된 모델 성능을 기대할 수 있다.